

# Network Virtualization, Control Plane and Service Orchestration of the ICT STRAUSS Project

R. Muñoz, R. Vilalta, R. Casellas, R. Martínez  
Optical Networks and Systems Department  
Centre Tecnològic de Telecom. de Catalunya (CTTC)  
Castelldefels (Barcelona), Spain

X. Cao, N. Yoshikane, T. Tsuritani  
Photonic Transport Network Laboratory  
KDDI R&D Laboratories Inc.  
Fujimino, Japan

L. M. Contreras, V. López, J.P. Fernández-Palacios,  
O. González de Dios  
Scalable Multilayer Photonics Networks  
Telefónica I+D  
Madrid, Spain

A. Autenrieth  
Advanced Technology  
ADVA Optical Networking  
Martinsried, Germany

S. Peng, M. Channegowda, R. Nejabati, D. Simeonidou  
High Performance Networks  
University of Bristol  
Bristol, United Kingdom

M. Schlosser  
Photonic Networks and Systems  
Fraunhofer Heinrich Hertz Institute  
Berlin, Germany

**Abstract**— Emerging cloud applications such as real-time data backup, remote desktop, server clustering, etc. require not only more traffic being delivered between datacenters, but also dedicated and application-specific virtual optical network (VON) services to support each application's QoS and SLA level. On the other hand, another requirement is to support end-to-end network service provisioning across multiple VONs comprising different transport (e.g. Flexi-grid DWDM OCS, OPS, etc) and control plane technologies (e.g., centralized OpenFlow or distributed GMPLS). This paper presents the preliminary architecture of the network virtualization, control and orchestration layers proposed in the STRAUSS project.

**Keywords**—transport network virtualization; OpenFlow; Active Stateful PCE; SDN network orchestration; datacenter connectivity.

## I. INTRODUCTION

In recent years, new high-performance Internet applications such as Cloud Computing and high-definition video streaming are emerging. These applications have a common requirement for a high capacity network infrastructure, which can be provided by optical networks. However, each of these applications has its own specific access and network resource usage patterns as well as quality of service (QoS), Service Level Agreement (SLA) and dynamicity requirements. Therefore, dedicated and application-specific optical network services are desired to support each application category. Optical network virtualization is a key technology for addressing this issue. It is able to partition and/or aggregate optical network resources into virtual resources, and then interconnect them together to compose multiple virtual optical networks (VONs) [1]. VONs can coexist using different network topologies and protocols and having their own specific QoS and SLA while sharing the same infrastructure.

In an optical network supporting network virtualization, each VON requires a control plane for the provisioning of dynamic, adaptive and fault-tolerant network services. Two control plane architectures are active subjects of research, namely GMPLS and OpenFlow. The GMPLS architecture is based on a distributed control plane (signaling, routing and link management), and has been extended to support delegating the path computation function to a path computation element (PCE) [2]. On the other hand, OpenFlow allows operators to control the network using software running on a logically centralized controller [3]. OpenFlow defines an open protocol that allows configuring a network device remotely.

When a physical infrastructure comprises heterogeneous optical transport (e.g., flexi-grid DWDM optical circuit switching -OCS- and optical packet switching -OPS-) and control plane technologies (e.g. centralized OpenFlow or distributed GMPLS), which do not naturally interoperate, an orchestration mechanism is required. Thereby, the orchestration mechanism allows the composition of end-to-end virtual transport infrastructures across different transport technologies as well as end-to-end network service provisioning across multiple VONs comprising different transport and control plane technologies. Software defined networking (SDN) is a key technology to address this requirement, since the separation of control and data planes makes the SDN a suitable candidate for end-to-end network service orchestration across multiple domains with heterogeneous control plane and transport technologies.

The ICT STRAUSS project addresses the above requirements by deploying a network virtualization, control and orchestration layers. This paper provides an overview of the preliminary architecture design and the main use case considered in the project.

