Demonstration of Monitoring and Data Analytics-triggered reconfiguration in partially disaggregated optical networks

Ll. Gifre[1*], F. Boitier[1], C. Delezoide[1], M. Ruiz[2], M. Buffa[3], A. Morea[3], R. Casellas[4], L. Velasco[2], P. Layec[1]
[(1)] Nokia Bell Labs, Nozay, France; [(2)] Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain; [(3)] Nokia, Vimercate, Italy; [(4)] Centre Tecnològic Telecomunicacions Catalunya (CTTC), Castelldefels, Spain.
e-mail address: lluis.gifre_renom@nokia-bell-labs.com

Abstract: We demonstrate a novel agent for optical disaggregated optical networks. When the Monitoring and Data Analytics detects a degradation, it recommends the SDN controller to trigger a network reconfiguration computed by a novel planning tool.

1. Introduction

To build agile and highly reliable optical networks in a cost-efficient way, the telecom industry moved from fixed infrastructure defined at the "design phase" to flexible and reprogrammable hardware capable of supporting numerous operating cases. Software Defined Networks (SDN) controller provides network devices abstraction through an open SouthBound Interfaces (SBI) which enables centralizing control and management tasks in a single system. It offers an easy management interface for network operators to interact with different devices and reprogram them.

Furthermore, monitoring the states of network components and dynamically tuning their configurations could reduce margins and entail a reduction in CAPEX [1]. This scenario demands, not only an SDN controller for reprogramming the devices, but also a specialized Monitoring and Data Analytics (MDA) system [2] to collate monitored data from the network components and to apply novel data analytics techniques, like Machine Learning, to extract knowledge and to take appropriate decisions. The network can be reprogrammed in a proactive or reactive manner when undesired conditions are predicted (proactive) or detected (reactive). These automated control closed-loops also reduce OPEX and improve the network reliability and performance [3].

Network disaggregation has been proposed in the last years aiming at improving the agility and cost efficiency of optical networks. To achieve this goal, open control interfaces using device models and protocols were standardized, thus enabling coordination between multi-vendor equipment such as in Open Line Systems [4,5]. Nevertheless, when disaggregation level is high, each single component will expose its interface to the SDN controller leading to complex management of network elements, e.g. a N-degree ROADM will then be a combination of multiple filter and amplifier cards. Similarly, the dual problem arises when a single vendor shelf integrates components belonging to different network nodes, such as transponders and N-degree ROADM in a single shelf. However, up to now no generic open agent for disaggregated/aggregated network elements is available.

In this demonstration, we propose a novel open device agent, exposing to the SDN controller virtual network elements constructed from logical aggregation or disaggregation of physical network components, named as ADONIS, which stands for Aggregator/Disaggregator for Optical Network equIpmentS. We demonstrate ADONIS on a 4-node meshed optical network on a closed-loop scenario integrating monitoring and analytics with an SDN controller and a planning tool.

2. Innovation

This live demonstration will exhibit new prototype components integrated with previous systems, namely CASTOR MDA system [2] and ONOS SDN controller using the Open Disaggregated Transport Network (ODTN) drivers. The most relevant innovations of this demonstration are: *i*) to propose an open device agent prototype capable of aggregating multiple physical (commercial) devices into a single virtual component and to disaggregate a single physical (commercial) device into multiple virtual components; *ii*) to extend existing open data models and interfaces to control the virtual devices obtained thanks to the ADONIS agent; *iii*) to implement closed-loop scenarios to reconfigure the virtual devices after an MDA system identifies a degradation; *iv*) to integrate the novel SMART-A planning tool with the ONOS SDN controller to provide the required reconfigurations of the network; and *v*) to demonstrate the abovementioned components in a 4-node testbed with ROADMs from 2 different vendors.

3. Architecture

The architecture for the proposed cognitive network controller is depicted in Fig. 1a; the four main components are:

