Demonstration of SDN Orchestration in Optical Multi-Vendor Scenarios

V. López[1], O. González de Dios[1], L.M. Contreras[1], J. Foster[2], H. Silva[2], L. Blair[2], J. Marsella[2], T. Szyrkowiec[3], A. Autenrieth[3], C. Liou[4], A. Sasdasivarao[4], S. Syed[4], J. Sun[5], B. Rao[5], F. Zhang[5] and J.P. Fernández-Palacios[1]

[1] Telefónica I+D/GCTO, Distrito Telefonica, c/Ronda de la Comunicación s/n, 28050 Madrid (Spain). Email: victor.lopezalvarez@telefonica.com

[2] Ciena, 7035 Ridge Road Hanover, MD 21076 U.S.A.

[3] ADVA Optical Networking SE, Fraunhoferstr. 9a, 82152 Martinsried, Germany

[4] Infinera, 169 W Java Dr, Sunnyvale, CA 94089 U.S.A

[5] Huawei Technologies Co.,Ltd., 518129, Shenzhen, China.

Abstract: SDN brings automation to network operation and abstracts the complexity of optical networks. An orchestration layer is required to support multivendor interoperability scenarios. This work demonstrates that an ABNO architecture enables SDN controlled domain interoperability.

OCIS codes: (060.0060) Fiber optics and optical communications; (060.4250) Networks.

1. Introduction

Automation is an old aim of network engineers to reduce the operational complexity (mainly in optics) and enable capacity to easily adapt to changing traffic conditions. Typically, network operation is done using vendor-specific, proprietary, and sometimes closed Network Management Systems (NMS). National transport networks within the Telefonica Group have regions employing a different vendor in each region. One of the problems associated with using such an NMS in each region is that it is not possible to automate end-to-end (E2E) provisioning.

Software-defined networking (SDN) for optical transport networks (OTN) is gaining attention in research projects and standardization bodies. The Open Networking Foundation (ONF) recently published a solution brief on "OpenFlow-enabled Transport SDN" [1], which introduces the Control Virtual Network Interface (CVNI) for exchanging the virtualized view of transport network resources between client and provider using an open API. Similarly, the Application Based Network Operations (ABNO) architecture, which is an architectural consideration at IETF [2], is addressing network virtualization and orchestration in multiple scenarios. In addition to that, the Abstraction and Control of Transport Networks (ACTN) [3] initiative in IETF proposes the coordination of underlying transport layers and their presentation as an abstracted topology to the applications, facilitating the control following the SDN paradigm. One of the primary reasons for this considerable interest, both in the industry and academia, is due to the reduction in CAPEX and OPEX in large service provider networks. Dynamic configuration of network resources minimizes capital investments and the time it takes to configure the network.

At this initial stage of SDN productization, there is an explosion of new controllers in the industry, including several implementations for OpenFlow controllers (e.g. Floodlight, Trema, Nox, Ryu). Moreover, there is an important huddle around the OpenDaylight project, which not only can support OpenFlow, but also other protocols like Path Computation Element Protocol (PCEP) or vendor-specific NMS interfaces.

2. Network Orchestration and Abstraction

There are various challenges in orchestrating OTNs over multiple domains: heterogeneous control plane and transport technologies inside the domain as well as various communication interfaces and protocols for external access need to be considered. However, the aim of the operator is typically to set up E2E IP/Ethernet services for their customers. To solve such a problem, a hierarchical approach was proposed in [4], where an orchestrator is put in charge of end-to-end connectivity provisioning, using an abstracted view of the network and covering inter-domain aspects. Each domain has a separate controller that is responsible for the configuration of its own region. Every controller knows the vendor-specific details of its own underlying products and technologies, so each vendor can optimize transmission performance across the optical layer. Furthermore, the optical layer technology does not have to be the same across the different domains. One domain can have integrated OTN switching capabilities while another domain may use WDM or even flexgrid optical switching. The only important fact to the orchestrator is that the controller offers four key services: (i) Provisioning, (ii) Topology Discovery, (iii) Monitoring and (iv) Path Computation.

