Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks

Raul Muñoz, Ricard Vilalta, Ramon Casellas, Ricardo Martinez, Thomas Szyrkowiec, Achim Autenrieth, Víctor López, Diego López

(Invited Paper)

***Abstract*—SDN and NFV have emerged as the most promising candidates to improve network function and protocol programmability and dynamic adjustment of network resources. On the one hand, SDN is responsible for providing an abstraction of network resources through well-defined Application Programming Interfaces. This abstraction enables SDN to perform network virtualization, that is, to slice the physical infrastructure and create multiple co-existing application-specific virtual tenant networks (VTNs) with specific quality of service (QoS) and Service Level Agreement (SLA) requirements, independent of the underlying optical transport technology and network protocols. On the other hand, the notion of NFV relates to deploying network functions that are typically deployed in specialized and dedicated hardware, as software instances (named virtual network functions - VNF) running on commodity servers (e.g. in data centers) through software virtualization techniques. Despite of all the attention that has been given to virtualizing IP functions (e.g. Firewall, AAA, etc) or LTE control functions (e.g., MME, SGW and PGW), some transport control functions can also be virtualized and moved to the cloud as a VNF. In this work we propose to virtualize the tenant SDN control functions of a VTN and move them into the cloud. The control of a VTN is a key requirement associated with network virtualization, since it allows the dynamic programming (i.e., direct control and configuration) of the virtual resources allocated to the VTN. We experimentally assess and evaluate the first SDN/NFV orchestration architecture in a multi-partner testbed to dynamically deploy independent SDN controller instances for each instantiated VTN and to provide the required connectivity within minutes.**

***Index Terms*—optical network virtualization; SDN and NFV orchestration; cloud and network resources, virtual network functions.**

I. Introduction

SOFTWARE Defined Networking (SDN) is defined as a logically centralized control framework that supports the programmability of network functions and protocols by decoupling the data plane from the control plane through a well-defined control protocol. OpenFlow is the flagship protocol for the southbound interface being standardized by the Open Networking Foundation (ONF) [1]. It offers a logical switch abstraction, mapping high-level instructions of the protocol to hide vendor-specific hardware details, which mitigates inter-operability issues commonly found in multi-vendor deployments [2]. The control entity (SDN controller) is responsible for providing an abstraction of the network forwarding technologies (e.g., packet/flow switching or circuit switching) through an Application Programming Interface (API). This abstraction enables to deploy a network hypervisor in order to perform network virtualization, that is, to slice the physical infrastructure and create multiple co-existing virtual tenant networks (VTN) independent of the underlying transport technology and network protocols. VTNs must be independently controlled by their own control plane instance (e.g., a tenant SDN controller) [3], allowing for specific quality of service (QoS) and service level agreement (SLA).

Typically, the tenant SDN controller of each VTN runs in a dedicated host. It can be deployed using several available open source SDN controller implementations such as OpenDaylight, Floodlight, Trema, POX, Ryu, ONOS, etc. Thus, when a new VTN is dynamically created, the tenant SDN controller implementation is required to be manually installed and configured on a dedicated server, as well as the required connectivity between the network hypervisor and the tenant SDN controller server, typically located in the tenant Network Operation Center (NOC). The whole process can take several days. In this work we propose to virtualize the tenant SDN control functions leveraging a Network Function Virtualization (NFV) architecture [4] and to move them into the cloud in order to dynamically deploy independent SDN controller instances and provide the required connectivity within minutes, whenever a new VTN is dynamically deployed. This approach also offers additional advantages such as the lack of hardware maintenance downtime (a virtual tenant SDN controller can be quickly and easily moved between physical hosts within a data center when hardware maintenance is required), and decreased recovery time in case of a disaster or failover (backups and snapshots of the virtual tenant SDN controllers taken throughout the day can be moved from one data center to another and redeployed easier and faster after a failure).

The notion of NFV relates to deploying network functions that are typically deployed in specialized and dedicated hardware, as software instances (named virtual network functions - VNF) running on commodity servers in data centers (DCs) (or in general, in computing distributed throughout the network) through software virtualization techniques. NFV is applicable to any data plane packet processing and control plane function in fixed and mobile network infrastructures. Examples of VNFs include IP network functions such as load balancers, firewalls, security or Authentication, Authorization and Accounting (AAA), LTE/EPC network functions, such as Mobility Management Entity (MME), Serving Gateway (SGW), and PDN Gateway (PGW). Despite

Raul Muñoz, Ricard Vilalta, Ramon Casellas, and Ricardo Martínez are with the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Av. Carl Friedrich Gauss 7, 08860 Castelldefels, Spain (e-mail: raul.munoz@cttc.es)

Thomas Szyrkowiec and Achim Autenrieth are with ADVA Optical Networking, Campus Martinsried, Fraunhoferstraße 9a, 82152 Martinsried/Munich, Germany

Victor Lopez and Diego López are with Telefonica I+D, c/ Don Ramon de la Cruz 84, 28006, Madrid, Spain

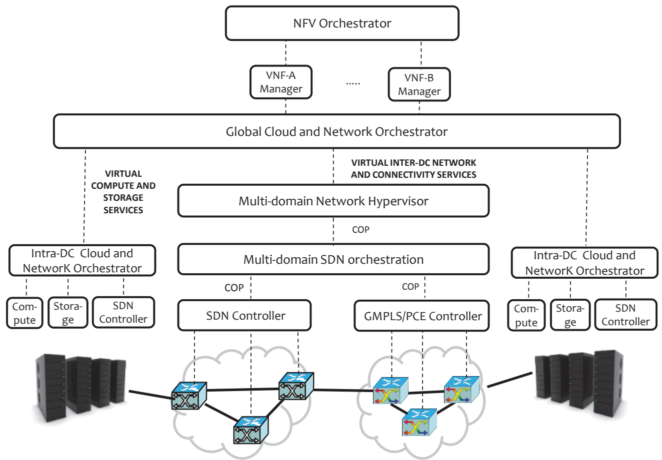


Fig. 1: Proposed Network Function Virtulization Management and Orchestration (NFV MANO) architecture

of all attention has been given to virtualizing IP functions or LTE control functions (SGW, PGW), some transport control functions can also be virtualized and moved to the cloud as a VNF. Previous works on control plane function virtualization for transport networks have been focused on the virtualization of the PCE [5].

