SIP Based OBS networks for Grid Computing[*]

A. Campi, W. Cerroni, F. Callegati

Department of Electronics, Informatics and
Systems, University of Bologna
Via Venezia 52 – 47023 Cesena ITALY
{fcallegati,acampi,wcerroni}@deis.unibo.it

G. Zervas, R. Nejabati, D. Simeonidou

Department of Electronic Systems
Engineering, University of Essex
Colchester - CO4 3SQ, UK
{rnejab, dsimeo, gzerva }@essex.ac.uk

Abstract. In this paper we discuss the use of the Session Initiation Protocol (SIP) as part of the control plane of an application aware Optical Burst Switching (OBS) network, able to support Grid computing applications. The paper presents the possible alternatives for the architecture of such control plane and reports of an experiment in an existing OBS test-bed where this approach was successfully tested in practice.

1 Introduction

In a Grid computing environment the network becomes part of the computation resources, therefore it plays a key role in determining the performance. Optical networks promise very high bandwidth interconnection and are a good candidate to support this sort of services [1].

Mostly optical circuit switching based on wavelength routing has been considered to this end but, if we imagine a future scenario with a broad range of user communities with diverse traffic profiles and connectivity requirements, providing network services with wavelength granularity is a solution that has drawbacks in terms of efficiency and scalability. In this case Optical Burst Switching (OBS) [2] is a candidate to implement a more scalable optical network infrastructure to address the needs of emerging Grid networking services and distributed applications [3]. The feasibility of a programmable or application aware OBS network, to support very high speed and “short lived” grid sessions, has recently been proposed and investigated [4].

In this paper we address the problem of implementing the control plane for an application aware optical network. In spite of its importance the research on this topic is rather limited and again mainly related to the wavelength routing scenario [5]. In this paper we propose an architecture that is based on three layers of standard protocols: the Job Subscription Description (JSDL) [6] language used by the application to communicate their service needs, the Session Initiation Protocol (SIP) [7] used to map the jobs requests into communication sessions and the Generalized Multiprotocol Label Switching (GMPLS) [8] used to control the networking

[*] This work is partially supported by the Commission of the European Union through the IST e-Photon+ project.

functions. We will show that, by properly interfacing these layers this architecture proves to be effective, flexible and scalable.

The paper is organized as follows. In section 2 we discuss the architecture of the control plane. In section 3 we describe the experimental set-up used to prove the feasibility of this concepts. In section 4 we describe the experiment results and in section 5 we draw some conclusions.

2 Application Aware Control Plane based on SIP

The Job Submission Description Language (JDSL) specifies a general syntax for documents used by the applications to notify to the infrastructure the requirements of the jobs they intend to submit. The result is the definition of an XML schema for documents to be exchanged among grid entities. In broad term JDSL can be considered the communication language used by the control plane components of the Grid at the highest application level.

At the same time network elements have their own control plane to guarantee proper forwarding. IP based routing is not easy to implement at very high speed. Forwarding based on a simpler function than full IP header processing is therefore desirable. Moreover traffic engineering and dedicated protection/restoration are not easy to implement with a pure connectionless best-effort network protocol like IP. This is what motivated the introduction of MPLS [8] and GMPLS [9] that are efficient telecom-oriented solution for the fast and automated provisioning of connections across multi-technology (IP/MPLS, Ethernet, SDH/SONET, DWDM, OBS etc.) and multi-domain networks, enabling advanced network functionalities for traffic engineering, traffic resilience, automatic resource discovery and management.

Indeed JDSL is a pure application protocol that is not meant to deal with networking issues while GMPLS is not natively designed to support service oriented functionalities. *As a consequence the technical challenge rises from the inter-working of the application platforms and of their resource management systems with the underlying next generation optical networks powered with GMPLS network control plane.*

The abovementioned problem is not peculiar to grid computing and is rather typical of emerging networking scenarios. Next generation networks will very likely deal with applications that require “communication services” within “communication sessions” rather independently with respect to the networking environment they are using. Such a problem is addressed by the IMS specifications [10]. In broad terms the goal of the IMS is to define a service control architecture that enables all sorts of multimedia services to be carried over an IP based networking infrastructure.

The IMS has the SIP protocol as the basic language to establish and control communication sessions. SIP is a session control protocol well known but indeed its adoption as a cornerstone of the IMS architecture has indeed largely increased the interest into it. SIP is a rather general protocol that can support almost any “user defined” functionality by means of suitable payloads. It provides all the messages that are necessary to control a long lasting communication session, including authentication, presence discovery, re-negotiation and re-routing etc.

