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

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

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

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

A. Autenrieth
Advanced Technology
ADVA Optical Networking
Martinsried, Germany

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]. These 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 centralized controller [3]. OpenFlow defines an open protocol that allows configuring the behavior of 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 to allow 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 plane and data plane makes the SDN a suitable candidate for end-to-end network service orchestration across multiple domains with heterogeneous and incompatible control plane and transport technologies.

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

II. STRAUSS ARCHITECTURE

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

- An optical infrastructure to support Ethernet transport beyond 100 Gb/s, 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 Virtual Infrastructure control plane, employing 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, multi-domain network.

III. DATACENTER USE CASE

In the considered use case (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., multivendor) core networks interconnecting distributed datacenters. In the datacenter 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.

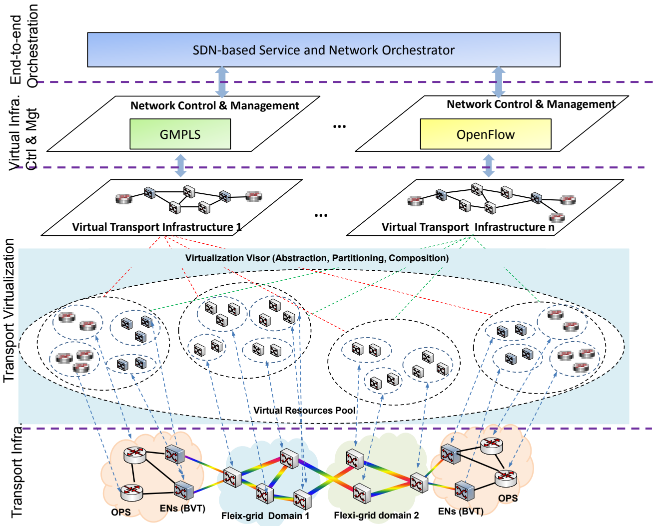

Fig. 1. STRAUSS network architecture

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, and to this end, 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 experimental evaluation of the devised architecture has been performed in an international testbed across Spain (CTTC), UK (UNIVBRIS) and Japan (KDDI) [4].

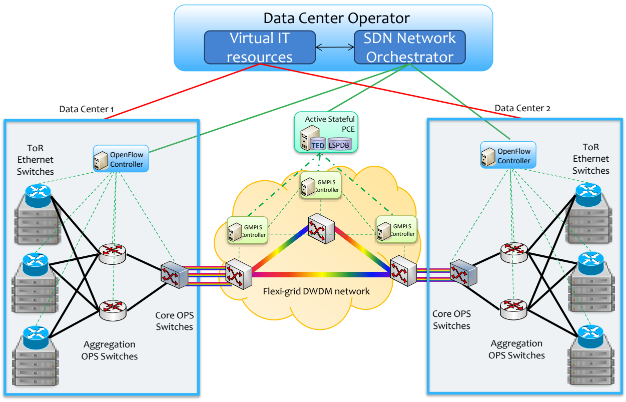

Fig. 2. Applicability of the STRAUSS architecture to datacenter connectivity

B. *VON resource allocation algorithms*

In STRAUSS, the physical substrate is a multi-domain 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 multi-technology scenario, the VON resource allocation algorithms need to consider the information of physical networks (e.g. the network topology, the inter-domain connectivity, and the availability of physical resources such as switch ports or spectrum slots of each domain) as well as technology domain specific attributes and constraints (e.g. spectrum continuity and impairments). The information will be stored in and retrieved from the Global Network TED module in the Virtualization Visor.

When a VON request is received from a client, indicating the requested virtual topology comprising virtual nodes & virtual links and also specifying the required capabilities and capacities (e.g. virtual node switching capabilities and virtual link bandwidth), the Resource Assignment module in the Virtualization Visor will be called, which contains an algorithm bundle composed of multiple algorithms designed for different scenarios (e.g. single domain/multi-domain Flexi-grid EON, OPS). After the required capabilities and capacities of the VON request are analyzed, 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 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 *Shortest Path* and *Load Balancing*. The required number of frequency slots (6.25GHz or 12.5GHz each) for each virtual link can be calculated using the $m=BW/6.25$ (or12.5), 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 specially taken into account. Finally, an end-to-end "bandwidth corridor" with m frequency slots and central frequency $193.1 + n\times0.00625$ (THz) will be established.