This paper is organized as follows. Sec.II presents the proposed integrated SDN/NFV management and orchestration architecture for multi-tenant transport networks to dynamically deploy VTNs and their corresponding tenant SDN controllers as VNFs in data centers. Sec.III shows the experimental assessment and performance evaluation in a multi-partner experimental setup involving cloud and network facilities from CTTC in Barcelona (Spain), ADVA Optical Networking in Gdynia (Poland) and Telefónica I+D in Madrid (Spain). Finally Sec.IV concludes the paper.

II. SDN/NFV management and orchestration architecture for dynamic VNF services

In this section we present the proposed integrated SDN/NFV management and orchestration architecture in order to deploy VNFs on top of an integrated cloud and network platform (i.e., NFV Infrastructure - NFVI). The NFVI is composed of heterogeneous multi-domain and multi-layer transport networks with heterogeneous transport and control technologies interconnecting distributed data centers, providing compute, storage and network resources. On top of this physical infrastructure, we deploy an NFVI virtualization layer responsible for virtualizing the compute, storage and network resources of the NFVI by means of a virtualized infrastructure manager (VIM). On top of this virtualization layer we deploy several VNF managers and the NFV orchestrator that is responsible for the dynamic deployment of virtual SDN-enabled networks. Fig.1 presents the functional blocks of the proposed SDN/NFV management and orchestration architecture.

A. Multi-domain SDN orchestrator

The multi-domain network orchestration mechanism acts as a unified transport network operating system (or controller of controllers) allowing the composition, at a higher, abstracted level, of end-to-end provisioning services across multiple domains with heterogeneous multi-layer transport network technologies regardless of the specific control plane technology employed in each domain (e.g., SDN/OpenFlow or GMPLS/PCE). The conceived multi-domain SDN network orchestrator architecture is based on the Application-based Network Operations (ABNO) architecture [6] proposed in the Internet Engineering Task Force (IETF). It has been experimentally validated for multi-layer and multi-domain network orchestration in [7] and [8].

Typically, the northbound interface (NBI) of a domain controller is technology and vendor dependent, so the multi-domain SDN network orchestrator has to implement different plugins for each of the domain controller's NBI. The STRAUSS project has defined a generic functional model of a "control plane" for the provisioning of connectivity, topology dissemination and path computation, and defines an associated protocol (the Control Orchestration Protocol - COP) [9]. Different use cases defined by European-funded research projects (IDEALIST, DISCUSS, COMBO and IN-SPACE) about how the COP could be used in their heterogeneous network scenarios are provided in [10]. The design of COP between the orchestration and control layers allows the simplification and optimization, in terms of scalability and compatibility between the different modules which compose the SDN architecture. COP unifies all the orchestration functionalities into a single protocol paradigm. The proposed COP provides a common NBI API so that all SDN controllers can be orchestrated using a single common protocol. The latest Optical Internetworking Forum (OIF) / Open Networking Foundation (ONF) Transport SDN API [11] is in line with COP's objectives. COP provides a research-oriented multi-layer approach using YANG/RESTconf, while OIF/ONF Transport SDN API is focused on standardization efforts for orchestration of REST NBI for SDN controllers. RESTCONF is a REST-line protocol defined by the IETF for accessing data defing in YANG using datastores defined in NETCONF. A draft of the COP definition is open for discussion and can be downloaded and contributed at [12].

In brief, COP is composed of three main base functions:

1) Call: The first common service identified as a COP requirement is the design of a common provisioning model which defines an end-to-end connectivity provisioning service. In the scope of COP the service Call is defined as the provisioning interface. A Call object must describe the type of service that is requested or served by it (e.g., DWDM, Ethernet, MPLS). It also contains the endpoints between whom the service is provided. The Call object also includes the list effective connections made into the data plane, to support the service call. A Connection object is used for a single network domain scope. It should include the path or route across the network topology the data traverses, which may be fully described or abstract depending on the orchestration/control schemes used. Each connection must be associated with a single control plane entity (e.g. a SDN controller) responsible for the configuration of the data path. Finally, the Call also introduces the necessary TE parameters (e.g., bandwidth) that the service requests.

2) Topology: The COP definition also covers the topological information about the network, which must include a common and homogeneous definition of the network topologies included in the TE Databases (TED) of the different control instances. A Topology object may consist of a set of nodes and edges, which form a tree structure. A Node must contain a list of ports or endpoints and their associated switching capabilities. An Edge object is defined as the connection link between two Endpoints. Due to the need of conforming to a common model among different transport network technologies, the definition of the three main objects described (Node, Edge, Endpoint) must be extensible, able to include TE extensions to describe different switching capabilities (i.e., time-slots, packets, wavelengths, frequency slots).

3) Path Computation: The Path Computation service should provide an interface to request and return Path objects which contain the information about the route between two Endpoints. Path computation is highly related to the previous group of resources. In the service Call, the Connection object has been designed to contain information about the traversed Path. The Path model should be the same in both, the service Call and at the Path Computation. Furthermore each component in the Path object is represented as an Endpoint with TE information associated to it.

B. Multi-domain Network Hypervisor

A network hypervisor is responsible for partitioning and/or aggregating the physical resources of the heterogeneous networks into virtual resources, interconnecting them to compose multiple end-to-end VTNs with different VNT topologies while sharing the same physical infrastructure. It is also responsible for representing an abstracted topology of each VTN (i.e., network discovery) to an independent tenant SDN controller, and for it to remotely control (i.e., dynamic provisioning, modification and deletion of connections) the virtual network resources allocated to their corresponding VTN, as if they were real resources, through a well-defined interface (e.g., OpenFlow protocol). The network hypervisor can dynamically create, modify and delete VTNs in response to application demands (e.g., through a traffic demand matrix describing resource requirements and QoS for each pair of connections).