The proposal presented in this paper conceptually follows the IMS approach applied to the Grid over optics scenario. A service boundary is kept between the user and the network and all the exchanges of information related to the management of the specific application service needs are done within SIP communication sessions. In such a view the SIP sessions become the key element for the application aware optical network. Sessions can be established after proper user authentication, suspended, continued, re-routed and their service profile can be modified adding or taking away resources or communication facilities according to the needs. In broad terms the introduction of the session provides the network with the status information that can be used to manage the data flows (retrieve, modify, suspend, etc.) both according to the network need and with reference to the application requirements.

Basically SIP will enable end-to-end dynamic service provisioning across a global heterogeneous optical network infrastructure. SIP will give to the middleware the capacity to exploit the network-oriented features of GMPLS on one side and the rich semantic of application oriented languages such as JDSL on the other.

SIP interworking with OBS

The OBS network is a transport cloud with edge routers at the boundaries. Edge routers provide access from legacy domains into the OBS network by performing burst assembly and by creating the burst control packets, while the OBS core routers are devoted to pure optical switching of the data bursts.

SIP is used to challenge, negotiate and maintain application sessions and brings the notion of session into the application aware OBS network. This notion provides a significant new spectrum of opportunities in terms of quality of the communication. Generally speaking the network layer deals with data flows. Since a communication session at the application layer may involve several data flows (either parallel or sequential or both) the introduction of SIP on top of the OBS transport layer enables the possibility to manage the communication requirements for the entire application session and not only for the individual burst/packet or stream.

In this paper we propose and demonstrate the use of the SIP protocol to support Grid networking over an OBS network. We will refer in the following to a Grid-aware SIP Proxy (GSP) as a network component that is able to satisfy the communication requests of Grid applications by exploiting the SIP protocol over the OBS network.

The GSPs are an integral part of the control plane of the Grid-aware OBS network. They works coupled with OBS edge and/or core routers by means of a middleware that process the SIP message payloads and talks with the OBS layers by means of suitable APIs. GSPs have SIP signalling functionalities to manage the application sessions and exploit the OBS network to provide the required communication facilities. This means that GSPs send the requests to open communication paths (for instance GMPLS Label Switched Paths or LSPs) to the optical nodes according to the applications requirements and leave to the network layer all the network related problems such as flow control, QoS guarantee etc.

The middleware between the GSPs and the OBS control plane is made of 3 main parts:

1. Interface with SIP servers to interpret the applications requests related to the sessions supporting Grid computing requests (for instance parsing and processing JSDL documents);
2. interface to the control plane (for instance GMPLS) of the OBS network;
3. internal engine which translate the application requests into network related communication instances (for instance JSDL request into GMPLS LSPs requests).

Sessions established by means of SIP signaling can be mapped on the network layer, by means of the middleware, according to two well known approaches, outlined in the following [5].

Overlay approach

The overlay approach keeps both physical and logical separation between the SIP (session) layer and the optical network. The non-network resource (i.e. Grid resource: computing and storage) and the optical layer resources are managed separately in an overlay manner. An IP legacy network carries the SIP signals and the OBS network carries the Grid data: the slow network (based on legacy technology) is used for the signalling while the fast network (optical OBS) is used only for data transmission.

The GSPs are placed into the edge routers only and use a legacy electronic connection to forward the SIP signals to the other GSPs (at other edges of the OBS cloud). GSPs negotiate the session and are composed by registrar, location and proxy servers in order to provide all the functions of a SIP network. The users (i.e. the application) use the SIP protocol to negotiate the Grid communication session. When the session is set the middleware is responsible to request a data path between the edge routers involved in the session to the optical network control plane. Then the session data cut through the OBS network and the SIP layer is not involved any more.

The main advantage of this approach is to use the various technologies for what they do best. The well-established legacy technology based on the current Internet is used to carry the signalling, i.e. low speed and rather low bandwidth data transmissions while the high speed OBS network is used to carry bulk data.

Integrated approach

The integrated approach enriches the optical control plane with SIP functionalities, to realize a pure OBS network that works controlled by the SIP protocol to negotiate the application (grid) sessions and by GMPLS to control the connection management functions. No legacy networks are into play any more and signalling and data share the same networking infrastructure. Since the signalling and the data plane share the same network infrastructure, all OBS node must at least have the capacity to read and forward a SIP message.