The abovementioned methodology is adopted for the on-line dynamic VON requests, that is, the VON request is received and dealt with one by one. If multiple VON requests are received or scheduled to be processed at the same time, the off-line optimal planning methodology will be adopted. After the algorithm bundle is called, the suitable to-be-allocated resources for the VON request will be sent to the Resource Configuration module in the Virtualization Visor for configuring equipment in corresponding domains.

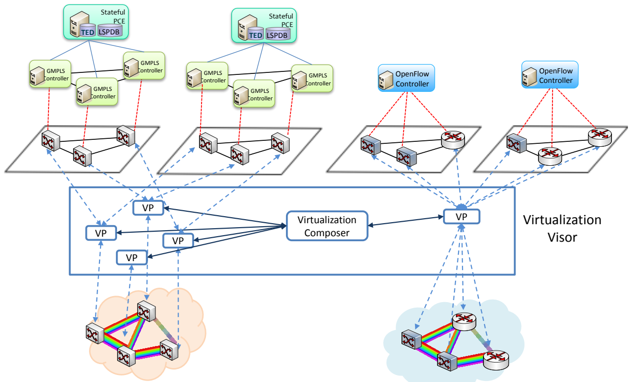

Fig. 3. Proposed Transport Network Virtualization layer architecture

V. TRANSPORT TECHNOLOGY SPECIFIC CONTROL PLANE

A. *OpenFlow control plane for OPS networks*

Fig. 4 shows an OpenFlow (OF) - Optical Packet Switch (OPS) network architecture. An OF-OPS network mainly comprises of a SDN Controller and Openflow-based OPS nodes. Due to the flow-based Openflow control, 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 Flowtable. Extended Openflow protocol is supported in the OF-OPS node, where Flowtable is embedded as the rule for packet forwarding. Each OF-OPS node will send unmatched packets to the 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 according to the instructions. Thanks to Openflow, traditional complex control functions (routing, resource allocation, etc.) can be offloaded to the centralized Controller, while OPS node is only in charge of 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. Actually, unfortunately, such an OF-OPS node described above is still not available at this moment. However, with the introduction of additional Openflow Agents (OFA) between SDN Controller and the regular OPS nodes [5], OPS network can still be controlled via Openflow. After a pioneer research conducted by National Institute of Information and Communications Technology (NICT) on interworking between Openflow network and an independent OPS network [6][7], here we explain how to control such an OPS network directly via Openflow. The regular OPS node we used attaches OP_ID (label) to packets according to its label-mapping table and forwards packets according to its own forwarding table. In order to configure its label-mapping table and forwarding table via Openflow, the OFA virtualizes the OPS node and interacts with the Controller. Whenever OFA receives a request from the Controller for 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 configurations.

B. *OpenFlow control plane for Flexi-grid DWDM networks*

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

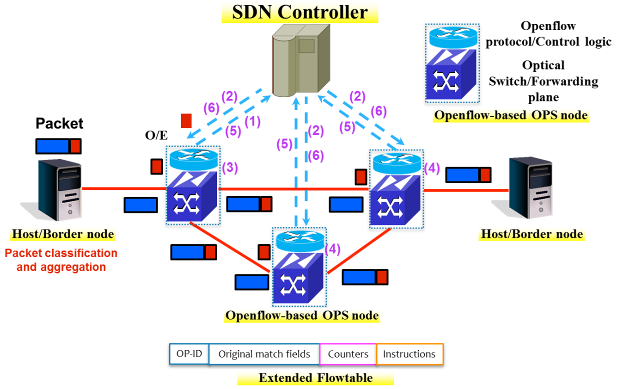

Fig. 4. OpenFlow – OPS network architecture

GHz each). This allows efficient, flexibly configurable and manageable optical spectral bandwidth 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 Software Defined Networking (SDN) based control plane can provide a centralized global view of the network using vendor agnostic protocol such as OpenFlow (OF). In order to control Flexi-grid networks in a SDN paradigm the following main challenges need to be dealt with: 1) to redefine the circuit 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 also 4) to develop applications which can utilize the SDN control plane abstraction.