The proposed system architecture relies on the network orchestrator, which provides a generic network abstraction mechanism for the different transport infrastructure resources (e.g., Ethernet, flexi-grid DWDM [13]). The proposed architecture of a multi-domain network hypervisor has been proposed and experimentally evaluated in [14].

C. Intra-DC Cloud and Network Orchestrator

Virtualization of cloud and network resources within a data center is provided by Intra-DC Cloud Orchestrators. Intra-DC Cloud Orchestrators may be deployed with different cloud computing software platforms, such as OpenStack, CloudStack or OpenNebula. The Intra-DC Cloud Orchestrator takes over the creation/ migration/ deletion of Virtual Machine (VM) instances (computing service), storage of disk images (image service), and the management of the virtual machines' network interfaces and the intra-DC network connectivity (networking service).

The computing service (e.g., Nova in OpenStack) manages the virtual machine in the compute hosts. A compute service agent is running in each host and controls the hypervisor (e.g., 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 virtual machine file systems; it also operates in a centralized manner by maintaining a copy of all the disk images. An image-service agent is running in each host to request the download of images when a new virtual machine instantiation requires it. It also permits to create new images of the currently working instances, a process known as snapshotting, which is used for virtual machine migration. Finally, the connectivity between virtual machines within the hosts and OpenFlow-enabled switches within a DC is managed by the networking service (e.g., Neutron in OpenStack). It creates the virtual interfaces, attaches them to the OpenFlow-enabled switches and offers a DHCP service for the VMs to get the assigned IP address. The Neutron plugin can be used to provide full control of the OpenFlow-enabled switches through an SDN Controller, and disabling reactive packet-in mechanism for unknown incoming connections [15].

D. Global Cloud and Network Orchestrator

The interconnection of different DC sites that are physically dispersed, but logically centralized, is one of the major challenges to face in order to provide global end-to-end cloud services. However, current federated cloud solutions do not take advantage of the dynamicity and flexibility provided by the transport networks. The network is considered a commodity providing fixed and static bandwidth pipes (e.g. L2) for bulk data transfer among DCs. Cloud federation targets the logical management of IT resources in a federation of multiple distributed cloud orchestrators, assuming that there are pre-provisioned inter-DC connections between distributed DCs through IP or L2 services.

Thus, there is the need to perform an integrated orchestration of distributed DCs and heterogeneous transport networks in order to dynamically and globally provision, migrate and delete virtual machines and provide the required end-to-end connectivity between distributed DCs across multi-domain, multi-layer network infrastructures. In our architecture, this function is provided by the Global cloud and Network Orchestrator, as shown in Fig.1 and is responsible for effectively coordinating the management of the IT resources in the distributed DCs, and the network resources in the heterogeneous networks, providing a unified cloud and network operating system towards the applications, such as the VNF managers. A preliminary architecture of an Intra-DC Cloud Orchestrator named SDN IT and Network Orchestrator (SINO) has been defined and evaluated in [16] and [17].

E. VNF Managers and NFV orchestrator

VNFs running in a virtual machine can be located in the most appropriate DC, and can be interconnected with each other and with the end-points in a certain way (forwarding graph) in order to achieve the desired overall end-to-end functionality or service. This is known as ”service chaining”. Thus, VNFs can be distributed over several DCs connected through multiple heterogeneous transport networks.

The VNF manager is responsible for the lifecycle management (i.e., creation, configuration, and removal) of a VNF. The VNF manager design and definition are being discussed at standardization bodies. A simple design for a VNF manager is proposed in [18]. In this work we propose a novel VNF manager called virtual SDN controller manager (vSDN manager). The vSDN manager is in charge of requesting the creation of VMs in a data center and the installation of an operating system image with a compiled OpenDaylight or Floodlight SDN controller (referred to as tenant vSDN controller in Fig1) from the Cloud controller.

Finally, the NFV Orchestrator is defined by the ETSI as being 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 to deploy end-to-end NFV forwarding graphs. The NFV Orchestration design and definition are subject to definition at standardization bodies, and still at discussion. In this work we propose an NFV orchestrator responsible for the management (provisioning, modification and deletion) of the virtual SDN-enabled optical networks by orchestrating the creation of the virtual tenant SDN controllers (vSDN controllers), the VNT, and the required connectivity with the dynamically deployed vSDN Controllers that are located in the cloud.

III. Experimental Assessment and Performance Evaluation

A. Network Scenario

We have developed a proof-of-concept prototype of the proposed NFV management and orchestration architecture in the multi-partner experimental setup shown in Fig.2, in order to experimentally assess and evaluate the dynamic provisioning of virtual SDN-enabled optical transport networks. The experimental setup is composed of a data center provided by CTTC in Barcelona (Spain), a DWDM network with an Optical Network Hypervisor (ONH) from ADVA in Poland, and a computer terminal in the NOC of Telefónica in Madrid (Spain). All these three premises are connected using OpenVPN tunnels over the Internet. Specifically, we have setup two OpenVPN tunnels, one between CTTC and Telefónica, and anther one between CTTC and ADVA. The CTTC data center is based on Openstack for the cloud computing platform, and OpenDaylight for the control of the OpenFlow-enabled network. The OpenStack Havanna release has been deployed into five physical servers with 2 x Intel Xeon E5-2420 and 32GB RAM each, one dedicated to the cloud controller and the other four as compute pool (hosts) for virtual machine instantiation. Eight OpenFlow switches have been deployed following a tree topology using standard custom off the shelf hardware (Intel Core i7-3770) with OpenVSwitch software, which can be controlled by OpenFlow 1.0. The ADVA ONH is a multi-tenant capable application that creates abstracted representations of the underlying optical transport network and exposes them to client SDN controllers. The supported models for the abstraction include a single virtual node and abstracted links spanning multiple devices. For the experiment only the former one has been used. This means that from the perspective of the exposed SDN interface, the ONH acts, for each VTN, as one virtual switch controlled by OpenFlow.