The GSPs in the edge nodes are full functional and logically identical to those mentioned before. On the other hand the GSPs in the core nodes can be equipped with a subset of functionalities, to satisfy the best performance/complexity trade-off. On one side the SIP functionalities could be limited to a light proxy with forwarding functions only. In such a way most of the intelligence of the SIP layer can still be segregated at the boundaries of the OBS network. On the other side the OBS core nodes could be equipped with full functional GRID proxies and therefore could take part in the operations of managing the application requests, for instance actively participating in the resource discovery process.

Resource discovery and reservation

As an example of application of the concepts presented in the previous sections, here we discuss how SIP can be used to implement resource discovery and reservation, a meaningful case to discuss.

We assume that Grid resources and users are divided into domains, an organization that is particularly indicated in a Grid network oriented to consumer applications where a large number of resources and users can be present. A domain can be the set of users connected to a single GSP as well as a subset or superset. SIP can be used to implement resource discovery and reservation in two different ways.

The former follows a single phase approach where both resource discovery and reservation are performed at the same time. The user with data to be processed remotely (e.g. Grid user, e-Science) sends a request to the GSP at its ingress edge router to the OBS cloud in the form of an INVITE SIP message. Encapsulated into the INVITE there is the JSDL document which describes the job requirement. The SIP server passes the job requests (JR) to the its underlying middleware that performs a resource discovery algorithm to find out whether there are enough computing resources available within its known resource. If the answer was “yes” the proxy would start establishing the session, if it was “no” it would forward the message to the other proxies in the network to look for the requested resources until the message arrive to an available resource or it is dropped. If a resource is found the INVITE is acknowledged and a session between user and resource is created, thus reserving the resource usage. At the same time a data path is created in the OBS network to carry the job related data.

In the latter approach two phases are used, resource discovery at first with a notifications mechanism and then direct reservation by the client. The basic concept behind this idea is the understanding that the resource management in a Grid network is similar to the presence notifications of a SIP network. A resource is associated with SIP address (i.e. a user of the SIP network) and has a set of proprieties with a state. The presence notifications messages of the SIP network can then be used to update and notify, support presence, messaging, state change detection, etc. Using messages like PUBLISH, SUBSCRIBE and NOTIFY the various management functions of a resource can be provided. Fig. 1. shows two simple call flows for resource discovery management.

- The scenario represented in Fig. 1.a describes a localised publishing approach. In this approach the Grid resources are published only to the attached SIP proxy with a PUBLISH message. A user requesting resources sends a SUBSCRIBE message to the nearest GSP. The GSP 1 checks the status of its own resources and if they can satisfy the request then notifies the client. Otherwise, the

SUBSCRIBE message is propagated to the other known GSPs either by utilizing sequential or parallel forking in order to discover the requested resources. The GSP with available resources sends a NOTIFY message back to user. The NOTIFY is used to communicate to the user the availability and location of the resources (i.e. address of the end point or GSP or domain).

- Fig. 1.b portrays the scenario in which the availability of Grid resources is distributed to all GSPs on the same domain by utilising PUBLISH message. Then the SUBSCRIBE message sent by the user to describe its request is sent to the nearest GSP. This GSP is aware of all resources of the domain and discovers the requested ones can send a NOTIFY message back to the user. Otherwise propagates the SUBSCRIBE message to other domain(s).

After the resource discovery the user knows the location (SIP name or network address) of the resource and can attempt a direct reservation by an INVITE message. The detailed description of the reservation process is presented in experimental description section.

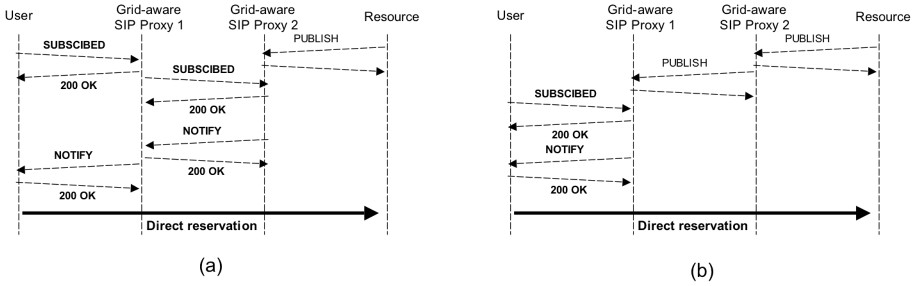

Fig. 1. Example of resource discovery and state notification. (a) Localized approach. (b) Distributed approach.

