Experimental Assessment of ABNO-based Network Orchestration of end-to-end Multi-layer (OPS/OCS) Provisioning across SDN/OpenFlow and GMPLS/PCE Control Domains

Raul Muñoz(1), Ricard Vilalta(1), Ramon Casellas(1), Ricardo Martínez(1), Frederic Francois(2), Mayur Channegowda(2), Ali Hammad(2), Shuping Peng(2), Reza Nejabati(2) and Dimitra Simeonidou(2), Noboru Yoshikane(3), Takehiro Tsuritani(3), Victor López(4), Achim Autenrieth(5)

(1) CTTC, Av. Carl Friedrich Gauss 7, 08860 Castelldefels (Barcelona), Spain, raul.munoz@cttc.es
(2) University of Bristol, Bristol, BS8 1TR, United Kingdom
(3) KDDI R&D Laboratories, 2-1-15 Ohara, Fujimino, Saitama, 356-8502 Japan.
(4) Telefónica I+D, Don Ramon de la Cruz 84, Madrid, Spain.
(5) ADVA Optical Networking, Martinsried (Munich), Germany.

Abstract *We present and experimentally assess in an international testbed an ABNO-based network orchestrator for end-to-end multi-layer (OPS and Flexi-grid OCS) and multi-domain provisioning across heterogeneous control domains (SDN/OpenFlow and GMPLS/Stateful PCE) employing dynamic domain abstraction based on virtual node aggregation.*

Introduction

When a physical infrastructure is comprised of multiple domains (e.g., provided by different vendors) with heterogeneous optical transport technologies and control plane technologies, an orchestration mechanism is required to enable the provisioning of end-to-end connectivity services. The ICT STRAUSS project is developing a network orchestration layer, using a SDN orchestrator based on the Application-Based Network Operations (ABNO)[1,2] to enable the seamless interworking between GMPLS/PCE and SDN/OpenFlow control plane entities for end-to-end provisioning of dynamic connectivity services across the targeted multi-layer (OPS and Flexi-grid OCS) and multi-domain network. The first proof-of-concept prototype of an ABNO-based orchestration for multiple SDN/OpenFlow controllers was presented in[3]. In that work, the ABNO-based orchestrator had a full view of the physical topology (i.e., node and links) of each domain.

Since this approach lacks of scalability (for a very large number of nodes) and confidentiality (SDN/OpenFlow controllers may not disclose internal topology within a domain), in this paper we extend the ABNO-based orchestrator to deal with abstracted views of the topology of each domain. In addition, we also consider, for the first time, the orchestration of end-to-end connectivity services across not only multiple SDN/OpenFlow controllers but also with GMPLS-controlled domains with active stateful PCE. Fig.1 shows the available transport and control plane technologies of each partner's network domain involved in the international testbed of the ICT STRAUSS project.

ABNO-based Orchestrator Architecture

Fig.1 presents also the six building blocks of the ABNO architecture that are required to support the multi-domain and multi-layer network orchestration considered in this paper. The ABNO controller runs the workflows and can interwork with the different blocks. The Topology Server recovers the topology exposed by each SDN/OpenFlow controller's North-Bound Interface (NBI) or the BGP-LS speaker of the GMPLS/PCE domain. The PCE handles the path computation across the network graph provided by the Topology Server and it has been extended to deal with OpenFlow datapath identifiers[4]. The Provisioning Manager is responsible for the actual flow establishment request to the OpenFlow controllers through each specific controller's NBI, and to the active stateful PCE[5] of the GMPLS domains. The Flow server is responsible for storing the state of the provisioned flows. Finally, the virtual network topology manager (VNTM) is responsible to coordinate in multi-layer networks the layered establishment of server connections (i.e., OCS) and its promotion as logical link in the client layer (i.e., OPS).

