Enriching Intent-based SDN to Ease Customer Service Management in Transport Networks

Antonio Marsico[1], Mohit Chamania[2], Roberto Doriguzzi-Corin[1], Chris Matrakidis[3], Dimitrios Klonidis[3], Pontus Sköldström[4], Abdul Ghafoor[4], Stéphane Junique[4], Victor Lòpez[5], Domenico Siracusa[1]

[1]FBK CREATE-NET, [2]ADVA Optical Networking, [3]Athens Information Technology, [4]RISE ICT/Acreo, [5]Telefònica

Abstract: Intent-based Software-Defined Networking can automate mapping of customer services to transport services. We demonstrate this using a multi-layer orchestrator that provisions a complex customer service over an IP/Optical testbed.

OCIS codes: (060.4250) Networks, (060.4258) Networks, network topology

1. Introduction

Network operators are driving the evolution of Software Defined Networking (SDN) to increase automation and facilitate low/zero touch provisioning. Automation will allow operators to better utilize their infrastructure, support dynamic service requests, while reducing operational costs in terms of reduced human intervention. To achieve this objective, SDN control planes should be able to abstract and replace some OSS (Operations Support System) functionalities in order to orchestrate service activation across multiple vendor and technology domains, as well as perform crucial actions for service fulfillment (reporting of failures, automatic restorations etc.).

A key challenge here requires orchestrators to handle both *customer* and *transport* services. Transport services typically refer to specific connections like an MPLS tunnel or an optical lightpath, that can be controlled and monitored via well-known SDN abstractions. Customer services, on the other hand, can be composed of one or more transport services, which are tailored to serve a specific application. As an example, we consider a datacenter interconnect customer service in which multiple endpoints must all be interconnected with each other. In a typical transport network, this service would include multiple point-to-point transport services between participating endpoints. To activate this service, OSSs would need to include the logic to instantiate the different point-to-point transport services, including logic to rollback operations in case provisioning of one of the required transport services fails.

This paper presents an SDN orchestrator that uses a novel application-centric intent interface to expose abstractions for customer services. Internally, the orchestrator hosts logic for service activation and assurance based on these abstractions, moving this complexity away from the OSS. To demonstrate this capability, the orchestrator will provide a customer service that creates a mesh for a data center interconnect application that connects three different regions (each one with two data centers). The SDN orchestrator provisions the customer service over a real IP/Optical testbed.

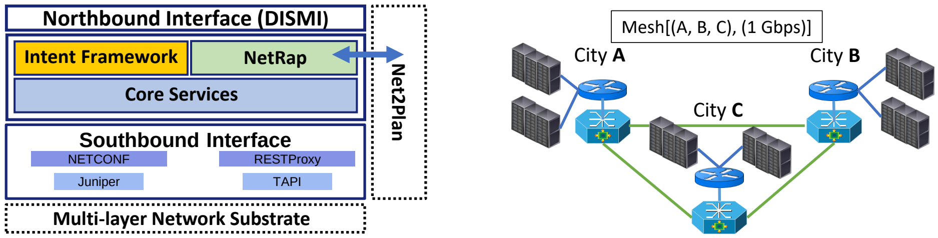
2. Approach

We propose a multi-layer network orchestrator, which exploits a SDN paradigm to manage an IP/Optical network. The orchestrator is an open-source, vendor-agnostic modular framework based on ONOS, which exposes a set of high-level primitives (intents) for specifying service requirements. The orchestrator also has integrations with online network planning and optimization tools to support in-operation multi-layer optimization.

Fig. 1a shows the architecture of the orchestrator. The DISMI intent interface [2] provides an abstraction layer where application-centric customer service requests are translated into simpler transport service intents. The DISMI intents provide a set of actions (e.g. connection, mesh, etc.) as connectivity services between several endpoints, each of which can have several requirements (e.g., minimum bandwidth, etc.). An endpoint in the request represents a region composed of one or more physical connection points (e.g. routers, hosts,etc.). DISMI decomposes the intents into simpler ones (e.g., a mesh connection into multiple point-to-point connections) and submits them to the underlying intent framework. DISMI hosts the logic to translate service activation and assurance capabilities for customer services, based on the status of underlying transport services as exposed by the SDN orchestrator.

The intents submitted by DISMI are processed by the NetRap [1] module and an external provisioning tool named Net2Plan [3]. The NetRap module intercept incoming intents and sends them to Net2Plan, which calculates a possible path for provisioning. If found, the calculated path is compiled into simpler commands understandable by network devices, and otherwise a failure is reported to DISMI.

