Comprehensive survey on T-SDN: Software-defined Networking for Transport Networks

Rodolfo Alvizu, *Student Member, IEEE*, Guido Maier, *Senior Member, IEEE*, Navin Kukreja, Achille Pattavina, *Senior Member, IEEE*, Roberto Morro, Alessandro Capello and Carlo Cavazzoni

***Abstract*—Paradoxically, with an ever-increasing traffic demand, today transport-network operators experience a progressive erosion of their margins. The alarms of change are set, and Software Define Networking (SDN) is coming to the rescue with the promise of reducing Capital expenditures (CapEx) and Operational expenses (OpEx). Driven by economic needs and network innovation facilities, today transport SDN (T-SDN) is a reality. It gained big momentum in the last years, however in the networking industry, the transport network will be perhaps the last segment to embrace SDN, mainly due to the heterogeneous nature and complexity of the optical equipment composing it. This survey guides the reader through a fascinating technological adventure that provides an organic analysis of the T-SDN development and evolution considering contributions from: academic research, standardization bodies, industrial development, open source projects and alliances among them. After creating a comprehensive picture of T-SDN, we provide an analysis of many open issues that are expected to need significant future work, and give our vision in this path towards a fully programmable and dynamic transport network.**

***Index Terms*—Software defined networking, optical transport network, transport SDN, software defined optical networking, network programmability, orchestration, transport API, network controller, network virtualization, network function virtualization, OpenFlow, GMPLS.**

I. Introduction

FOR a telecom operator or a carrier, the possibility of having the full control of his network in his own hands, without depending on another company (e.g. the vendor of network equipment) for any upgrade or new feature, has been a dream for a long time. Now, turning this dream into reality seems to be at hand. Telco carriers are seeking the opportunity of extending the success that Software Defined Networking (SDN) met in the data-center world to the large-scale geographic networks: that is essentially the Transport-SDN (T-SDN) phenomenon. However, as we will try to explain in this survey, exporting the SDN model to carrier networks is all but simple, for several technical, procedural and economic reasons we will present in the following.

Certainly, the T-SDN (r)evolution, if implemented in a short time, will have the potential of bringing enormous advantages for the carriers and the network operators, especially in terms of cost reduction and revenue amplification: those are exactly the moves operators need to escape the current impasse which is jeopardizing their future growth.

The need for cost reduction is particularly urgent for the operators of transport networks, i.e. those large networks of substantial geographical extension (regional, national, continental or even intercontinental) providing infrastructure (links and equipment) to move large and aggregated data traffic (in the form of streams of packets). Transport-network infrastructure is expensive, especially in terms of operating costs, and must constantly be upgraded and expanded to keep the pace with the increase of traffic. This rapid growth is generated by the applications offered from the giants of Internet (the so-called over-the-tops) to users (typically, for free) and that users are more and more eager to enjoy. Therefore, taken in the vise of ever-increasing traffic and practically flat revenues from the subscribers, transport-network operators are struggling. They assist to the progressive erosion of their margins, while the same over-the-tops at the basis of their worries are capturing the largest share of the ICT market value.

So, the only possibility to preserve margins is reducing Capital Expenditure (CapEx) and Operational Expenses (OpEx). However, current technologies and architectures are constrained by their operational complexity and static nature, that lead to inefficient network utilization and over-provisioning of resources. As a consequence, operators have set the alarms of change and have started to look at Software-Defined Networking (SDN). SDN is buzzing the networking world with the promise of increasing the programmability and innovation pace as well as reduction of OpEx and CapEx.

As we will show in the paper, there are strong reasons to believe that SDN can actually be a cost-slashing technology, and in fact, it has already proven to be such for datacenter operators and at the edge of the transport networks. The scenario is more difficult in case of operators of transport networks, especially because of the heterogeneous nature of the equipment composing the core network, compared to the relative uniformity of switches used in the datacenters and at the edges. As we will convey in the following parts, the main problem is to apply the SDN concept in an environment that was not natively developed to support this new control technology. Therefore, in our paper we will not speak about SDN in general, but we will focus on the specific Transport-SDN (or T-SDN) scenario, investigating the topic under the point of view of large transport-network operators such as TIM (the Italian incumbent operator).

R. Alvizu, G. Maier and A. Pattavina are with Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Milan, Italy. E-mail: (rodolfoenrique.alvizu, guido.maier, achille.pattavina)@polimi.it

R. Alvizu, G. Maier and A. Pattavina are also with SWAN networks, spin-off company of Politecnico di Milano, Italy.

N. Kukreja was with DEIB, Politecnico di Milano at the time this work was done. E-mail: navin.kukreja@gmail.com

R. Morro, A. Capello and C. Cavazzoni are with TIM S.p.A., Turin, Italy. E-mail: (alessandro.capello, carlo.cavazzoni, roberto.morro)@telecomitalia.it

As the reader will soon realize, the structure of the following is quite articulated and segmented, because the matter itself is complicated. To provide a comprehensive picture of T-SDN development we need to consider many types of system architectures and the interplay of many contributions coming from different sides: academic research, standardization bodies, large international projects, industrial development, industrial and open source alliances, and so on. We hope we will be able to guide the reader in a smooth navigation (also with an aid from the figures), providing an organic analysis.

At the end, this survey paper is about a fascinating technological adventure, in which an innovation, initially underestimated, then exalted as salvific, is now currently undergoing a careful redesign process to clear many details that before have been overlooked. All in a nutshell of years, because the time-to-market is a critical factor, under the pressure of cost reduction. And the end of the transport-networks' softwarization process seems still far away.

A. Programmability: a shift of paradigm in networking

Fig. 1 depicts the legacy network architectures, which are based on purpose and vendor-specific systems composed by highly integrated and specialized forwarding chips, proprietary operating systems and predefined features. In order to apply new network policies, an operator has to configure each device using vendor-specific command line interfaces (CLI)s. To provide a new feature, an operator may wait for a long period before the device vendor releases a software upgrade that supports that expected feature. The distributed network intelligence makes hard to understand the current state of the network and to apply new forwarding rules. The integrated system presented in Fig. 1 imposes a challenge towards innovation and network evolution. A clear example is the conversion from IPv4 to IPv6 that after more than 10 years is not near to be fully accomplished.

On the other hand, SDN is based on open systems, and purpose and features are provided through development of software and applications. The Open Networking Foundation (ONF) [1], an organization dedicated to the promotion and adoption of SDN, defines it as a programmable network architecture where the control plane is separated from the data plane (forwarding hardware) as depicted in Fig. 2.

By decoupling control and data planes, the network intelligence and state can be logically centralized, the forwarding infrastructure can be conveniently abstracted to the application plane, and innovation is boosted independently at each plane [2].

Before the formal definition of SDN, in 2008 McKeown et al. [3] proposed the OpenFlow switch. OpenFlow was created to foster innovation in campus networks by allowing researchers to test their ideas in an isolated "slice" of the real network [3]. Such approach breaks the limitations of an "ossified" network infrastructure by separating its control and forwarding planes. In the same year, Gude et al. [4] proposed a network operating system called NOX. A Network Operating System (NOS) provides centralized programming interfaces to the network (called Northbound Interfaces: NBI). Using the NBIs provided by a NOS, the applications are able to exploit the logically centralized view of the network. OpenFlow and NOX paved the way to the definition of SDN architecture (initially called the NOX-based network).

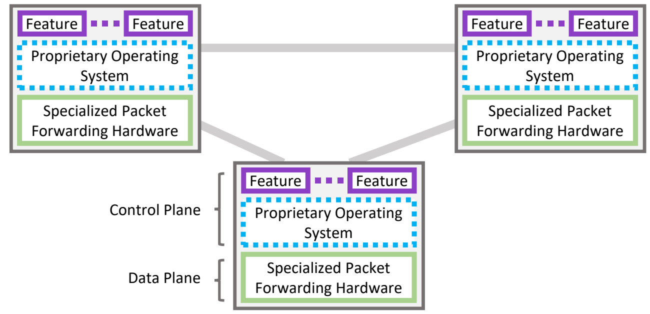

Fig. 1. Legacy network architecture, composed by equipment that integrates proprietary specialized forwarding hardware, proprietary operating system and predefined set of distributed control and management features.

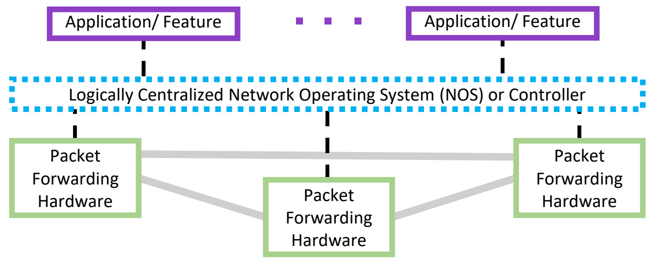

Fig. 2. Basic SDN network architecture, composed by commodity-hardware packet forwarding devices, a logically centralized controller that collects network state and push forwarding rules. Control and management features are implemented as applications.

In 2009 the first OpenFlow switch Specification version 1.0 was published [5]: it describes an open and standard interface for the communication between a NOS implementation (e.g., NOX) and simple data plane Network element (NE). OpenFlow provides a data plane abstraction based on flow tables. OpenFlow was conceived in [3] and [5] for packet switching networks. Thus, a flow is a set of packets that share a traffic relation identified by a combination of packet headers. The flow abstraction was later extended to cover also circuits.

The ONF [1] is standardizing OpenFlow, the today's de facto SDN technology [6]. There are more than 30 SDN controllers in the market today, and a growing support from vendors that already presented commercial solutions for packet switching domains [7].

The evolution of SDN is motivated by three main markets: enterprises, cloud service providers (datacenters) and telecommunication service providers [8]. As depicted in Fig. 3, datacenters and enterprises (which mostly use networks based on packet switching) have experienced a fast development of SDN solutions and worldwide SDN deployments [6][7]. Data centers and big companies like Google were the first to deploy SDN-based solutions [9].

