Dynamic Virtual Network Reconfiguration over SDN Orchestrated Multi-Technology Optical Transport Domains

A. Aguado[1], S. Peng[1], M. V. Alvarez[2], V. Lopez[2], T. Szyrkowiec[3], A. Autenrieth[3], R. Vilalta[4], R. Muñoz[4], R. Casellas[4], R.Martínez[4], N. Yoshikane[5], T. Tsuritani[5], R. Nejabati[1], D. Simeonidou[1]

(1) High Performance Networks Group, University of Bristol, a.aguado@bristol.ac.uk

(2) Telefónica I+D, Don Ramon de la Cruz 84, Madrid, Spain.

(3) ADVA Optical Networking, Fraunhoferstr. 9a, 82152 Martinsried/Munich, Germany

(4) Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Castellefels (Barcelona), Spain

(5) KDDI R&D Laboratories, Fujimino, Saitama, Japan

Abstract: *Application-based Network Operations consists of existing components offering an application-driven network management. We validate its applicability to multi-tenant networks in multi-technology optical domains based on congestion detection and failure recovery by demonstrating fast reconfiguration, while keeping the upper layer unaware.*

Introduction

Optical transport networks are emerging as a key solution to the ever-growing demand for supporting new applications and services (e.g. High Definition Video Streaming, etc.), which require high bandwidth and low latency network connectivity. Optical packet switching (OPS) fits into metro access networks and significantly reduces the number of O/E/O conversions. Elastic optical networks (EONs), based on flexi-grid optical circuit switching (OCS), are able to provide flexible bandwidth. On the other hand, different sets of network control protocols (such as OpenFlow (OF) and Generalized Multi-Protocol Label Switching (GMPLS)) have been defined to enable/enhance network programmability. The Application-Based Network Operations (ABNO) architecture [1] aims to provide network orchestration over multiple domains and technologies, whilst using standard protocols. A Multi-domain Network Hypervisor (MNH) [2] that runs on top of the ABNO architecture provides an abstracted and virtual view and requests end-to-end (E2E) connections from the ABNO.

This paper presents the experimental assessment of the applicability of the ABNO architecture for two different use cases: (i) Failure recovery and (ii) Congestion detection. For the first time, both are applied over multi-technology optical networks. We demonstrate the replanning of network services previously requested by the MNH, thus making the virtualization layer unaware of any network reconfiguration. These results show the capabilities of the ABNO architecture, which is able to reconfigure E2E connections. Further we show the capabilities to create new E2E backup paths when an existing service is moved to the current backup path.

Architecture and Interfaces

The ABNO architecture is composed of a set of well-defined modules that encapsulate different network functionalities. They utilize standard protocols and interfaces for both internal (PCEP, RESTful) and external communications (RESTful NBI and SBI, GMPLS protocols such as PCEP and BGP-LS). The main components used in this work are shown in Fig. 1.

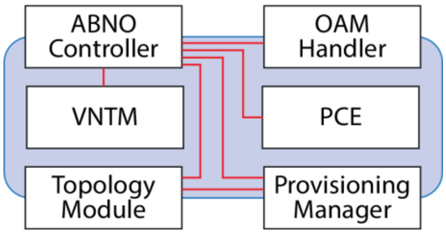


Fig. 1 ABNO Architecture (Main Modules)

The ABNO controller is the main gateway to the architecture. It is able to receive requests through a REST interface and orchestrates the other modules in the architecture to map the incoming request to a specific workflow. The PCE is responsible for the calculation of the path through the network, based on the current status and the requirements of the request. The Virtual Network Topology Manager (VNTM) is the module that stores the multilayer information, sets up or tears down lower-layer Label Switched Paths (LSPs) as well as upper-layer virtual links consequently. The Topology Module has at least one topology database with information about the available network elements such as node types and addresses, controller types and their address, edges, port information, etc. It can gather information through different protocols and interfaces (BGP-LS, OSPF-TE and/or REST APIs from different SDN controllers). The Provisioning Manager is the module that handles the E2E path by configuring the controllers. It splits the explicit route object (ERO) according to the different domains, sending the configuration using different protocols, i.e. OpenFlow, RESTful and/or PCEP Initiate for GMPLS domains.

