

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

Ramon Casellas, *Senior Member, IEEE, Member, OSA*, Ricardo Martinez, *Senior Member, IEEE*, Ricard Vilalta, *Senior Member, IEEE*, and Raul Muñoz, *Senior Member, IEEE*

(Invited 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 in order to scale, or due to confidentiality and/or interoperability reasons, and the need to virtualize the (multi-domain) transport network emerges as a key requirement to support functions such as network slicing 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 supporting such requirements, such as the *Abstraction and Control of Traffic Engineering Networks of the Internet Engineering Task Force*, known as ACTN. In this paper, 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 testbed implementation.

Index Terms—Multi-Domain Optical Network Control, ACTN, Software Defined Networking, OpenROADM device model

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 terms of 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 or to control the allocated partitions [5]–[7]. Tenants are given abstracted topology views of the physical underlying network and are allowed to utilize and independently control allocated virtual network resources as if were real. In this regard, questions related to the efficient and optimal abstraction of the underlying networks or how to support virtualization and slicing remain open.

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

R. Casellas, R. Martinez, R. Vilalta, and R. Muñoz are with CTTC/CERCA, Optical Networks and Systems Department, Castelldefels, Barcelona, Spain e-mail: ramon.casellas@cttc.es

This work has been partially supported by the EU-H2020 Metro-Haul Project with G.A 761727 and the Spanish AURORAS (RTI2018-099178) project.

Manuscript received November 15, 2019; revised February X, 2020.

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 mode. 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 the support of 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 control and management aspects (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 a 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. The data is arranged into one or multiple configuration datastores. A configuration datastore is the complete set of configuration information that is required to get a device from its initial default state into a desired operational state. 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. When accessing configuration or state data, with NETCONF operations, subtree filter expressions can select subtrees, providing a great degree of flexibility. 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 orchestration of the network resources to support the customer services, facilitates 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 main 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 MPI is achieved by a range of YANG modules dependent on the technology of each underlying network. The applicability of YANG models to ACTN can be found in [12].

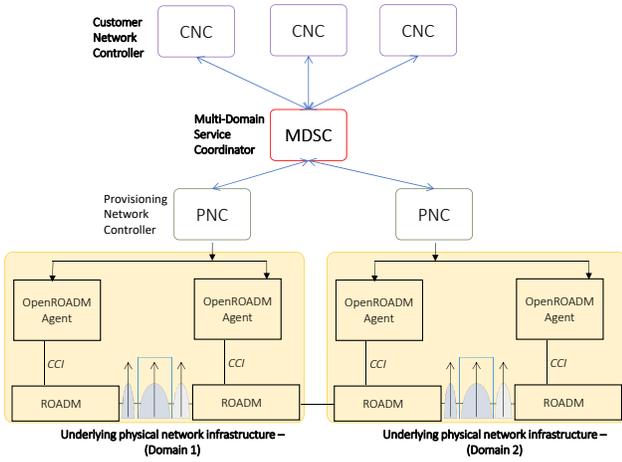


Fig. 1. Abstraction and Control of Traffic Engineered Networks architecture.

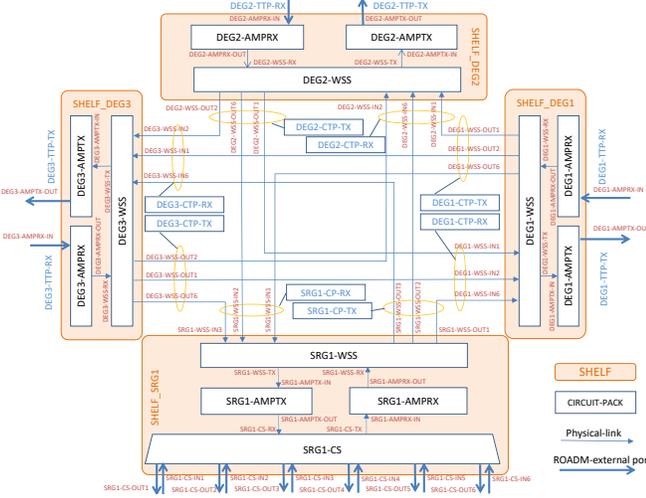


Fig. 2. Basic sketch of the OpenROADM device model with 3 degrees (DEG) and a single add-drop Shared Risk Group (SRG).

