Enabling Network Slicing Across a Disaggregated Optical Transport Network

Ramon Casellas[1], Alessio Giorgetti[2], Roberto Morro[3], Ricardo Martínez[1], Ricard Vilalta[1], Raül Muñoz[1]

[1]*CTTC/CERCA (Spain), *[2]*CNIT/SSSUP (Italy), *[3]*TIM (Italy)*

ramon.casellas@cttc.es

Abstract: We propose and implement a network virtualization architecture for open optical (partially) disaggregated networks, based on a device hypervisor and OpenConfig and OpenROADM data models, in support of 5G network slicing over interconnected NFVI-PoPs.

OCIS codes: (060.4250) Networks;

1. Introduction

Network slicing has been considered a major selling point of 5G networks as initially proposed by the NGMN [1], to deal within the heterogeneous requirements of services offered to a multiplicity of vertical industries, in terms of e.g., latency, bandwidth, security, or resiliency. A *network slice* is macroscopically defined as logical network of interconnected functions and resources, supported over a shared infrastructure, generalizing and unifying previous concepts such as virtual private networks (which become mechanisms by which slicing is deployed and enforced). The relationship between slicing and the underlying transport network is still a topic of research: for slicing approaches based on the NFV framework across multiple interconnected NFVI-PoP (Fig.1), the transport network needs to become a resource under the orchestrating system. Second, it is expected that the *virtualization of the transport network* itself [2], defined as the partitioning and composition of the underlying physical optical infrastructure to create co-existing optical virtual networks (OVN) provides isolation: an OVN underlies and provides connectivity to component functions of a network slice (NS).

In this paper, we consider the virtualization of an open transport network, with the aim of supporting the aforementioned NFV-based slicing (Fig.1), and assuming a partial disaggregation model. We cover devices modeled as OpenConfig terminal devices/Optical Platforms [3] or OpenROADM devices [4] with their component Degrees (DEG) and Shared Risk Groups (SRG, OpenROADM naming for the add/drop component). As main contributions: 1) we propose and implement a virtualization/slicing architecture based on an *OVN manager*, which instantiates and controls the lifetime of the OVNs and a *per-device hypervisor entity*, which partitions a device according to its *standardized device data model*, into multiple virtualized devices (e.g., vROADMs). It ensures isolation and acts as a (restricted) NETCONF/YANG proxy to the physical device, so a *per-OVN* dedicated controller can control the OVN independently. 2) We extend the ONOS SDN controller with OpenConfig and OpenROADM device drivers and applications exporting a TAPI North Bound Interface (NBI). 3) We demonstrate the provisioning of a Network Media Channel (NMC) across the OVN.
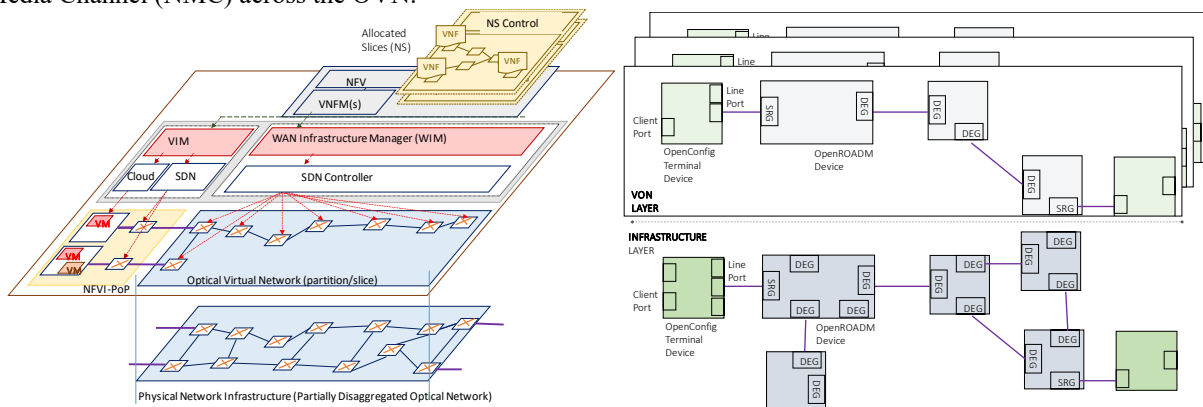

Fig. 1 Supporting Network Slice (NS) over multiple NFV Infrastructure Points-of-Presence interconnected by a virtualized Disaggregated Optical Transport Network (left). Logical view of Virtual Optical Disaggregated Networks over a shared infrastructure (right).