## II. STRAUSS ARCHITECTURE

STRAUSS proposes a next-generation software defined optical Ethernet transport network architecture (Fig. 1), composed of four layers:

- An optical infrastructure layer to support Ethernet transport beyond 100 Gb/s by suitably combining the high-capacity flexi-grid OCS networks with flexible spectrum management and the high-throughput and statistically multiplexed OPS systems.
- A transport network virtualization layer which virtualizes the heterogeneous data plane resources. The physical infrastructure is partitioned and/or aggregated into virtual resources (i.e., virtual nodes and links), and virtual resources from different domains are selected to compose end-to-end virtual transport infrastructures.
- A control plane layer, employing both GMPLS and customized network control based on OpenFlow sits over each virtual transport infrastructure, providing independent control functionalities in order to handle both covered switching technologies (i.e., OPS and flexi-grid OCS).
- A service and network orchestration layer, using SDN-based orchestrator to enable the seamless interworking between GMPLS and OpenFlow control planes for the automatic provisioning of end-to-end Ethernet transport services spanning the targeted multi-layer and multi-domain network.

## III. DATACENTER USE CASE

In the considered use case (depicted in Fig. 2), distributed datacenter domains are interconnected by means of a (potentially virtualized) optical core transport infrastructure. The datacenter domains are controlled by OpenFlow and the optical core transport infrastructure is controlled by GMPLS. While OpenFlow is specially adapted to single domain intra-datacenter networks (packet level control, lots of routing exceptions), a standardized GMPLS based architecture would enable dynamic optical resources allocation and restoration in multi-domain (e.g., multi-vendor) core networks interconnecting distributed datacenters. In the datacenter

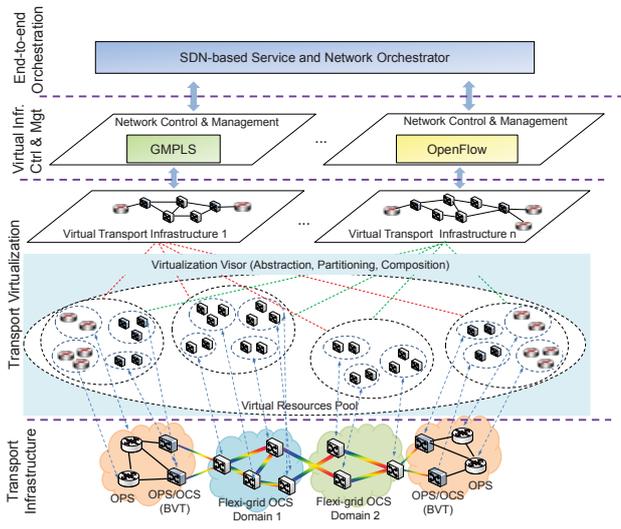


Fig. 1. STRAUSS network architecture

domains, conventional ToR Ethernet switches are attached to optical packet switching (OPS) elements. Each OPS node is used to efficiently aggregate the generated Ethernet data traffic into optical packets. The resulting optical packets form then the data flow that the bandwidth-variable transponders (BVT) equipped at the edge OPS core nodes will inject into the flexi-grid OCS network to be transported towards the remote datacenter site. The automatic control functions such as the provisioning within a domain, the equipment configuration, intra-domain path computation and domain traffic engineering (TE) decisions are performed by the control plane instances managing each network domain. To provide end-to-end control functions, it is necessary to have the interoperability and coordination provided by the orchestrator.

## IV. TRANSPORT NETWORK VIRTUALIZATION

### A. Preliminary architecture

The virtualization layer provides a mechanism for virtualizing transport nodes and links, by means of partitioning and aggregation techniques, and offering them as network slices. Partitioning enables physical infrastructure providers to partition their physical resources into multiple independent slices (virtual resources) with each virtual resource exactly mimicking functionality and performance of the real physical resource slices. The partitioning of the resources is technology dependent. Thereby, we propose specific technology abstraction mechanisms for the different Ethernet transport infrastructure resources (e.g., OPS, flexi-Grid optical networks).