The telecommunication service providers are far behind SDN deployments due to the challenges and complexity of transport networks. Transport networks are composed by heterogeneous multi-layer, multi-domain and multi-vendor archi-

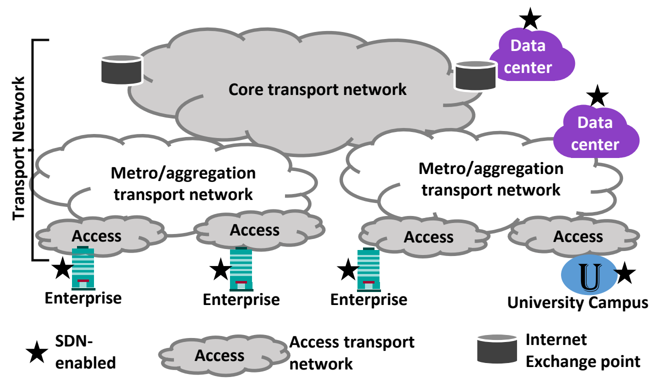

Fig. 3. SDN has been deployed in packet based networks: datacenters, over-the-top companies and enterprises. The implementation of SDN in transport networks (access, aggregation, metro and core), know as transport-SDN (T-SDN), still represents a challenge.

tectures. While SDN was conceived for packet-oriented layers, the transport network also involves circuit-oriented layers and must control the complexity of optical and/or wireless domains. Moreover, the optical network elements have vendor-specific implementations that lead to an heterogeneous data plane that is not easily represented with OpenFlow semantics.

Transport Software-Defined Networking (T-SDN) is an SDN-based architecture for the control and management of transport networks, that could involve multi-layer, multi-domain and multi-vendor scenarios. Transport networks have features usually not present in computer networks where the SDN paradigm arose, like resilience (protection or restoration mechanisms to assure Service Level Agreements sometimes implemented in coordination between data and control plane), more sophisticated architectures and heterogeneous technologies (OTN, OCh, MPLS, PBB, ...[1]), that need to be taken into account when applying the concepts introduced by SDN. In this sense, T-SDN is an extension of SDN introducing different abstractions, interfaces, protocols, and control plane elements to cope with transport networks peculiarities and to overcome the limitations of OpenFlow in this field.

Telecommunication operators and transport service providers have strong interests in deploying T-SDN to migrate their transport infrastructure from an “ossified” and static architecture to a dynamic and programmable system, possibly saving the huge investments made during the last decades. However, T-SDN still represents a major challenge and there is no consolidated commercial solution nor stable standards so far. Some optical vendors, in collaboration with Open Source projects, are at the initial stage of their T-SDN solutions. Standardization bodies are working to guarantee the interoperability of vendor solutions, but the standardization process is far from being completed.

B. Related surveys and tutorials

Multiple survey papers on SDN have been recently published [6], [7], [10], [11]. However, those works present a broad overview of SDN-technologies in multiple areas, from datacenters to wireless and optical networks. A survey and categorization of hypervisors for SDN is provided by [12].

The surveys on optical transport SDN solutions started with the first stage of its evolutionary path, where the focus was to implement SDN concepts into single domain optical networks for a unified control of optical and IP layers [13]–[17]. This necessary evolutionary step is called in literature Software-defined Optical Networks (SDON). However, apart from being multi-layer, transport networks are composed by multiple domains given by the segment (access, metropolitan and core/backbone), the technology, and even delimited by vendor islands. For instance, Cvijetic et al surveyed the specific case of SDN and OpenFlow for optical access networks [18], [19]. The Authors of [20] provide an overview that includes: monolithic control plane architectures for single domain transport networks (SDON) and the second evolutionary steps of T-SDN based on hierarchical control plane architectures for multi-domain transport networks. Ref. [20] focused on SDN research activities for optical access, metro and core networks, with a special focus on passive optical access.

In [21], the author gives an overview on SDN orchestration for the multi-domain optical datacenter networking. In [22] was presented a survey that focused on the interoperability issues of network management for carrier-grade networks, with a focus on Multi-Technology Operations System Interface (MTOSI), NETCONF and the advent of SDN.

C. Aim and organization of this paper

The aim of this paper is to provide the complete picture, classification and historical evolution of T-SDN developments taking into account the whole ecosystem of transport network players composed by: academic research, standardization bodies, large international projects, industrial vendors, and open source and industrial alliances. T-SDN is an open subject with a very fast innovation pace, thus this paper provides list of areas in T-SDN architecture that are expected to need significant future work, and present our understanding of the T-SDN future.

This survey uses the sections as building blocks to generate a full picture of T-SDN. Section II explains Software Defined Networking in a nutshell. Section III describes the enabling technologies and challenges of extending SDN over optical transport networks (T-SDN). Section IV provides a brief summary of T-SDN historical evolution to better guide the reader through the rest of this paper. Section V expounds the first academic research efforts in T-SDN, that are classified as Monolithic Control Plane architectures (SDON). Section VI details on hierarchical control plane architectures for T-SDN (HT-SDN) that are more suitable for the multi-vendor, multi-layer and multi-domain nature of transport networks. Section VII introduces virtualization architectures, algorithms and strategies for Network Function Virtualization (NFV) in T-SDN. Section VIII describes research efforts on protection, restoration, segment routing and emulation of T-SDN. Section IX present the activities from the main standardization bodies regarding T-SDN architectures. Section X provides an insight

[1] Optical Transport Network (OTN), OTN Optical Channel (OCh), Multi-Protocol Label Switching (MPLS), Provider Backbone Bridge (PBB)

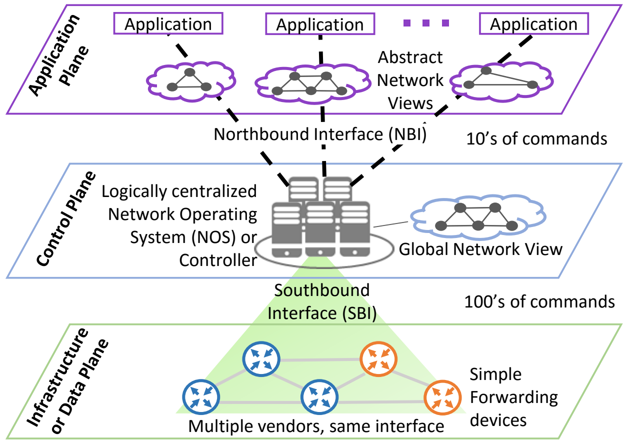

Fig. 4. SDN architecture is composed by a data plane that forwards the traffic and provides open interfaces (SBI) to the control plane. This control plane maintains a global network view, installs forwarding rules in the data plane and exposes open interfaces (NBI) to the application plane. The applications implement the network intelligence based on abstract network views.

into the main Open Source T-SDN related projects. Section XI lists and compares the most influential vendors in T-SDN, including *black-box* and *white-box* solutions. Section XII identifies open research areas, while section XIII gives our vision on the future of this topic and concludes this survey on the path towards a fully programmable and dynamic transport network. At the end of this paper we provide an appendix of abbreviations.

II. Software Defined Networking in a nutshell

SDN is an emerging architecture for designing, building and managing networks. The basic idea is to abstract the underlying complexity and decouple the control plane and management plane from data plane. SDN is changing every aspect of today's networking world and is driving disruptive innovation in every networking sectors - packet switching, wireless access network, enterprise network, datacenter, cloud computing, virtualization, and finally also optical switching and transport networks, that are the focus of this review.

A. SDN-Architecture Planes

The following subsections describe the basic SDN architecture: Fig. 4 shall be used as reference. Generally, SDN is composed by three main planes with specific functionalities and interfaces. The components inside each plane may vary from those presented in Fig. 4. Covering every aspect of SDN is out of our scope, for a deeper review on SDN-architecture we refer the interested reader to [7], [23]–[25].

Starting from the bottom of Fig. 4 and moving towards the upper part, we identify the following planes:

1) Data Plane (DP): the DP is at the bottom of the SDN architecture, it is responsible for handling packets in the datapath based on policies received from the Control Plane (CP). The data plane is composed by physical or virtual traffic forwarding and processing network elements (NE)s like switches, routers, and middleboxes. While in conventional networking, data and control plane are implemented in the firmware of NEs, in SDN the control functionalities are decoupled from the NEs. In SDN the NEs does not perform any complex distributed algorithms, allowing to implement them with economic Commercial off-the-shelf (COTS) devices.

The data plane forwards, drops and modify packets according to policies imposed by the control plane. This is possible thanks to the definition of proper interfaces called the Southbound Interfaces (SBI)s. Through the SBIs, the data plane exposes visibility and control of its processing and forwarding capabilities to the control plane.

2) Control Plane (CP): SDN moves out the control plane from the firmware of NEs and implements it as software. The software-based control plane enables programmatic access to network resources and forwarding policies, and makes network administration agile and flexible. The control plane is composed by a logically centralized NOS or SDN controller. It is the "brain" of the SDN architecture that controls the communication between applications (business logics and intelligence) and network devices.

The NOS provides essential functionalities such as network topology storage, state information, notifications and device management, security, and shortest paths routing: these are the basic building blocks that most of the network applications need.

Additionally, the controller abstracts low-level details of forwarding plane, and provides a pool of APIs called Northbound interfaces (NBI)s or NB-APIs to the application plane. The control plane translates the requirements from SDN applications down to the data plane by distributing the configuration of forwarding elements using the SBIs.

3) Application Plane: the application plane is where applications that define the network forwarding rules reside (software). Such software programs consume the programmable platform provided by controllers' NBIs to obtain network state, and are able to change the network behavior by modifying the data plane forwarding rules.

