Abstraction and Control of Multi-Domain Disaggregated Optical Networks With OpenROADM Device Models

Ramon Casellas, Senior Member, IEEE, Member, OSA, Ricardo Martínez, Senior Member, IEEE, Ricard Vilalta, Senior Member, IEEE, and Raül Muñoz, Senior Member, IEEE

(Highly-Scored Paper)

Abstract—Network operators are evolving their optical transport networks in order to make them cost effective. In some scenarios, this means considering adopting software-defined networking principles along with open and standard interfaces, leveraging the underlying hardware programmability while, at the same time, considering the benefits of (partial) disaggregation, in view of the potential benefits of decoupling terminal devices from the line systems or of separating the hardware from the controlling software. In this evolution, operators often segment their networks into domains. Reasons include the need to scale, or confidentiality and/or vendor interoperability constraints. Additionally, the need to virtualize the (multi-domain) transport network has emerged as a key requirement to support functions such as network slicing and partitioning, and to empower end users to control their allocated partitions, enabling new business models related to multi-tenancy. In this context, several standards-defining organizations have been working on architectures, interfaces, and protocols to support requirements, such as the Abstraction and Control of Traffic Engineering Networks of the Internet Engineering Task Force, known as ACTN. In this article, we experimentally validate a control plane architecture for multi-domain disaggregated transport networks that relies on the deployment of network elements compliant with the OpenROADM multi source agreement device model. We demonstrate the abstraction and control of such networks in line with the ACTN framework and we show the applicability of the approach with a proof-of-concept tested implementation.

Index Terms—ACTN, multi-domain optical network control, OpenROADM device model, software defined networking.

I. INTRODUCTION

The majority of deployed optical transport networks are proprietary, behaving as a closed, highly coupled, single-vendor managed domain. Even if their respective control and management systems may export high-level and open north-bound interfaces (NBI), the internal interfaces are not systematically disclosed or open. Although this level of system integration and the expected vendor support has clear advantages, a trend known as disaggregation has steadily emerged, driven by the requirements of telecommunication and data-center operators, mostly in the need to keep costs down while supporting a sustained traffic increase.

Disaggregation involves composing and assembling open and available components, devices and sub-systems into optical infrastructures and networks, combining “best-in-class” devices, tailored to specific needs. Several models are possible within this trend in optical transport networks, allowing more flexible, re-configurable and elastic architectures [1], such as a partial disaggregation where the control of the transceivers and terminal devices is decoupled from the control of the (open) line system (OLS), or a full disaggregation, based on white boxes, where different optical network elements (such as ROADMs [2], [3], transponders, line amplifiers, etc.) can be provided by different vendors.

Second, operators often segment their networks into domains in order to scale, or due to confidentiality and/or interoperability reasons. In either case, the control of such multi-domain transport networks typically relies on hierarchical control models with a hierarchy of controllers [4]. This raises new challenges related to the limited topology visibility across domains, having a clear impact on the optimality of control plane aspects such as path selection or resource allocation.

Finally, the need to virtualize the transport network has emerged as a key requirement to support functions such as network slicing and empower end users to control the allocated partitions [5]–[7]. Tenants are given abstracted topology views of the underlying physical network and are allowed to utilize and independently control allocated virtual network resources as if were real. Open questions remain open regarding the efficient and optimal abstraction of the underlying networks or how to support virtualization and slicing.

This results in the need to design, implement and validate SDN control plane architectures supporting such requirements: i) controlling disaggregated optical transport networks based on open and standard interfaces; ii) support multi-domain scenarios
with limited topology visibility (with and without electrical conversion at the domain boundaries) and, iii) supporting secure network sharing and virtualization. The former requirement has resulted in a major adoption of data modeling languages and open transport protocols, while the last two have driven the design of SDN architectures, such as the Abstraction and Control of Traffic Engineered Networks (ACTN), described later, and which is used as a reference framework for our work. This paper, extending [8], is structured as follows: we complete the introduction with an overview of the concepts of a device data model along with other key elements of a model driven development approach to SDN, such as the YANG data modeling language or the NETCONF and RESTCONF protocols and a high-level description of the ACTN architecture and the OpenROADM device model. In Section II we present the architecture of the system, including aspects such as topology aggregation modes (Section II-A), and network virtualization (Section II-B). Section III elaborates on the dynamic workflow and message exchange for the provisioning of connectivity services (network media channels) between multiple domains and Section IV details the experimental evaluation of our implementation. Finally, Section V concludes the paper.