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). 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. 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 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 within *abstracted ROADMs (aROADMs)* are latter mapped and induce an 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 synthesize 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,

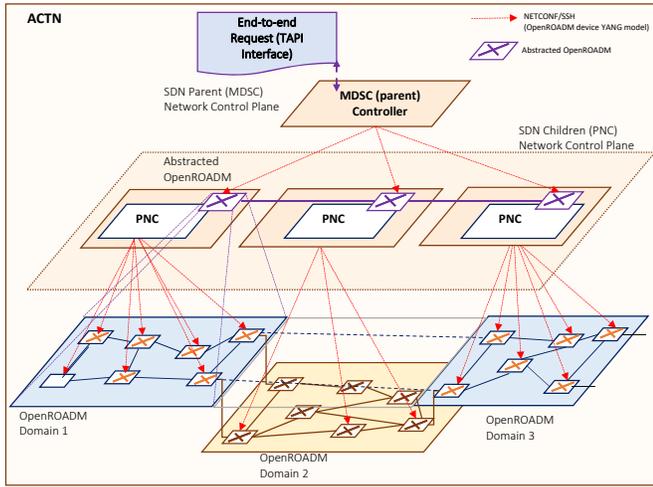


Fig. 3. Control Plane Architecture using ACTN for a Multi-domain flexi-grid network based on the OpenROADM device model

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 (MSDC) 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 MSDC 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.

2) *Link mesh abstraction*: Due to limitations related to the optical technology, a single node abstraction may not be 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

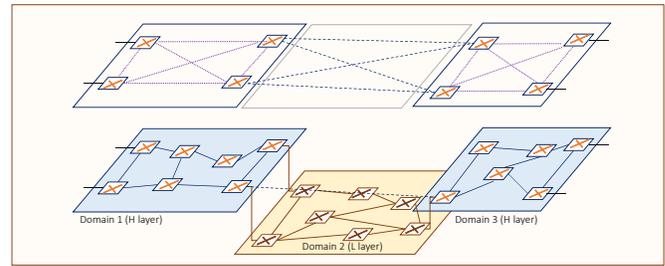


Fig. 4. Illustration of topology aggregation using a virtual link (full) mesh topology.

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).

Let us note that, in the ACTN framework, topology aggregation methods are referred to as topology abstraction types, and cover: i) Native/White Topology, in which PNC provides the actual network topology to the MSDC without any hiding or filtering of information; ii) black topology, where there is a minimal representation of the edge-to-edge topology without disclosing any internal connectivity (e.g. the entire domain network may be abstracted as a single abstract node) and iii) Grey topology, where the PNC exposes an abstract topology containing all PNC domain border nodes and an abstraction of the connectivity. Type A implies a full mesh of TE links and type B implies a more detailed network comprising internal abstract nodes and abstracted links.

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 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.

To illustrate the topology abstraction 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).