(i) *Provisioning* enables the set-up, tear down and modification of connections in the network. Its most basic feature is to set up a point-to-point connection between two locations. However, there are other characteristics that a client interface can have like (a) excluding or including nodes/links for traffic engineering, (b) defining the

		REST API 1	REST API 2	REST API 3	REST API 4
Provisioning	End Points	Supported	Supported	Supported	Supported
	Explicit Route	Roadmap	Roadmap	Supported	Supported
	Route Restrictions	Supported	Roadmap	Supported	Supported
	Service Protection	Supported	Roadmap	Roadmap	Supported
	Bandwidth	Supported	Roadmap	Supported	Supported
	Disjoint paths	Supported	Roadmap	Supported	Supported
Topology	Network Identifiers	Supported	Supported	Supported	Supported
	Physical Links	Supported	Roadmap	Supported	Supported
	Virtual Links	Supported	Supported	Roadmap	Roadmap
	Resource occupation	Supported	Roadmap	Supported	Supported
	Physical impairments	No use cases	Not supported	No use cases	Supported
Monitoring	Network status	Supported	Not supported	Supported	Supported
	Alarms, events	Roadmap	Not supported	Supported	Roadmap
	Path Computation	Supported	Roadmap	Supported	Supported

Table. 1. Main features in current REST APIs

protection level, (c) defining its bandwidth or (d) defining its disjointness from another connection. (ii) *Topology Discovery* requires, at a minimum, that the interface exports network topology information with unique identifiers. However, network identifiers (such as IPv4 or datapath-IDs) help to carry out path computation and to integrate the nodes for an end-to-end scenario. Further, the controllers can provide information about the links in the domain (physical or virtual), their utilization or even information about physical impairments, which the orchestrator may apply to a physical impairments computation model [5]. It is clear that the more information is shared, the less abstracted the network appears. So it is important to highlight that the North Bound Interface (NBI) of the orchestrator should be more abstract than the NBI of the controller. (iii) *Monitoring* the connections that have been created is very useful in a multi-controller scenario, where after a failure in one domain, the domain's controller may request another connection. (iv) *Global Path Computation* is a critical and fundamental feature because individual controllers in each domain are only able to share abstracted information that is local to their domain. An orchestrator with its global end-to-end view can optimize end-to-end connections that individual controllers cannot configure. Without a path computation interface, the orchestrator is limited to carrying out a crank-back process.

Table 1 shows the main features of the REST APIs from each of the controllers used in this activity. Based on the information in the table, it is clear that there is not a homogeneous implementation of the NBI. This happens not only in the commercial environment, but also in the standardization process. There are some efforts within ONF with the CVNI or the NBI but there is no clear solution for a common API that would provide a consistent interface to the controllers of each vendor. From the research community, the EU FP7 STRAUSS project proposes the so-called "Control Orchestration Protocol" (COP). The aim of the COP protocol is to define a solution to enable a standardized communication between orchestrator and controllers, but it is not defined in any standardization fora.

3. Application-Based Network Operations Architecture

This demonstration follows a SDN architecture based on the Application Based Network Operations Architecture (ABNO) framework, defined in the IETF [2]. This framework groups together a number of standard components and protocols able to control and manage the network in a coordinated way, taking into consideration requests from external applications. The ones relevant for this experiment are: (i) the *ABNO Controller* which is the module

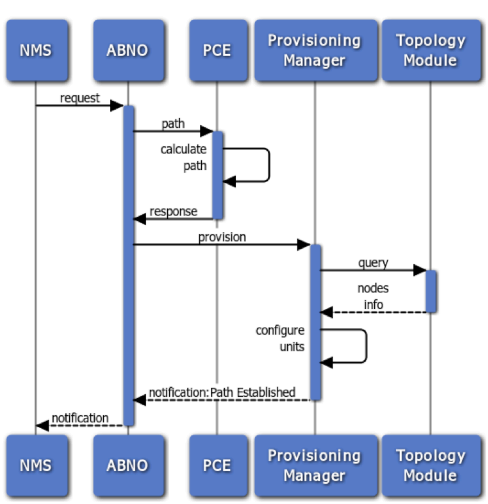

Fig. 2. ABNO interactions for optical service provisioning use case

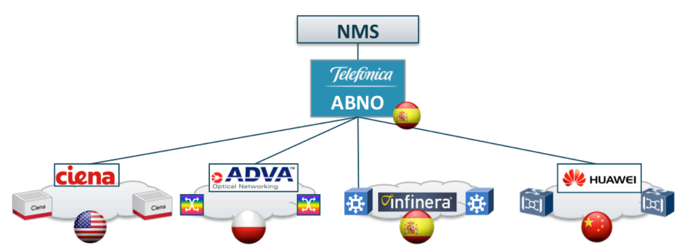

Fig.1. Hierarchical architecture using ABNO as orchestrator

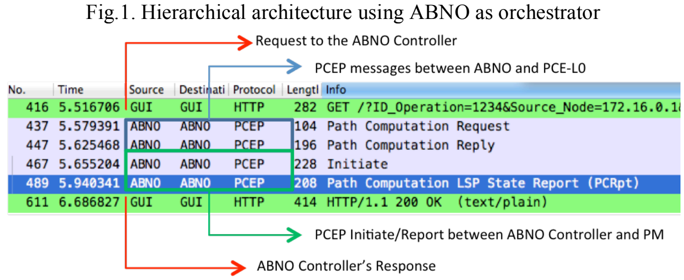

Fig. 3. Message flow in optical service provisioning use case.

executing the workflows and coordinating the individual actions of the rest of ABNO components; (ii) the *Path Computation Element* (PCE) which is in charge of providing the best path in a domain, according to specific constraints; (iii) the *Topology Module* (TM) that maintains an updated view of the topology graph; and, (iv) the *Provisioning Manager* (PM) which directly interacts with the network elements (either directly or via an intermediate control plane element).