The architecture for the virtualization of the transport network is shown in Fig.3. The STRAUSS project proposes a virtualization approach based on the concept of a Virtualization Visor (VV), an entity responsible for the virtualization of optical transport infrastructure domains. The proposed VV partitions each domain resources (i.e. links and nodes) into virtual optical resources. Later, the obtained virtual optical resources are composed into actual VONs, controlled by either a GMPLS or an OpenFlow control plane. The VV consists of two elements: a Virtualization Partitioner (VP), which can be distributed or centralized, and a Virtualization Composer (VC), which is responsible for creating virtual network slices, by composing the offered resources of the different VPs, upon request. A first implementation and experimental evaluation of the devised architecture has been performed in an international testbed across Spain (CTTC), UK (UNIVBRIS) and Japan (KDDI) [4].

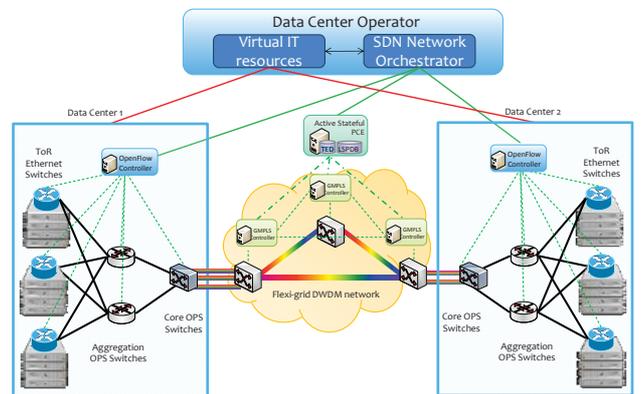


Fig. 2. Applicability of the STRAUSS architecture to datacenter connectivity

### B. VON resource allocation algorithms

In STRAUSS, the physical substrate is a multi-domain and multi-technology network comprising heterogeneous transport technologies (e.g. Flexi-grid EON, OPS). In order to create multiple coexisting but isolated VONs across such a multi-domain and multi-technology scenario, the VON resource allocation algorithms need to consider the information of physical networks (e.g. the topology, the inter-domain connectivity and the availability of physical resources such as switch ports and available optical spectrum slots within each domain) as well as the technology domain specific attributes and constraints (e.g. the spectrum continuity and physical impairments). The information will be stored in and retrieved from the global network Traffic Engineering Database (TED) module in the Virtualization Visor.

When a VON request is received from a client indicating the requested virtual topology (i.e., virtual nodes and links) and the required capabilities and capacities (e.g., virtual node switching capabilities and link bandwidth), the Resource Assignment module in the Virtualization Visor is called. This contains an algorithm bundle composed of multiple algorithms conceived for different scenarios (e.g., single/multi-domain flexi-grid EON, OPS, etc.). Once the required capabilities and capacities of the VON request are processed, the suitable algorithm in the bundle will be executed to allocate the resources in the targeted domain scenario, taking into account the physical resource availability information and the optical layer constraints. In the flexi-grid domain, the virtual nodes are mapped to the physical nodes, while the virtual links are mapped to the physical lightpath calculated by routing algorithms such as the *Shortest Path* and the *Load Balancing*. The size of the frequency slot in terms of slices (6.25GHz or 12.5GHz) for each virtual link can be calculated using the  $m=BW/6.25$  (or 12.5), where  $BW$  is the requested bandwidth of the virtual link. In order to set up an end-to-end feasible lightpath, the spectrum continuity needs to be ensured. Finally, an end-to-end "bandwidth corridor" (i.e., frequency slot) will be established. The frequency slot has a nominal frequency defined by  $193.1 + n \times 0.00625$  (THz) which is used to calculate the central frequency of a frequency slot while  $m \times 12.5$  GHz yields the slot width, where  $n$  is an integer and  $m$  is a positive integer.