3 Experimental set-up

In this section we describe an experimental set-up that demonstrates the feasibility of using SIP as the application signaling component in a GRID over OBS network. It refers to the overlay approach with one way reservation that has been tested at first, since it does not require to embed the SIP functionalities into the OBS nodes and to encapsulate SIP messages into optical bursts. Nonetheless the experiment here reported is, in our view, significant to prove the feasibility and effectiveness of the proposed concept. Moreover this choice is also motivated by the fact that the overlay approach will likely be the first choice in the actual deployment of such solution. The experiment exploits an existing OBS test-bed whose main characteristics have been already reported in the literature [4].

In the experiment the end user intending to submit a job request to the Grid environment prepares a JDSL document. It registers to its service access point, i.e. a

GSP, and sends its request by encapsulating the JSDL document into a SIP message. The message is processed at the GSP in the ingress edge node. By analyzing the JDSL the GSP execute the resource discovery and then forwards the SIP request to the SIP proxy connected to the egress edge node leading to the computational resources. Upon acceptance of the job request, the OBS network is used to carry the job submission (application data) over optical bursts.

The overlay network testbed architecture utilizes SIP protocol on a higher physical and logical layer to the OBS network testbed and it is illustrated in Fig. 1. The testbed operates at 2.5 Gbps both for the Optical Burst Ethernet Switched (OBES) control channels [11] as well as the out-of-band data channels which both operate in asynchronous mode. It also provides full-duplex communication between two Edge Routers and one Core Router. Both edge and core OBS routers are equipped with an embedded network processor, implemented with a high-speed Field Programmable Gate Arrays (FPGA). The data plane generates variable sized bursts with variable time intervals and operates in bursty mode.

The GSP are based on a SIP stack called pjsip [14] that runs on a PC. In the experiment we implemented a GSP for any edge router. In the specific case of the testbed this means two GSP are coupled with the two edge routers.

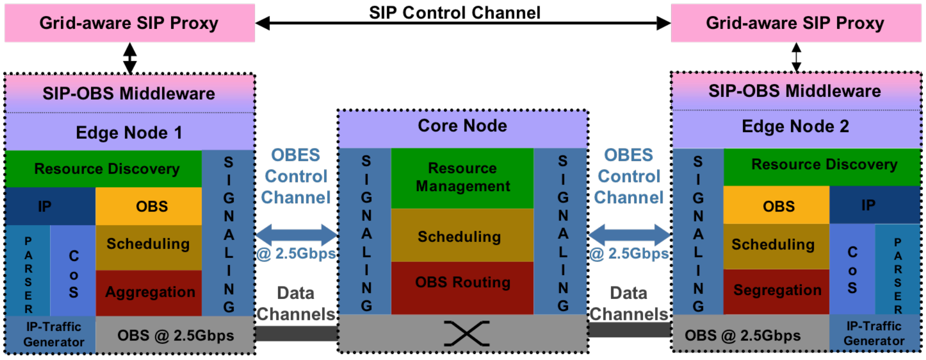

Fig. 2. Overlay network architecture utilizing SIP over OBS. It incorporates two OBS Edge Routers equipped with Grid-aware SIP Proxies on top and one Core Router. The Testbed operates in full-duplex mode.

Both edge routers operate as ingress and egress nodes by utilizing Xilinx high-speed and high-density VirtexII-Pro FPGA prototype boards. At the ingress side, the edge routers utilize one fast and widely tunable SG-DBR laser each for wavelength allocation agility. Both OBS edge nodes are connected to the GSP running in the two PCs. The PCs are connected by a normal Ethernet network, thus realizing the separate signaling interconnection for SIP messages between remote proxies.

The core router integrates an FPGA and a 4×4 all optical switching matrix (OXS). The FPGA controls the OXS. It extracts the incoming BCHs, processes it and in turn drives the OXS with appropriate control signals to realize the switching path required by the BCH. Finally it reinserts a new BCH towards the egress edge router. The 4×4

OXS matrix has extremely low crosstalk levels of < -60 dB and fast chip switching time of ~1.5 ns have been achieved [11].

4 Experiment Description

The network concept demonstrated in the test-bed provides application layer resource discovery and routing of application data or user's jobs to the appropriate resources across the optical network. The OBS control plane architecture comprises a resource discovery stage and a traditional OBS signalling stage using Just-In-Time (JIT) bandwidth reservation scheme [2].