B. Example of a dynamic deployment of virtual SDN-enabled optical transport networks

Let us consider the example in Fig.3.a to show the workflow between the involved functional blocks of the proposed NFV management and orchestration architecture. First (step1), the NFV Orchestrator requests the provisioning of a new virtual tenant SDN controller from the vSDN manager, specifying the desired tenant SDN controller implementation (e.g., OpenDaylight, Floodlight, etc.). To this

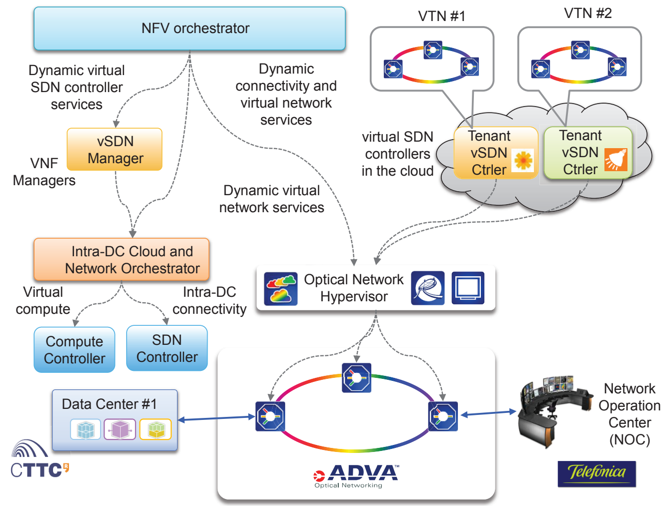
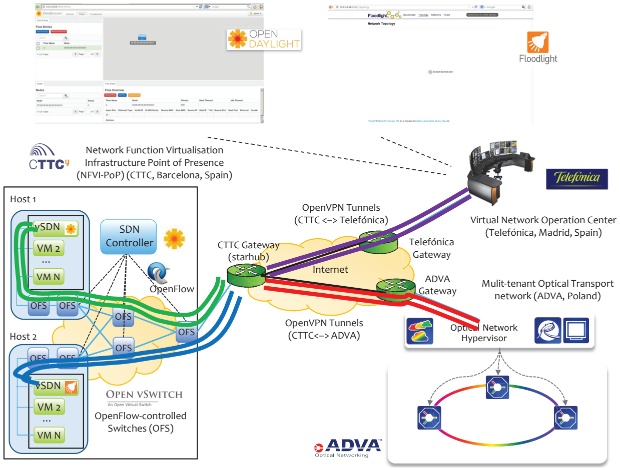


Fig. 2: Experimental network scenario setup composed of CTTC, ADVA and Telefónica facilites

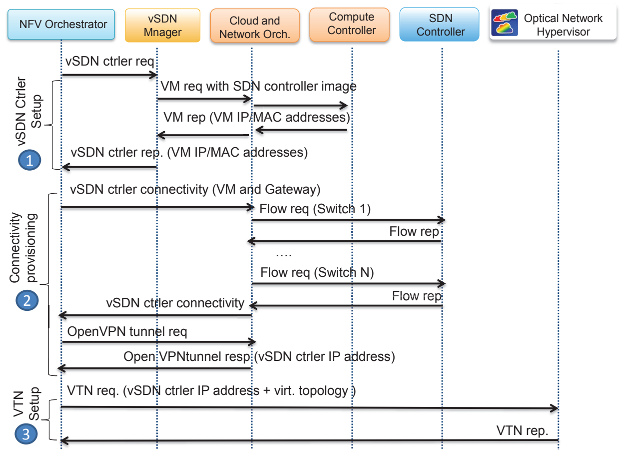
end, the vSDN manager sends a request to the cloud and network orchestrator to create a new virtual machine with the requested pre-installed SDN controller. Then, the cloud and network orchestrator requests the provisioning of the new virtual machine from the compute controller. In a distributed environment with several data centers, the NFV orchestrator could request to deploy the vSDN controller in the data center nearest to the NOC to minimize the latency. Once the requested vSDN controller is up and running, the vSDN manager controller notifies it and provides the IP address of the virtual machine. Then (step2), the NFV orchestrator sends a request to the cloud and network orchestrator to provision the required connectivity between the vSDN controller and the NOC and the ONH (Fig.3.b). To this end, the cloud and network orchestrator first requests the SDN controller to provision two flows, one between the deployed virtual machine and the CTTC gateway, and another one from the CTTC gateway to the deployed virtual machine. Second, it configures in the deployed virtual machine two OpenVPN tunnels on top of the deployed flows. Once the OpenVPN tunnels to the CTTC gateway are deployed, the cloud and network orchestrator informs the NFV orchestrator about the IP address of the vSDN controller. Finally (step3), once the connectivity is provided, the NFV orchestrator requests the ONH to provision a VTN with the IP address of the vSDN controller and the requested virtual topology graph. This virtual topology representation can be based either on virtual node or virtual link aggregation mechanisms. The former hides internal connectivity issues by representing the VTN topology as a single virtual node. The latter represents the VTN topology as a set of virtual nodes and virtual links, where each virtual link is a connection across physical nodes. Once the VTN has been successfully provisioned, the ONH notifies the NFV Orchestrator. At this point, the virtual tenant SDN-enabled optical network is ready to be used through a virtual tenant SDN controller located in the cloud.