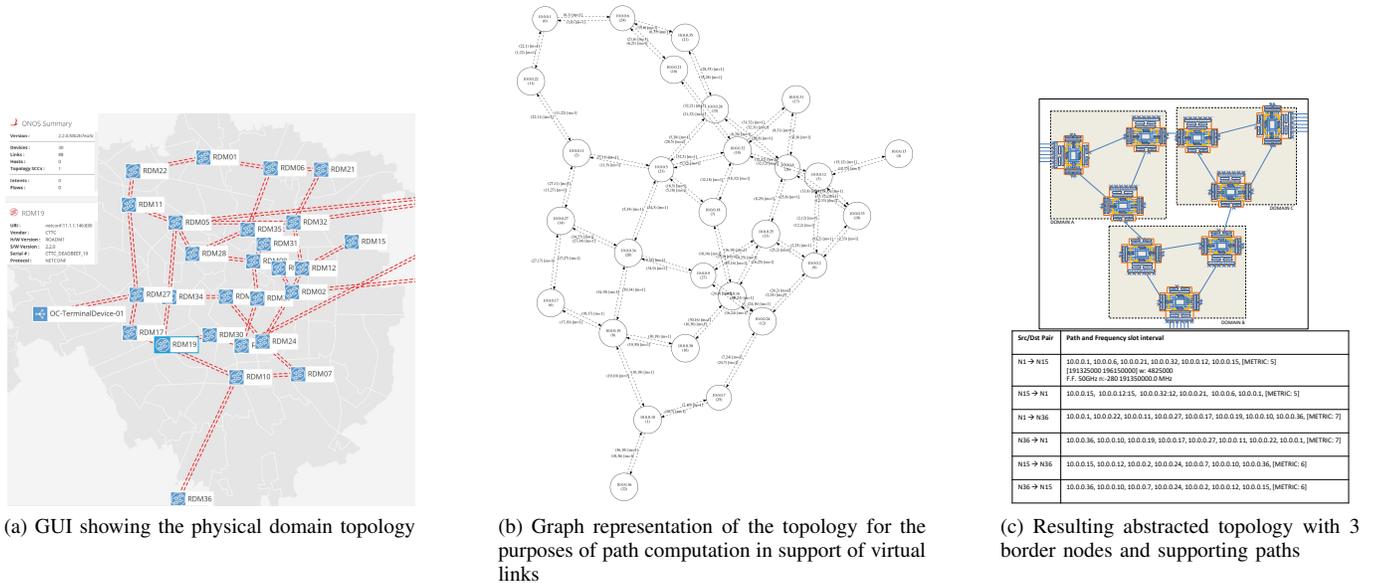


Fig. 5. Topology used to illustrate the network aggregation in terms of virtual link mesh, based on a typical Italian/European big city (courtesy of TIM).

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. In a Intel PC i7 7700, with 32Gb RAM, the implemented algorithm is able to compute the mesh of 756 paths in $O(20)$ seconds. However, let us note that the common case is that no all nodes in such a transport network are border nodes. 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 $O(200)$ ms. 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 MSDC level.

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 or 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 ROADMs). 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 (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 a link management and automated neighbor discovery, links between ROADMs in a domains 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 ROADMs 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 50GHz 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 ROADM degrees; 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 `jedit-config` operation) over the OMS interfaces (by convention named `MC-TTP-DEG1-TTP-RX-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 $O(800)$ 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 list of available Service Interface Points, followed by the request to establish a network media channel connectivity service, using a REST POST operation) followed