A. Device Data Models and the YANG Modelling Language

A device Information Model macroscopically describes the device capabilities, in terms of operations and configurable parameters, using high level abstractions without specific details on aspects such as a particular syntax or encoding. A Data Model determines the structure, syntax and semantics of the data that is externally visible. YANG [9] is a data modeling language, where a model includes a header, imports and include statements, type definitions, configurations and operational data declarations as well as actions (RPC) and notifications. The language is expressive enough to structure data into data trees within the so called datastores, by means of encapsulation of containers and lists, and to define constrained data types (e.g., following a given textual pattern); to condition the presence of specific data to support optional features and to allow the refinement of models by extending and constraining existing models (by inheritance/augmentation), resulting in a hierarchy of models. Although initially conceived to model configuration and state data for network devices, YANG has become the data modeling language of choice for multiple network management systems (covering devices, networks, and services, even preexisting protocols) due in part, for its features and flexibility and the availability of tools.

B. NETCONF and Restconf

An associated protocol offers primitives to view and manipulate the data, providing a suitable encoding as defined by the data-model. For YANG, the NETCONF protocol [10], enables remote access to a device, and provides the set of rules by which multiple clients may access and modify a datastore within a NETCONF server (e.g., device). It is based on the exchange of XML-encoded Remote Procedure Call (RPC) messages over a secure connection (commonly secure shell, or SSH). NETCONF enabled devices include a NETCONF server, Management applications include a NETCONF client and device Command Line Interfaces (CLIs) can be wrapped around a NETCONF client. The layering mode relies on having configuration or notification data (Content Layer) that is exchanged between a client and a server, with a set of well-defined operations (e.g.,<get-config>, <edit-config>, within the Operations Layer) encapsulated in RPC messages or notifications (Message Layer) and using a Secure Transport. After establishing a session over a secure transport, both entities send a hello message to announce their protocol capabilities, the supported data models, and the server’s session identifier.

NETCONF differentiates between configuration data, data that describes operational state and statistics. Configuration data is the data that is provided by a client to enable a device to behave as desired, setting it into a functioning or running state. It excludes data that the device can learn by itself, such as read-only data or statistics. For example, an optical transceiver model may expose configuration data (modulation format to use) and operational data (Bit Error Rate, temperature) and statistics (packets received).

The data is arranged into one or multiple configuration datastores. A configuration datastore is a conceptual place to store the complete set of configuration information that is required to get a device from its initial default state into a desired operational state, and may be implemented, for example, using files, a database, etc. A datastore is the target or the source of a NETCONF operation, as described later. Having multiple datastores enables operations to be done in e.g., a “candidate” datastore and commit the changes as a whole (so called transactional semantics). Datastores are named, and NETCONF defines three: running, startup and candidate.

When accessing configuration or state data, with NETCONF operations, filter expressions can select subtrees, providing a great degree of flexibility. For example, when retrieving the statistics of a device port, the client may specify the port name the request refers to, either by explicitly encoding the subtree in the request or using an xpath expression.

Alternatively, RESTCONF (an effort to map NETCONF operations to REST operations over HTTP following REST model) can also be applied, arguably simpler but less complete. A significant number of initiatives are defined around the use of YANG models yet the number of different, often partially overlapping, models is increasing, and this is likely to remain an issue for the foreseeable future. There is little experience effectively using such models and the underlying complexity needs to be managed.

C. Abstraction and Control of Traffic Engineered Networks

The Abstraction and Control of Traffic Engineered Networks (ACTN) architecture, relying on Software Defined Networking (SDN) principles with a hierarchy of controllers, was initially conceived to cover (optical) transport networks and was later extended to cover other Traffic Engineering (TE) technologies. The framework as been recently been published by the IETF [11]. The main objective is to provide a mechanism for
customers of a (transport) network operator to request and then manage a virtual network built on top of TE connections across such operator’s network. The reference architecture is shown in Fig. 1. The Customer Network Controller (CNC) is a software component under control of the customer. It makes a request to the network operator for a virtual network by sending a message over the CNC-MDSC Interface (CNI). The Multi-Domain Service Coordinator (MDSC) is responsible for orchestrating the allocation of network resources to support the customer services, facilitating underlay transport resources to be abstracted and virtual network instances to be allocated, and coordinates one or multiple Provisioning Network Controllers (PNC), which are responsible for controlling one of the underlying domains, using the MDSC-PNC Interface. Each PNC controls a given domain, with a domain-specific South Bound Interface (SBI). Each domain may be different administrative domains, technology domains, or even virtual networks thus enabling a hierarchy. ACTN facilitates heterogeneous domain transport networking and control/management, while enabling a logically centralized multi-domain orchestration, using a hierarchical architecture to scale. Multiple data models can be applied to the ACTN framework (such as Layer 2 or Layer 3 service models). The MDSC-PNC Interface (MPI) is achieved by a range of YANG modules dependent on the technology of each underlying network. The actual YANG device model is quite comprehensive. Macroscopically, it defines a first section related to the device information (common language node identifiers, vendor, model, serial number, geolocation coordinates, etc.) followed by a section that includes a list of circuit-packs, describing the physical architecture including their components ports and naming, as well as the correspondence in terms of actual racks and shelves.