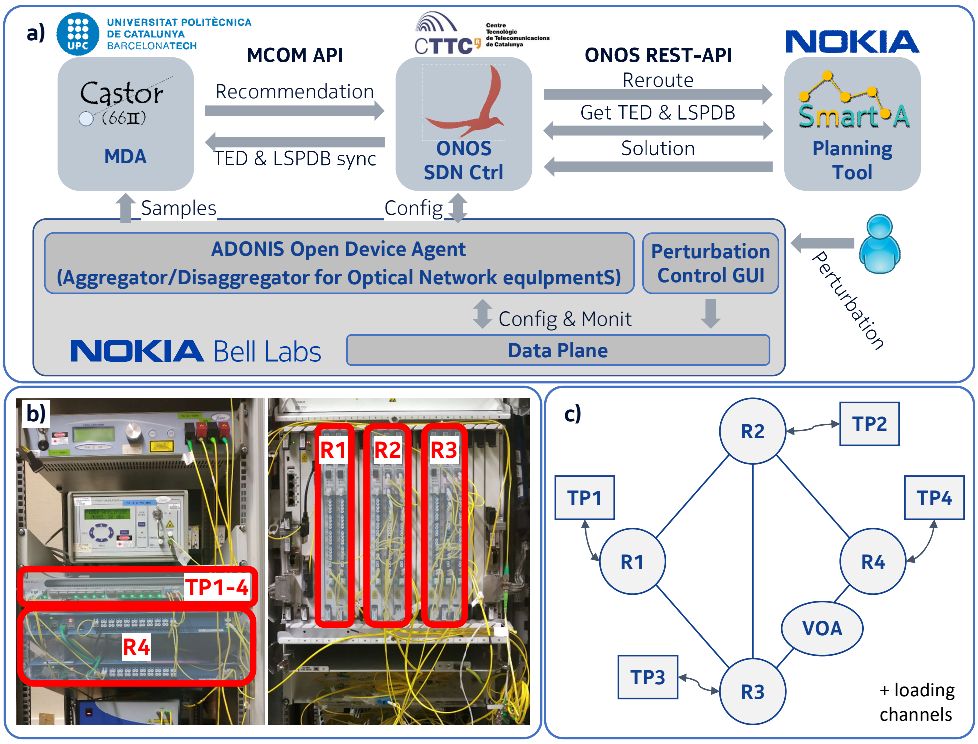

Figure 1. Proposed architecture: (a) software components of the cognitive network controller, (b) physical equipment and (c) virtual devices and network topology

i) The ADONIS open device agent abstracts the physical devices into virtual devices and interfaces them with ONOS. ADONIS is implemented in Python by Nokia Bell Labs. The NorthBound Interface (NBI) relies on NETCONF/YANG offering either OpenConfig [6] interfaces for transponders and OpenROADM [7] for ROADMs. We illustrate the behavior of ADONIS thanks to Fig. 1b showing the physical (commercial) data plane and to Fig. 1c depicting the virtual devices used in the 4-node network topology. The physical data plane consists of one Nokia 1830 PSI-2T 4-lineport transponder (Fig. 1b, middle left) configured by ADONIS as 4 independent virtual transponders (labeled TP1 to TP4 in Fig. 1c); two Lumentum RDM20 degree boxes (Fig. 1b, bottom left) as one virtual 2-degree ROADM (labeled R4); one Nokia 1830 PSS-32 shelf (Fig. 1b, right) containing 8 iROADM9R degree cards configured as 3 virtual ROADMs (one 2-degree node labeled R1, two 3-degree node labeled R2 & R3). For R1, R2 and R3, the ADONIS first needs to disaggregate the shelf into its individual cards and then to aggregate these cards into 3 virtual ROADMs. Finally, a Variable Optical Attenuator (VOA) is deployed in link R3-R4 to emulate optical fiber perturbations and is controlled by an independent GUI. Additionally, the testbed is loaded with some loading channels to emulate additional lightpaths.

ii) The CASTOR MDA system is in charge of collecting and processing monitored data from the network equipment and issuing appropriate recommendations to the SDN controller. It is implemented in Python by UPC following a micro-services architecture for the backend. In this demonstration, the MDA is extended with an application to implement the closed-loop. The M-COM interface [8] is used to populate the MDA Traffic Engineering Database (TED) and Label Switching Path Database (LSPDB) and to issue recommendations to the SDN controller. To build a monitoring system scalable and performant given the heterogeneity of network equipment and the volume of monitored samples, we choose the IPFIX protocol to conveyed data to the CASTOR SBI. We extended this protocol, designed to convey packet layer-related monitoring data, with additional fields, such as transmitted and received power, and Bit Error Ratio (BER), to reflect optical layer devices states.

iii) The ONOS SDN controller [9] manages the virtual devices forming the network topology. In this demonstration, we use the mainstream version of ONOS with the ODTN drivers which provides the NETCONF/YANG–based SBIs interfacing with ADONIS. An application implementing the M-COM interface is used to enable the coordination between CASTOR and ONOS. We implemented an additional application for ONOS to capture recommendations emitted by CASTOR and issue appropriate computation requests to SMART-A.

iv) The SMART-A planning tool, configured as a routing tool for this experiment, is developed by Nokia Italy in