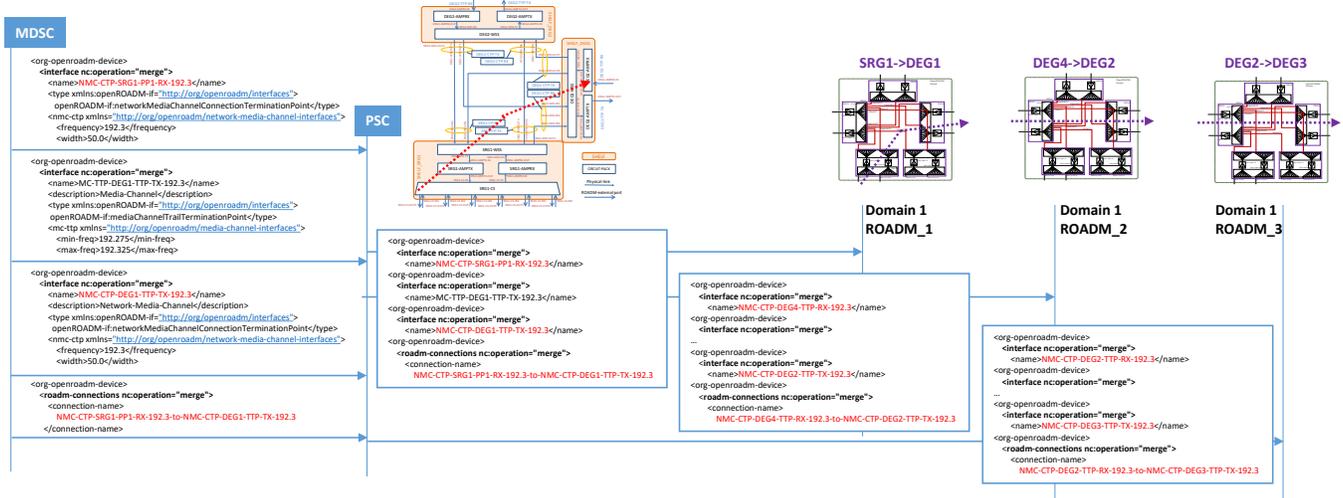


Fig. 6. NETCONF messages for the provisioning of a connection across a domain, involving MSDC-PNC and PNC-devices.

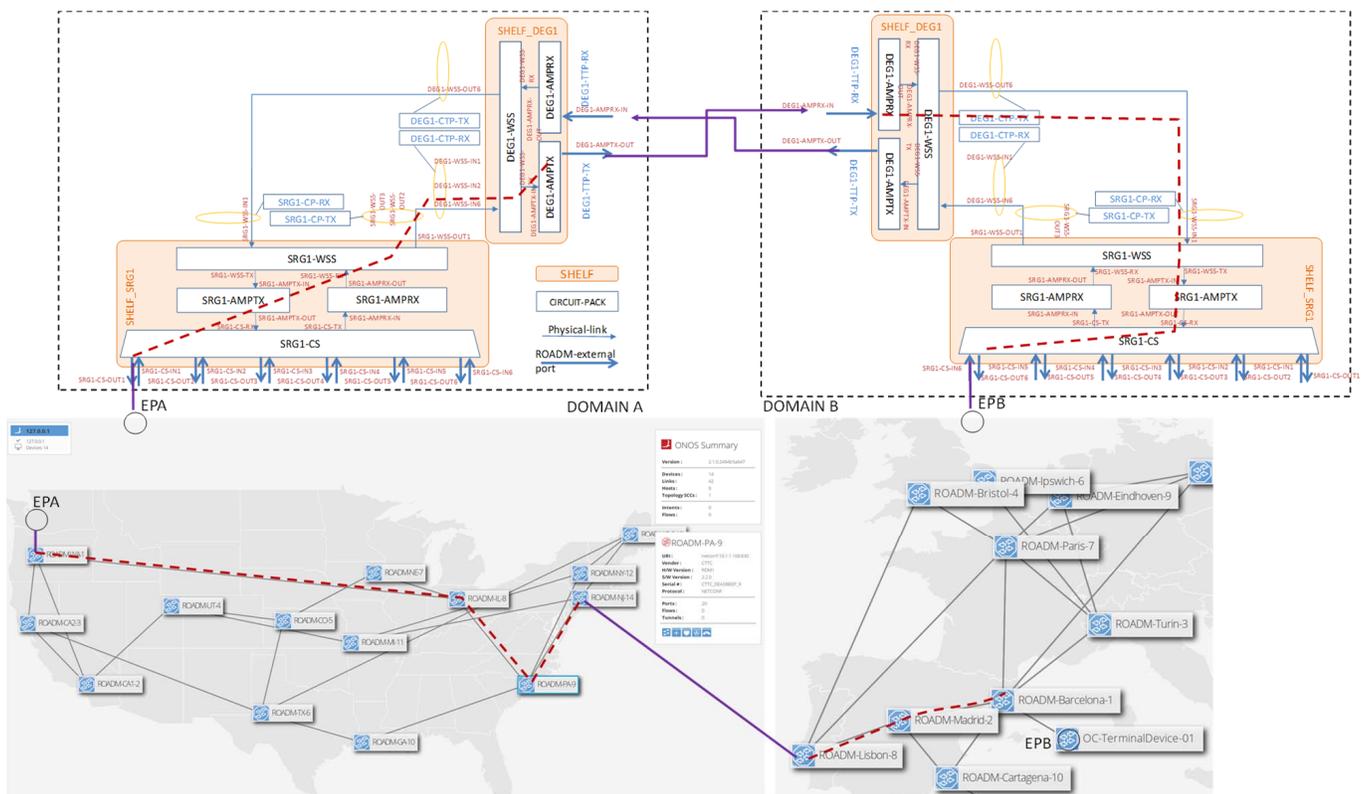


Fig. 7. Experimental evaluation scenario, encompassing 2 domains. The left one emulates the 14-node NSFNET topology and a 10-node from the Metro-Haul project.

