The SDN/NFV Cloud Computing Platform and Transport Network of the ADRENALINE Testbed

Ricard Vilalta, Arturo Mayoral, Raul Muñoz, Ramon Casellas, Ricardo Martínez
Optical Networks and Systems Department
Centre Tecnològic de Telecomunicacions de Catalunya, CTTC
Av. Carl Friedrich Gauss 7, 08860 Castelldefels, Spain
Email: ricard.vilalta@cttc.es

***Abstract*—This work extends NFV paradigm to transport networks, known as Transport NFV. This paper presents a detailed overview of the SDN/NFV services that are offered on top of the Cloud Computing platform and transport network of the ADRENALINE Testbed. On the one hand, we propose a generic architecture for SDN/NFV services deployed over multi-domain transport networks and distributed data centers. On the other hand, we present two use cases of possible NFV services: a virtual Path Computation Element (vPCE) and the deployment of virtual SDN controllers (vSDN) on top of virtualized transport networks.**

I. Introduction

Network Functions Virtualization (NFV) aims at using Information Technology (IT) virtualization techniques to virtualize entire classes of network node functions. A Virtualized Network Function (VNF) consists of a network function running as software instances running on commodity servers, instead of having custom hardware appliances [1]. Examples of VNFs include load balancers, firewalls, security or Authentication, Authorization and Accounting (AAA) network functions.

Software Defined Networks (SDN) has emerged as the most promising candidate to improve network programmability and dynamic adjustment of the network resources. SDN proposes a centralized architecture where the control entity (SDN controller) is responsible for providing an abstraction of network resources through programmable Application Programming Interfaces (API). One of the main benefits of this architecture resides on the ability to perform control and management tasks of different network forwarding technologies such as packet/flow switching, circuit switching (e.g., optical wavelength switched transport), by means of the same network controller. The OpenFlow (OF) protocol is the most commonly deployed protocol for enabling SDN [2].

NFV and SDN are a key concepts to understand the evolving of current networks. In this regard, the main objective pursued by telecommunication operators is to gradually adopt the innovations carried out in the IT industry in the last ten years. The objective can be simplified into achieve the separation between infrastructure (hardware) and operational resources (software) by means of partitioning the infrastructure resources and virtualization of the operational resources.

Several innovations in the telecommunications industry have been done in the last years since the developing of the SDN and NFV concepts. New software-based switching solutions for packet based networks had been implemented ready to be installed in non-dedicated hardware. This had led into a great improvement in the network flexibility, allowing multi-tenancy, and service isolation.

The European Telecommunications Standards Institute (ETSI) formed an industry group to define the NFV architecture [1]. In the proposed view, Ethernet connectivity is expected and virtual networks might be deployed using classical virtualization mechanisms, such as VLANs. This limited view of the network, does not take into account the necessary specifics of transport network (e.g., typically optical networks). The efforts to provide programmability to transport networks via SDN architectures have been successfully demonstrated [5]. NFV can benefit from the provided network programmability, in order to take into account the needs of network specific functions (e.g., in terms of latency and throughput). The usage of SDN/NFV in transport networks will break the typical usage of a transport network as flat pipes.

Is in this context, we present a NFV Management and Orchestration (NFV MANO) architecture, in order to deploy VNFs on top of an integrated IT and Network orchestration platform, which can be located in different geographically distributed NFV Infrastructure Points of Presence (NFVI-PoP). These locations can be understood as NFV-enabled Data Centers (DC). These NFVI-PoP may be interconnected using multi-domain and multi-technology transport networks. There is the need for offering this IT and Network resources as a whole, by means of an SDN IT and Network Orchestrator (SINO), which acts as a Virtualized Infrastructure Manager (VIM). The SINO which is enabled with the flexibility provided with an underlying transport network. Several research projects, such as T-NOVA [3] and UNIFY [4] are proposing similar architectures.

This paper presents the SDN/NFV cloud computing platform and transport network of the ADRENALINE Testbed. It provides NFV services thanks to an integrated orchestration of IT and network resources, in order to provide intra/inter-DC network connectivity and deploy Virtual Machines (VMs) using OpenStack cloud computing system. The paper is organized as follows: in Section II, we describe the implementation on our testbed of the proposed architecture for offering NFV services upon distributed DCs. In Section III, we propose two use cases, which have been validated, on top of the testbed. Finally, we conclude the paper in Section IV.