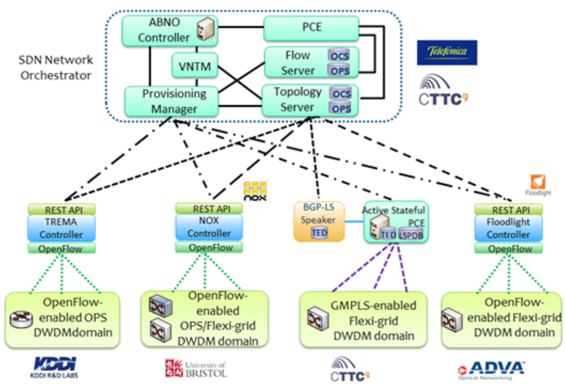

Fig. 1: International Testbed of the STRAUSS project

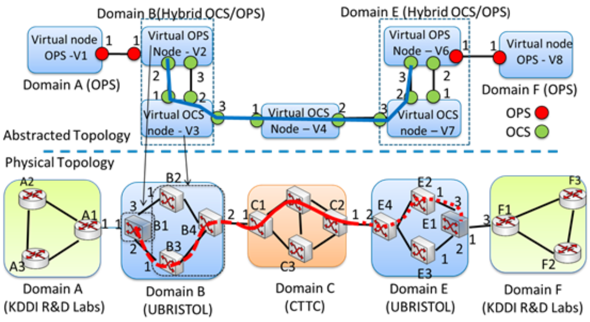

Fig. 2: Abstracted and physical network view

Domain abstraction: Virtual node

The Topology Server builds an abstracted topology of the domains based on the virtual node aggregation mechanism. This abstraction mechanism hides internal connectivity issues by representing each domain as a virtual node. In the example of Fig.2, each domain is seen as a node for the ABNO's Topology Server, with the exception of the two hybrid OPS/OCS domains (Domain B and E), where each one is represented with two virtual nodes, one for each switching technology. Then, the border nodes of each domain are seen as ports of the virtual nodes, and are connected through inter-domain links. Each SDN/OpenFlow controller or the BGP-LS speaker in the GMPLS domain is responsible for performing the actual mapping of the virtual node/ports with the real nodes/ports, and to expose the virtual topology (node_id, port_id, port_type, supported range of OPS Labels or flexi-grid nominal central frequencies.

Logical OPS link creation

Let us consider that we need to provision an OPS connection between a pair of OPS nodes from Domain A and F (Fig.2). Since there is no OPS connectivity between Domain B (V2) and Domain E (V6), the ABNO controller requests to the VNTM the creation of a new logical OPS link between virtual nodes *V2* and *V6* (step 1). First, the VNTM requests to the PCE a Flexi-grid OCS path from *V2* to *V6* for a defined bandwidth based on the ABNO's policy (step 2). In this case, the PCE can find a path solution across Domains B, C and E (Fig. 2) and it replies with the spatial path (i.e., virtual nodes and links), but also assigning a frequency slot to each virtual link corresponding to inter-domain links. In this scenario we have made the assumption that each domain border node is equipped with full 3R regeneration and therefore, no spectrum continuity constraint must be satisfied from end-to-end. Then, the ABNO controller requests to the Provisioning Manager the setup of the end-to-end OCS flow for the computed path (step 3). The first action performed by the Provisioning Manager is to identify the domains that will be involved in the actual provisioning of the end-to-end OCS flow. Then, the second action is to

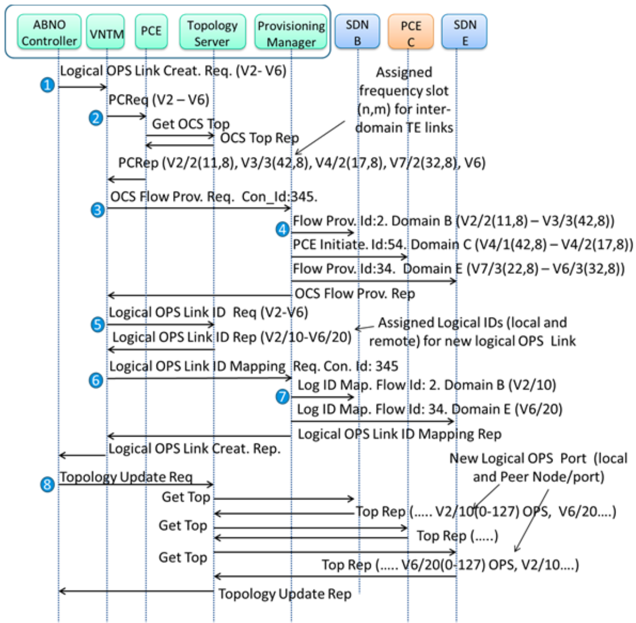

Fig. 3: Workflow for a Logical OPS Link creation (V2 and V6)