Use Cases Context

1. *Failure Recovery*

The ABNO architecture is proposed as a solution for orchestrating networks that have multiple controllers and is able to handle different network events. In this context, the ABNO architecture is able to receive failure alarms and start recovery workflows to create new E2E paths. Moreover, ABNO can create and provision a new backup connection after the first failure is detected. The Operation, Administration and Maintenance (OAM) handler receives failure alarms and triggers an internal workflow, and allows the ABNO controller to obtain information about the affected services and to configure new E2E connections.

2. *Congestion Detection and Replanning*

A set of monitoring tools has been created for detecting congestion in network interfaces. We have also developed a tool that acts as an application sitting on top of the SDN controller to retrieve flow information and statistics. It builds a bandwidth usage graph (see Fig.6) that allows setting adaptive threshold to control the traffic load. In the event that the threshold is exceeded, the application notifies the OAM handler, thereby starting a replanning workflow, moving the affected service to less congested interfaces and/or network domains.

Experimental Setup and Demonstration

The experimental setup is shown in Fig. 2, where Bristol, KDDI and ADVA domains are OF-enabled and controlled by various SDN controllers (i.e. ODL, POX), while CTTC is a GMPLS domain managed by an active and stateful PCE. The controllers hide the internal setup of each domain from the TID's ABNO [4], allowing an abstract view of the experimental setup (domain as a single node). The abstract topology consists of two OPS domains (KDDI and Bristol) and two flexi-grid OCS domains (ADVA and CTTC). At the edge, an SDN-enabled opto-electrical interface [3] was implemented for interconnecting the OPS and flexi-grid OCS domains. Such interface can be seen as a L2 switch that retrieves statistics from packet counting. The E2E paths through multiple domains are computed, delegating the internal computation to each network controller.

Use case 1: Failure Recovery

When the network is up and running, multiple virtual networks requested by different tenants have already been set up and E2E connections have been established. The SDN controller (e.g. POX in KDDI) monitors the traffic via optical power meters, therefore it is aware of the status of its own domain. Once a failure is detected at an interface, the SDN controller sends a POST message to the OAM handler in the ABNO. The alarm is sent in a Javascript object notation (JSON), with the following format:

{"event":"alarm", "id":"PORT_FAILURE", "body": {"dpid":"00:00:00:00:00:00:FF:01","port":"3"}}

The OAM handler receives the JSON, parses the event, and maps it to a specific ABNO workflow.

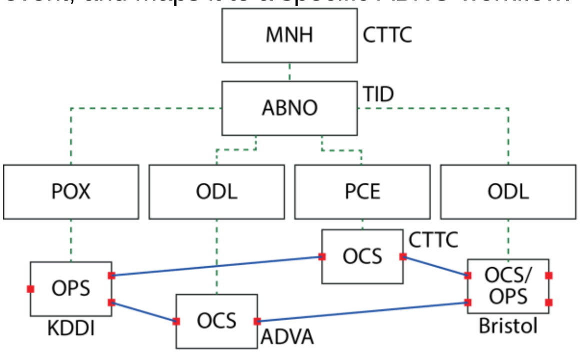


Fig. 2 Experimental Setup

The ABNO controller module gets all the affected E2E connections from the service table and sends a PCEP request message with a sub-object extension of the exclude route object (XRO). The XRO contains the information about the failed interface or node, so that the PCE can exclude this resource and calculate the new paths. We use XRO because updating the PCE TED through an REST API or IGP discovery could take more time. The ABNO controller receives the PCEP response and then sends the new path information to the provisioning manager which in turn configures the new path. Fig. 3 shows the message flow among the modules of the ABNO. For instance, referring to Fig. 2, if there are flows from KDDI to Bristol through CTTC, which fails, all the flows will be reconfigured to the other domain (i.e. ADVA) and services will be restored. Fig. 4 shows the exchange of messages for both use cases.