In general, the applications obtain a simplified (abstracted) network view from the SDN controller (that hides and deal with the complexity of data plane), and based on that, implement the control-logic to make decisions that will be translated by the controller into commands to program the network. Applications may themselves expose another layer of abstracted network control.

Through the programmability offered by SDN, innovation is possible at the application plane. There is a wide range of applications already proposed and tested for different network domains that were categorized in [7] as: traffic engineering, mobility and wireless, measurement and monitoring, security and datacenter.

The application plane represents one of the key aspects of SDN since it offers the possibility to write code in diverse languages (that consume the network APIs) to perform business intelligence, optimization and provide new services. Such programmatic-networks interface was not available before SDN, and it is changing the whole networking ecosystem:

1) network operators have now the chance to avoid vendor lock-in by developing their own vendor-agnostic applications (or re-using open source ones);
2) third-party vendors have the possibility to enter the SDN market with applications for Orchestration, analytics, monitoring and optimization;
3) for big vendors (e.g. Cisco, Nokia, Juniper, Ciena, NEC, among others) the application plane represents another ground-field for generating value to their customers and differentiation towards competitors.

B. SDN-Architecture Interfaces

Two main classes of interfaces can be envisaged, the Southbound Interface (SBI), between the data and the control plane, and the Northbound Interface (NBI), between the application and the control plane.

1) Southbound Interfaces (SBI)s: also called Southbound Application Programming Interfaces (SB-API)s, allow NEs to exchange control and state information with the SDN controller. They provide programmatic control of all forwarding operations, device-capability advertisements, statistics reports and event notifications.

An SBI is defined by the forwarding elements that support it. Thus it is important to have open and standard SB-APIs to foster interoperability among different vendors and break the vendor lock-in that was the norm in legacy networks.

OpenFlow promoted by ONF [1] is the first open standard SDN SBI [3], and it is today's most accepted and used SDN SB-API, while other open SDN SBIs like OVSDB [26], ForCES [27], Protocol-Oblivious Forwarding (POF) [28] and OpenState [29] are less popular.

In order to allow backward compatibility and multi-technology control, some SDN controllers include legacy SBIs such as: PCEP [30], SNMP, BGP [31] and NETCONF [32]. We will discuss these protocols more in depth later on.

2) Northbound Interfaces (NBI)s: are the communication channels between control plane and applications. The NBIs or Northbound Application Programming Interfaces (NB-API)s represent the global management APIs offered by an SDN controller.

The NB-APIs allow the applications to exploit the abstract view of the network provided by the control plane and to collect statistics information, taking the requirements to enforce business logics and network intelligence from applications.

The NBIs facilitate innovation, automation and management of SDN networks, and are expected to be implemented in open, vendor-neutral and interoperable way. The most common NBIs provide RESTful JSON APIs as communication protocol. The ONF is working in the definition of standard NBIs and a common information model [33].

C. SDN-Architecture Abstractions

The separation of control and data planes is based on the introduction of a common abstraction model of the data plane accompanied with protocols or interfaces to enable the control and configuration. SDN abstractions are based on the data modeling of SDN infrastructure and services to extract simpler and common representations of the network using different representations that range from specific NEs to network-wide services. The abstractions reduce complexity and increase efficiency of automation from high-level applications and SDN controller. The network abstractions allow hiding all but the relevant characteristics from a network, device or service.

Network-wide and service abstractions allow SDN applications to consume SDN services that are abstracted from the underlying network technologies. Network-device abstractions enable the control plane to support heterogeneous equipment and technologies in the data plane.

The abstraction is the key to programmability and rapid innovation in SDN. The SDN abstractions should provide the right amount of information for the right amount of control. There are multiple layers of abstractions in the SDN-architecture, and they play an important role in addressing the success of this technology as each abstraction layer impose constraints and loss of information and control.

Two essential abstraction layers are always present in the SDN-architecture:

1) The Device and resource Abstraction Layer (DAL): the network devices provide the DAL to hide hardware-specific implementation details. It allows different vendors with heterogeneous hardware implementations to provide a common representation of the device towards the SDN controller. Thus, the DAL allows deploying standard interfaces between control plane and heterogeneous data plane. The DAL is commonly provided by an agent running on top or inside the data plane devices.

2) The Service Abstraction Layer (SAL): recursively, the control plane provides another abstraction layer to hide the complexity of distributing data-plane configuration and forwarding rules, as well as collecting the current state of the network (topology, links state, failures, and in some cases delay and jitter) through multiple SBIs and DALs. One of the main goals of SAL is to separate the NBIs from the protocol-specific SBIs. The SAL provides to internal-controller services and applications a standard set of packet-processing functions over a simplified graph-based view of the network.

The two main architectures for SAL are the Model-Driven Service Abstraction Layer (MD-SAL) and the API-Driven Service Abstraction Layer (AD-SAL)

Other abstraction functions have been proposed in SDN for: distributed updates, modular composition, virtualization, formal verification and even network programming languages [34].

D. OpenFlow (OF)

OpenFlow was proposed by the authors of [3] to decouple the control from the forwarding plane of Ethernet networks, and to allow full programming capabilities of the network devices. OpenFlow defines:

- a communication protocol between controller and switch;
- specification of components and functionalities of the switch, in order to allow an SDN controller to gather a common logical view from heterogeneous network devices.

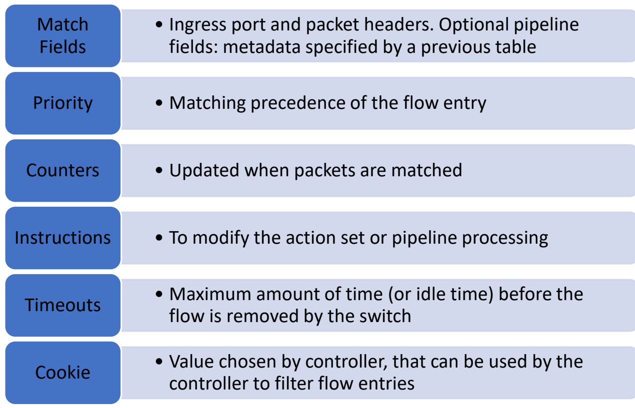

Fig. 5. Main components of a flow entry in a flow table as defined by OpenFlow v.1.4.0 [35].

The first version OpenFlow v1.0 was released by Stanford University [5]. From version 1.2 the OpenFlow specifications are released by the ONF, and it has become the de facto standard to implement SDN [2].

As the name suggests, OpenFlow provides a data plane abstraction based on flow tables. A flow is a set of packets that share a traffic relation identified by a combination of packet headers. In general, a flow table matches incoming packets to identify a specific traffic flow (packet lookup) and specifies the set of actions or rules to perform, for instance packet forwarding. A NE can have several flow tables to perform pipeline processing, and a group table that triggers multiple actions to group of flows. The OpenFlow node can create more than one (secured) OpenFlow channel to communicate with several controllers. Flow tables are populated by SDN controllers. The SDN controller can configure the forwarding rules by adding, updating and deleting the so called flow entries that constitute the flow tables. As can be seen in Fig. 5, a flow entry is identified by its match fields and priority. The flow entry contains the set of instructions to be executed upon a packet match. For a better understanding of OpenFlow we refer the reader to the OpenFlow switch specifications [1].

In the OpenFlow Switch Specification Version 1.4.0 (Wire Protocol 0x05) from October 2013 [35], a set of port properties were introduced to add support for optical ports. The properties included fields to configure and monitor the transmit and receive frequency of a laser, as well as its power. OpenFlow Version 1.4.0 established the first step towards the introduction of optical capabilities. However, those new fields are not sufficient to control the heterogeneous and complex nature of optical networks, as discussed in section III. More optical capabilities are expected in future OpenFlow switch specification release.

The new releases on OpenFlow will also have to match the fast evolution of OpenFlow switch technology. For instance, the use of new smart TCAM memory [36] will allow to deploy enriched routing functionality, speeding-up the interaction between controller and switches.

III. Transport Software-Defined Networking (T-SDN)

Telecommunication operators and transport service providers showed strong interests in the deployment of SDN in their optical transport networks. Providers can use SDN to provide automated and efficient connectivity in order to meet new service and application needs. However, the protocols and SDN-architecture extensions needed to control and manage the transport networks (called Transport SDN or T-SDN) represent a major challenge, due to the heterogeneous multi-domain (vendor, technology), multi-layer and some times even analog nature of transport networks.

A. *Formal definition of T-SDN*

Transport SDN (T-SDN) is an SDN-based architecture for the control and management of transport networks, that could involve multi-layer, multi-domain and multi-vendor scenarios. The Optical Internetworking Forum (OIF) defines Transport SDN (T-SDN) as a subset of SDN-architecture functions comprising the transport network relevant components [37].

Transport networks have features usually not present in computer networks where the SDN paradigm arose, like resilience, sophisticated architectures and heterogeneous technologies (OTN, OCh, MPLS, PBB, among others) and optical domain impairments, that need to be taken into account when applying the concepts introduced by SDN. In this sense, we define T-SDN as a subset of SDN-architecture that comprises extensions to abstractions, interfaces, protocols, and control plane elements to cope with transport networks peculiarities and to overcome the limitations of OpenFlow in this field.

The ONF Optical Transport Working Group (ONF-OTWG), renamed as the Open Transport Working Group, proposed what they called OpenFlow-enabled Transport SDN-architecture as described in [38], which is mainly based on OpenFlow. At the end of 2013 the ONF published the OpenFlow Switch Specification v1.4.0 [35] that introduced for the first time support for optical ports. Nevertheless, the work is still in progress in ONF-OTWG to define stable and standard NBI and SBI specification for SDN/OpenFlow-based T-SDN including: extensions for management and control of optical transport [8], [38]–[40], and wireless transport [41]. This work focuses on the optical transport network, rather than in the wireless transport, which is an area of great interest with the advent of 5G mobile networks, the Internet of things and mobile cloud era.