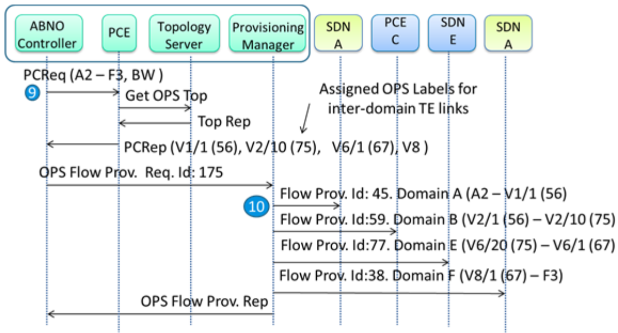

Fig. 4: Workflow for an OPS flow provisioning (A2-F3)

segment the received path into the corresponding domains. For each domain segment, the Provisioning Manager must also identify the input virtual node/port, and the output virtual node/port, as illustrated in Fig.3. Finally, the Provisioning Manager requests the actual provisioning of the domain segments using each specific SDN controller's Restful API NBI or the PCEP interface for the instantiation of LSPs through an active stateful PCE in the GMPLS domain (step 4). Each domain must then map the received virtual nodes/ports endpoints into real nodes/ports (e.g., *V2/2 – V3/3* in Domain B to *B1/3 – B4/2*) and expand the received segment, that is, to compute an explicit path based on the real topology information of the domain, as well as to assign the required frequency slot to satisfy the bandwidth requirements.

Once all domain segments have been successfully provisioned, the established OCS Flow can be then used as a data link for the OPS layer. To create a logical OPS link associated to the OCS flow that can be used by the PCE when computing paths and performing traffic grooming (i.e., multiple OPS flows can be may be grouped over a single OCS flow), the VNTM requests to the Topology Server an

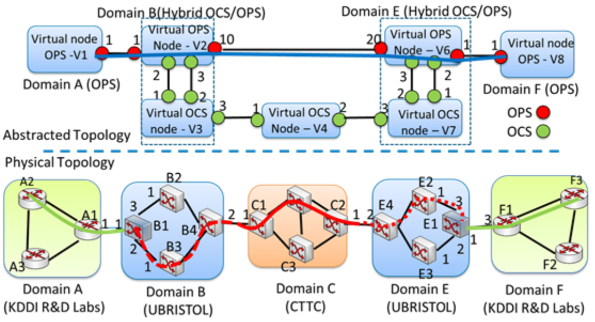

Fig. 5: Abstracted/physical topology with the logical OPS link

available port identifier for node *V2* and *V6* (step 5). These logical identifiers will be used to unambiguously identify the created logical OPS link. Then, the VNTM requests to the Provisioning Manager (step 6) the mapping of the provisioned Flexi-grid OCS flow and the logical OPS link, specifying the assigned port identifiers (i.e, *V2/10* and *V6/20*). The Provisioning Manager identifies the domains associated to each virtual node, and sends a request to map the assigned logical port to the established OCS flow at the specified node (step 7). To this end, it has been required to extend the RestFul API of the SDN controller located in Domains B and E with a new command: add_port(dpath, port, flow_id, type, peer_dpath, peer_port). Once the mapping has been performed successfully, the VNTM notifies the ABNO controller, and it requests the Topology Server to update its topology databases (step 8) to learn about the new logical OPS link. This trigger is required because the RestFul API interfaces of the SDN controller do not support asynchronous notification of changes in the network topology, unlike BGP-LS or OSPF-TE routing protocols.

OPS flow provisioning

The ABNO controller requests to the PCE the computation of a path between the OPS nodes A2 and F3 (Fig.4). Now, the PCE can compute a path through virtual nodes *V1, V2, V6* and *V8* using the new logical OPS link (*V2/10-V6/20*), as observed in Fig. 5 (blue line). Thus, the PCE replies the spatial path and the assigned OPS labels to each inter-domain link. As mentioned before, the assignment of either the frequency slot or the OPS labels for the virtual links corresponding to inter-domain links is responsibility of the ABNO-based orchestrator. Then, the ABNO controller requests to the provisioning of the computed OPS path to the provisioning Manager, performing exactly the same actions as in previous described step 3.