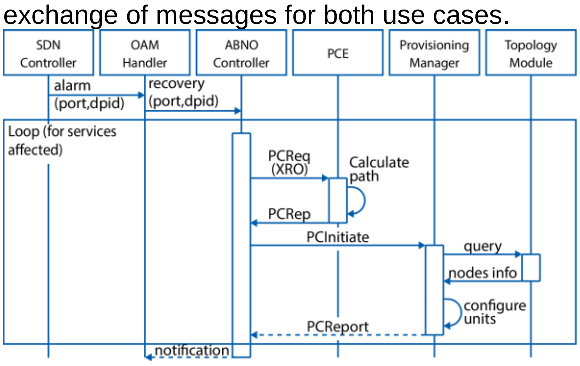
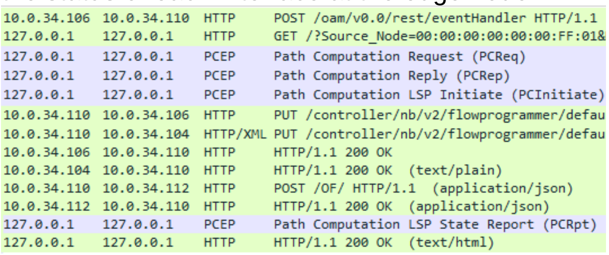


Fig. 3 Failure Recovery Workflow

Use case 2: Congestion Detection and Replanning

We assume an initial network configuration with several E2E connections between the OPS domains. End users start generating traffic using available network services. Our aforementioned packet-counting-based network monitor application sits on top of the SDN controller (e.g. ODL-based in Bristol), which is able to get traffic statistics from the edge interfaces (seen as L2 switches by the controller). It builds a graph with the status of each interface at the edge node.



10.0.34.106	10.0.34.110	HTTP	POST /oam/v0.0/rest/eventHandler HTTP/1.1
127.0.0.1	127.0.0.1	HTTP	GET /?Source_Node=00:00:00:00:00:00:FF:01&
127.0.0.1	127.0.0.1	PCEP	Path Computation Request (PCReq)
127.0.0.1	127.0.0.1	PCEP	Path Computation Reply (PCRep)
127.0.0.1	127.0.0.1	PCEP	Path Computation LSP Initiate (PCInitiate)
10.0.34.110	10.0.34.106	HTTP	PUT /controller/nb/v2/flowprogrammer/defau
10.0.34.110	10.0.34.104	HTTP/XML	PUT /controller/nb/v2/flowprogrammer/defau
10.0.34.106	10.0.34.110	HTTP	HTTP/1.1 200 OK
10.0.34.104	10.0.34.110	HTTP	HTTP/1.1 200 OK (text/plain)
10.0.34.110	10.0.34.112	HTTP	POST /OF/ HTTP/1.1 (application/json)
10.0.34.112	10.0.34.110	HTTP	HTTP/1.1 200 OK (application/json)
127.0.0.1	127.0.0.1	PCEP	Path Computation LSP State Report (PCRpt)
127.0.0.1	127.0.0.1	HTTP	HTTP/1.1 200 OK (text/html)

Fig. 4 Failure recovery (one single service) and congestion replanning capture

Once an interface exceeds the predefined load threshold (configured by the domain infrastructure provider), the application sends a JSON message. The message specifies the overloaded interface that should be avoided by the PCE in the computation of the replanned path and the flow information to detect the affected E2E service. The JSON format is:

{"event":"alarm", "id":"CONGESTION", "body": {"dpid":"00:00:00:00:00:00:30:04","port":"3",“flow ”:{“dpid”:”00:00:00:00:00:00:03:04” ,”ingressPort”:”1”,”output”:”3”}}

The OAM handler detects the alarm and sends the event to the ABNO controller which will then start a replanning workflow and obtain the network service that matches the flow from the service table. A single PCEP request with the XRO is sent to the PCE, which avoids the overloaded interface. The PCEP response with the E2E path is forwarded from the ABNO controller to the provisioning manager, which splits the path according to the boundaries of each traversed domain and then sends the configuration to the responsible controllers. Fig. 5 shows the internal workflow among the modules inside the ABNO.