2. Disaggregated Network and Slicing Model

Our reference network is aligned with the ODTN project phase II [5], and corresponds to a partially disaggregated network [6]. Generally, virtualizing an optical network can be performed using a combination of a given device support (that is, hardware support) and/or the use of a virtualization layer (commonly referred to as hypervisor) that

extends or emulates such support. This, in turn, can be done directly at the device level or at the SDN controller level. Examples of the latter are the ONF SDN architecture or ACTN [7]. In any case, there are multiple degrees of freedom in defining an OVN/partitioning the devices: assigning a device's degrees (DEG) and SRGs, assigning the internal and external ports as well as internal and physical links and partitioning the usable spectrum. Fig 2 shows a potential partition of a sample OpenROADM device into 2 logical ROADMs.
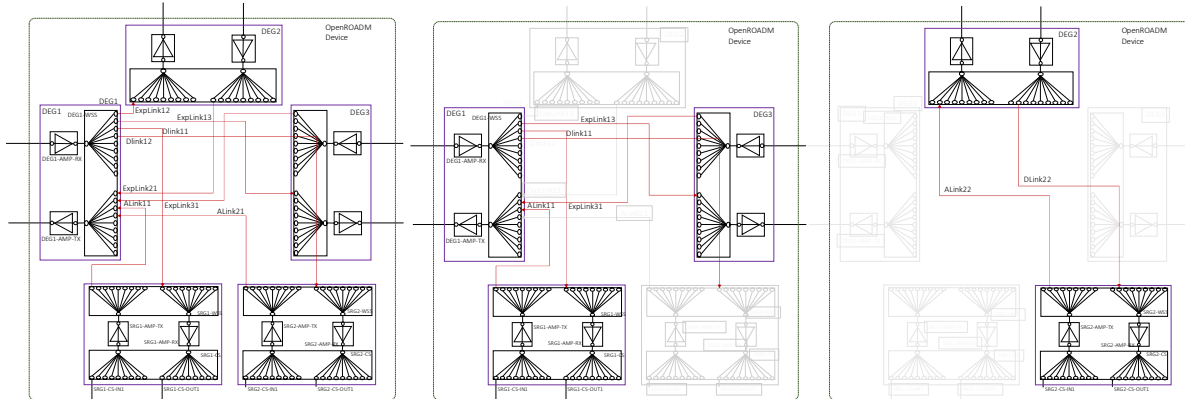

Fig. 2. Sample OpenROADM device with 3 DEG and 2 SRGs (left); A potential partitions resulting in two virtual ROADMS: 2 DEG, 1 SRG (center), and 1 DEG, 1 SRG (left).

A key element of our architecture is the hypervisor (Fig.3), which is responsible for coordinating access to the underling physical device agent, so each virtual device only sees and operates on a partial (restricted) view of the data model configuration and operational datastore. The hypervisor behaves as multiple NETCONF agents (one per OVN), as defined by the partition. When a new OVN has to be created the OVN manager is used to create the vROADMs/partitions, configuring the hypervisors and allocating a new SDN controller instance, configured to connect to the virtual devices. During the lifetime of the OVN, such SDN instance can provide end-to-end connectivity upon request between (virtualized) transceivers.
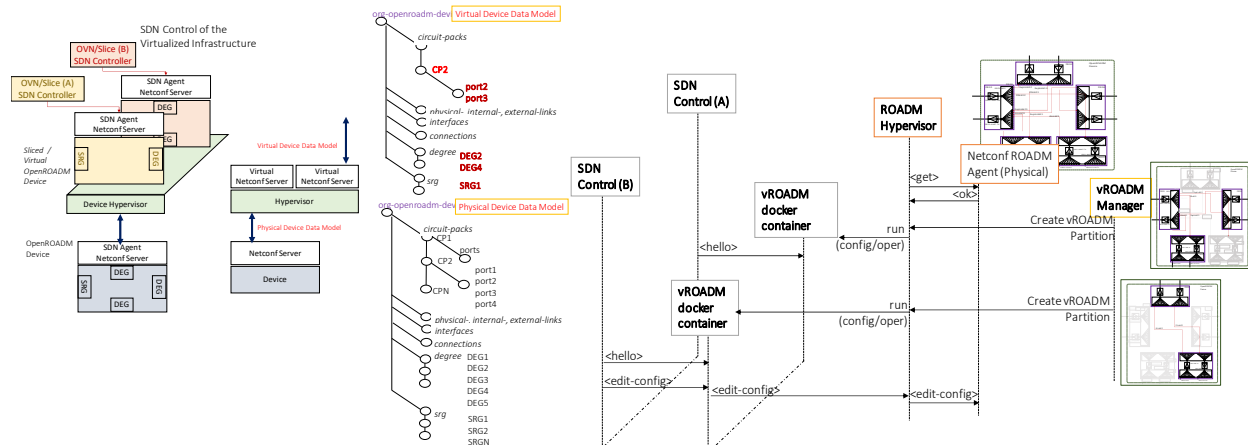