978-1-4799-7899-1/15/$31.00 © 2015 IEEE

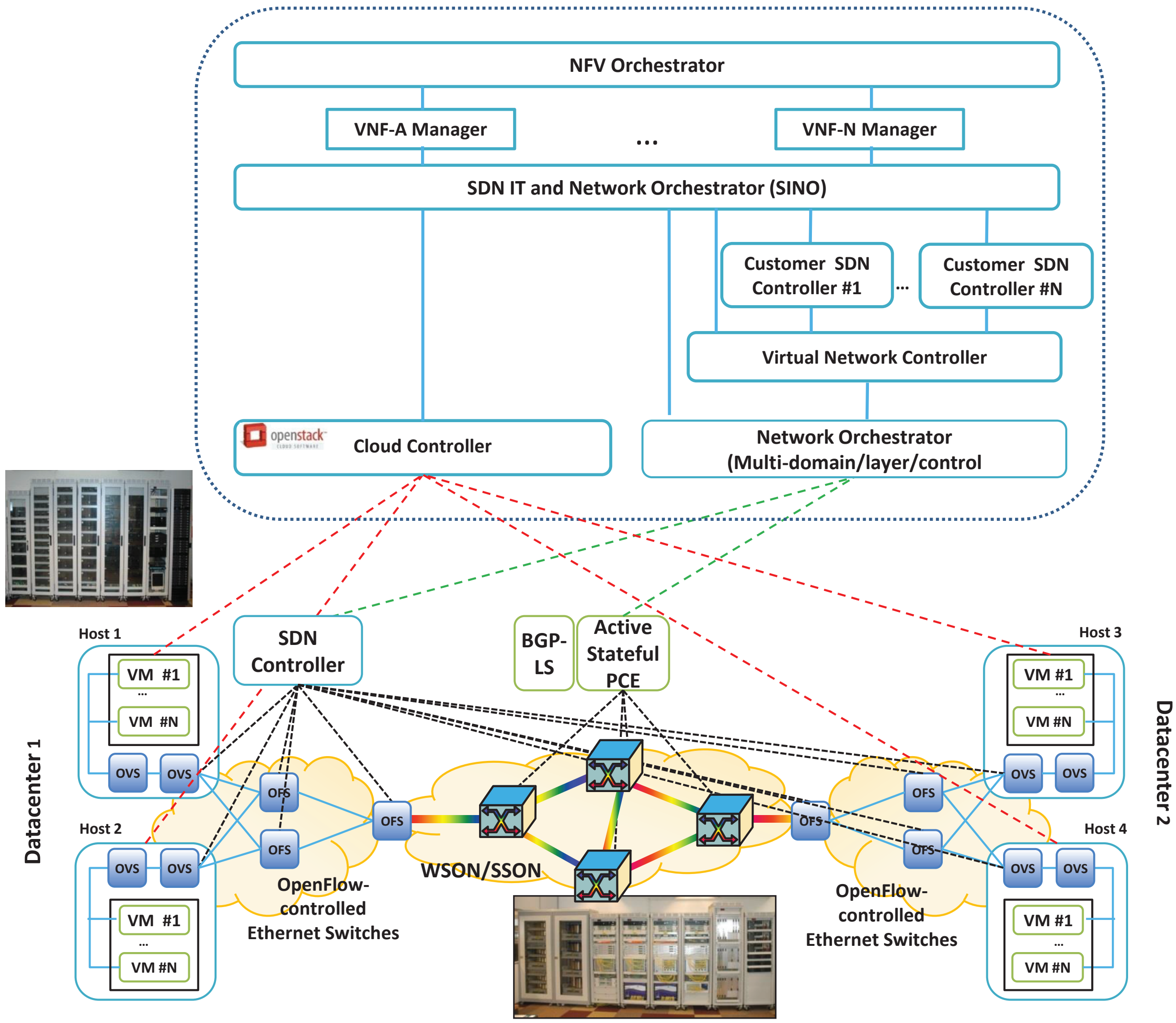

Fig. 1: The SDN/NFV Cloud Computing platform and Transport Network of the ADRENALINE Testbed

II. The SDN/NFV Cloud Computing platform and Transport Network of the ADRENALINE Testbed

This section details the implementation of the proposed Transport SDN/NFV architecture in the cloud computing platform and transport network of the ADRENALINE Testbed. We start describing the testbed in detail, including hardware and software. Later, the design of the different proposed components for the Transport SDN/NFV architecture is presented.