The above mentioned methodology is adopted for dynamically serving VON requests on an one-by-one basis. If multiple VON requests are received or scheduled to be processed at the same time, an off-line optimal planning methodology will be adopted. After the algorithm bundle terminates, the computed resources for the VON request are sent to the Resource Configuration module in the Virtualization Visor for configuring the equipment in the corresponding

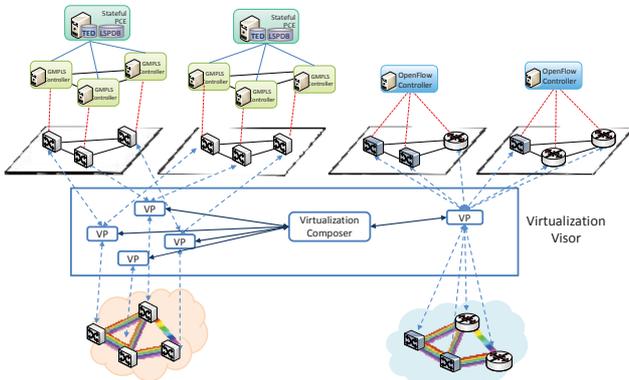


Fig. 3. Proposed Transport Network Virtualization layer architecture

domains.

## V. TRANSPORT TECHNOLOGY SPECIFIC CONTROL PLANE

### A. OpenFlow control plane for OPS networks

Fig. 4 shows the OpenFlow (OF)-based control for the OPS network. The OF-OPS network mainly comprises of a SDN Controller and OF-based OPS nodes. Incoming packets are classified and aggregated into flows at the border node and assigned with a unique label, which is included in the matching field of the extended OF Flowtable. The extended OF protocol is supported in the OF-OPS node, where the Flowtable is embedded to dictate the rules for the packet forwarding. Each OF-OPS node sends unmatched packets to the SDN Controller, where available resources are calculated and provided for each flow. For the following packets of this flow, a match against the label will be found in the Flowtable and packets will be switched accordingly. Thanks to the control and data plane decoupling provided by the OpenFlow protocol and SDN concept, traditional complex control functions (e.g., routing, resource allocation, etc.) can be offloaded to the centralized Controller, while the OPS node is only in charge of its own distributed Flowtable matching and packet forwarding. To decrease contention and also enhance network survivability, OF-OPS nodes will dynamically report their link/port status to the SDN controller for path recalculation and Flowtable adjustment. The above described OF-OPS node is not available yet. However, with the introduction of additional OpenFlow Agents (OFA) between the SDN Controller and the regular OPS nodes [5], the OPS network can be controlled via OF protocol. After the world's first research conducted by National Institute of Information and Communications Technology (NICT) for the interworking between an OF network and an independent OPS network [6][7], herein we explain how to directly control such an OPS network via the OF protocol. The used regular OPS nodes attach the OP\_ID (label) to packets according to its label-mapping table, and then forwards packets according to its own forwarding table. To configure both the label-mapping table and forwarding table via the OF protocol, the OFA virtualizes the OPS node and interacts with the Controller. Whenever the OFA receives a request from the Controller for a Flowtable modification, it abstracts the corresponding information (label, ports, etc.) and translates it into standard commands, which are sent to the OPS node for table configuration.