The following subsection briefly describe some transport network technologies to help the reader to better understand the challenges related to the extensions of SDN principles to optical transport networks presented in subsection III-C.

B. *Enabling transport network technologies*

The transport networks involve many different technologies across multiple layers:

- *layer 3 and layer 2*: IP, Ethernet, MPLS [42] and MPLS-TP [43] that provides statistical multiplexing at packet level (L2).

- *Layer 1 and layer 0*: at layer 1 OTN [44] that supports ODUk electrical Time Division Multiplexing (TDM), and at layer 0 optical Wavelength Division Multiplexing (WDM) and the new flexible grid technologies.

Traditionally, routing, signaling and protection functionalities were placed at the IP layer, and the optical layer provided static connectivity for the layer 3 and layer 3 devices. However, flexibility and dynamic capabilities of state-of-the-art optical devices allow us to avoid the hop-by-hop IP processing by efficiently and dynamically adapting the optical connections to the traffic demands and by-passing the IP layer whenever it is possible. Optical by-pass allow us to avoid the energy consumption, costs, delays and complexity of hop-by-hop IP processing. We now describe some of the optical network technologies that enable a flexible and dynamic optical layer.

- Transparent optical networks: composed by optical devices that are capable of switching signals in the optical domain such as Reconfigurable Optical Add-drop Multiplexers (ROADM), Wavelength cross-connects (WXC) and Photonic cross-connects (PXC).
- Elastic Optical Network (EON): consists of Bandwidth Variable Optical cross-connect (BV-OXC)s and Bandwidth Variable optical Transponder (BVT)s. In the EONs, the previously fixed WDM grid becomes flexible (flexi-grid) by introducing spectral and modulation format flexibility, allowing lightpaths to meet the variable requirements of services and applications, as described in G.694.1 [45]. In flexi-grid the optical channels are identified by port, central frequency, frequency slot bandwidth and type of signal [46], [47].
- Generalized Multi-Protocol Label Switching (GMPLS): GMPLS is a control plane technology (RFC 3945 [48]) proposed by the Internet Engineering Task Force (IETF) to manage heterogeneous switching modes including packets, time slots, wavelengths and fibers. GMPLS is a distributed control plane based on a pool of protocols standardized by IETF (e.g. OSPF, IS-IS, RSVP-TE [2]) and it is the most used control plane in current optical transport networks. The Path Computation Element (PCE) [49] is playing an important role in the interoperability between GMPLS and SDN, as explained in section IX-C2.
- Network Management System (NMS) and Element Management System (EMS): the optical network equipment are typically controlled and managed through a centralized NMS/EMS. NMS/EMS provides a highly reliable optical resource allocation (lightpath provisioning) in a manual and semi-static fashion. The NMS computes optical reach, configures the devices, and performs monitoring to ensure proper signals quality. The NMS provides a Northbound interface to the operations support system (OSS) (or applications) usually based on the Simple Network Management Protocol (SNMP), Common Object Request Broker Architecture (CORBA) or Extensible Markup Language (XML) [50].

[2]Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS), Resource Reservation Protocol - Traffic Engineering (RSVP-TE)

C. The challenges of Transport SDN

SDN was specifically defined for packet-switched networks at layers 3 and 2 [3]. Today, standardization for OpenFlow-based SDN is strongly supported by ONF [1], and there is a growing market of commercial OpenFlow-based SDN solutions [51].

On the other hand, T-SDN involves support of layers 3 and 2 and additional support for circuit-switched networks at layers 1 (SONET/SDH & OTN[3]) and 0 (optical), which entails significant challenges when compared with SDN solutions that focus only on layers 3 and 2. Therefore, the standardization process of T-SDN has been slower and remains an open issue. Nonetheless, there are early-stage vendor solutions, mainly based on reuse of legacy technologies as presented in section XI. Table I summarizes the characteristics of T-SDN and the challenges imposed by the optical infrastructure.

SDN programmability depends on the definition of common data plane abstractions and standard Southbound and Northbound interfaces [3]. At layers 3 and 2, accomplishing such features was relatively easy: indeed, in these layers the data plane abstractions can be defined upon well standardized packet headers. The exploitation of this advantage was the basis to deploy a common Southbound interface like OpenFlow. Consequently, for packet-oriented networking manufacturers, it was simple to produce OpenFlow-enabled devices, that can be supported by commodity hardware, and a simple OpenFlow agent. Finally, OpenFlow agents could benefit from the well-consolidated techniques for packet classification based on standard layer 2 and layer 3 packet-fields.

At layer 1, composed mainly by OTN and its predecessor SONET/SDH technologies, it is as-well relatively easy to embrace SDN support [52]. Layer 1 OTN involves switching time slots in the electrical domain. Thus, all the signals are converted to the electrical domain, undergoing optical-to-electrical (OE) and electrical-to-optical (EO) conversions, on a hop-by-hop basis. OTN layer is well standardized by OIF, International Telecommunication Union (ITU) and IETF, with standard compliant vendors' solutions.

The optical Layer 0, composed by fixed and flexi-grid (D)WDM technologies is the major challenge. We may say that optical switching, that involves configuring OXCs at wavelengths and fibers, as in OTN is relatively easy. The optical switching capability allow us to perform optical bypass. Thus, all optical paths i.e. lightpaths, are established to avoid OE-EO conversions on a hop-by-hop basis.

In the following we list some of the reasons that make the optical layer more complex than the layers above.

- The optical layer is transmission dependent. Differently from electrical infrastructure, in optical networks transmission limitations translates into routing constraints for the logical layer of the lightpaths, i.e., transmission reach and wavelength continuity constraints. Therefore, at the optical layer not all the paths are feasible.
  – The quality of signals in the optical layer is affected by photonic impairments such as chromatic and po-

[3]Synchronous Optical Network (SONET)/ Synchronous Digital Hierarchy (SDH) & OTN

TABLE I
TRANSPORT SDN CHARACTERISTICS

	Layer 3 and Layer 2	**Layer 1 (OTN)**	**Layer 0 (optical)**
Traffic model	Electronic packet-switching	Electronic TDM circuit-switching	Optical WDM circuit-switching
Data Plane operations	Packet header lookup, and packet operations (forwarding, encapsulation, pipeline processing, statistics collection)	Operations over time slots, signal transmission, detection and switching. Performance monitoring	Fiber switching, wavelength conversion, signal transmission (modulation format), detection, amplification and regeneration on fixed and flexi-grid technologies. Performance monitoring
Complexity	Low complexity: digital operations, based on packets headers	Relatively low complexity: digital operations, based on time slots	High complexity: analogical operations, sensitive to physical layer constraints
Data Plane implementation	Homogeneous: based on standard protocols & specifications, vendor agnostic. Suitable for COTS devices	Homogeneous: based on standard protocols & specifications	Heterogeneous: vendor-specific features & configuration, administratively independent vendor islands
Data Plane abstraction	Easy-to-define standard abstractions	Relatively easy-to-define standard abstractions	Hard-to-define low-level standard abstractions
Southbound interface	Standardized SBI (e.g., OpenFlow)	Non standard SBI, reuse of GMPLS and vendor-specific interfaces, multiple extensions proposed for OpenFlow (OpenFlow+)	
Control Plane	Standard OpenFlow-based control	Vendor-specific interface control, SDN/GMPLS and ASON, OpenFlow-based control	
Maturity	Standard commercial solutions and rollouts, based on OpenFlow	Non standard commercial solutions. Some OpenFlow standardization covered	Non standard commercial solutions

larization mode dispersions, fiber nonlinearities and Amplified Spontaneous Emission (ASE) noise [53].
  – The optical systems are characterized by vendor-specific technologies and features like: switching constraints, power equalization, recovery mechanisms, and elastic transponder capabilities. For instance, among switching capabilities there is colored/colorless, directed/directionless, and blocking/contentionless [54]–[57].
- The optical networks continue to evolve, and present a gap between standardization and vendor implementations [50].

To cope with such complexity, optical network solutions rely on a vendor-specific management system (e.g., NMS and EMS) that performs optical resource allocation, lightpath's reach computation, devices configuration, and quality of signals monitoring. As depicted in Fig. 6, current optical networks implement the GMPLS protocol suite as distributed control plane for dynamic path setup. The NMS together with GMPLS provide a "big switch" abstraction that hides the optical complexity and topology to the OSS and applications.

Historically, the optical network equipment providers have increased their solutions' competitive advantages by: introducing proprietary technologies with new features, and improving their management systems. This behavior led to heterogeneous data planes, with interoperability issues among diverse vendors' equipment. In consequence, the transport network of service providers is composed by administratively isolated vendor islands, each controlled by a centralized NMS. This heterogeneous scenario represents a big challenge to define common abstractions for T-SDN, and to gather detailed visibility and control over the multi-layer, multi-vendor, and multi-

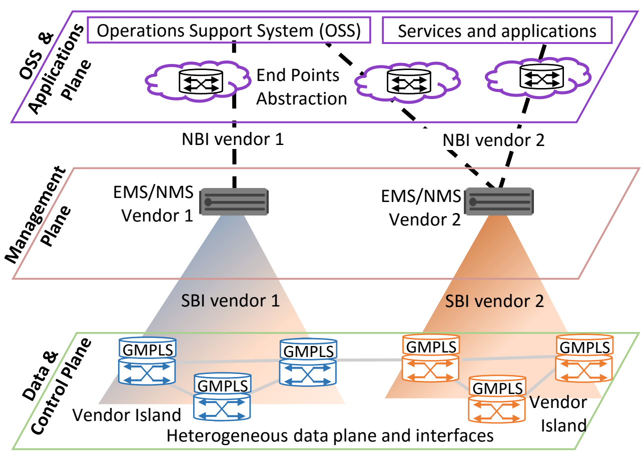

Fig. 6. Legacy transport network architecture. Notice that data and control plane are like in Fig. 1, but the management features are centralized in vendor specific EMS/NMS. SBI and NBI are both vendor specific.

domain optical transport networks.