In STRAUSS, some actions have been taken to overcome the challenges, such as the flow definition in a Flexi-grid network, the extensions to the OF protocol and the design of a OF controller (POX), which together can introduce 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. The frequency slot has a nominal frequency defined by 193.1 + n×0.00625 (THz) which is used to calculate the central frequency of a frequency slot while m×12.5 GHz yields the slot width, where n is an integer and m is a positive integer. In the SDN/OpenFlow 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.

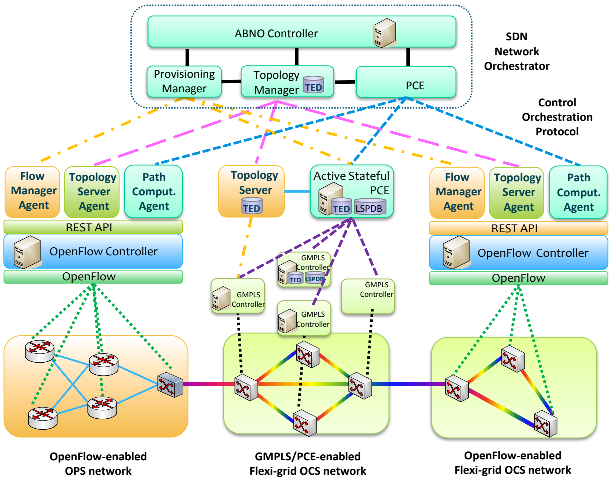

Fig. 5 Proposed SDN network orchestration for multiple domains with heterogenous transport and control planes technologies

In order to implement the SDN paradigm to support the Flexi-grid optical network, we have extended the Switch_Feature, CFlow_Mod and CPort_Status, Error messages of the Circuit addendum of 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 a BVT or BV-WXC, only m and n values are exchanged between the controller and the device via CFlow_Mod messages. In order 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 for the introduction of dynamics and adaptation in GMPLS-based distributed control plane for flexi-grid DWDM networks, as well as for enabling the standardized deployment of the GMPLS control plane in the SDN-based orchestration architecture. A stateful PCE allows for optimal path computation considering both the network state (TED) and the global LSP state (LSPDB) [8]. Additionally, and 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.

We propose an active stateful PCE as a key enabler for Optical Transport SDN. 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 (e.g., such as short restoration times or distributed LSP provisioning).
- The PCE has also been extended and adapted in order to become the entity responsible for dynamically, and upon request, allocate network resources for virtual optical networks (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 single domain and mono vendor solutions. However, network operators usually have in place multiple technologies (provided by different vendors) in their networks and 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 considering and architecture that should deal with OpenFlow and GMPLS domains at the same time.

The reference architecture for a controller follows the Application-Based Network Operations (ABNO) architecture which is being defined in IETF based on standard building blocks [10]. This approach is well aligned with operators' objectives, because it allows the implementation for each of the different blocks by different third parties, and it enables the reutilization of some of such blocks (like the PCE) that is already active in some scenarios. Moreover, the utilization of the PCE facilitates the computation of a path in multi-domain scenarios, like the scenario for this project with domains controlled by different technologies. 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 it can to talk with the different blocks, while the PCE computes the paths between the different domains. The view of the PCE can be the physical network or an abstracted network. Moreover, the PCE is able to talk with each of the 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 to have a method to ask for connections or topological information. The users may not be able to request for 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 (the OpenFlow protocol) 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 works 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, but specific "agents" must be developed (on a technology-specific basis), in order 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 collocated 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

The research leading to these results has received funding from the EC FP7/2007-2013 under grant agreement nº 608528 and by the Japanese MIC and NICT through the EU-Japan coordinated project STRAUSS.

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-l02, l997.

[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.