A. Testbed description

The cloud computing platform and transport network of the ADRENALINE Testbed (Fig. 1) is an experimental testbed at CTTC premises in Castelldefels (Barcelona, Spain).

For the cloud-computing platform, we have deployed OpenStack Havana into five physical Commercial Off The Shelf (COTS) servers with 2 Intel Xeon E5-2420 processors, 32GB RAM each, and 2 TB. One server is dedicated to the OpenStack controller and the other four act as OpenStack Compute Hosts for VM instantiation. Two availability zones have been defined in order to obtain distributed data centers, which are interconnected.

For the intra-data center networks, four OpenFlow switches have been deployed using standard COTS hardware with multiple 1Gb Network Interface Cards (NICs) and running OpenVSwitch (OVS), which is an OpenFlow software switch. Each Data Center border switch is implemented using COTS hardware with several 1G NICs and a 10 Gb/s XFP tunable transponder, and OVS.

Finally, the inter-data center interconnection is provided through a GMPLS-controlled optical network. This is composed of an all-optical WSON with 2 ROADMs and 2 OXCs providing re-configurable (in space and in frequency) end-to-end lightpaths, deploying a total of 610 km of G.652 and G.655 optical fiber, with six DWDM wavelengths per optical link.

The intra-data center network is controlled with OpenDayLight (ODL) SDN Controller, Hidrogen Service Provider release. The Active Stateful Path Computation Element (AS-PCE) is responsible for the inter-data center network connectivity. An AS-PCE is a PCE which maintains not only the traffic engineering information (link and node states), but also the state of the active connections in the network [6]. The AS-PCE is granted to manage the active paths controlled by the nodes, allowing the PCE to modify or tear down the established connections. More details on the AS-PCE might be found in [7].

B. Network Orchestrator

The Network Orchestrator (NO) is introduced in order to support end-to-end connectivity by orchestrating the different network domains through per-domain SDN/OpenFlow or GMPLS/PCE controllers. The NO must take into account the heterogeneous underlying network resources (e.g., multi-domain, multi-layer and multi-control). The NBI of a Physical SDN Controller (PSC) are typically technology and vendor dependent, so the NO shall implement different PSC plugins for each of the NBI. It is assumed that the PSCs are able to provide network topology information and flow programming functions.

The NO architecture is based on the proposed Application-based Network Operations (ABNO) [8], and has been validated for multi-layer multi-domain network orchestration in [9].

The Network Orchestration Controller is the component responsible for handling all the processes involved and to provision end-to-end connectivity services. It also exposes a NBI to offer its services to applications, such as the VNC.

C. Virtual Network Controller

Each customer (i.e., tenant) network is handled by a Customer SDN Controller (see Fig. 1), which is a SDN controller run by a virtual network customer for controlling its own deployed virtual network. The Virtual Network Controller (VNC) is the responsible for providing the abstraction and virtualization of the underlying network resources. The VNC is a network hypervisor, which is introduced to dynamically deploy multi-tenant virtual networks on top of an abstract network provided by the Network Orchestrator (NO).

The proposed VNC system architecture has been described in [10]. Four hierarchical control levels are identified: Customer SDN Controller (CSC), VNC, NO, and multiple PSC. A CSC is a SDN controller run by a Virtual Network (VN) customer for controlling its deployed VN. As mentioned before, a PSC is the centralized instance of control in charge of a physical infrastructure (i.e., SDN controller or an AS-PCE).

The VNC is responsible for receiving VN requests, processing them and allocating physical resources. Moreover, the VNC is responsible for the mapping between the allocated physical resources and the abstracted resources that are offered to the CSCs, and the control of such abstract networks, acting as a proxy for the OF protocol between a CSC and the underlying PSC. The partitioning of the resources is performed by the VNC, and to this end, the proposed system architecture relies on the NO, which provides a generic network abstraction mechanism for the different transport infrastructure resources (e.g., Ethernet, flexi-grid DWDM).

Each tenant is able to request a VN. Once the VN has been correctly setup, the CSC acts as a standard SDN controller where the controlled VN is an abstracted slice of the different allocated physical resources, which are managed by their corresponding PSCs.