### B. OpenFlow control plane for flexi-grid DWDM networks

Traditional fixed grid networks use fixed channel spacing of 50 GHz or 100 GHz while in flexi-grid optical networks the frequency slot width is flexible (typically 6.25 GHz or 12.5

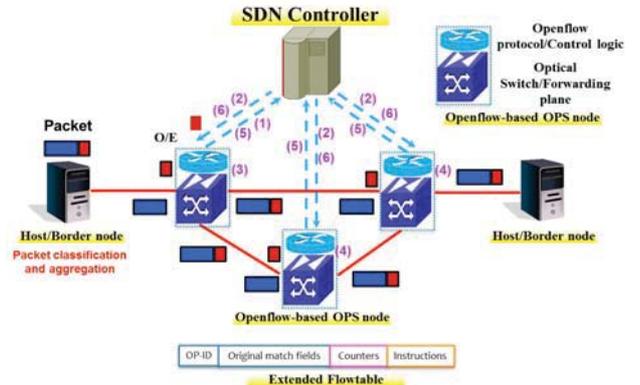


Fig. 4. OpenFlow – OPS network architecture

GHz each). This allows efficient, flexibly configurable and manageable optical spectrum for elastic bitrate systems. However, it also raises concerns in terms of fragmentation and routing which can be addressed by an intelligent control plane. The SDN-based control plane can provide a centralized global view of the network using vendor agnostic protocol such as OF. In order to control Flexi-grid networks in a SDN control paradigm the following main challenges need to be dealt with: 1) to redefine the circuit-oriented OF protocol for the new resource type, 2) to extend the flow concept to Flexi-grid networks, 3) to cater to its interactions with other technology domains (fixed grid, packet switched networks, etc.), and 4) to develop applications which can utilize the SDN control plane abstraction.

In STRAUSS, some actions have been taken to overcome the challenges namely: the flow definition in a Flexi-grid network, the extensions to the OF protocol and the design of a OF controller (based on POX). Such combined capabilities allow introducing Flexi-grid functionalities for seamless path computation between different technology domains.

In Flexi-grid optical networks, a flow is identified by a Flow Identifier comprising port, central frequency (CF), frequency slot bandwidth (FSB), modulation format and type of signal fields associated with the switch. The flows are processed based on OF specifications detailed in version 1.0. In the SDN/OpenFlow control paradigm for Flexi-grid optical networks, an OF-enabled device issues a Switch\_Feature message to describe the Flexi-grid switch capabilities that in turn are used by the controller to build the network topology and the associated constraints. Once the network properties are collected, based on the application request/demand, the controller issues a flow modification message called CFlow\_Mod to the devices in order to program the devices and in turn the network to accommodate the demand. The allowable granularity of  $m$  and  $n$  for Flexi-grid equipment and the available modulation format can be determined using the Switch\_Feature messages.

In order to implement the SDN paradigm to support the Flexi-grid optical networks, we have extended the

Switch\_Feature, CFlow\_Mod, CPort\_Status and Error messages of the circuit addendum of the OF protocol. The CFlow\_Mod messages are extended to support the Flexi-grid domain based on the ITU-T G.694.1 recommendation. To control both BVTs and BV wavelength cross connects (BV-WXC), only  $m$  and  $n$  values are exchanged between the controller and the device using the CFlow\_Mod messages. To utilize these extended messages, we have also designed and implemented an extended OF controller based on POX supporting packet switched networks, fixed grid and Flexi-grid optical networks.

### C. GMPLS control plane for Flexi-grid DWDM networks

An active stateful Path Computation Element (PCE) is a key element facilitating the use of dynamic and adaptable mechanisms within GMPLS-based distributed control plane for flexi-grid DWDM networks. Additionally, the active stateful PCE enables the standardized deployment of the GMPLS control plane within the SDN-based orchestration architecture. A stateful PCE allows for optimal path computation considering both the network state (TED) and the global connection (Label Switched Path – LSP) state in a repository referred to as LSPDB [8]. Furthermore, an active stateful PCE can control the state (e.g., increase of bandwidth, rerouting) of the stored LSPs, and expose the capability of setting up and releasing new LSPs. It is known as active stateful PCE with instantiation capabilities.