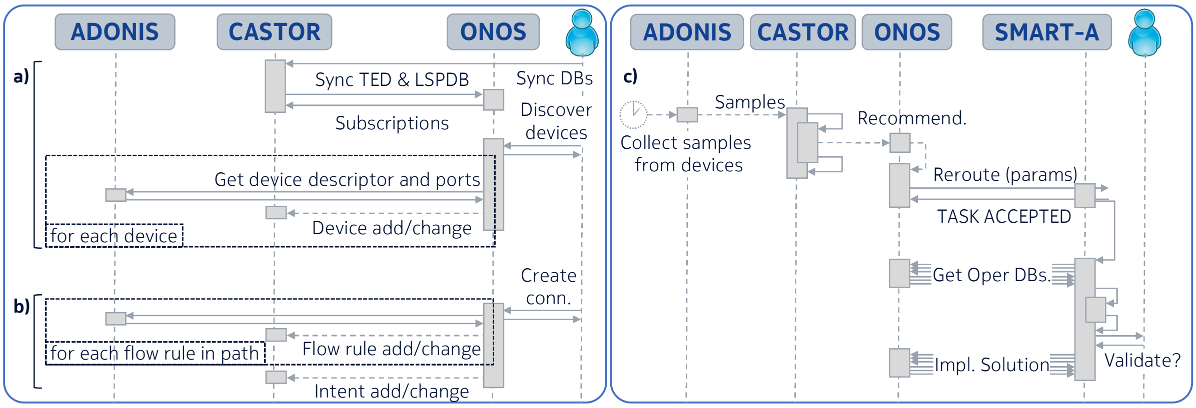

Figure 2. Proposed workflows: (a) initial set-up, (b) per-connection creation, and (c) closed-loop

C++. It is responsible for solving the optimization problems requested by ONOS, optionally asking the network operator for validation, and implementing the computed solution in the network through ONOS. It integrates a REST API to receive computation requests from ONOS and uses ONOS standard NBI to retrieve up-to-date information available regarding devices, links and connections, and to update connections after a solution is validated.

4. Demonstration

Three workflows, reproduced in Fig. 2, are defined for this demonstration. The first workflow (a), manually triggered by the network operator when the cognitive network controller is initialized, consists in an initial synchronization of the operational databases, i.e. TED and LSPDB, belonging to ONOS and CASTOR. Then, ONOS discovers virtual devices belonging to the network topology. For each device, it retrieves its details by means of ADONIS and issues an operational databases update notification to CASTOR. The second workflow (b), manually triggered by the network operator every time a new connection is created in ONOS, configures the appropriate rules in the devices along connection path and issues an operational databases update notification to CASTOR.

The third workflow (c) implements the closed-loop to be demonstrated. ADONIS periodically retrieves monitoring data samples from the physical devices, associates them to the appropriate virtual devices and forwards it to CASTOR by means of IPFIX messages; CASTOR stores the samples on its internal repository and forwards them to the soft failure detection and localization algorithm. The algorithm correlates the samples with the network components and analyzes trends to identify soft failures. In case of detecting a soft failure, it uses the computed correlation data to localize the origin of the soft failure and issues a reconfiguration recommendation to ONOS including, among others, the component(s) to be avoided. Upon ONOS receives CASTOR recommendation the former sends a request to SMART-A, which first retrieves from ONOS an up-to-date copy of the operational databases and solves the optimization problem to reroute the affected connections aiming at minimizing their reconfiguration downtime. If a solution is found, SMART-A optionally asks for validation to the network operator. When validated, SMART-A provides the appropriate requests to ONOS to implement the reconfiguration actions.

During this demonstration, we will have the opportunity to explain and show the ADONIS agent and the extended open data models and interfaces. The integration of the 4 key components validates the novel agent on commercial equipment in the framework of disaggregated optical network using monitoring and closed-loop reconfigurations.

The research leading to these results has received funding from the EC through the METRO-HAUL (G.A. nº 761727).

References

[1] S. Oda, et al., "A Learning Living Network With Open ROADMs", IEEE/OSA JLT, vol. 35, pp. 1350-1356, 2017.
[2] Ll. Gifre et al., "Autonomic Disaggregated Multilayer Networking", IEEE/OSA JOCN, vol. 10, pp. 482-492, 2018.
[3] D. Rafique, et al., "Cognitive Assurance Architecture for Optical Netowk Fault Managenent", IEEE/OSA JLT, vol. 36, pp. 1443-1450, 2017.
[4] N. Sambo, et al., "Experimental Demonstration of a Fully Disaggregated and Automated White Box Comprised of Different Types of Transponders and Monitors", IEEE/OSA JLT, vol. 37, pp. 824-830, 2018.
[5] F. Paolucci, et al., "OpenConfig Control of 100G/400G Filterless Metro Networks with configurable Modulation Format and FEC", in Proc. IEEE/OSA Optical Fiber Communication Conference (OFC), 2019.
[6] OpenConfig [online] available 2019. http://openconfig.net/
[7] OpenROADM MSA [online] available 2019. http://www.openroadm.org/
[8] L. Velasco et al., "Building Autonomic Optical Whitebox-Based Networks", IEEE/OSA JLT, vol. 36, pp. 3097-3104, 2018.
[9] Open-source Network Operating System (ONOS) project [online] available 2019. https://onosproject.org/