by the NETCONF over SSH NETCONF 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 $O(500)$ 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 involved NETCONF over SSH exchanges between the PNC of domain B, the terminal device B, as well as the NETCONF agents for the ROADMs in Lisbon, Madrid, and Barcelona. The captured number of IP packets is $O(100)$, since the PNC exchanges multiple NETCONF edit-config messages with the ROADMs as well as the receiving terminal device. The required throughput is relatively low, and such

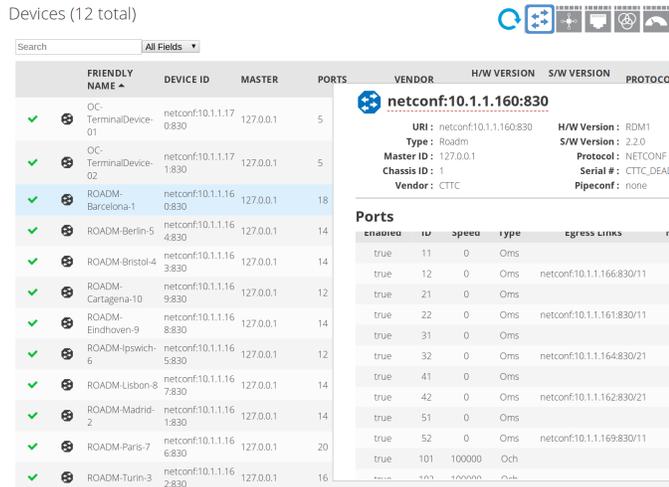


Fig. 8. ONOS GUI showing the detail of the Domain B, including each node NETCONF identifiers, port count, etc.