C. Results

In the experimentation, we request the NFV orchestrator to provision a virtual SDN-enabled optical transport networks either controlled by OpenDaylight or Floodlight with

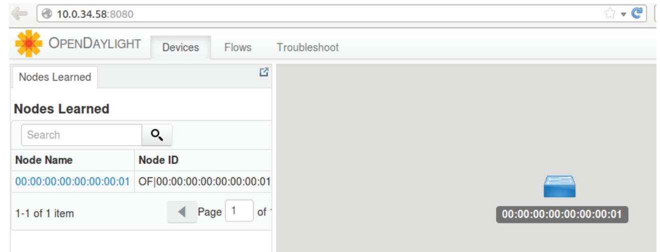


(a) Workflow for provisioning a first virtual SDN-enabled optical network

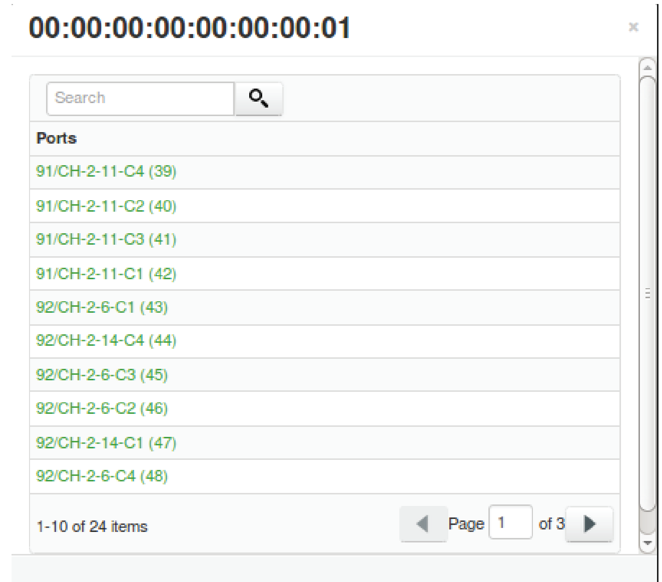


(b) Provisioned vSDN controllers and flows in the experimental scenario for two virtual SDN-enabled optical network controlled with OpenDaylight and Floodlight

Fig. 3: Example of dynamic deployment of virtual SDN-enabled optical transport networks



(a) View of the VTN topology represented as a single virtual node



(b) View of the available virtual ports for switching in the single virtual node

Fig. 5: OpenDaylight's Graphical User Interface (GUI) screenshot of a vSDN controller running in the cloud

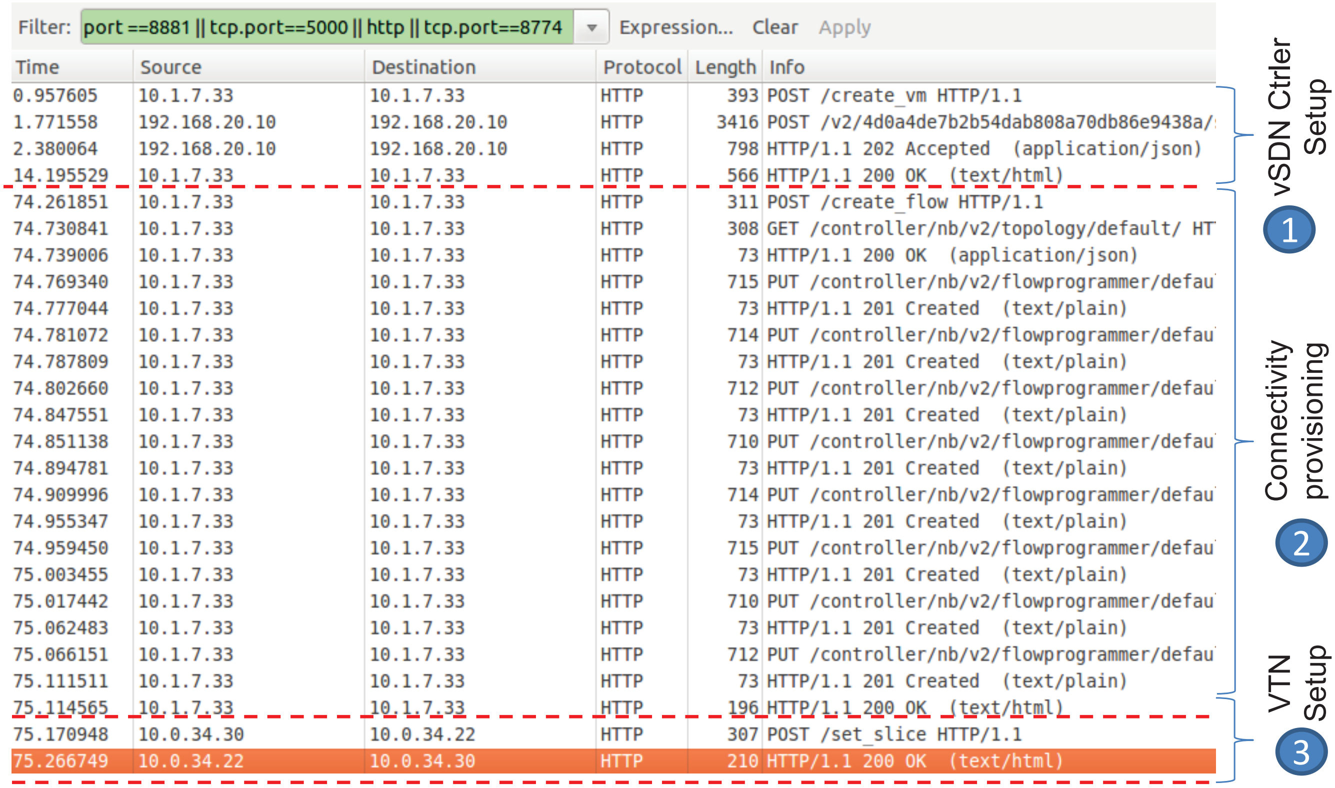
abstracted topologies based on the single virtual node model.

Fig.4 shows a wireshark capture with the exchange of messages to provision a virtual SDN-enabled optical network as shown is Sec.III-B involving the NFV orchestrator, vSDN Manager, cloud and network orchestrator, OpenDaylight and OpenStack controllers, and the ONH. In this experiment, all these modules have been implemented in a single server (10.1.7.33), with the exception of the ONH that is located in Poland (10.0.34.22). In particular, Fig.4 shows first the vSDN controller setup through the exchange of a HTTP POST request/response between the NFV orchestrator and VNF orchestrator to the Openstack controller. Then, the NFV orchestrator configures the eight OpenFlow-enabled switches to offer one bidirectional flow between the deployed virtual machine and the CTTC's gateway by sending eight HTTP PUT messages to the SDN controller. Finally, the NFV orchestrator requests the actual provisioning and configuration of a slice of the ONH through the exchange of HTTP POST request/response messages.