D. Cloud Orchestrator

The Cloud Orchestrator takes over the creation/migration/deletion of VM instances (computing service), disk images storage (image service), and the management of the VM network interfaces (networking service). The Cloud Orchestrator handles multiple Cloud Controllers in a federation of multi-cloud testbeds. In our testbed a single Cloud Controller based on OpenStack is used.

The computing service (e.g., Nova in OpenStack) manages the VM into the compute hosts (Hosts 1-4 in Fig. 1). A compute service agent is running in each host and controls the computing hypervisor (KVM) used for the creation/deletion of the VMs. The image service (e.g., Glance in OpenStack) handles the disk images which are used as templates for VM file systems. The connectivity between VMs and virtual switches within the hosts is managed by the networking service (e.g., Neutron in OpenStack). It creates the virtual interfaces, attaches them to the virtual switches and offers a DHCP service for the VMs to get the assigned IP address.

The SINO controls the Cloud Orchestrator through a RESTful API, used to both trigger the Cloud Orchestrator actions and get the necessary information about the running VM instances. In our testbed, we have modified Neutron OVS plugin to provide full control of the OVS to the SDN Controller, through the NO, and disabling reactive packet-in mechanism for incoming unknown connections.

E. SDN IT and Network Orchestrator

The SINO is the responsible of handling the different IT and Network resources. Virtualization of IT resources is provided by means of a Cloud Orchestrator. The network resources are handled by means of a NO, which orchestrates the different network domains (e.g., packet and optical domains).

The SINO provides the following services: VM Create, Read, Update, Delete (CRUD) mechanism; Network CRUD mechanism; and VM migration.

The VM CRUD mechanism allows, via a REST API, to create, read, update or delete a VM. A VM might be requested based on its availability zone, its hardware resources (i.e., flavor), or the disk image to be loaded. A VM is also allocated inside a network. A second management network is provided by default to the VM to provide management access to the VM by the different applications (e.g., VNF Managers).

The network CRUD mechanism allows to create, read, update or delete a L3 network, which might include a valid IP range, from which an IP address is assigned to a VM NIC. To create the network, end-to-end paths are requested between each VM attached to the network.

The process to provide an end-to-end path between two VMs is as foolows. The SINO issues flow requests between VM1 and VM2 to NO. After computing the route, the NO is aware of either the positive reachability of the computing resources through the packet network (intra-DC) or whether an inter-DC connection is needed. In the first case, the NO is ready to send the command to the SDN Controller to establish the forwarding rules to the OVS Host switches and into the intra-DC switches. In the second case, the NO needs to establish an optical connection between the DCs, via AS-PCE PCInitiate message. When the optical connection has been established, the SDN Controller is is able to discover the new L2 link established between the DCs. A new path computation will be triggered and its results will derive into the establishment of the necessary forwarding rules to the SDN controller.

Different types of VM migration exist, being the most common live and block migrations. The former allows moving a VM without interrupting the processes running inside it. The latter involves a disruption service time wherein the VM is stopped. VMs do not run isolated. Typically, a VM is connected with other VMs in the same network to offer a joint service. If one of the VMs from a network is migrated, its connection state must be maintained, which is known as a VM seamless migration, and is performed by the SINO [11].

F. *VNF Manager*

The VNF manager is responsible for the lifecycle management (i.e., creation, configuration, and removal) of a Virtual Network Function. The VNF manager design and definition are being discussed at standardization bodies. We propose a simple design, based on the necessary functionalities to be covered in general by a VNF manager.

The VNF controller offers the interface to NFV Orchestrator to control the VNF. The Virtual IT resources component interacts with the SINO to obtain the necessary IT and network resources to deploy a VNF. In order to control the VNF, the VNF Manager has also access to the obtained IT resources (e.g., the VNF Manager has access to the allocated VM). Finally, VNF life cycle component manages the life cycle of the VNF, monitors it and notifies of incidences to the VNF controller.

We have implemented several VNF Managers (see the proposed use cases in Section III) as python scripts which interact with the SINO.

G. *NFV Orchestrator*

The NFV Orchestrator is defined by the ETSI as the responsible for managing the life cycle of the physical and software resources to support the infrastructure virtualization and the life cycle of the different VNFs [1]. The NFV Orchestration design and definition are subject to definition at standardization bodies, and still at discussion.

In our testbed, we have implemented python scripts in order to request the necessary VNF managers depending on the proposed use case. A mechanism for providing SFC will be incorporated.