The particular orchestration of active stateful PCEs from an SDN Controller becomes an opportunity which is motivated by the following reasons:

- A flexible, mature and feature-complete protocol (PCE communication protocol, PCEP) acting as a SDN southbound interface. The addition of LSP instantiation capabilities to the PCEP protocol allows the end-to-end LSP provisioning within the domain controlled by the active stateful PCE.
- The leverage between network programmability and key GMPLS benefits such as short restoration times or distributed LSP provisioning.
- The PCE has also been extended and adapted to become the entity responsible for dynamically and, upon request, allocate network resources for virtual VON as detailed in [9].

## VI. SDN-BASED CROSS-LAYER AND CROSS-TECHNOLOGY ORCHESTRATION

### A. Preliminary SDN orchestrator architecture

Most of the solutions for SDN are based on a single domain comprising unique vendor network solutions. However, network operators usually rolled out multiple technologies (provided by different vendors) in their networks which are fragmented into multiple domains to cope with administrative and regional organizations. A single SDN controller cannot configure the whole network of an operator for scalability and reliability issues. This is even more complicated when

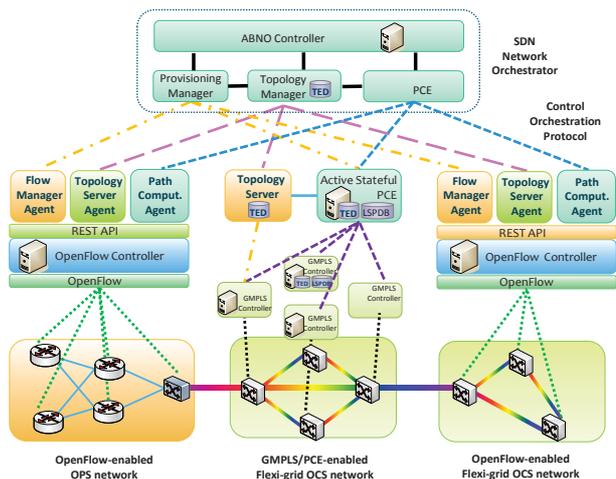


Fig. 5 Proposed SDN network orchestration for multiple domains with heterogeneous transport and control planes technologies

considering and architecture that should deal with OpenFlow and GMPLS-controlled domains at the same time.

The reference architecture for a controller follows the Application-Based Network Operations (ABNO) architecture which is being defined by IETF based on standard building blocks [10]. This approach is well aligned with operators' objectives since it allows the implementation of each block by separated third parties, and enables the reutilization of specific blocks (e.g., PCE) already applied in some scenarios. Moreover, the utilization of the PCE facilitates the path computation in multi-domain scenarios as targeted in STRAUSS. Fig. 5 presents four building blocks of the ABNO architecture which are required to support the inter-datacenter use case considered in this project. The *ABNO Controller* runs the workflows and can interwork with the different blocks, while the PCE computes the paths through different domains. The view of the PCE can be either the physical or an abstracted network. Moreover, the PCE is able to interoperate with each of the peer PCEs in the domains to build an end-to-end path. The *Topology Module* is in charge of retrieving the information of the network status. There are several protocols that are suitable for this interface depending on the scenario. The work to be done in the project is to select which is the interface to be used for our use case. Finally, the *Provisioning Manager* configures the network elements depending on the control plane technology that the device supports.

Regarding the relation between the controllers, two main approaches have been considered to solve the multi-control plane problem of the project. The first one is to define a hierarchy of controllers, where there are multiple SDN controllers interacting with a SDN orchestrator hierarchically placed on top of them. A second approach is to have multiple controllers at the same hierarchical level and with a westbound interface to enable the communication of the specific controllers for each of the multiple technologies or domains. The interaction of controllers of the same level requires the definition of some of the functionalities of such westbound interface. The relationship between the SDN controllers could be similar to the one shown by the interactions of the users and the SDN controller. Each controller requires having a mechanism to ask for connections or topological information. The users may not be able to request paths to a SDN controller, but this functionality is required between controllers to enable end-to-end path optimization.