D. *T-SDN classification*

In Fig. 7 we present a classification of T-SDN solutions by their control plane architecture in: monolithic T-SDN (SDON), hierarchical T-SDN (HT-SDN) and flat/mesh T-SDN (FT-SDN).

1) Monolithic architecture (SDON): the SDON was proposed by research efforts, and was the first step in the evolution of transport SDN. In literature we can find the term SDON (Software Defined Optical Networking) that refers to:

- single SDN controller over a single optical domain, based on extensions that enable SDN at the optical layer 0

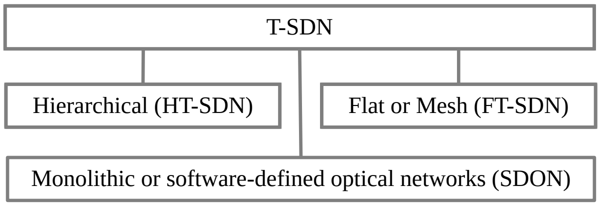

Fig. 7. Classification of T-SDN solutions based on control plane architecture: monolithic (SDON), hierarchical T-SDN (HT-SDN) and flat or mesh T-SDN (FT-SDN).

comprising software-defined transceivers, ROADMs and OXCs along with extensions to SDN control plane and Southbound interfaces (e.g., OpenFlow) [14][58];
- single SDN controller over a multi-layer network that provides Unified Control Plane (UCP) of IP and Optical layers. With an SDN-enabled optical layer, SDON is able to exploit the benefits of a UCP for IP and optical layers [58].

2) Hierarchical architecture (HT-SDN): the standardization bodies involved in SDN and transport networks (mainly ONF and OIF) agreed on a hierarchical architecture of controllers for transport SDN (HT-SDN) [38] [59]. The hierarchical architecture better suites the multi-domain nature of transport networks, where multiple domain controllers (SDN-based and legacy-based) are orchestrated either by a parent controller or by the transport network orchestrator. An SDON controller becomes a domain controller in the HT-SDN architecture.

3) Flat or mesh architecture (FT-SDN): a flat control plane architecture (FT-SDN) is composed by multiple domain controllers with a peer-to-peer coordination. Therefore, in opposite to HT-SDN that uses Northbound and Southbound interfaces for inter-controller communication, FT-SDN uses East/West interfaces for the peer-to-peer interaction between SDN controllers.

Flat control plane architectures were not the focus of T-SDN early-stage development. The standardization of the east/west interfaces is far behind the achievements in Northbound and Southbound interfaces. For instance an inter domain protocol for the east/west interface between two EON domains was proposed in [60]. Peer-to-peer relations are expected to gain more interest for:

- control plane clustering, which is supported by the latest version of Opendaylight controller (ODL) and by ONOS controllers, but is not well studied in literature;
- inter provider coordination, where flat architectures are expected to be created among service providers [37].

IV. Historical Brief Summary of T-SDN

In this section we briefly describe the evolutionary path of research activity to enable SDN in transport networks that mainly focuses on monolithic (SDON) and hierarchical (HT-SDN) architectures. Another important component that we briefly describe in this section are the standardization efforts and open source control plane frameworks.

To provide a big picture of T-SDN activities, tables II, III, IV, V, and VI summarize the timeline and proposed our classification of the evolution in T-SDN research solutions, and standardization activities. Table IV provides the timeline and classification of research contributions on virtualization, as well as Virtual Network Embedding (VNE) algorithms in T-SDN scenario and T-SDN with network function virtualization solutions (T-SDN-NFV). Table V summarizes the main efforts on standardization of T-SDN and VI specifically presents the standardization of data models for transport-networks APIs.

A. Monolithic architecture: SDON

The research activity in T-SDN control planes begins in 2009 at Stanford University with the so called Packet and Circuit Convergence (PAC.C) extensions to OpenFlow [52], [61]. The first task was to theoretically and experimentally prove the viability and usefulness of migrating to SDN. Section V-A1 describes PAC.C, a solution that aims at fully centralized architecture with a single SDN controller based on native support of OpenFlow in network elements, and OpenFlow extensions to manage circuit-flows and packet-flows. PAC.C leads to convergence of packet and circuit domains into a flow-switched network with a UCP that benefits from the visibility and control over IP and optical domains.

In 2009, PAC.C established a baseline approach for transport SDN. Up to 2014, most of the research efforts shared a common target: to enable the optical data plane to be directly controlled by an SDN/OpenFlow controller [15], [58].

We classify solutions characterized by the use of a single SDN controller to manage multi-layer transport networks as SDON (section V). Based on the type of SBI, we further classify the SDON solutions into:

- Single southbound interface
  - Extended OpenFlow (OF+) with native support (section V-A1).
  - OF+ with agent-based support (section V-A2).
  - NETCONF with agent-based support (section V-A3).
- Hybrid SDN/GMPLS interfaces
  - Legacy interfaces towards GMPLS CP (section V-B1).
  - OF+ towards GMPLS CP (section V-B2).
  - Legacy interfaces (RSVP-TE and OSPF) and agent-based OF+ towards GMPLS CP (section V-B3).

In table II we propose a timeline of research efforts on SDON that are classified by the SBI used towards the optical domain.

B. Hierarchical architecture: HT-SDN

After successful SDON proof-of-concepts, the focus shifted towards hierarchical controller architectures that we call in this work HT-SDN (section VI).

The main rationale behind this approach is that a complex system with heterogeneous domains and equipment provided by multiple vendors can be better controlled by a modular and hierarchical control plane.

Hierarchical architectures increase scalability and allow a better integration of the heterogeneous domain/layer environment of transport networks by placing specialized controllers for each domain and layer on a hierarchical architecture. A

TABLE II
TIMELINE OF RESEARCH EFFORTS ON MONOLITHIC CONTROL PLANE ARCHITECTURE SOLUTIONS (SDON), CLASSIFIED BY THE SOUTHBOUND INTERFACE USED TOWARDS THE OPTICAL DOMAIN

Year	Single Southbound interface (SBI): Extended OpenFlow (OF+)	Single Southbound interface (SBI): NETCONF	Hybrid SDN and GMPLS (SDN/GMPLS) SBIs: Legacy interface towards GMPLS CP	Hybrid SDN and GMPLS (SDN/GMPLS) SBIs: OF+ towards GMPLS control plane (CP)	Hybrid SDN and GMPLS (SDN/GMPLS) SBIs: GMPLS CP interfaces and OF+ towards GMPLS CP
2009	[52], [61]				
2010	[62]–[64]				
2011	[65]–[68]		[69], [70]		
2012	[71]–[74]		[75]	[75]	[72]
2013	[76]–[82]	[83], [84]			[76]
2014	[85]	[86], [87]		[85]	
2015	[88]	[89], [90]		[88]	

TABLE III
TIMELINE OF RESEARCH ON HIERARCHICAL CONTROL PLANE ARCHITECTURE SOLUTIONS (HT-SDN), CLASSIFIED BY THE SBI USED TOWARDS THE OPTICAL DOMAINS AND THE NBI USED AMONG THE HIERARCHY OF CONTROLLERS

Year	Single interface-based hierarchy: Extended OpenFlow (OF+)	Single interface-based hierarchy: Hierarchy of Stateful-Hierarchical PCEs (SH-PCE)	Hybrid SDN/GMPLS Hierarchy: SDN (ODL) controller over PCE-based controller	Hybrid SDN/GMPLS Hierarchy: SDN (ABNO) controller/orchestrator over heterogeneous controllers	Hybrid SDN/GMPLS Hierarchy: Control Orchestration Protocol (COP)-based Orchestrator over heterogeneous controllers
2013	[91]				
2014			[92], [93]	[94]–[96]	
2015		[97]		[98]–[100]	[101]
2016					[102]

TABLE IV
TIMELINE OF RESEARCH ON VIRTUALIZATION SERVICE OVER T-SDN, WITH A CLASSIFICATION OF T-SDN VIRTUALIZATION ARCHITECTURES

Year	Classification of T-SDN virtualization architecture: Distributed (domain controllers)	Classification of T-SDN virtualization architecture: Centralized (parent controller or orchestrator)	Classification of T-SDN virtualization architecture: Abstracted (application on top of orchestrator)	T-SDN virtualization-related efforts: Virtual Network Embedding (VNE) Algorithms	T-SDN virtualization-related efforts: T-SDN and Network Function Virtualization (NFV)
2012	[103]				
2013		[83]	[59]	[104], [105]	
2014	[106]–[108]	[38]			
2015			[109]	[110]	[111]
2016			[112]	[113]	[114]–[116]

TABLE V
TIMELINE OF STANDARDIZATION ON T-SDN ARCHITECTURE, CONTROL PLANE AND SOUTHBOUND INTERFACES

Year	Interfaces: ONF OpenFlow Optical Extensions	Interfaces: IETF Interfaces	HT-SDN: Requirements	HT-SDN: Reference Architecture	HT-SDN: Virtualization and abstraction	HT-SDN: OIF-ONF Global Demonstration
2013	[35]		[59]			
2014			[40]	[38]		[8]
2015	[39]	[30]–[32]		[37], [49], [117]	[118], [119]	
2016		[120]				[121]

TABLE VI
TIMELINE OF STANDARDIZATION ON DATA MODELS AND APIS FOR TRANSPORT NETWORKS

Year	Network-wide Data Models for Northbound Transport APIs: Common Information Model ITU-T G.7711 [33], [122]: ONF	Network-wide Data Models for Northbound Transport APIs: Common Information Model ITU-T G.7711 [33], [122]: IETF	Network-wide Data Models for Northbound Transport APIs: Information model based on ITU-T ASON: OIF	Optical Network-element Data Models for Southbound Transport APIs based on YANG: OPEN-ROADM	Optical Network-element Data Models for Southbound Transport APIs based on YANG: OpenConfig
2015			[37]		
2016	[123]	[124]–[131]		[132], [133]	[134], [135]

domain refers to an autonomous network area defined by a specific: layer, vendor, data plane or control plane technology. As shown by Fig. 8, the optical layer can be composed of multiple domains given by vendor islands, heterogeneous data plane (e.g., fixed-grid, flexi-grid, Optical Packet Switching (OPS), etc.) and control plane (SDN and GMPLS) technologies.