The user with data to be processed remotely (e.g. Grid user, e-Science) sends a request to the GSP of the ingress edge router to the OBS cloud. The request carries the job specification and resource requirements (i.e. computational and network) in the payload, in the form of a JSDL document. The SIP server classifies the incoming job requests (JR) and then either process it or forward it to the next proxy which has knowledge of the resources available.

In this experiment, due to the very simple network topology, the distribution of resources is static. Client applications are connected to the edge router 1 while resources are connected to the edge router 2. Therefore all messages with requests are sent by the clients to GSP 1 coupled with edge router 1. Since GSP 1 does not have any available resource, the requests are forwarded to the GSP 2, coupled with edge router 2, from where the resources are reachable.

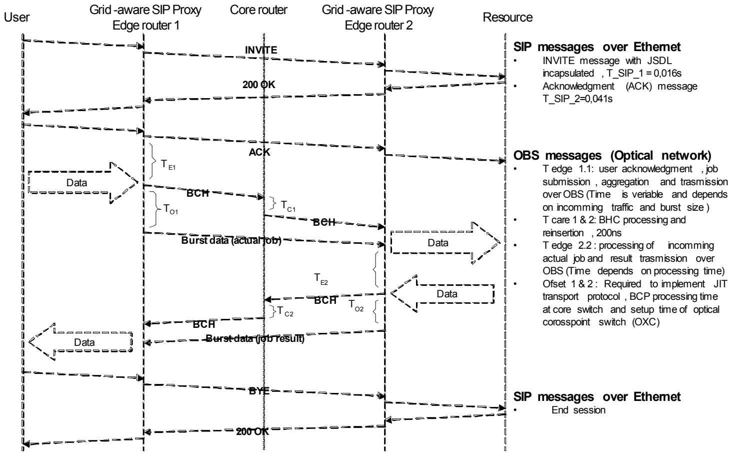
User
Grid -aware SIP Proxy Edge router 1
Core router
Grid -aware SIP Proxy Edge router 2
Resource
INVITE
200 OK
ACK
T E1
Data
BCH
T O1
T C1
BCH
Burst data (actual job)
Data
T E2
Data
BCH
T O2
T C2
BCH
Data
Burst data (job result)
BYE
200 OK
SIP messages over Ethernet
• INVITE message with JSDL incapsulated , T_SIP_1 = 0,016s
• Acknowledgment (ACK) message T_SIP_2=0,041s
OBS messages (Optical network)
• T edge 1.1: user acknowledgment , job submission , aggregation and trasmission over OBS (Time is veriable and depends on incomming traffic and burst size)
• T care 1 & 2: BHC processing and reinsertion , 200ns
• T edge 2.2 : processing of incomming actual job and result trasmission over OBS (Time depends on processing time)
• Ofset 1 & 2 : Required to implement JIT transport protocol , BCP processing time at core switch and setup time of optical corosspoint switch (OXC)
SIP messages over Ethernet
• End session

Fig. 3. Session control messages and data flows in a grid job transfer over the OBS network. The figure shows the messages exchanged between edge router 1 and edge router 2 via the OBS core router, by means of either Et hernet (SIP) or the OBS testbed (grid job data).

The message flow is presented in Fig. 3. The user sends the job specification with an INVITE message to the proxy of the edge router. Encapsulated into the INVITE there is the JSDL document which describes the job requirement. The proxy reads the INVITE and performs a resource discovery algorithm and forward the message to the GSP2. After the INVITE is processed at GSP 2 the user is informed about the result of the resource discovery, with a positive reply message (200 OK), that is received by GSP 1 T_{S1} = 0.016 seconds (experimentally measured) after sending the INVITE message. In this case GSP 1 forwards the OK message to the client that will then set up the application to sent the job before acknowledging the session establishment. In this case a computational resource reservation signalling (ACK) message is sent to GSP 2 after T_{S2} = 0.041 s. The time elapsed between the arrival of the OK message and the departure of the ACK is due to the signalling between GSP 1 and the client (not reported in the figure) and of the set up of the application to transfer the data referring to the communication session under negotiation. Then the user sends the actual job to edge router 1 for transmission to the reserved resources attached to edge router 2.

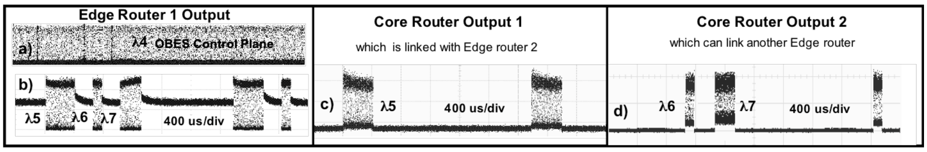

Fig. 4. Experimental results. a) OBES Control Plane, b) Asynchronously transmitted variable sized bursts, c) burst routed through core router towards edge router 2 and d) bursts routed towards emulated router.