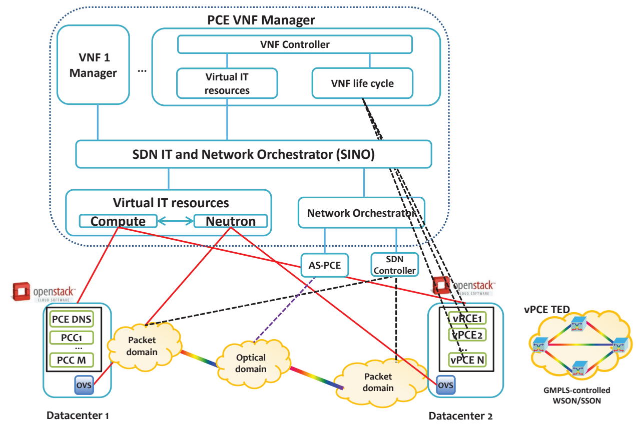

Fig. 2: Virtual PCE use case

III. Use Cases

After having presented the proposed SDN/NFV transport network architecture, and described it in the context of the SDN/NFV cloud computing platform and transport network of the ADRENALINE Testbed, we propose two use cases that consider the whole architecture.

The first use case focuses on the PCE, and how it can be offered in a NFV context. The second use case provides the foundations for offering multi-tenant virtual SDN controller (vSDN).

A. *Use Case: vPCE*

A transport Path Computation Element (PCE) is a transport network function, which is able to perform constrained path computation on a graph representing a network (TED). The PCE can be run as an application on top of COTS equipment. The initial driver for the deployment of PCEs was its use in multi-domain networks with limited visibility and, in the scope of optical transport networks, which increase the complexity of path computation.

In this use case, we propose the adoption of the NFV architecture to deploy a PCE dedicated to path computation of a transport network as a VNF. A PCE VNF Manager is introduced, so that the proposed transport PCE NFV is able to handle intense peak loads of path computation requests.

Figure 2 shows how the use case fits in the previously presented SDN/NFV transport network architecture. The PCE NFV Manager dynamically deploys virtual PCEs (vPCEs) on demand to keep the quality of the VNF, in terms of latency, request processing time, or dedicated algorithms. A vPCE is a PCE instance, which is run as a software application on a cloud computing environment (e.g., a virtual machine). A PCE DNS is introduced in order to offer the deployed vPCEs as a single VNF perceived by the different Path Computation Clients (PCC).

A PCC can use DNS to discover a PCE only when it needs to compute a path and does not require any other node in the network to be involved. In case of an intermittent PCEP session, which are systematically opened and closed for

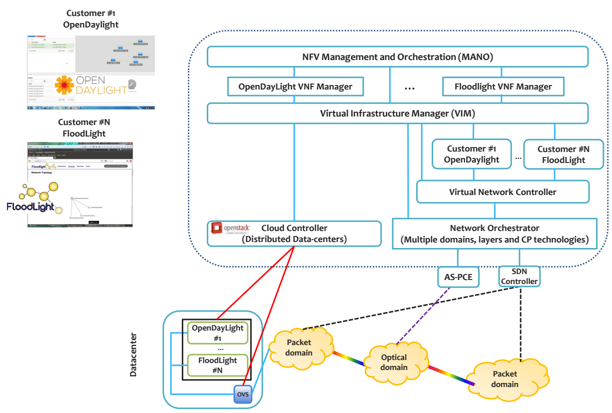

Fig. 3: Multi-tenant vSDN use case

each PCEP request, a DNS-based query-response mechanism is suitable. Moreover, DNS supports load balancing where multiple vPCEs (with different IP addresses) are known in the DNS for a single PCE server name and are seen for the PCC as a single resource. Requests are load-balanced among vPCEs without any complexity at the PCC.

Further information and results on this proposed use case, might be found in [12].

B. Use Case 2: Multi-tenant vSDN

Typically, the SDN controller of each VN runs in a dedicated host. It can be deployed using several available open source software implementations such as OpenDaylight, Floodlight, Trema, POX, etc. Thus, when a new VN is dynamically created through the VNC, it is required to manually install and configure a SDN controller implementation on a dedicated server, as well as to provide connectivity between the VNC and the SDN controller servers, typically located in a Network Operation Center (NOC). The whole process can take several hours.