In hierarchical T-SDN-architecture, domain controllers are in charge of intra-domain path computation and management of the optical domain complexity. Each domain controller provides to the parent controller or orchestrator an abstract representation of its domain. The parent controller or a transport network orchestrator application coordinates the domain controllers, is in charge of inter-domain path computation and end-to-end service creation.

Table III presents a timeline and the proposed classification of HT-SDN based on the interface used to interact among the hierarchy of controllers:

- hierarchy of OF+ controllers (section VI-A);
- hierarchy of Stateful-Hierarchical PCE (SH-PCE)s (section VI-B);
- hybrid SDN/GMPLS hierarchy:
  - SDN parent controller over PCE-based domain controller (section VI-C1);
  - SDN controller/orchestrator over heterogeneous domain controllers (section VI-C2);
  - Control Orchestration Protocol (COP)-based Orchestrator over heterogeneous domain controllers (section VI-C3).

C. Standardization

As presented in Table V, it was only after 2013 that Standards Developing Organization started to release documents related to T-SDN. In the following we briefly describe the main SDOs involved in T-SDN:

- Open Networking Foundation (ONF) supports a fully OpenFlow-based T-SDN and has already started to introduce basic optical features into the OpenFlow specifications (see section IX-A1). ONF-TAPI is the main Standards Developing Organization (SDO) working on development of standard APIs for the Northbound interface of T-SDN controllers (see section IX-E2);
- Optical Internetworking Forum (OIF) aims at establishing well defined Northbound interfaces (controllers APIs), to assure interoperability in multi-domain, multi-layer, multi-technology and multi-vendor environments (see section IX-A2).
- Internet Engineering Task Force (IETF) has also contributed with multiple protocols and technologies that can be reused to achieve T-SDN, e.g. GMPLS, Path Computation Element Protocol (PCEP) and NETCONF (see section IX-A3). Another important contribution from IETF is the multi-domain PCE-based SDN orchestration Architecture for Application-Based Network Operations (ABNO), described in section IX-D. YANG data modeling language RFC 6020 [136], is becoming an essential component for standardizing controllers data models, as well as producing standard Northbound and Southbound interfaces (see section IX-E3).

Open Application Programming Interfaces (API)s to control and manage transport networks are a major topic of interest for network providers in order to foster programmability to lower CapEx and OpEx of their multi-layer and multi-vendor transport infrastructure. The standardization efforts on Transport APIs (TAPI)s, can be classified in two types:

- Network-wide Data Models for Northbound APIs: provided by the control plane and standardized by IETF, ONF and OIF (sections IX-E1, IX-E2 and IX-E3);
- Network-element Data Models for Southbound APIs: provided directly by the transport network devices, and standardized by the Open ROADM project and OpenConfig Working Group (sections IX-F1 and IX-F2).

Though we do not provide a timeline table for vendor solutions, multiple optical networking manufacturers, ahead of the standardization efforts, came up with vendor-specific implementations of controllers, orchestrators, YANG data models, protocols and interfaces for T-SDN (see section XI).

D. Open Source Networking Frameworks

Another important component in the evolution of T-SDN are the open source networking frameworks. In section X we focus on the main open source projects that support real T-SDN solutions. The two main open source carrier-grade controllers are:

- OpenDaylight: that is becoming a common platform for vendors' solutions, and pioneering demonstrations (section X-A1);
- Open Network Operating System (ONOS) a younger player that specifically focuses on Service Providers, and is taking off rapidly in T-SDN market (section X-A2).

Such controllers, together with ABNO specifications (from IETF) are filling the gap of slow standardization process on T-SDN technologies. Other open source networking efforts that are also part of the software-defined transformation are:

- Open-Orchestrator (Open-O) [137]: the first open-source end-to-end service orchestrator project to support integration of both NFV and SDN;
- Open Sourced Enhanced Control, Orchestration, Management and Policy (ECOMP) architecture [138]. Its goal is to support full automation and incrementally reduce dependencies on the Legacy OSS;
- Open Network Automation Platform (ONAP) Project: Merger of Open Source ECOMP [138] and OPEN-O [137];
- Open Platform for NFV (OPNFV): development and evolution of NFV components across various open source ecosystems [139];
- Open source Platform for Network Data Analytics (PNDA) a big data analytics platform for networks and services [140].

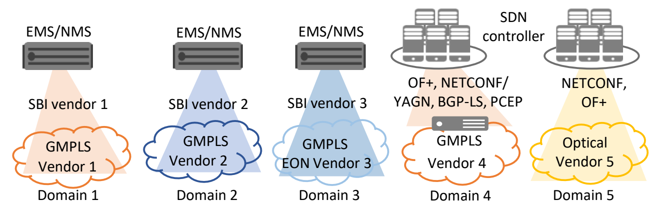

Fig. 8. Example of optical domains in a transport network.

V. Research Efforts on Monolithic Control Plane Architectures (SDON)

This section presents an overview of the main research efforts towards SDON, that is the first approach to be proposed for transport SDN. Fig. 9 presents the classification of SDON solutions based on the SBI used towards the optical domain(s).

A. SDON Architectures with a Single SBI towards Optical Domains

*1) **Extended OpenFlow (OF+) with native support model**:* the authors of PAC.C evidenced that the OpenFlow data plane abstraction, based on packet flow as atomic traffic element, can be extended to support circuit flows, by adapting the cross-connect tables of transport switches to circuit-flow tables. In OpenFlow, the flow table is used to perform packet lookup for classification and forwarding actions. For layer 1 and 0 optical infrastructure, a circuit flow-table is more appropriated. The PAC.C circuit-flow table is used to configure the switching matrix of layer 1 and 0 optical device.

The OpenFlow Circuit Switched Addendum v.03 [62] detailed a model to deploy circuit-flow tables into circuit-switching network elements at layer 1 and 0. In order to enable the flow abstraction, a circuit-switching flow table operating at layer 1 or layer 0 must be implemented into the optical equipment (separated from the packet flow table). Fig. 10 shows the circuit-flow table proposed in [62], where the circuit flows are defined by four fields per input and output ports, namely: port number, wavelength, virtual port associated with the Virtual Concatenation Group (VCG) and starting time slot for SONET/SDH. Interconnection between packet and circuit domains is achieved by mapping packets to circuit-flows using VCGs.

Accordingly, OpenFlow protocol extensions (v 1.0) were proposed to support the circuit-flow table depicted in Fig. 10 [62]. These extensions allow for wider and flexible definition of flow identifiers, which are defined as combination of headers from Layer 2 to 4 (fields in the packet header) and circuit identifiers from Layer 1 and 0 (position in time, spectrum and space). PAC.C extensions led to UCP for the management of OpenFlow-enabled packet, circuit and hybrid switches. An SDN controller can dynamically update the circuit-switching flow tables in the cross-connects, increasing adaptability of transport networks to traffic pattern variations or failures.

A PAC.C proof of concept for convergence of Ethernet/TDM was presented in [64] using SDN and OpenFlow as unifying architecture and abstraction, respectively. The

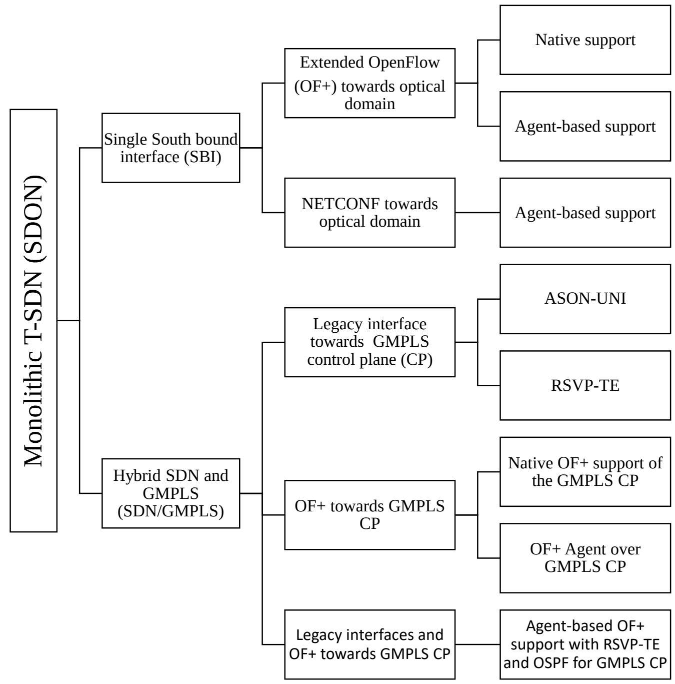

Fig. 9. Classification of SDON solutions based on the SBI used towards the optical domain(s).

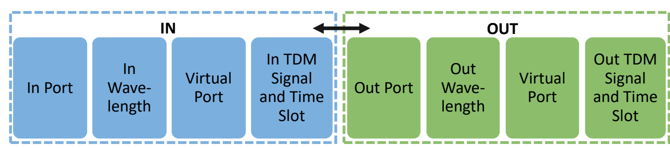

Fig. 10. Circuit flow table used to define the state of the switching matrix. It was proposed in the OpenFlow Circuit Switched Addendum v.03 [62].

demonstration was based on three Ciena switches that were modified to natively support OpenFlow for packets and circuits. Using an extended NOX controller [4], the authors presented an application to monitor the network performance based on the switch port and flow statistics, and to react upon network congestion by fully managing L1 (TDM) and L2 (Ethernet) flows on-demand. In [63] the authors added lightpath configuration capabilities to PAC.C.