The communication with network devices is handled by the Providers, which are responsible for presenting an abstracted view of device-specific configuration, control and management operations. The orchestrator implements a NetConf provider for the Juniper routers (physical and virtual), and Restproxy provider for interacting with the Optical controllers via the T-API protocol [4].



(a) (b)

Fig. 1. (a) The orchestrator architecture, (b) Multi-site data center scenario.

3. Scenario

Fig. 1b depicts the demonstration scenario, named as Data Center Interconnection (DCI). Every region (i.e. City A, B, and C) is composed of two data centers in order to provide fault tolerance. A customer can exploit the orchestrator to interconnect via a *unique* service request all the regions. Net2Plan calculates the IP and Optical paths required to fulfill the request and orchestrator manages the installation in the IP/Optical network devices. DISMI selects as connection point only one data center per region and, in the case of a failure, automatically swaps the connection to the other data center. By exploiting the DISMI abstractions, users can (i) directly request a service composed of many different point-to-point connections via a single request and exploit (ii) the endpoint abstractions for a zero-touch management of failures, by switching to alternate endpoints defined for the pair of regions.

4. Performance Evaluation

The DCI scenario has been implemented on a Optical/IP testbed composed of three ADVA FSP3000 Reconfigurable Optical Add-Drop Multiplexers (ROADM) and three Juniper MX240 IP routers, each one attached to a different ROADM. The three ROADMS are interconnected with each other to form a ring topology. We have measured the time taken by the orchestrator to establish mesh interconnectivity as described in Sec. 3. Without considering hardware latencies, the time spent by the orchestrator is about 1 second (Fig. 2(a)). It includes decomposition of the DISMI mesh intent into low-level intents (DISMI, 20 ms), path computation (Net2Plan, 900 ms) and generation of protocol-specific commands (RestProxy and NETCONF, 90 ms). The total provisioning time is presented in Fig. 2(b). It is divided into orchestrator time (blue box), and hardware configuration time (gray boxes) which involves setup of three lightpaths (around 24 seconds each), and six IP routes, two for each adjacency created with the lightpaths (2 seconds each). The two SBIs of the orchestrator work in parallel and therefore only the IP routes linked to the last configured lightpath contribute to the total provisioning time.

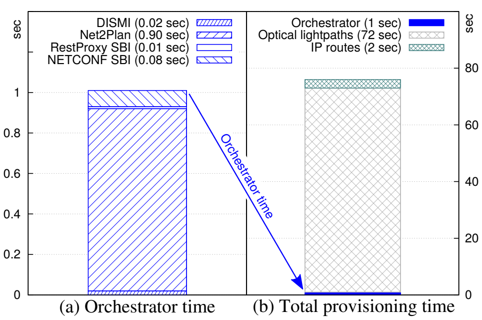


Fig. 2. DCI provisioning time.

5. Conclusions

This paper demonstrates the use of and SDN orchestrator with application-centric intents to establish complex customer services over a multi-layer network. The abstract representation of a customer service enables the orchestrator to understand and operate according to the requirements of a customer service, *reducing human intervention* (in the form of multiple individual transport service requests) required to establish this service. *Service fulfillment* logic, such as handling of transport service failures that are part of a customer service can also be built into the orchestrator. Finally, these abstractions facilitate *reuse of common service intents* such as mesh services for other customer applications (e.g. managed WAN) with low overhead.

Acknowledgement: The research leading to these results has received funding from the European Commission within the H2020 Research and Innovation programme, ACINO project, Grant Number 645127, www.acino.eu.

References

1. P. Sköldström *et al.*, "Making Powerful Friends: Introducing ONOS and Net2Plan to Each Other," in *Proc. of the International Conference on Transparent Optical Networks (ICTON)*, 2017.
2. P. Sköldström *et al.*, "DISMI - an intent interface for application-centric transport network services," in *Proc. of the International Conference on Transparent Optical Networks (ICTON)*, 2017.
3. P. Pavon-Marino *et al.*,"Net2plan: An open source network planning tool for bridging the gap between academia and industry", IEEE Network, vol. 29, no. 5, pp. 90-96, 2015.
4. V. Lopez *et al.*, "Transport API: A Solution for SDN in Carriers Networks," in *Proc. of the European Conference on Optical Communication (ECOC)*, 2016.