HTTP	SOURCE	DESTINATION	PROTOCOL	LENGTH	INFO
7	192.168.1.1	10.1.1.160	HTTP	290	GET /onos/restconf/data/tapi-common:...
11	10.1.1.160	192.168.1.1	HTTP	75	HTTP/1.1. 200 OK (application/yang-d...
19	10.1.1.160	192.168.1.1	HTTP	1864	POST /onos/restconf/operations/tapi-...
21	192.168.1.1	10.1.1.160	HTTP	659	HTTP/1.1. 200 OK (application/yang-d...
26	10.1.1.160	10.1.1.160	TCP	404	34452 -> 830 [PSH, ACK] Seq=1 Ack=1 W...
27	10.1.1.160	10.1.1.160	TCP	116	830 -> 34452 [PSH, ACK] Seq=1 Ack=337 W...
28	10.1.1.160	10.1.1.160	TCP	68	34452 -> 830 [ACK] Seq=337 Ack=49 Win...
29	10.1.1.160	10.1.1.160	TCP	1516	830 -> 34452 [ACK] Seq=49 Ack=337 Win...
30	10.1.1.160	10.1.1.160	TCP	68	34452 -> 830 [ACK] Seq=337 Ack=1497 W...
31	10.1.1.160	10.1.1.160	TCP	1516	830 -> 34452 [ACK] Seq=1497 Ack=337 W...
32	10.1.1.160	10.1.1.160	TCP	68	34452 -> 830 [ACK] Seq=337 Ack=2945 W...
33	10.1.1.160	10.1.1.160	TCP	500	830 -> 34452 [PSH, ACK] Seq=2945 Ack=...
34	10.1.1.160	10.1.1.160	TCP	68	34452 -> 830 [ACK] Seq=337 Ack=3377 W...
35	10.1.1.160	10.1.1.160	TCP	612	34452 -> 830 [PSH, ACK] Seq=337 Ack=3...
36	10.1.1.160	10.1.1.160	TCP	116	830 -> 34452 [PSH, ACK] Seq=3377 Ack=...
37	10.1.1.160	10.1.1.160	TCP	1516	830 -> 34452 [ACK] Seq=3425 Ack=881 W...
38	10.1.1.160	10.1.1.160	TCP	68	34452 -> 830 [ACK] Seq=881 Ack=4873 W...
39	10.1.1.160	10.1.1.160	TCP	1516	830 -> 34452 [ACK] Seq=4873 Ack=881 W...
40	10.1.1.160	10.1.1.160	TCP	1228	830 -> 34452 [PSH, ACK] Seq=6321 Ack=...
41	10.1.1.160	10.1.1.160	TCP	68	34452 -> 830 [ACK] Seq=881 Ack=7473 W...
42	10.1.1.160	10.1.1.160	TCP	740	34452 -> 830 [PSH, ACK] Seq=881 Ack=7...
43	10.1.1.160	10.1.1.160	TCP	116	830 -> 34452 [PSH, ACK] Seq=7473 Ack=...
44	10.1.1.160	10.1.1.160	TCP	1124	58244 -> 830 [PSH, ACK] Seq=1 Ack=1 W...
45	10.1.1.160	10.1.1.160	TCP	1124	34742 -> 830 [PSH, ACK] Seq=1 Ack=1 W...
46	10.1.1.160	10.1.1.160	TCP	116	830 -> 34742 [PSH, ACK] Seq=1 Ack=105 W...
47	10.1.1.160	10.1.1.160	TCP	68	34742 -> 830 [ACK] Seq=105 Ack=49 Win...
48	10.1.1.160	10.1.1.160	TCP	116	830 -> 58244 [PSH, ACK] Seq=1 Ack=105 W...
49	10.1.1.160	10.1.1.160	TCP	68	34452 -> 830 [ACK] Seq=1553 Ack=7521 W...
50	10.1.1.160	10.1.1.160	TCP	68	58244 -> 830 [ACK] Seq=1553 Ack=49 Win...
51	10.1.1.160	10.1.1.160	TCP	916	830 -> 34464 [PSH, ACK] Seq=1 Ack=1 W...
52	10.1.1.160	10.1.1.160	TCP	244	830 -> 34452 [PSH, ACK] Seq=7521 Ack=...
53	10.1.1.160	10.1.1.160	TCP	68	34452 -> 830 [ACK] Seq=1553 Ack=7697 W...
54	10.1.1.160	10.1.1.160	TCP	836	830 -> 34760 [PSH, ACK] Seq=1 Ack=1 W...
55	10.1.1.160	10.1.1.160	TCP	244	830 -> 34742 [PSH, ACK] Seq=49 Ack=10...

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

control plane connectivity can be provided over a diversity of technologies, being latency the critical aspect. Setup delays are computed without taking into account hardware configuration delays with real ROADMs, which can be on the order or seconds/minutes. Setup delay due to control plane only has little impact on the performance.

V. CONCLUSION

We have experimentally validated a control plane for multi-domain optical networks based on the ACTN framework and the OpenROADM device model, used in the abstraction of each domain and the actual devices. An end-to-end network media channel has been demonstrated hierarchically, using TAPI NBI, requested between transponders client ports. The delay and control plane throughout is within targeted values, validating the proposed approach. Note that one of the takeaway messages is that having a common data model

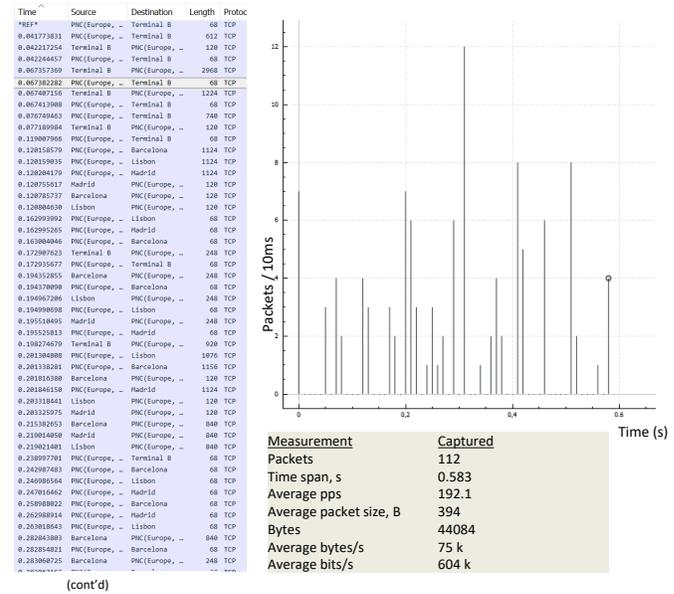


Fig. 10. Wireshark capture and I/O plot of the exchanges between the PNC controller of the domain B and the involved optical devices