Fig. 3. Hypervisor: deployed as a layer over the device agent behaving as multiple device agents; each virtual device data model is a subset of the data model (right). The vROADM manager can request a partition and the hypervisor allocates a container to run the virtual device agent (right).

3. In-Slice Service Provisioning

During the lifetime of the OVN, the manager provides the OVN SDN controller with the credentials of the (virtual) devices, so NETCONF sessions are established and the devices' capabilities discovered (tunability and switching capabilities). NMC are set up upon request between two terminal devices, using either native ONOS API or a standard such as TAPI [8], in which terminal devices client ports are mapped to Service Interface Points (SIPs). This triggers a routing and spectrum assignment process that finds the k-shortest path between the devices and performs first fit spectrum allocation. For the Terminal Device, a logical channel association is instantiated within the device between a client (transceiver) port and an optical channel component bound to the line port of the device. For each of the OpenROADM devices across the path, a ROADM internal connection is requested: OTS and OMS (optical transport and optical multiplex) interfaces are created within each degree (if not existing) and supporting Media Channel and NMC interfaces are created, followed by the creation of a roadm-connection object (Fig.4, right).
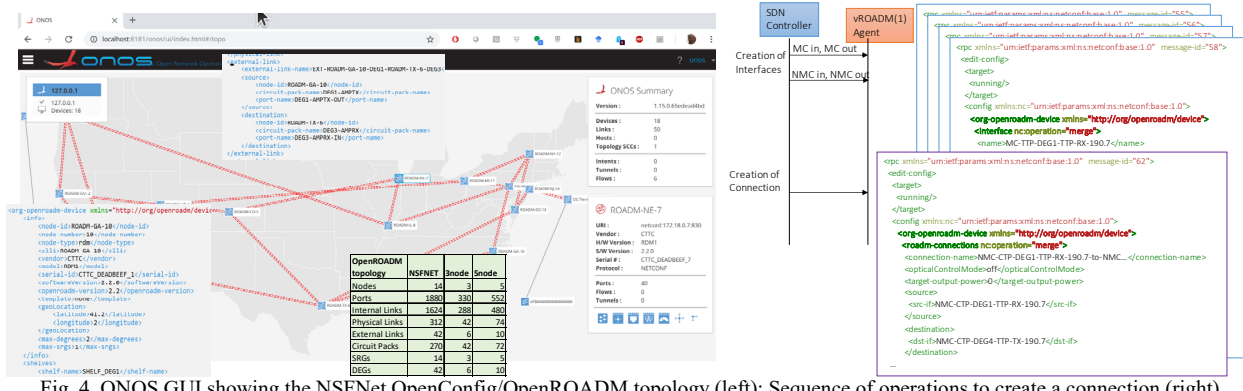
Fig. 4. ONOS GUI showing the NSFNet OpenConfig/OpenROADM topology (left); Sequence of operations to create a connection (right)