D. The OpenROADM Device Model

The device model proposed by OpenROADM [13] (in this work we cover device model v2.2) is sketched in Fig. 2. From a device perspective, a ROADM is composed of a number M of directions or degrees (DEG) and a number N of Shared Risk Groups or SRGs (OpenROADM terminology for Add/Drop stages). In the figure, there is a single SRG (SRG1), and 3 degrees (DEG1, DEG2, DEG3). From a physical perspective, a given component (degree, or SRG) is implemented in terms of circuit-packs, briefly defined as a field replaceable unit within electronic switching equipment. For example, a given degree has Reception (RX) and transmission (TX) amplifiers and a Wavelength Selective Switch (WSS) to multiplex and demultiplex the optical signals coming to/from external links, other degrees or add/drop stages. DEG1 includes DEG1-WSS, DEG1-AMPTX, and DEG-AMRX circuit packs (see figure). SRGs combine WSS, amplifiers and Combiners/Splitters. All the different elements are interconnected by physical and internal links. The actual YANG device model is quite comprehensive. Macroscopically, it defines a first section related to the device information (common language node identifiers, vendor, model, serial number, geolocation coordinates, etc.) followed by a section that includes a list of circuit-packs, describing the physical architecture including their components ports and naming, as well as the correspondence in terms of actual racks and shelves.

The OpenROADM device model follows ITU-T terminology when defining connections, and this affects the subsequent naming of interfaces, and other objects in the device model. In simple terms, the basic OTN layers are visible in the Optical Transport Network (OTN) transport structure and consist of the OCh, Optical Multiplex Section (OMS), and the Optical Transmission Section (OTS). Macroscopically, a Physical Termination Point (PTP) or Port is an access point on an network to which a link is attached. It is the representation of a physical port. A Connection Termination Point (CTP) represents the actual or potential endpoint of a cross-connection, link connection, or circuit. A CTP is contained within a PTP. A Trail Termination Point (TTP) is a reference point that represents a location of insertion/extraction of monitored and adapted information characteristic to a given layer network (as opposed to the information presented by the client of the layer network). When a cross-connection is to be created in a ROADM, for example, between incoming port of degree DEG1 and outgoing port of DEG2, first the OTS and OMS interfaces need to be created, basically specifying that such port is attached to
transmission section and specifying the optical frequency band enabled for that port. Next, Media Channel (MC) interfaces are created, that represent TTP for the data to be switched (named, for example, MC-TTP-DEG1-TTP-RX and MC-TTP-DEG2-TTP-TX). Finally, an internal connection is to be established so 2 network media channel (NMC) interfaces are created that support the CTP that are the endpoints of the cross-connection.

The next block in the device model details the set of ROADM interfaces (logical connection or trail termination points that can be supported over either underlying logical interfaces or directly physical ports), resulting in an interface hierarchy. Another section lists the internal links (links between ports of a given component or circuit pack), the physical links (links between different components, such as a link between degrees or between degrees and SRGs, commonly named express, add or drop links) and external links (links that are instantiated to reflect connectivity between ROADMs). Next, the model also includes two lists for the main components: a list of numbered degrees, which defines the ROADM overall degree, and a list of SRGs including a list of add drop port pairs (client ports towards the terminal devices). Finally, the roadm-connection list includes the connections that are active (established) in the device, between two logical interfaces. Setting or removing ROADM connections involves several steps and NETCONF messages, creating or deleting supporting (logical) interfaces and later creating connection objects between logical interfaces, as detailed in Section III.