Once a virtual SDN-enabled transport network is deployed, the NOC's computer terminal can connect to the Graphical User Interface (GUI) of the vSDN controller located in the cloud. Fig.5 presents a screenshot of OpenDaylight's GUI showing a dynamically deployed vSDN controller. In particular, this figure depicts both the representation of the VTN topology as a single virtual node (Fig.5.a), as well as the available virtual ports of the virtual single node (Fig.5.b). The shown port names are based on the last octet of the IP address and the path of the module. This allows informed connection decisions despite the condensed abstraction.

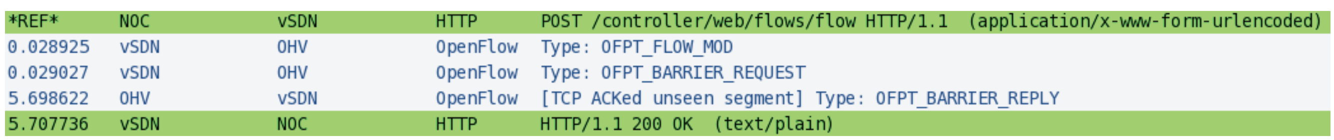
Using the vSDN controller's GUI we can dynamically request the provisioning of flows within the allocated VTN. Fig.6 shows a wireshark capture at a vSDN controller with the exchange of messages for provisioning a flow between the NOC's computer terminal (webpage) and the ONH, once the virtual SDN-enabled optical network is up: HTTP POST request from NOC computer terminal to vSDN controller, OpenFlow FLOW_MOD and BARRIER messages between the vSDN controller and the ONH (to configure the virtual switch), the OpenFlow BARRIER_REPLY from the ONH to the vSDN controller, and finally the HTTP POST response to the NOC. Since the virtual optical switch, representing the optical equipment, requires quite a long time to be configured, the time the controller waits for a response after sending a Barrier Request or a Statistics Request message has been increased from 2s (default) to 5s.

Finally, Tab.I shows the evaluated performance metrics, namely, the average delay for creating a virtual machine and configuring a vSDN controller (74s), for creating a VTN



Time	Source	Destination	Protocol	Length	Info
0.957605	10.1.7.33	10.1.7.33	HTTP	393	POST /create_vm HTTP/1.1
1.771558	192.168.20.10	192.168.20.10	HTTP	3416	POST /v2/4d0a4de7b2b54dab808a70db86e9438a/:
2.380064	192.168.20.10	192.168.20.10	HTTP	798	HTTP/1.1 202 Accepted (application/json)
14.195529	10.1.7.33	10.1.7.33	HTTP	566	HTTP/1.1 200 OK (text/html)
74.261851	10.1.7.33	10.1.7.33	HTTP	311	POST /create_flow HTTP/1.1
74.730841	10.1.7.33	10.1.7.33	HTTP	308	GET /controller/nb/v2/topology/default/ HT
74.739006	10.1.7.33	10.1.7.33	HTTP	73	HTTP/1.1 200 OK (application/json)
74.769340	10.1.7.33	10.1.7.33	HTTP	715	PUT /controller/nb/v2/flowprogrammer/defau
74.777044	10.1.7.33	10.1.7.33	HTTP	73	HTTP/1.1 201 Created (text/plain)
74.781072	10.1.7.33	10.1.7.33	HTTP	714	PUT /controller/nb/v2/flowprogrammer/defau
74.787809	10.1.7.33	10.1.7.33	HTTP	73	HTTP/1.1 201 Created (text/plain)
74.802660	10.1.7.33	10.1.7.33	HTTP	712	PUT /controller/nb/v2/flowprogrammer/defau
74.847551	10.1.7.33	10.1.7.33	HTTP	73	HTTP/1.1 201 Created (text/plain)
74.851138	10.1.7.33	10.1.7.33	HTTP	710	PUT /controller/nb/v2/flowprogrammer/defau
74.894781	10.1.7.33	10.1.7.33	HTTP	73	HTTP/1.1 201 Created (text/plain)
74.909996	10.1.7.33	10.1.7.33	HTTP	714	PUT /controller/nb/v2/flowprogrammer/defau
74.955347	10.1.7.33	10.1.7.33	HTTP	73	HTTP/1.1 201 Created (text/plain)
74.959450	10.1.7.33	10.1.7.33	HTTP	715	PUT /controller/nb/v2/flowprogrammer/defau
75.003455	10.1.7.33	10.1.7.33	HTTP	73	HTTP/1.1 201 Created (text/plain)
75.017442	10.1.7.33	10.1.7.33	HTTP	710	PUT /controller/nb/v2/flowprogrammer/defau
75.062483	10.1.7.33	10.1.7.33	HTTP	73	HTTP/1.1 201 Created (text/plain)
75.066151	10.1.7.33	10.1.7.33	HTTP	712	PUT /controller/nb/v2/flowprogrammer/defau
75.111511	10.1.7.33	10.1.7.33	HTTP	73	HTTP/1.1 201 Created (text/plain)
75.114565	10.1.7.33	10.1.7.33	HTTP	196	HTTP/1.1 200 OK (text/html)
75.170948	10.0.34.30	10.0.34.22	HTTP	307	POST /set_slice HTTP/1.1
75.266749	10.0.34.22	10.0.34.30	HTTP	210	HTTP/1.1 200 OK (text/html)

Fig. 4: Wireshark capture when provisioning a virtual SDN-enabled optical network



REF	NOC	vSDN	HTTP	POST /controller/web/flows/flow HTTP/1.1 (application/x-www-form-urlencoded)
0.028925	vSDN	OHV	OpenFlow	Type: OFPT_FLOW_MOD
0.029027	vSDN	OHV	OpenFlow	Type: OFPT_BARRIER_REQUEST
5.698622	OHV	vSDN	OpenFlow	[TCP ACKed unseen segment] Type: OFPT_BARRIER_REPLY
5.707736	vSDN	NOC	HTTP	HTTP/1.1 200 OK (text/plain)

Fig. 6: Wireshark capture at a virtual tenant SDN controller when provisioning of a flow.