4. Experimental Demonstration

For the experimental evaluation, we consider an NSFNet topology with 14 OpenROADM devices (emulating the physical infrastructure, see Fig.4), with 42 links. A partition/OVN is defined selecting 10 nodes and a subset of the OpenROADM degrees, resulting on a sub-graph of the original topology. Virtual agents are allocated in Docker containers in such a way that devices in the same OVN belong to the same IP subnet. A TAPI request is sent to the OVN SDN controller for the NMC. The experiment is triggered by posting a RESTConf Remote Procedure Call (RPC) requesting a TAPI connectivity service (Fig.5, left). The NMC is set up in $O(100)$ms, by exchanging multiple edit-config messages with the agents. Fig 5 shows the Wireshark capture of NETCONF/SSH (center) and the I/O plot of traffic in pps (right). Hardware delays are not taken into account, which would increase the overall latency.
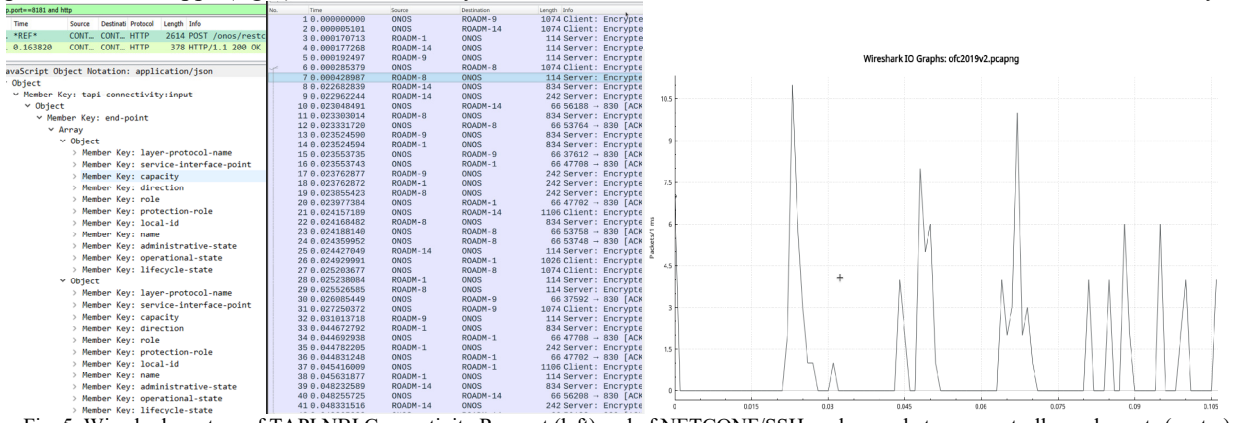

Fig. 5. Wireshark capture of TAPI NBI Connectivity Request (left) and of NETCONF/SSH exchanges between controller and agents (center);

5. Conclusions

We have demonstrated the virtualization of an open and disaggregated optical network, along with the provisioning of network media channels within a given OVN instance, relying on device hypervisors and standard data models. The implemented extensions to ONOS SDN controller for OpenConfig or OpenROADM can be used regardless of the infrastructure is real or virtual. The PoC demonstrates the setup of a connection / optical connectivity intent – not taking account hardware delays –, largely satisfying the target values and demonstrating the feasibility of a model driven SDN control for a partially disaggregated network.

6. Acknowledgements

7. References

[1] NGMN White Paper on 5G https://www.ngmn.org/5g-white-paper/5g-white-paper.html
[2] R. Vilalta, A. Mayoral, R. Muñoz, R. Martínez, R. Casellas, "Optical Networks Virtualization and Slicing in the 5G era", OFC2018.
[3] OpenConfig project and data models http://openconfig.net and https://github.com/openconfig/public/tree/master/release/models
[4] The Open ROADM Multi-Source Agreement (MSA) http://www.openroadm.org
[5] The Open Disaggregated Transport Network project, ONF, https://www.opennetworking.org/odtn/
[6] E. Riccardi et al, "An Operator view on the Introduction of White Boxes into Optical Networks", JLT, v36, I15, pp3062-3072, 2018.
[7] R. Casellas et al, "Experimental Validation of the ACTN architecture for flexigrid optical networks w Active Stateful H PCEs", ICTON2017.
[8] ONF Transport API (TAPI) https://www.opennetworking.org/wp-content/uploads/2017/08/TAPI-2.0-Updates-Overview.pdf