II. CONTROL PLANE ARCHITECTURE

The generic ACTN does not enforce a given protocol and can be applied or mapped to existing entities, interfaces and protocols, adopting or supporting existing deployments. A particular example is the Path Computation Element (PCE) and PCEP protocol [14]. In this paper, we implement and demonstrate the ACTN approach for a multi-domain OpenROADM network in which the same unified open data model is used to abstract a domain, enabling an arbitrary hierarchy, where internal connections with abstracted ROADMs (aROADMs) trigger the corresponding intra-domain (Network) Media Channels or (N)MC. In other words, a given optical network (flexi-grid) domain is presented to the MDSC as one or more aROADMs, and the same SDN controller can be used regardless of whether the underlying domain is a physical device or abstracts a network topology, without changes. This is shown in Fig. 3

A. Topology Aggregation Methods

A key challenge in hierarchical architectures is the topology abstraction method. Several methods are known in the literature e.g., [15]. From a research perspective, topology abstraction methods were initially conceived for packet-switching with Quality of Service (QoS) networks [16], [17] aiming at synthesizing domain-internal state for inter-domain dissemination. Optical networks raise additional constraints, starting with multi-domain dense Wavelength Domain Multiplexing (DWDM) networking [18], [19] proposed a hierarchical inter-domain solution for Automatically Switched Optical Networks (ASON) based upon a simple node (coarse) abstraction policy. Additional works extends it presenting a new Routing and Wavelength Assignment (RWA) strategy based upon the stochastic estimation of effective number of available wavelengths along inter-domain paths. More generically, the abstraction is commonly referred to the process in which a domain state is represented using a subset of the domain state, using only relevant information. In our concrete architecture, each domain controller (PSC) is responsible for abstracting its domain as a (set of) device(s), in such a way that the parent controller (MDSC) manages an abstracted network and can perform end-to-end abstracted path computation. At each domain, each PSC obtains and manages the domain topology by querying the device (including OpenROADM external links) and manages services instantiated in its domain mapping configuration operations requested on the set of abstracted devices to configuration operations over the underlying physical devices. The MDSC manages domain-connectivity and endpoint reachability. Common abstraction models rely on either a virtual node, a mesh of virtual links or an optimized layout of virtual links according to a per-domain abstraction optimization objective function.

1) Node Abstraction: The straightforward abstraction is to represent a domain as a single node, due to its relative simplicity and the fact that, for hierarchical models, the path computation aspect is commonly specified in terms of a two-step process in which a first step is a domain selection followed by segment expansion in each domain. In such (often referred to as Virtual Node) case, an abstracted OpenROADM device represents the underlying domain connectivity. From a high level perspective, domain demarcation/entry points are mapped to ports in the abstract node. In our specific architecture and device model, such OpenROADM device has as many SRGs as the SRGs in the domain, and as many degrees as inter-domain links. It is the responsibility of the PNC to keep the association between abstracted SRGs and DEGs and the underlying physical resources.

![Fig. 3. Control Plane Architecture using ACTN for a Multi-domain flexi-grid network based on the OpenROADM device model.](image-url)
2) Link Mesh Abstraction: Due to limitations related to the optical technology, a single node abstraction may not considered sufficient, unless the internal connectivity is indeed also reflected in the node (e.g., the abstract node is not an opaque node). To overcome this limitation, a link mesh abstraction e.g., [20] is commonly used, trying to reflect internal domain connectivity across endpoints (often referred to as Virtual Links). A given set of links model the internal domain connectivity. With a full mesh ($N^2$) between edge nodes, the abstraction introduces new edge nodes that have the same number of SRGs and as many DEG elements as (virtual) links towards edge nodes. Advanced methods dynamically update such link abstraction by computing paths between endpoints and mapping path attributes to (virtual) link attributes. A particular case is the full mesh, in which a (pair of) unidirectional links is exported between each domain border node (see Fig. 4).

In the case of a virtual link (e.g., full mesh) abstraction, there is an issue on how service interface points on all nodes inside the domain are abstracted and presented in the logical topology. In our case, the constraint is that if a node TTP or SIP is to be exposed by a domain controller and used by higher layers, that node must appear in the topology (becomes a border node). For example, a physical linear topology A-B-C, if B has client ports that need to be visible to higher layers, it is not possible to have a virtual mesh A-C and B needs to be present in the abstraction. Let us also mention that there are constraints to be taken into account when performing path computation over abstracted topologies. Not all such constraints can be accurately reflected and additional policies, constraints or heuristics (suboptimal) may need to be applied. Our approach to topology abstraction can use both aforementioned methods. Note that, when using the node abstraction, the internal connectivity of the node can reflect the underlying paths between domain endpoints. If no path is available between two border nodes for a given frequency slot width, the corresponding internal link may be removed from the device model. For the virtual link mesh, the approach is as follows: for each source/destination path, we compute the $K$-shortest path using Yen’s $k$-shortest path algorithm. The number of exported links is configurable by policy. For each (order) potential path, we iterate the Optical Multiplex Section (OMS) links – network links between nodes – and compute a a set of available frequency intervals (ranges) by performing the intersection of available frequency slots at each link and removing the occupied frequency slots. The resulting frequency intervals (in the form of one or more frequency ranges $[f_1, f_2]$ are announced as TE attributes of the virtual links, and provide an aggregated view of the underlying available network resources.