### B. Control Orchestration Protocol

The main objective is to define a new interface and protocol that abstracts the particular control plane technology of a given domain. In this sense, the proposed architecture applies the same abstraction and generalization principles that OpenFlow/SDN have applied to data networks: much like OpenFlow identifies an abstracted, generic model of packet switch that can be used regardless of a particular vendor or technology, and provides a protocol (OpenFlow) to query and set its forwarding state. The project defines a generic functional model of a "control plane" for the provisioning of connectivity and defines an associated protocol (the Control Orchestration Protocol). The orchestrator function operates under the assumption that each domain is composed of a (potentially

virtualized) data plane controlled by an instance of a given control plane technology, but transport and/or control plane technologies for each domain can be different. The main functionalities of the orchestrator are abstract and not technology related. However, specific "agents" need to be developed (on a technology-specific basis) to map the abstracted control plane model into the specifics of the underlying control plane technology. In other words, the agent will act as a protocol gateway, interpreting orders (and updating status) from the orchestrator (and back) and will apply the involved control plane procedures. The actual placement of such per-technology agent is dependent on the underlying control plane instance (e.g., collocated in controllers in the GMPLS case or in the OpenFlow controller – Fig. 5). This control plane abstraction must enable the provisioning of data services using the underlying configuration technology, and could typically address several main blocks, once the requirements are clearly identified, notably a) network resource discovery, including topology management, adaptation and virtualization, while providing isolated and secure access to the underlying hardware (topology server); b) connectivity provisioning (flow manager) and c) path computation.

## VII. CONCLUSIONS

The STRAUSS architecture addresses the provisioning of end-to-end network services across multiple domains with heterogeneous transport and control plane technologies. The proposed architecture applies new SDN principles to enable cost reduction and reduced time to market of new services.

## ACKNOWLEDGMENT

Work funded by EC FP7/2007-2013 (G.A. 608528) and by Japanese MIC and NICT through EU-Japan project STRAUSS and Spanish MINECO project FARO (TEC2012-38119).

## REFERENCES

- [1] S. Peng, et al., Application-aware and Adaptive Virtual Data Centre Infrastructure Provisioning over Elastic Optical OFDM Networks, in Proc. of 39th ECOC 2013, September 22-26 2013, London, U.K.
- [2] R. Muñoz, et al., PCE: What is It, How Does It Work and What are its Limitations? , IEEE/OSA Journal of Lightwave Technology, Vol. 32, No. 4, pp. 528 - 543, February 2014.
- [3] M. Channegowda, et al., Experimental demonstration of an OpenFlow based software-defined optical network employing packet, fixed and flexible.. , Optics Express , Vol. 21, No. 5, pp. 5487-5498, March 2013.
- [4] R. Vilalta, et al., Dynamic Multi-domain Virtual Optical Networks Deployment with Heterogeneous Control Domains, in Proc. of Optical Fiber Conference (OFC), San Francisco, March 2014.
- [5] Renaud M., et al., " Networkand System Concepts for Optical Packet Switching," *IEEE Comm. Magazine.*, Vol.35, No.4, pp. 96-102, 1997.
- [6] T. Miyazawa, "Architecture for Interworking between an Optical Packet & Circuit Integrated Network and OpenFlow- based Networks," iPOP 2013, Tokyo, Japan, May 2013.
- [7] H. Harai, "Optical Packet and Circuit Integrated Networks and SDN Extensions," ECOC 2013, Mo.4.E.1, September 2013
- [8] E. Crabbe, et al., "Pcep extensions for stateful pce," IETF draft-ietf-pce-stateful-pce (work in progress), October 2013.
- [9] R. Vilalta, et al., "Dynamic virtual gmpls-controlled wson using a resource broker with a vnt manager on the adrenaline testbed," Optics Express, vol. 20, no. 28, pp. 29 149– 29 154, December 2012.
- [10] D. King et al, A PCE-based Architecture for Application-based Network Operations", IETF draft-farrkingel-pce-abno-architecture, Feb 2014