Das et al. [65], [66] exploited the OpenFlow flow granularity to implement dynamic application-based traffic aggregation to create bundles of video, voice and HyperText Transfer Protocol (HTTP) traffic. Using such application-based bundles and the dynamic circuit-switching capabilities, application-aware traffic engineering and failure recovery at the circuit-switched network were demonstrated.

A hybrid packet-circuit switch architecture (see Fig. 11) was proposed as a replacement for backbone routers in order to achieve fully meshed IP core that can exploit the UCP of the converged packet-and-circuit architecture [141]. For a typical backbone operator use case, the hybrid nodes potentially allow up to 60% of cost savings [141].

PAC.C focused on achieving efficient UCP with a native integration of OpenFlow into optical NEs, that wager for a disruptive model from current network elements of transport

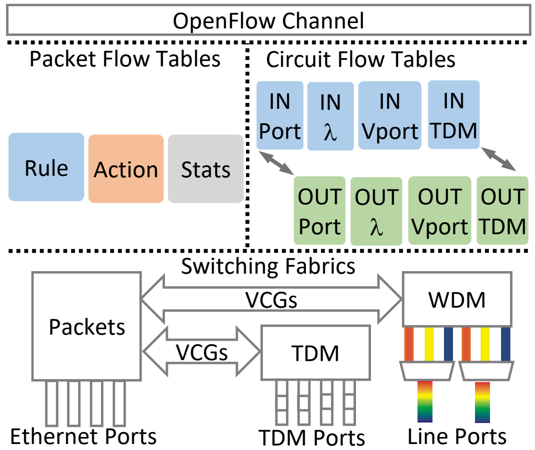

Fig. 11. Hybrid packet-circuit switch architecture. VCG: Virtual Concatenation Groups.

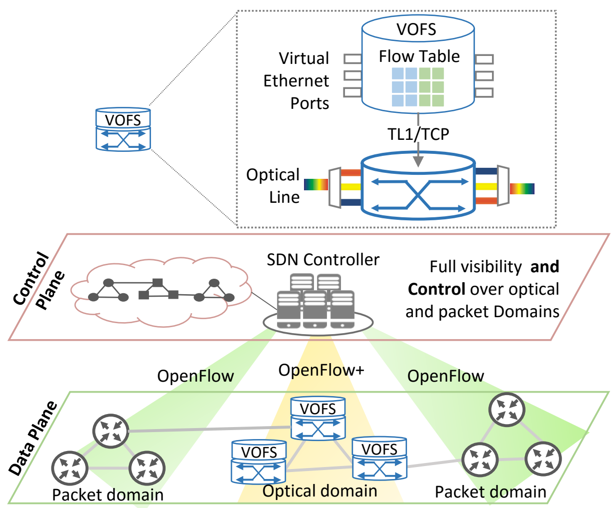

Fig. 12. Architecture of the first pure extended OpenFlow (OF+) agent-based model that provides full visibility and control of optical domains [67]. The agent (virtual OpenFlow switch; VOFS) interacts with the optical device management interface using TL1 and abstracts the optical node. In this work we call OpenFlow+ (or OF+) any extension of OpenFlow in support of transport networks.

technologies. In PAC.C the optical network features such as switching constraints, power equalization and optical impairments, were not considered.

*2) **Extended OpenFlow (OF+) with agent-based support models**:* in order to avoid disruptive native-OpenFlow support as required by PAC.C (see section V-A1), the OpenFlow agent-based model was introduced in [67].

The OpenFlow agent bridges the lack of OpenFlow support at hardware level in legacy Network Equipment (NE). The idea is similar to the architecture presented in Fig. 16 but without the GMPLS control plane. The OpenFlow agent converts legacy NEs into OpenFlow capable devices, and allows a smooth transition path towards SDON. In the following we describe the first works that proved the viability of SDN/OpenFlow for legacy transport networks by the adoption of OpenFlow agents.

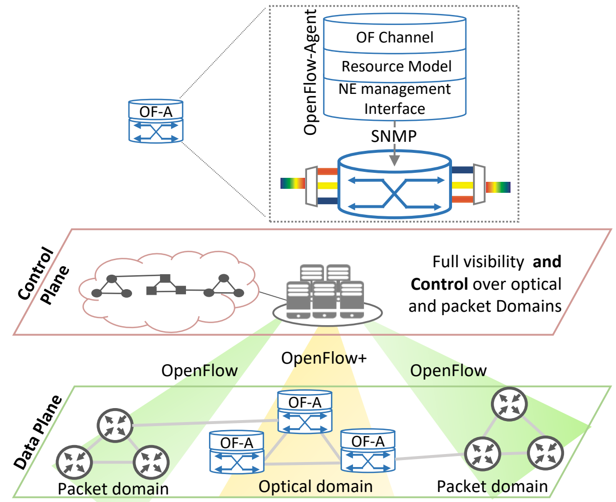

Fig. 13. Pure extended OpenFlow (OF+) agent-based model. The OpenFlow agent (OF-A) enables the control of legacy devices and directly interacts with the NEs through SNMP.

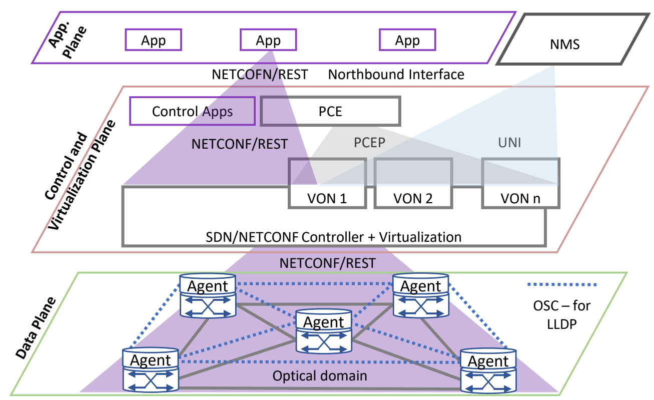

Fig. 14. The NETCONF-based controller [83], merges the virtualization functionalities into the controller.

- *The ML-MG OpenFlow agent:* ML-MG (Multi-Layer and Multi-Granularity capabilities for transparent networks) is the result of collaborative works done by KDDI R&D Laboratories (Japan) and Beijing University of Posts and Telecommunications (China), later joined by Centre Tecnològic Telecomunicacions Catalunya (CTTC) research institution (Spain) and University of California-Davis (USA). ML-MG focused on development and testing of OpenFlow-based control plane for transparent optical networks [67].
  In [67] ML-MG proposed the first OpenFlow agent for optical NEs (depicted in Fig. 12), as a virtual switch composed by n virtual Ethernet interfaces associated with n physical ports, and a circuit-flow table (similar to PAC.C) to abstract the NE. This agent was later called the Virtual OpenFlow Switch (VOFS) in [71]. The VOFS provides virtualized view and OpenFlow interface to install rules in the circuit-flow tables of an optical NE. The agent translates installed rules into standard Transaction Language 1 (TL1) commands to configure the cross connection of optical devices. Authors of [68] accomplished the first

proof-of-concept of dynamic lightpath allocation over a transparent optical network (with wavelength continuity constraint) controlled by SDN/OpenFlow. Four OXCs with VOFSs on top provided transport services to interconnect two IP domains.