The abstraction of the topology using a full mesh involves the potential computation of $K \times N (N – 1)$ paths with $N$ being the number of nodes. However, let us note that the common case is that no all nodes in such a transport network are border nodes. This process can be repeated at each service arrival or departure (or configurable by policy) in order to update the status of the domain at the MDSC level. To illustrate the algorithm, consider the network topology of Fig. 5. It is composed of 28 devices (OpenROADM devices) and 2 terminal devices, with 88 total external links (links interconnecting 2 OpenROADM degrees).

In a Intel PC i7 7700, with 32 Gb RAM, the implemented algorithm is able to compute the mesh of 756 paths in $\sim 20$ seconds. Assuming 3 border nodes (1, 15 and 36), the computation of 6 virtual links results in 6 paths (with TE metrics 6, 7, corresponding to the hop count) and is carried out in $\sim 200$ ms.

B. Virtualizing the Network

The architecture encompasses the ability to virtualize the multi-domain topology, to share the underlying infrastructure between different CSC controllers. In our specific context, our approach relies on supporting virtualization directly at the device level. A key advantage is that with full virtualization support, the SDN controller and other functional elements may operate without change, unaware of whether they operate on virtualized physical devices. It relies on the concept of OpenROADM hypervisor [7]. Conceptually, it behaves like a hypervisor in the computing domain: a hypervisor is a functional element (e.g., computer software, firmware or hardware) that instantiates and enables the controlled execution of virtual machines - referred to as guests - running over a common physical server - referred to as host - and presents the guest operating systems with a virtual operating platform. Multiple instances may thus share the virtualized hardware resources. An OpenROADM hypervisor allows multiple virtual ROADM devices to behave independently and to be controlled by a dedicated SDN Controller using the same open interfaces and protocols as the physical device. It partitions a device according to its standardized device
data model, into multiple virtualized devices (e.g., virtualized ROADM devices). This hypervisor ensures isolation and acts as a (restricted) NETCONF/YANG proxy to the physical device). The role of the ROADM hypervisor is to coordinate access to the underlying 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 actual implementation is based on the idea that for each active partition, a virtualized NETCONF server is instantiated and a running datastore is created encompassing only the elements included in the partition. SDN Controllers interact with agents running in dedicated containers. NETCONF operations are processed by the NETCONF front-end (over the partial view of the device), and forwarded to the hypervisor.

III. CONTROL PLANE WORKFLOW

Let us detail the workflow involved in the provisioning of end-to-end NMC. At each domain, and from the point of view of networks operation, the procedure still requires initial configuration: the network operator configures the PNC SDN controller with the list of devices, including aspects such as IP addresses, NETCONF credential, etc.

A. Network Topology Discovery/Management

The PNC establishes a persistent NETCONF session with the OpenROADM devices, which implies an initial capability exchange, where the PNC discovers which data models and features are supported. Next, the PNC commonly issues <get> and <get-config> messages as needed to retrieve information and operational state of the device (for example, a <get> operation with a subtree filter of <org-openroadm-device><info> allows retrieving basic data to add the device into the SDN controller device manager. Similar operations may be carried out in order to retrieve the composing circuit packs and ports, obtaining the internal connectivity and to discover port capabilities.

The OpenROADM device model allows each device to announce its direct neighbors by listing its external-links. If such information is not available, and given the lack of link management and automated neighbor discovery, links between ROADM devices in a domain must be configured at the level of the PNC of the domain. Similarly, the inter-domain topology is provisioned at the level of the MDSC.

B. Inter-Domain Service Provisioning

The provisioning of an end-to-end NMC involves an interaction between the MDSCs and the PNCs using abstracted multi-domain topology, including an optimal domain selection and subsequent segment expansion. After the path computation is complete, the MDSC coordinates the provisioning across the domains. To instantiate an end-to-end service, each PSC entity maps configuration requests coming from the MDSC, defined over the abstracted ROADMs (aROADM) into an intra-domain connection supported over several physical ROADM cross-connections within the underlying domain, creating the appropriate interfaces accordingly.