Experimental Assessment

We have developed a proof-of-concept prototype of the ABNO-based orchestrator in the international testbed of the STRAUSS project shown in Fig.1, in order to experimentally assess, at the control plane level (i.e., no

No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.0.34.30	10.0.34.34	PCEP	PATH COMPUTATION REQUEST MESSAGE
2	1.748137	10.0.34.34	10.0.34.30	PCEP	PATH COMPUTATION REPLY MESSAGE
3	1.798084	10.0.34.30	10.0.34.6	HTTP	GET /set_flow/?dpath_src=00-00-00-00-00-00-FF
4	1.864323	10.0.34.30	10.0.34.14	PCEP	PATH COMPUTATION INITIATE MESSAGE
5	3.667313	10.0.34.30	10.0.34.6	HTTP	GET /set_flow/?dpath_src=00-00-00-00-00-00-FF
6	3.760976	10.0.34.30	10.0.34.6	HTTP	GET /add_port/?dpath_src=00-00-00-00-00-00-FF
7	3.854475	10.0.34.30	10.0.34.6	HTTP	GET /add_port/?dpath_src=00-00-00-00-00-00-FF
8	3.917795	10.0.34.30	10.0.34.34	PCEP	PATH COMPUTATION REQUEST MESSAGE
9	5.674728	10.0.34.34	10.0.34.30	PCEP	PATH COMPUTATION REPLY MESSAGE
10	6.032277	10.0.34.30	10.0.34.10	HTTP	GET /set_flow_action/?%7B%22flow_id%22:%221%2
11	6.755098	10.0.34.30	10.0.34.26	HTTP	GET /set_flow_action/?%7B%22flow_id%22:%221%2
12	7.165105	10.0.34.30	10.0.34.6	HTTP	GET /set_flow/?dpath_src=00-00-00-00-00-00-FF
13	7.261782	10.0.34.30	10.0.34.6	HTTP	GET /set_flow/?dpath_src=00-00-00-00-00-00-FF

Fig. 6: Whireshark capture at the ABNO controller

hardware configuration is performed at the domains) the end-to-end multi-layer and mult-domain provisioning across heterogenours SDN/OpenFlow controllers and GMPLS/PCE domains. In this experimentation, all the modules of the ABNO-based orchestrator, with the exception of the PCE (IP:10.0.34.34), have been implemented in a single server (IP:10.0.34.30). Domain's controllers are connected to the ABNO-based orchestrator through OpenVPN. Five domains are involved in the experimentation: Two OpenFlow-controlled OPS domains from KDDI R&D Labs (IP: 10.0.34.10 & 10.0.34.26), two OpenFlow-controlled OPS/OCS domains from BRISTOL (IP: 10.0.34.6 port 8080 and 8081) and a GMPLS-controlled OCS domain with active stateful PCE from CTTC (10.0.34.14), as shown in the network topology of Fig.2. First, we request to provision an OPS flow from A2 to F3, requiring to create a logical OPS link (as shown in the previous example). Fig.6 shows a Wireshark capture at the ABNO controller with the involved workflow: OCS path computation to the PCE (step 2 Fig.2), OCS flow provisioning for each domain (step 4), Logical OPS link ID mapping (step 7), OPS path computation to PCE (step 9), and OPS flow provisioning per domain (step 10). The provisioning time is above 7 seconds in average. We have also test the provisioning of multi-layer and multi-domain connections that reuse the provisioned OCS flow (i.e., traffic grooming), and the provisioning time is reduced to 3 seconds in average.

Conclusions

This paper has experimentally assessed the SDN-based ABNO architecture to perform the orchestration of end-to-end multi-layer provisioning in a multi-vendor environment, where each domain has its own control plane.

Acknowledgements

EU FP7 STRAUSS (FP7-ICT-2013-EU-Japan 608528) and MINECO FARO (TEC2012-38119).

References

[1] D. King et al, "A PCE-based Architecture for Application based Network Operations", IETF draft, 2014.

[2] A. Aguado et al., Th3I.5, OFC 2014.

[3] Y. Yoshida et al., pdp Th5A.2. OFC 2014.

[4] R. Casellas et al., OW4G.2. OFC 2013.

[5] E. Crabbe, "PCEP Extensions for Stateful PCE", IETF draft 2014.