both for physical and and abstracted ROADMs simplifies the overall process, allowing the same controller instances to be applied over real or abstracted ROADMs. Beyond the fact that the experimental validation supports the feasibility of the approach, let us also note that the required throughput of the SDN interfaces for configuration is relatively low, and such control plane connectivity can be provided over a diversity of technologies.

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 Giorgetti 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] Emilio Riccardi, Paul Gunning, Óscar González de Dios, Marco Quagliotti, Víctor López, and Andrew Lord, *An Operator view on the Introduction of White Boxes into Optical Networks*, J. Lightwave Technol. 36, 3062-3072 (2018)
- [2] N. Calabretta, K. Prifti, N. Tessema, X. Xue, B. Pan, R. Stabile, *Photonic integrated WDMcross-connects for optical metro and data center network*, in Proceedings of Metro and Data Center Optical Networks and Short-Reach Links II; 1094603 (2019) <https://doi.org/10.1117/12.2508897>, SPIE OPTO, 2019, San Francisco, California, United States
- [3] Jan Kundrát, Ondřej Havlíš, Jaroslav Jedliňský, and Josef Vojtěch, *Opening up ROADMs: Let Us Build a Disaggregated Open Optical Line System*, J. Lightwave Technol. 37, 4041-4051 (2019)
- [4] Ramon Casellas, Ricardo Martínez, Ricard Vilalta, and Raül Muñoz, *Control, Management, and Orchestration of Optical Networks: Evolution, Trends, and Challenges*, J. Lightwave Technol. 36, 1390-1402 (2018)