Having a common data model both for physical and abstracted ROADM networks simplifies the overall process. According to the OpenROADM specification [13], the creation of ADD, DROP or EXPRESS internal connections is carried out by the dynamic creation of a hierarchy of supporting interfaces, including OTS and OMS (optical transport and optical multiplex) interfaces if they are not pre-existing, of Media Channel (MC) and Network Media Channel (NMC) interfaces for the media channel with its parameters such as upper and lower frequencies and slot width and nominal central frequency, followed by the creation of the roadm-connection encompassing the two involved logical interfaces. For example, to create a
unidirectional express connection supporting a 50 GHz signal centered at 190.7 GHz from degree 1 to degree 4, the SDN controller proceeds as follows. First, assume that the OTS and OMS interfaces are configured at the ROADMs; these interfaces correspond to the lower layers of the OTN model; OTS corresponds to the Optical Transmission Section, e.g., between amplified fiber sections and OMS to the Optical Multiplex section. Next, create Media Channel (MC) interfaces (via a \texttt{<edit-config>} operation) over the OMS interfaces (by convention named MC-TTP-DEG1-TTP-TX-190.7 and MC-TTP-DEG1-TTP-RX-190.7, with a min-freq: 190.675 THz and max-freq: 190.725 THz, which specifies the supporting media channel. Next, create Network Media Channel (NMC) interfaces over the MC interface, specifying a center frequency and width, inducing Connection Termination Points (CTPs). Finally, establish the unidirectional connection between CTPs, from the interface NMC-CTP-DEG1-TTP-RX-190.7 to NMC-CTP-DEG4-TTP-TX-190.7.

If the device corresponds to a physical ROADM device in a given domain, the parameters conveyed in the NETCONF messages allow the proper configuration of the hardware (see, for example [21]). At the abstracted level, a (for example) ADD connection over an aROADM involves creating MC and NMC interfaces in the abstracted device (represented by the PNC) and connecting NMC interfaces. Such operations are mapped into one ADD connection and several EXPRESS ones across the different physical ROADMs in the domain, each involving the creation of several interfaces and the respective connection objects. This is shown in Fig. 6: the MDSC decides to setup a connection over the abstracted node, which is mapped internally within the domain to the configuration of connections within the underlying ROADMs that along the physical route/path.

IV. TESTBED PROOF-OF-CONCEPT AND VALIDATION

A. Implementation

The system has been implemented and demonstrated in a control plane testbed. The MDSC is implemented in terms of an Open Networking Operating System (ONOS) instance extended with OpenROADM drivers and a Transport API (TAPI) v2.1 NBI [23], covering the photonic media layer [22]. The PSC controllers combine an application based on the ConfD project [24], which provides a framework for NETCONF agent implementation, thus providing a NETCONF front-end towards the MDSC and allowing the PNC to represent the domain as an abstracted ROADM (in particular, the datastore that supports the NETCONF abstracted OpenROADM device) is based on the topology aggregation. The application also registers for notifications from ConfD regarding the creation and deletion of data nodes in the OpenROADM device tree that corresponds to interface and cross-connections and later in interacts with a per-domain ONOS dedicated instance, consuming its NBI for intra-domain topology abstraction and connectivity configuration.

B. Experimental Evaluation

To validate the approach, an end-to-end request for a network media channel is requested across 2 domains (Fig. 7), using as domain A the 14 node NSFNET and, as domain B, a 10 node Metro-Haul [25] topology. The details of the second domain, as seen by the PNC and dedicated ONOS instance is shown in Fig. 8. In this proof-of-concept, each domain is abstracted as a single aROADM. Two endpoints (endpoint A or EPA and endpoint B or EPB), modelled as OpenConfig Terminal devices are attached to their respective ROADMs nodes of each domain. Note that such endpoints are also configured at the MDSC, in order to be able to request services between them at the level of the MDSC).

The process for the end-to-end provisioning starts with a customer sending a TAPI connectivity request, and involves the processes of domain selection (path computation based on the abstracted topology), and the subsequent (parallel) provisioning within the 2 domains. The service provisioning delay, measured at the the MDSC NBI interface, is roughly \(\sim 800\) ms, as shown in Fig. 9. The Wireshark capture shows the initial TAPI request (implemented as two operations, retrieving the TAPI context using a REST GET operation in order for the client to retrieve the
Fig. 7. Experimental evaluation scenario, encompassing 2 domains. The left one emulates the 14-node NSFNET topology and a 10-node from the Metro-Haul project.

Fig. 8. ONOS GUI showing the detail of the Domain B, including each node NETCONF identifiers, port count, etc.

Fig. 9. Wireshark capture at the MDSC with the TAPI NBI request followed by the NETCONF over SSH NETCONF exchanges.

list of available Service Interface Points, followed by the request to establish a network media channel connectivity service, using a REST POST operation) followed by the NETCONF over SSH exchanges with the abstract ROADMs (partial list).