Fig. 6 shows the load of the interface over time. When the first flow starts generating more traffic, the total load of the interface overpasses the threshold. This event is detected by the application. The application then automatically starts the appropriate workflow through the ABNO using the OAM handler’s RESTful server. Figs. 6(a) and (b) show how the flow that generates the highest amount of the traffic (flow from interface 2 to 3) is moved from one interface to the other available one (number 4) within the OPS domain.

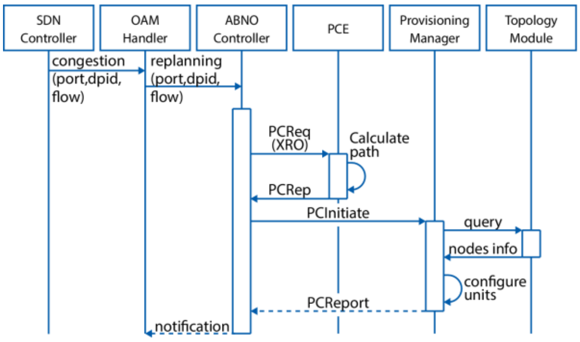


Fig. 5 Congestion detection and replanning workflow

Conclusions

This work for the first time presents two scenarios: failure recovery and congestion replanning over SDN orchestrated multi-technology optical domains. The virtual network replanning is hidden from the virtualization layer as well as the end user. This work validates the ABNO architecture as a key part in replanning multi-technology optical network domains.

Acknowledgements

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

References

[1] D. King and A. Farrel “A PCE-Based Architecture for Application-Based Network Operations”, RFC 7491.

[2] R.Vilalta et al., “Network Virtualization Controller for Abstraction and Control of OpenFlow-enabled Multi-tenant Multi-technology Transport Networks” OFC 2015.

[3] Y.Yoshida et al., “First international SDN-based Network Orchestration of Variable-capacity OPS over Programmable Flexi-grid EON” OFC 2014.

[4] A.Aguado et al., “ABNO: A feasible SDN Approach for Multivendor IP and Optical Networks”, JOCN 2015.

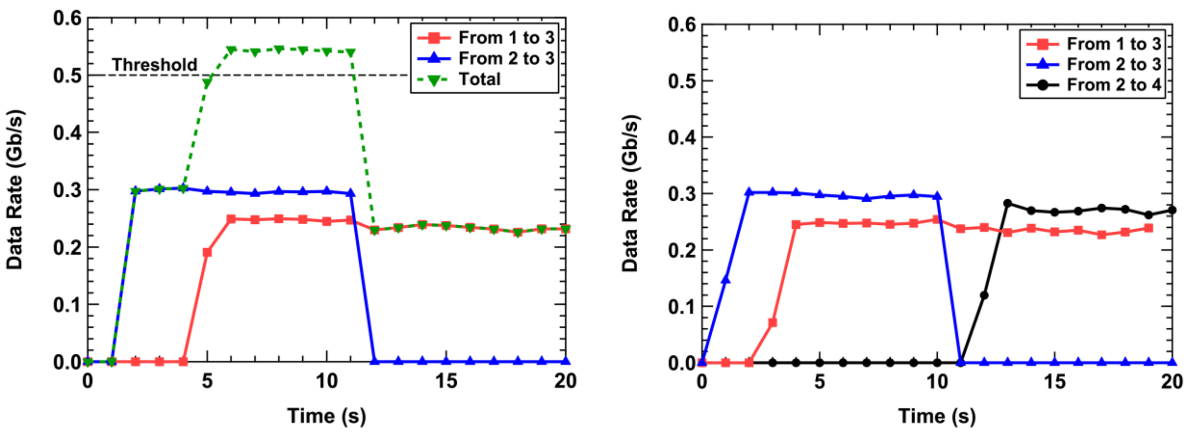


Fig. 6: a) Load of Interface 3 **b)** Load generated by all flows