The multi-domain scenario described here requires interaction with different Traffic Engineering Databases (TEDs), one per network domain, in such a way that multi-domain path computation can be performed.

In this demonstration the interaction with the network elements is performed through a dedicated controller per domain that is in charge of each vendor's network element configuration. To do that, the PM is adapted to interact with each of those controllers through the specific REST API offered by each of them.

4. Experimental Demonstration for Multi-vendor Orchestration

Telefonica's implementation of the ABNO architecture [6] has been extended to interface with the REST APIs of each vendor. Two modules are impacted by this adaptation: the provisioning module (PM) and the topology module (TM). The PM enables the configuration of each technology, while the TM retrieves the view of each domain. To do the experiment, four domains were connected: two Ciena 5400 OTN switches were located in the United States, four ADVA FSP 3000 optical nodes were located in Poland, two Huawei OSN 8800 T32 WDM/OTN nodes were located in China, and three Infinera DTN-X nodes were located in Telefonica's lab in Madrid (Fig. 1). Ciena's Multilayer WAN Controller (MLWC), Infinera's SDN Open Transport Switch, ADVA's optical network hypervisor and Huawei's T-SDN Controller are used in each region respectively. The ABNO orchestrator in Telefonica's lab interfaced with the vendors' controllers located in Ciena's and ADVA's labs using VPN tunnels.

Fig. 2 explains the interactions between the modules to provision an optical connection and Fig. 3 illustrates the packet capture trace showing the PCEP and HTTP messages exchanged in one execution (some messages like KeepAlive and Open messages are omitted to improve readability). The operator requests a new optical connection from the ABNO orchestrator using its North Bound Interface (NBI). This request contains the operation type or workflow (optical connection), its ID and the parameters for this operation (source and destination). The ABNO orchestrator sends a PCRequest message to the PCE for a path between the two optical nodes, and the PCE sends a PCResponse with the computation. Once the ABNO orchestrator knows the path, it requests the Provisioning Module (PM) to configure the connection. Note that the PCE is stateless and does not have instantiation capabilities in this experiment. The PM receives a PCInitiate message that contains the Explicit Route Object (ERO) with the whole path. The PM queries the TM to obtain the information about each node along the path. With this information, the PM knows the vendor and the technology for each node. There is a node type field in the request to the TM, where the vendor is specified. Based on the vendor and the technology, the PM selects the specific REST/API commands for each vendor. The configuration is performed per-domain, so the node sequence in the ERO is split into a request for each vendor domain. Once each segment of the path is set up the PM notifies the ABNO orchestrator using a PCReport message and the ABNO orchestrator notifies the NMS as well.

5. Conclusions

End-to-end interoperability between vendors is fundamental for network operators to take advantage of all features of SDN. The main contribution of this work is the experimental demonstration of end-to-end provisioning using an ABNO orchestration architecture with commercial SDN solutions (Ciena, Adva, Huawei and Infinera) across multiple control domains, and the validation of PCEP extensions to support the set-up of multi-domain connections. Thanks to this work, we showed that the operational complexity of using multiple configuration technologies with different vendors can be significantly reduced by the combination of ABNO orchestration and SDN control. However, a standard protocol is required to support this functionality, ONF or IETF within the ACTN can work on this direction.

6. Acknowledgements

The research is partially funded by the European Community's in the STRAUSS project (608528). Ciena authors would like to acknowledge Tim Ford, Amit Lad, Nadeem Akhtar, and David Boymel. This proof-of-concept was made possible through their commitment and support.

7. References

[1] M. Zimmerman, et al., "Openflow-enabled SDN and network functions virtualization," ONF, Feb 2014.
[2] D. King et al., draft-farrkingel-pce-abno-architecture-11.
[3] D. Ceccarelli, et al., draft-ceccarelli-actn-framework-01
[4] Y. Yuki, et al., PDP Th5A.2, OFC 2014.
[5] P. Poggiolini, J. Lightwave Technology, 30, 3857 (2012).
[6] A. Aguado, et al., Th3I.5, OFC 2014.
[7] R. Muñoz, et al., "Network Virtualization, Control Plane and Service Orchestration of the ICT STRAUSS Project" EUCNC 2014.