Next, the cross-connection in Europe (Domain B) abstracted node is mapped to 3 cross-connections in nodes Lisbon Madrid Barcelona, following the domain internal RSA (shortest path) as shown in Fig. 10. Within the domain, the provisioning is ~500 ms. Note that there is an overhead, given the XML encoding, the SSH transport for NETCONF and the need to send multiple messages (creation of MC, NMC interfaces, and connections, OMS and OTS interfaces at each degree port are pre-created). The figure shows a (partial) capture of the
ACKNOWLEDGMENT

We would like to thank the members of the ODTN and Metro-Haul projects for their insightful comments. In particular, we would like to thank Adrian Farrel for contributions on ACTN and its applicability to our problem domain along with the different related information and data models. Alessio Giorgi from CNIT/SSUP has contributed with fine tuning the ONOS OpenROADM device drivers and Roberto Morro and Emilio Riccardi from TIM have provided an additional implementation of OpenROADM enabled ROADM with netopeer and sample OpenROADM topologies.

REFERENCES

[1] E. Riccardi, P. Gunning, O. González de Dios, M. Quagliotti, V. López, and A. Lord, “An operator view on the introduction of white boxes into optical networks,” J. Lightw. Technol., vol. 36, pp. 3062–3072, 2018.
[2] N. Calabretta, K. Prifti, N. Tessema, X. Xue, B. Pan, and R. Stabile, “Photonic integrated WDM cross-connects for optical metro and data center network,” in Proc. Metro Data Center Opt. Netw. Short-Reach Links II; SPIE OPTO, San Francisco, California, United States, 1094603 (2019). [Online]. Available: https://doi.org/10.1117/12.2508897.
[3] J. Kundra, O. Havlís, J. Jedliński, and J. Vojtěch, “Opening up ROADMs: Let us build a disaggregated optical line system,” J. Lightw. Technol., vol. 37, pp. 4041–4051, 2019.
[4] R. Casellas, R. Martínez, R. Vilalta, and Raül Muñoz, “Control, management, and orchestration of optical networks: Evolution, trends, and challenges,” J. Lightw. Technol., vol. 36, pp. 1390–1402, 2018.
[5] M. Chanmegowda, R. Nejabati, and D. Simeonidou, “Software-defined optical networks technology and infrastructure: Enabling software-defined optical network operations,” J. Opt. Commun. Netw., vol. 5, no. 10, pp. A274–A282, 2013.
[6] X. Li et al., “5G-crosshaul network slicing: Enabling multi-tenancy in mobile transport networks,” IEEE Commun. Mag., vol. 55, no. 8, pp. 128–137, Aug. 2017.
[7] R. Casellas, A. Giorgetti, R. Morro, R. Martínez, R. Vilalta, and R. Muñoz, “Enabling network slicing across a disaggregated optical transport network,” in Proc. Opt. Fiber Commun. Conf. (OFC), San Diego, CA, USA, 2019, pp. 1–3.
[8] R. Casellas, R. Martínez, R. Vilalta, and R. Muñoz, “Abstraction and control of multi-domain disaggregated OpenROADM optical networks,” in Proc. Eur. Conf. Opt. Commun. (ECOC), Dublin, Ireland, 2019, pp. 1–4, Paper ECOC2019-W.3.E.2.
[9] M. Bjorklund, Ed., “The YANG 1.1 data modeling language,” IETF Request for Comments RFC7950, Aug. 2016.
[10] R. Enns, Ed., “Network Configuration Protocol NETCONF,” IETF Request for Comments RFC6261, Jun. 2011.
[11] D. Ceccarelli and Y. Lee, Eds., “Framework for abstraction and control of TE networks (ACTN),” IETF Request for Comments RFC8453, Aug. 2018.
[12] Y. Lee et al., “Applicability of YANG models for abstraction and control of traffic engineered networks,” Internet draft, work in progress. [Online]. Available: https://www.ietf.org/id/draft-ietf-teas-actn-yang.txt. Accessed on: Nov. 13, 2019.
[13] The open ROADM multi-source agreement (MSA). [Online]. Available: http://www.openroadm.org. Accessed on: Apr. 12, 2019.
[14] R. Casellas, R. Vilalta, R. Martínez, R. Muñoz, H. Zheng, and Y. Lee, “Experimental validation of the ACTN architecture for flexi-grid optical networks using active stateful hierarchical PCEs,” in Proc. Int. Conf. Transparent Opt. Netw., Girona, Spain, Jul. 2–6, 2017, doi: 10.1109/JCTON.2017.8025182.
[15] N. Neo Antoniades, G. Ellinas, and I. oannis Roudas, Eds., WDM Systems and Networks: Modeling, Simulation, Design and Engineering, New York, NY, USA: Springer, Science and Business Media, Dec. 7, 2011 - Technology and Engineering.
[16] Q. Liu et al., “Inter-domain routing scalability in optical DWDM networks,” in Proc. Int. Conf. Comput. Commun. Networks. Sep. 2008, pp. 1–7, doi: 10.1109/ICCNC.2008.ECP.22.
[17] Q. Ma and P. Steenkiste, “On Path selection for traffic with bandwidth guarantees,” in Proc. Int. Conf. Netw. Protocols, Atlanta, GA, Oct. 1997, pp. 191–202.
[18] I. Iliadis, “Optimal PNNI complex node representations for restrictive costs and minimal path computation time,” *IEEE/ACM Trans. Netw.*, vol. 8, no. 4, pp. 493–506, Aug. 2000.