(0.1s), and for configuring two OpenFlow flows and two OpenVPN tunnels between the vSDN controller and the CTTC's gateway node (0.9s), as depicted in Fig.3.b. The total average setup delay for provisioning a virtual SDN-enabled optical network is 75s. Once the virtual SDN-enabled optical transport network is set up, the average flow provisioning delay is 6s.

IV. Conclusion

We have presented the first SDN and NFV management and orchestration architecture for multi-tenant transport networks. The proposed architecture allows to dynamically deploy virtual optical transport networks and their corresponding virtual SDN controllers as VNFs in data centers. This architecture addresses two main challenges. First, it provides a multi-domain network orchestration and virtualization mechanism to offer dynamic and flexible end-to-end connectivity and virtual network provisioning services across multi-domain and multi-technology networks, integrating all transport network segments with heterogeneous optical technologies. The second main challenge is to perform an integrated orchestration of distributed cloud resources (virtual compute and storage) and network resources to dynamically deploy virtual machines and VNF instances and provide the required network connectivity between DCs and end-points across heterogeneous multi-domain transport networks. An experimental performance evaluation has been carried out in a multi-partner experimental setup involving CTTC in Barcelona (Spain), ADVA Optical Networking in Gdynia (Poland) and Telefónica I+D in Madrid (Spain).

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement nº 608528 through the project STRAUSS and the Spanish Ministry of Economy and Competitiveness (MINECO) through the project FARO (TEC2012-38119)

References

[1] "Sdn architecture 1.0," *Open Networking Foundation (ONF), https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf*, 2014.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

Av. VSDN Ctlrer Setup Delay		Av. Connectivity prov. delay	Av. virtual network setup delay	Av. Total deployment delay	Av. Flow prov. delay
VM creation	vSDN Ctrler Conf.				
73.6s	0.4s	0.9s	0.1s	75s	6s

TABLE I: Performance evaluation in terms of setup delays

[3] A. Autenrieth, T. Szyrkowiec, K. Grobe, J.-P. Elbers, P. Kaczmarek, P. Kostecki, and W. Kellerer, "Evaluation of virtualization models for optical connectivity service providers," in *Optical Network Design and Modeling, 2014 International Conference on*. IEEE, 2014, pp. 264–268.

[4] "Network function virtualization (nfv): Architectural framework," *ETSI GS NFV 002 v.1.1.1,*, 2013.

[5] R. Vilalta, R. Muñoz, R. Casellas, R. Martínez, V. López, and D. López, "Transport network function virtualization," *Journal of Lightwave Technology*, vol. 33, no. 8, pp. 1557 – 1564, 2015.

[6] D. King and A. Farrel, "A pce-based architecture for application-based network operations," IETF RFC 7491, March 2015.

[7] Y. Yoshida, A. Maruta, K. Kitayama, M. Nishihara, T. Takahara, T. Tanaka, J. Rasmussen, N. Yoshikane, T. Tsuritani, I. Morita *et al.*, "Sdn-based network orchestration of variable-capacity optical packet switching network over programmable flexi-grid elastic optical path network," *Journal of Lightwave Technology*, vol. 33, no. 3, pp. 609–617, 2014.

[8] R. Munoz, R. Vilalta, R. Casellas, R. Martinez, F. Francois, M. Channegowda, A. Hammad, S. Peng, R. Nejabati, D. Simeonidou *et al.*, "Transport network orchestration for end-to-end multi-layer provisioning across heterogeneous sdn/openflow and gmpls/pce control domains," *Journal of Lightwave Technology*, vol. 33, no. 8, pp. 1540–1549, 2015.

[9] R. Muñoz, R. Vilalta, R. Casellas, and R. Martínez, "Sdn orchestration and virtualization of heterogeneous multi-domain and multi-layer transport networks: The strauss approach," in *In Proc. IEEE Black Sea Conference on Communications and Networking (IEEE BlackSeaCom)*, 2015.

[10] R. Vilalta, V. López, A. Mayoral, N. Yoshikane, M. Ruffini, D. Siracusa, R. Martínez, T. Szyrkowiec, A. Autenrieth, S. Peng, R. Casellas, R. Nejabati, D. Simeonidou, X. Cao, T. Tsuritani, I. Morita, J. P. Fernández-Palacios, and R. Muñoz, "The Need for a Control Orchestration Protocol in Research Projects on Optical Networking," in *in Proceedings of European Conference on Networks and Communications (EuCnC)*, 2015.

[11] "Global transport sdn prototype demonstration," *Optical Internetworking Forum (OIF) / Open Networking Foundation (ONF) White Paper*, 2014.

[12] "Control Orchestration Protocol (COP)," https://github.com/ict-strauss/COP.

[13] O. Gonzalez de Dios, R. Casellas, "Framework and Requirements for GMPLS-based control of Flexi-grid DWDM networks," draft-ietf-ccamp-flexi-grid-fwk-05, work in progress, May 2015.

[14] R. Vilalta, R. Muñoz, R. Casellas, R. Martinez, F. Francois, S. Peng, R. Nejabati, D. Simeonidou, N. Yoshikane, T. Tsuritani, I. Morita, V. Lopez, T. Szyrkowiec, and A. Autenrieth, "Network virtualization controller for abstraction and control of openflow-enabled multi-tenant multi-technology transport networks," in *In Proc. Optical Fiber Communication Conference (OFC)*. OSA, 2015.