In this use case, we also propose to virtualize the network control functions (i.e., SDN controller) and move them into the cloud to dynamically deploy independent SDN controller instances within minutes, whenever new VNs are dynamically deployed. This approach offers additional advantages such as the lack of hardware maintenance downtime (i.e., a virtual SDN controller can be quickly and easily moved between physical hosts within a data center when hardware maintenance is required), along with decreased recovery time in case of a network disaster or failover.

Figure 3 presents the functional blocks of the proposed SDN/NFV orchestration architecture for deploying SDN-enabled VNs. On top of the SINO we deploy several VNF managers. We propose novel VNF managers called virtual SDN controller manager (vSDN manager). The vSDN manager is in charge of requesting to the Cloud Controller, the creation of VMs in a data center and the installation of an operating system image with a compiled OpenDaylight or Floodlight SDN controller.

On top of the proposed SDN/NFV architecture we find the MANO, which is responsible for the request of a VN to the SINO, which in its turn will request the VN to the VNC. The NFV Orchestrator orchestrates the creation of the vSDN controllers (located in the cloud), the virtual networks, and the connectivity between the vSDN Controllers (located in the cloud) with the VNC.

Further information and results on this proposed use case, might be found in [13].

IV. Conclusion

We have presented a SDN/NFV architecture for providing NFV services, which we have detailed for multi-domain transport networks and distributed DCs. We have provided the details of our implementation in the SDN/NFV cloud computing platform and transport network of the ADRENALINE Testbed. Two use cases have been provided, in order to validate the feasibility of the proposed architecture: a virtual Path Computation Element and the deployment of virtual SDN controllers on top of virtual networks.

Acknowledgment

This work was partially funded by EU FP7 STRAUSS (FP7-ICT-2013-EU-Japan 608528), EU FP7 PACE (619712), and Spanish MINECO project FARO (TEC2012-38119).

References

[1] M. Chiosi, et al., ”Network Functions Virtualisation, An Introduction, Benefits, Challenges & Call for Action”, *SDN and Openflow World Congress*, 2012.

[2] N. McKeown, et al., ”OpenFlow: enabling innovation in campus networks”, *ACM SIGCOMM Computer Communication Review*, Vol. 38, Num. 2, 2008.

[3] G. Xilouris, et al., ”T-NOVA: A marketplace for virtualized network functions”, Networks and Communications (EuCNC), 2014 European Conference on, vol., no., pp.1,5, 23-26 June 2014.

[4] A. Csaszar, et al., ”Unifying Cloud and Carrier Network: EU FP7 Project UNIFY”, IEEE/ACM 6th International Conference on Utility and Cloud Computing (UCC), Dec. 2013.

[5] L. Liu, et al., ”Field trial of an OpenFlow-based unified control plane for multilayer multigranularity optical switching networks”, *Journal of Lightwave Technology*, Vol. 31, Num. 4, 2013.

[6] E. Crabbe, et al., ”PCEP Extensions for Stateful PCE”, *draft-ietf-pce-stateful-pce-09*, IETF, 2014.

[7] R. Casellas, et al., ”Control and Management of Flexi-Grid Optical Networks with an Integrated Stateful PCE and OpenFlow controller [Invited]”, in *Journal of Optical Communications and Networking* Vol. 5, No. 10, pp. A57-A65, 2013.

[8] D. King, and A. Farrel, ”A PCE-based Architecture for Application-based Network Operations”, IETF draft *draft-farrkingel-pce-abno-architecture-13*, 2014.

[9] Y.Yoshida, et al., ”First international SDN-based Network Orchestration of Variable capacity OPS over Programmable Flexi-grid EON”, in *Optical Fiber Conference*, ThA.2, 2014.

[10] R. Vilalta, et al., ”Network Virtualization Controller for Abstraction and Control of OpenFlow-enabled Multi-tenant Multi-technology Transport Networks”, accepted at *OFC Conference*, 2015.

[11] A. Mayoral, et al., ”Experimental Seamless Virtual Machine Migration Using an Integrated SDN IT and Network Orchestrator”, accepted at *OFC Conference*, 2015.

[12] R. Vilalta, et al., ”Transport PCE Network Function Virtualization”, in *European Conference on Optical Communications*, We.3.2.2, Cannes, 2014.

[13] R. Muñoz, et al., ”SDN/NFV orchestration for dynamic deployment of virtual SDN controllers as VNF for multi-tenant optical networks”, accepted at *OFC Conference*, 2015.