[19] X. Bruin *et al.*, “Hierarchical routing with QoS constraints in optical transport networks,” in *Proc. 3rd IFIP-TC6 Netw. Conf.*, Athens, Greece, May 2004, pp. 662–674.

[20] R. Casellas *et al.*, “A control plane architecture for multi-domain elastic optical networks: The view of the IDEALIST project,” *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 136–143, Aug. 2016.

[21] R. Morro *et al.*, “Automated end to end carrier ethernet provisioning over a disaggregated WDM metro network with a hierarchical SDN control and monitoring platform,” in *Proc. 44th Eur. Conf. Opt. Commun.*, Roma, Sep. 23–27, 2018, pp. 1–3.

[22] A. Giorgetti, R. Casellas, R. Morro, A. Campanella, and P. Castoldi, “ONOS-controlled disaggregated optical networks,” in *Proc. Opt. Fiber Commun. Conf. Exhib.*, San Diego, CA, USA, 2019, pp. 1–3.

[23] Open Networking Foundation (ONF), Transport API project. [Online]. Available: https://wiki.opennetworking.org/display/OTCC/TAPI. Accessed on: Apr. 12, 2019.

[24] ConfD, management agent software framework for network elements. [Online]. Available: https://www.tail-f.com/confd-basic/. Accessed on: Apr. 12, 2019.

[25] The Metro-haul project. [Online]. Available: http://metro-haul.eu

**Ramon Casellas** (Senior Member, IEEE) graduated from the Polytechnic University of Catalonia, Barcelona, Spain, in 1999 and from ENST, Paris, France, with an Erasmus student exchange program Grant. He received the Ph.D. degree from ENST, in 2002, where he was an Associate Professor. In March 2006, he joined CTTC Optical Networking Department, where he is currently a Senior Researcher. His research interest areas include traffic engineering, control and management of optical transport networks (GMPLS/PCE, SDN, and NFV). He has been involved in several international, national, and industrial R&D projects and has coauthored five book chapters, more than 200 journal and conference papers, and five IETF RFCs. He is an ONF contributor and member of the ONF ODTN project.

**Ricardo Martínez** (Senior Member, IEEE) received the Ph.D. degree in telecommunications engineering from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 2007. He has been actively involved in several public-funded (national and European Union) research and development projects as well as industrial technology transfer projects. Since 2013, he has been a Senior Researcher with the Communication Networks Division, CTTC, Castelldefels, Spain. His research interests include service and resource (cloud and network) orchestration, control and network management architectures for packet and optical transport networks in multiple network segments.

**Ricard Vilalta** (Senior Member, IEEE) received the telecommunications engineering degree and the Ph.D. degree from the Polytechnic University of Catalonia, Barcelona, Spain, in 2007 and 2013. He is a Senior Researcher with the Communication Networks Division, Centre Tecnologic Telecomunicacions Catalunya, Barcelona. His research is focused on SDN/NFV, network virtualization, and network orchestration. He has been involved in international, EU, national and industrial research projects, and has authored and coauthored more than 200 journals, conference papers, and invited talks. He is also involved in standardization activities in ONF, IETF, and ETSI.

**Raül Muñoz** (Senior Member, IEEE) graduated in telecommunications engineering and received the Ph.D. degree in telecommunications from UPC-BarcelonaTech, Barcelona, Spain, in 2001 and 2005, respectively. He is the Head of the Optical Networks and Communications Networks Division, CTTC, Barcelona. He has coordinated the H2020-MSCA-ITN ONFIRE project and the EU-Japan FP7-ICT STRAUSS project. He has authored and coauthored more than 250 journal and international conference papers. His research interests include control and service management architectures (SDN, NFV, and network slicing) for disaggregated optical/packet transport networks with edge/cloud computing.