[15] T. Szyrkowiec, A. Autenrieth, P. Gunning, P. Wright, A. Lord, J. Elbers, and A. Lumb, "First field demonstration of cloud datacenter workflow automation employing dynamic optical transport network resources under openstack & openflow orchestration," in *39th European Conference and Exhibition on Optical Communication (ECOC 2013)*, 2013, pp. 22–26.

[16] A. Mayoral, R. Vilalta, R. Muñoz, R. Casellas, and R. Martinez, "Experimental seamless virtual machine migration using an integrated sdn it and network orchestrator," in *In Proc. Optical Fiber Communication Conference (OFC)*. OSA, 2015.

[17] A. Mayoral, R. Vilalta, R. Munoz, R. Casellas, R. Martinez, and J. Vilchez, "Integrated it and network orchestration using openstack, opendaylight and active stateful pce for intra and inter data center connectivity," in *In Proc. of European Conference on Optical Communication (ECOC 2014)*, 2014.

[18] R. Vilalta, A. Mayoral, R. Munoz, R. Casellas, and R. Martinez, "The sdn/nfv cloud computing platform and transport network of the adrenaline testbed," in *in Proceedings of 1st IEEE Conference of Network Softwarization (NetSoft)*, 2015.

Raül Muñoz (SM'12) graduated in telecommunications engineering in 2001 and received a Ph.D. degree in telecommunications in 2005, both from the Universitat Politècnica de Catalunya (UPC), Spain. He is Senior Researcher and Head of the Optical Network and System Department at Telecommunications Technological Center of Catalonia (CTTC). Since 2000, he has participated in over 20 R&D projects funded by the European Commission's Framework Programmes (H2020, FP7, FP6 and FP5) and the Spanish Ministries, as well as technology transfer projects with Industry. He has led several Spanish research projects, and currently, he leads the European Consortium of the EU-Japan project STRAUSS. His research interests include control and service management architectures, protocols and traffic engineering algorithms for future optical networks. He has published over 150 journal and international conference papers in this field.

Ricard Vilalta graduated in telecommunications engineering in 2007 and received a Ph.D. degree in telecommunications in 2013, both from the Universitat Politècnica de Catalunya (UPC), Spain. He also has studied Audiovisual Communication at UOC (Open University of Catalonia) and is a master degree on Technology-based business innovation and administration at Barcelona University (UB). Since 2010, Ricard Vilalta is a researcher at CTTC, in the Optical Networks and Systems Department. His research is focussed on Optical Network Virtualization and Optical Openflow. He is currently a Research Associate at Open Networking Foundation.

Ramon Casellas (SM'12) graduated in telecommunications engineering in 1999 by both the UPC-BarcelonaTech university and ENST Telecom Paristech, within an Erasmus/Socrates double degree program. After working as an undergraduate researcher at both France Telecom R&D and British Telecom Labs, he completed a Ph.D. degree in 2002. He worked as an associate professor at the networks and computer science department of the ENST (Paris) and joined the CTTC Optical Networking Area in 2006, with a Torres Quevedo research grant. He currently is a senior research associate and the coordinator of the ADRENALINE testbed. He has been involved in several international R&D and technology transfer projects. His research interests include network control and management, the GMPLS/PCE architecture and protocols, software defined networking and traffic engineering mechanisms. He contributes to IETF standardization within the CCAMP and PCE working groups. He is a member of the IEEE Communications Society and a member of the Internet Society.

Ricardo Martínez (SM'14) graduated and PhD in telecommunications engineering by the UPC-BarcelonaTech university in 2002 and 2007, respectively. He has been actively involved in several public-funded (national and EU) R&D as well as industrial technology transfer projects. Since 2013, he is Senior Researcher of the Communication Networks Division (CND) at CTTC. His research interests include control and network management architectures, protocols and traffic engineering mechanisms for next-generation packet and optical transport networks within aggregation/metro and core segments.

Thomas Szyrkowiec received his M.Sc. degree in informatics from the Technische Universität München, Germany, in 2013. He joined ADVA Optical Networking, Munich, in 2013, where he is working as an Engineer in the Advanced Technology team toward his Ph.D. degree in cooperation with the Chair of Communication Networks at Technische Universität München. His research interests include software-defined networking and network virtualization in flexible optical networks.

Achim Autenrieth is currently Principle Research Engineer Advanced Technology in the CTO Office at ADVA Optical Networking, where he is working on the design and evaluation of multilayer networks, control plane, and SDN concepts. He received his Dipl.-Ing. and Dr.-Ing. degree in Electrical Engineering and Information Technology from the Munich University of Technology, Germany, in 1996 and 2003. Achim is a member of IEEE and VDE/ITG, he authored or co-authored more than 75 reviewed and invited scientific publications and he is technical program committee member of ECOC, DRCN, and RNDM.

Victor López received the M.Sc. (Hons.) degree in telecommunications engineering from Universidad de Alcalá de Henares, Spain, in 2005 and the Ph.D. (Hons.) degree in computer science and telecommunications engineering from Universidad Autónoma de Madrid (UAM), Madrid, Spain, in 2009. In 2004, he joined Telefónica I+D as a Researcher. In 2006, he joined the High-Performance Computing and Networking Research Group (UAM) as a Researcher in the ePhoton/One+ Network of Excellence. He worked as an Assistant Professor at UAM. In 2011, he joined Telefonica I+D as Technology specialist. He has co-authored more than 100 publications and contributed to IETF drafts. His research interests include the integration of Internet services over IP/MPLS and optical networks and control plane technologies (PCE, SDN, GMPLS).

Diego López received his MS from the University of Granada in 1985, and his PhD degree from the University of Seville in 2001. Diego joined Telefonica I+D in 2011 as a Senior Technology Expert on network infrastructures and services. He is currently in charge of the Technology Exploration activities within the Transversal Projects and Innovation Unit. Diego is currently focused on identifying and evaluating new opportunities in technologies applicable to network infrastructures, and the coordination of national and international collaboration activities. His current interests are related to network infrastructural services, new network architectures, and network programmability and virtualization.