- [5] Channegowda, M., Nejabati, R., Simeonidou, D. "Software-defined optical networks technology and infrastructure: Enabling software-defined optical network operations", *Journal of Optical Communications and Networking*, 5, (10), A274-A282, 2013
- [6] Li, X., Casellas, R., Landi, G. et al, *5G-Crosshaul Network Slicing: Enabling Multi-Tenancy in Mobile Transport Networks*, *IEEE Communications Magazine*, special issue on Network Slicing in 5G systems, August 2017, Vol. 55, No. 8, August 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 *Optical Fiber Communication Conference (OFC) 2019*, OSA Technical Digest (Optical Society of America, 2019), paper Tu3H.3.
- [8] Ramon Casellas, Ricardo Martínez, Ricard Vilalta, Raul Muñoz, "Abstraction and Control of Multi-domain disaggregated OpenROADM Optical Networks", in *proceedings of the European Conference in Optical Communications (ECOC)*, Dublin, Ireland, 2019.
- [9] M. Bjorklund, Ed., "The YANG 1.1 Data Modeling Language", *IETF Request for Comments 7950*, August 2016.
- [10] R. Enns, ed. "Network Configuration Protocol NETCONF", *IETF Request for Comments 6241*, June 2011.
- [11] Ceccarelli, D., Lee, Y., editors, "Framework for Abstraction and Control of TE Networks (ACTN)", *IETF RFC8453*, august 2018.
- [12] Y. Lee et al., "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", internet draft, work in progress <https://www.ietf.org/id/draft-ietf-teas-actn-yang.txt> Accessed: 2019-11-13
- [13] The Open ROADM Multi-Source Agreement (MSA) <http://www.openroadm.org> Accessed: 2019-04-12
- [14] R. Casellas, R. Vilalta, R. Martínez, R. Muñoz, H. Zheng, Y. Lee, *Experimental Validation of the ACTN architecture for flexi-grid optical networks using Active Stateful Hierarchical PCEs*, in *Proc. 19 International Conference on Transparent Optical Networks (ICTON2017)*, 2-6 July, 2017, Girona, Spain, July 2017.
- [15] Neophytos Neo Antoniadis, Georgios Ellinas, Ioannis Roudas, *WDM Systems and Networks: Modeling, Simulation, Design and Engineering*, Springer Science and Business Media, Dec 7, 2011 - Technology and Engineering
- [16] Q. Liu, C. Xie, T. Frangieh, N. Ghani, A. Gumaste, N .S. V. Rao, T. Lehman, *Inter-Domain Routing Scalability in Optical DWDM Networks*, September 2008, DOI: 10.1109/ICCCN.2008.ECP.22
- [17] Q. Ma, P. Steenkiste, "On Path Selection for Traffic with Bandwidth Guarantees", *IEEE ICNP 1997*, Atlanta, GA, October 1997.
- [18] I. Iliadis, *Optimal PNNI Complex Node Representations for Restrictive Costs and Minimal Path Computation Time*, *IEEE/ACM Transactions on Networking*, Vol. 8, No. 4, August 2000.
- [19] X. Bruin, et al, *Hierarchical Routing With QoS Constraints in Optical Transport Networks*, 3rd IFIP-TC6 Networking Conference, Athens, Greece, May 2004.
- [20] R. Casellas, O. González de Dios, F. Paolucci, R. Morro, V. López, D. King, R. Martínez, F. Cugini, R. Muñoz, A. Farrel, R. Vilalta, *A Control Plane Architecture for Multi-Domain Elastic Optical Networks: the view of the IDEALIST project*, *IEEE Communications Magazine*, Vol. 54, No. 8, pp. 136-143, August 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 *Proceedings of 44th European Conference on Optical Communication (ECOC 2018)*, 23-27 September 2018, Roma (Italy).
- [22] A. Giorgetti, R. Casellas, R. Morro, A. Campanella, and P. Castoldi, *ONOS-controlled Disaggregated Optical Networks*, in *Optical Fiber Communication Conference (OFC) 2019*, OSA Technical Digest (Optical Society of America, 2019), paper Tu3H.2.
- [23] Open Networking Foundation (ONF), *Transport API project* <https://wiki.opennetworking.org/display/OTCC/TAPI> Accessed: 2019-04-12
- [24] ConfD, management agent software framework for network elements <https://www.tail-f.com/confd-basic/> Accessed: 2019-4-12.
- [25] The Metro-haul project <http://metro-haul.eu>

Ramon Casellas Ramon Casellas (SM'12) graduated in 1999 from UPC, Barcelona and from the ENST, Paris, with an Erasmus student exchange program grant. He completed a PhD degree in 2002 and worked at the ENST as an Associate Professor. In March 2006, he joined the CTTC Optical Networking Dept., where he is currently holding a Senior Researcher

position. 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 co-authored 5 book chapters, over 200 journal and conference papers and 5 IETF RFCs. He is an ONF contributor and member of the ONF ODTN project.

Ricardo Martínez Ricardo Martínez (SM'14) received Ph.D. degree in 2007 in telecommunications engineering, from the Universitat Politècnica de Catalunya, Barcelona, Spain. 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 is Senior Researcher of the Communication Networks Division at 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 Ricard Vilalta (SM'17) has a telecommunications engineering degree (2007) and Ph.D. degree (2013), at UPC, Spain. He is senior researcher at CTTC, in the Communication Networks Division. 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 published more than 200 journals, conference papers and invited talks. He is also involved in standardization activities in ONF, IETF and ETSI.

Raül Muñoz Raül Muñoz (SM'12) graduated in telecommunications engineering in 2001 and received a Ph.D. degree in telecommunications in 2005, both from UPC-BarcelonaTech, Spain. He is Head of the Optical Networks and Communications Networks Division Manager at CTTC (Barcelona, Spain). He has coordinated the H2020-MSCA-ITN ONFIRE project and the EU-Japan FP7-ICT STRAUSS project. He has published over 250 journal and international conference papers. His research interests include control and service management architectures (SDN, NFV, network slicing) for disaggregated optical/packet transport networks with edge/cloud computing.