Edge router 1 aggregates the job into optical bursts that are sent to the reserved resources by utilizing JIT bandwidth reservation scheme. The edge router is able to send data from different users to different reserved resources across the network as shown in Figure 4.b. Before that, a BCH has to be transmitted in order to setup the path (Figure 4.a) The first burst is routed through core router output port 1 (Figure 4.c) to be sent to edge router 2. The following two bursts are routed through core router output port 2 (and Figure 4.d) which is used as an additional link to emulate an additional edge router (e.g. edge router 3). The offset time required between BCH and Burst Data is 10.2 μs in order to incorporate core router processing time plus switch setup time (T_{C1} = T_{C1} = 200 ns) and optical crosspoint switch switching time (10 μs, Figure 4.d). Upon reception of the job, edge router 2 requires T_{E2} to read the incoming burst and emulate the forwarding to the resource and the processing time. Then it sends an emulated job result back to the user through the core router and edge router 1 (Figure 4.e).In this experiment, the data plane transmitted variable length optical bursts (from 60 μs up to 400 μs) with their associated BCHs over OBES control plane (Figure 4.a), on three different wavelengths (5=1538.94 - 6=1542.17nm- 7=1552.54 nm).

5 Conclusions

This paper proposes to introduce the concept of session in the control plane of a Grid enabled OBS network. To this end the Session Initiation Protocol (SIP) is used on top of the OBS network control layer (for instance based on GMPLS). SIP is used to establish application sessions according to the application requirements and then the network control plane is triggered to create the high bandwidth data paths, by properly controlling the OBS nodes. This approach in principle enriches the network with a number of application oriented features thanks to the rich session oriented semantic of SIP.

The paper analyzed the various architectural approaches of such solution and its possible development. Then an experimental set-up of an overlay architecture was implemented on an OBS test-bed to validate this approach. The results of the experiment reported in this paper prove the feasibility and effectiveness of the proposed solution for the control plane.

References

1. D. Simeonidou, et al., "Optical Network Infrastructure for Grid" Open Grig Forum document GFD.36, August 2004.
2. C. Qiao, M. Yoo, “Optical Burst Switching - A new Paradigm for an Optical Internet”, Journal of High Speed Networks, vol. 8, no. 1, pp. 36-44, Jan. 2000.
3. D. Simeonidou, et al., “Grid Optical Burst Switched Networks (GOBS),” Global Grid Forum Draft, May 2005.
4. D. Simeonidou et al. “Dynamic Optical-Network Architectures and Technologies for Existing and Emerging Grid Services”, IEEE Journal on Lightwave Technology, Vol. 23, No. 10, pp. 3347-3357, 2005.
5. R. Nejabati, D. Simeonidou, “Control and management plane considerations for service oriented optical research networks”, IEEE 8th International Conference on Transparent Optical Networks, June 18-22, 2006 - Nottingham, UK.
6. A. Anjomshoaa et al., “Job submission description language (JSDL) specification v. 1.0”, Open Grig Forum document GFD.56, November 2005.
7. J. Rosenberg et al., “SIP: Session Initiation Protocol”, IETF RFC 3261, June 2002.
8. E. Rosen, A. Viswanathan, R. Callon, “Multiprotocol Label Switching Architecture”, IETF RFC 3031, January 2001.
9. E. Mannie (Ed.), “Generalized Multi-Protocol Label Switching (GMPLS) Architecture”, IETF RFC 3945, October 2004.
10. M. Poikselka, et al., “The IMS: IP Multimedia Concepts and Services”, 2nd ed., Wiley, 2006.
11. Georgios Zervas, et.al., “QoS-aware Ingress Optical Grid User Network Interface: High-Speed Ingress OBS Node Design and Implementation”, OFC 2006, paper OWQ4, Anaheim, California, USA.
12. G. Zervas, et. al., “A Fully Functional Application-Aware OpticalBurst Switched Network Test-Bed”, to be published in OFC 2007, paper OWC2, Anaheim, California, USA.
13. Zhuoran Wang; Nan Chi; Siyuan Yu, Lightwave Technology, Journal of,Volume 24, Issue 8, Aug. 2006 Page(s):2978 – 2985.
14. http://www.pjsip.org