- *The OFELIA OpenFlow agent (OF-A)*: the European project OFELIA (OpenFlow in Europe Linking Infrastructure and Applications), was a main contributor into the evolution of transport SDN [142]. OFELIA proposed an OpenFlow agent (OF-A) composed by three vertical modules as depicted at the top of Fig. 13 [72]. The first module is responsible for the establishment of a secure channel with the controller. The second module creates a generic abstraction of the optical data plane composed by: circuit-flow tables, multi-domain mapping information (e.g., mapping packet to circuit) and the vendor-specific NE parameters (switching and power constraints, recovery mechanisms and optical layer impairments). The third module translates and configure the abstracted rules in the data plane via NE's management interface, e.g., SNMP or vendor-specific APIs.
- *First SDON controlling commercial ROADMS (OFELIA project)*: using the OF-A of Fig. 13, the authors of [72] implemented the first SDON architecture capable of controlling ROADMS (ADVA FSP3000). The network demonstrated in [72], was composed by three ROADMs in a ring topology that interconnects two packet-oriented SDN domains. An SDN controller based on Open Virtual Switch (OVS) and NOX, takes into account switching constraints and optical equalization for path computation with Quality of Transmission (QoT) assurance. [72] proposed a purely-extended OpenFlow model (Fig. 13): it is based on OpenFlow agents and extended OpenFlow (OpenFlow+). Switching constraints and power equalization messages were included in such extended OpenFlow. This allows the controller to manage cross-connection flow tables of ROADMs by exchanging *CFLOW_MOD* (circuit Flow_Mod) messages with the OF-A, thus controlling computation, establishment and release of the lightpaths.
- *Multi-layer and Multi-granularity capabilities for transparent networks (ML-MG group)*: extensions to the architecture of Fig. 12 allowed to achieve a UCP over multiple domains comprising packet switching, Optical Burst Switching (OBS) [143], and Optical Circuit Switching (OCS) [71].
  The first SDON field-trial connecting Japan, China and Spain demonstrated dynamic establishment, tear down and restoration of end-to-end paths across multiple layers and granularities [71], [144]. Failure-alarm monitoring for transponder control was also addressed by translating TL1 messages of optical NEs into OpenFlow *Packet In* messages. Transponder control information is specified through a *flow entry-transponder control information* translation table. Such table needs to be added into the SDN controller and into each OF-enabled transponder. Thus, upon link failure, the SDN controller obtains its description and is able to compute and establish a restoration path.
- *EON - Flexible grid WDM capabilities (ML-MG group)*: SDN/OpenFlow-based EON was also addressed with OpenFlow extensions and a control plane named OpenSlice [73], [81]. In each BVT an OpenSlice converter works as the OpenFlow agent with EON capabilities. The OpenSlice converter translates path set-up requests (PSR) into *Packet in* messages, and sends them to the SDN controller. It also translates *Slice Mod* messages sent from the controller, into vendor-specific commands (e.g., TL1) to establish: transceiver's Central Frequency (CF), slot width and modulation format. In the BV-OXC, an OpenSlice module maintains the cross-connection table with flexi-grid configuration parameters (In/Out Port, CF, slot width and modulation format) similar to the VOFS of Fig. 12. Such table is managed by the controller through *Slice Mod* messages.
  OpenSlice and GMPLS control planes were compared using a simple experimental test-bed in [81]. The results showed that for routes with more than 3 hops SDN achieved fastest path provisioning than GMPLS, thanks to its centralized nature. The authors claimed that OpenSlice reduces the complexity of GMPLS for dynamic path provisioning and IP traffic offloading to the EON domain.
  Authors of [78] extended the *Flow Mod* message to provide Bit Error Rate (BER) information to the controller, so the controller can perform computation of path, wavelength, symbol rate and modulation format. Upon link failure or signal degradation, an alarm (*Packet in* message) can trigger the controller to compute the reconfiguration of a working or protection path with proper symbol rate and modulation format.
  Some control-plane functionalities and algorithms for effectively avoiding spectrum fragmentation in EONs were validated using simulations and experimental set-up in [145].
- *Fixed and Flexible grid WDM domains (OFELIA project)*: Channegowda et al. [76] demonstrated, for the first time, extensions for a multi-domain packet network over a fixed and flexi-grid optical network. Such extensions are based on a previous work presented in [72]. To support fixed grid OXCs (WDM-OXCs), BV-OXCs and BVTs, the following OpenFlow messages were extended: *Switch_Feature*, *CFlow_Mod* and *CPort_Status*. Moreover, the authors deployed intra-domain and inter-domain flow tables in a NOX-based controller, and defined multi-domain mapping rules to handle multi-domain constraints.
  Using the former extensions, an application was presented in [76] for virtual slicing across multi-technology (fixed and flexi-grid DWDM, and packet domains) and geographical domains.
  To further exploit SDN capabilities, a cloud use case with storage and Virtual Machine (VM) migration was demoed in [146]. Such application assigns fixed-grid flows to narrow bandwidth VM migration services and flexi-grid channels to bandwidth-hungry storage migration services.
- *Path-computation offloading using PCE (ML-MG and*

OFELIA): the scalability of SDN paradigm represents a major concern for transport SDN, due to the complexity of multi-domain optical transport networks. As a solution, [74] proposed to offload the Impairment-Aware Routing (IA-R) tasks from the controller to a dedicated PCE [49]. To this end, a Path Computation Client (PCC) was integrated into a NOX controller. Through the PCC, the controller can send path-computation requests to a dedicated stateless PCE (one PCE dedicated to each domain), via PCE communication protocol (PCEP) [30]. An extended OpenFlow modifies the circuit-flow entries of OpenFlow-enabled optical nodes, following the work presented in [67]. The PCC is informed of the successful lightpath establishment using the PCEP protocol.

Ref. [74] exploited for the first time the PCEP [30] to improve the interoperation between SDN and GMPLS control planes (see section IX-C2). However, the PCE was not fully aware of the current state of the network. Thus, in Refs. [147], [148] a topology server was included into the SDN controller. Such server updates the topology information that the PCE employs for optical path computation.

In other work from OFELIA the PCE module was placed as an application on top of the SDN controller. Such PCE is able perform constraint-aware lightpath computation based on the gathered resource and switching-constraint information [72].

In [77] and [78] the OpenFlow integrated stateless-PCE SDN controller proposed in [147] was extended to support dynamic path computation and restoration in wavelength switched EONs. Using the Traffic Engineering Database (TED) updated by the controller, the PCE includes Optical Signal to Noise Ratio (OSNR) information for Impairment-Aware Routing and Wavelength Assignment (IA-RWA).

The authors of [79] and [80] upgrades the stateless-PCE used in [74] to a fully integrated Stateful Path Computation Element (S-PCE) inside the SDN controller. The S-PCE have access to the active paths on the network stored in a Label Switched Path Database (LSP-DB), allowing to improve the effectiveness of the path computation algorithms.

- *Other OpenFlow agent-based SDON proposals*: works presented in [85] and [88] experimentally evaluated: two OpenFlow-based UCP (*OpenFlow Messages Mapping* and *OpenFlow Extensions*), and a GMPLS-PCE UCP. In line with previous works [146][144], the OpenFlow agent acts as a virtual switch and translates among OpenFlow messages and TL1 commands. A PCE was implemented as a network application using the Northbound interface (NBI) of the controller.
  - OpenFlow Extensions model: it is based on the Addendum to OpenFlow protocol specification (v1. 0) [62].
  - OpenFlow Messages Mapping model: the OpenFlow messages are not extended but mapped into optical switch commands.

The authors of [82] demonstrated the control of inter-data center network, with an OpenFlow+ controller that interacts with agents on top of flexi-grid optical network devices. Optical domain controller interacts with an application (or datacenter) controller in a flat architecture using an proposed interface called application-transport interface.

The authors of [88] successfully demonstrated end-to-end lightpath establishment and restoration within a small experimental setup. Additionally, the authors developed an event-driven simulator that allowed to test their unified control planes in larger networks (NSFNet and COST 239 topology [4]). In accordance with precedent works, the *OpenFlow Extension* model improved the performance of the control plane and reduced the lightpath setup time, when compared to *OpenFlow Messages Mapping model*.

*3) **NETCONF/YANG with agent-based support model**:* the Network Configuration Protocol (NETCONF) RFC 6241 [32] is a standardized network management protocol that provides mechanisms to install, manipulate, and delete the configuration of network devices. While OpenFlow was conceived to program the data-plane forwarding-rules, NETCONF was conceived to configure the data-plane devices.

The T-SDN controller architecture depicted in Fig. 14, adopts NETCONF/Representational State Transfer (REST) as Southbound interface for configuration of optical equipment and advertisement of operational data [83]. The controller communicates with agents on top of optical NEs. Each agent is composed by a NETCONF modeling language YANG [136] database that provides proper abstraction of optical NEs. The agent also uses an Optical Supervisory Channel (OSC), to implement Link Layer Discovery Protocol (LLDP) to populate the YANG database.

An optical network abstraction maintained at the controller provides for policy-based operations of EON. The controller provides Virtual Optical Network (VON)s using network resource slicing based on wavelengths, nodes and links.

The authors of [84] demonstrated how the GMPLS can be deployed as a virtual control plane for each node of the VONs. This allows deployment of independent control planes for each VON and complexity reduction of optical network equipment. The authors of [84] claimed that the NMS can run on top of the virtual control plane via the User Network Interface (UNI).

The NETCONF-based controller have been used to prototype multiple control applications for the management of VONs over complex optical data planes. [86] presents a global equalization algorithm for ROADMs, [87] describes a PCE based IA-RWA algorithm, while in [89] an application reconfigures the transmission modulation format according to pre-established OSNR thresholds. The NETCONF-based controller [90] was extended to enforce context-aware policy-based control of network applications. Such extensions allows adaptive optical configuration (e.g., equalization and gain control), VONs provisioning and VONs restoration.

While OpenFlow has been the main focus for SDON research development, today NETCONF is recognized as the

[4]National Science Foundation Network (NSFNet) and Ultra-High Capacity Optical Transmission Networks (COST)

main SBI to be deployed in T-SDN. In section IX-E we report the efforts spent to define open APIs to control and manage transport networks (Transport APIs) that are supported by the NETCONF protocol.

B. SDON Architectures with multiple SBIs from SDN and GMPLS (Hybrid SDN/GMPLS models) towards optical domains

OpenFlow was conceived for packet domain, and optical equipment most of the time do not provide any circuit-switching flow-table as the one proposed by [62]. On the other hand, GMPLS was the most used control plane in legacy optical transport networks, and it is still wide-spread in current optical-equipment product lines.

GMPLS was also proposed to be used as part of the UCP for the IP and the transport networks using an overlay model [149]. In the aforementioned model, IP/MPLS network is managed as an overlay on top of the transport network. Thus, IP and optical control planes are separated and do not share topology information. The IP control plane requests services to the optical domain through the GMPLS UNI defined in RFC 4208 [149], according to the Automatic Switched Optical Networks (ASON) standards of the ITU [150].

Despite a long-lasting standardization process and a long list of GMPLS-compliant transport-equipment implementations, today there are no major commercial deployments of GMPLS as alternate UCP, due to its high level of complexity [151]. Das et al. [151] presented a comparison between SDN/OpenFlow and GMPLS/UNI as UCP, where control plane complexity, lack of visibility and flexibility provided by GMPLS/UNI and the use of vendor-specific interfaces are key shortcomings improved by SDN/OpenFlow model to tackle a UCP. However, SDN and GMPLS integration for a UCP spanning over circuit and packet domains represents a less disruptive approach than PAC.C (described in section V-A1). GMPLS can be reused as the control plane of the optical domain, while extended SDN/OpenFlow control plane can: 1) interface with GMPLS, 2) control the OpenFlow-enabled packet domains, and 3) provide centralized network view and intelligence. In this section we present a classification of the first proposals that envisioned the SDN/GMPLS control plane integration [69], [70], [75], [152].

Another approach used to create a UCP with GMPLS is based on the Path Computation Element (GMPLS/PCE) architecture [49]. The PCE provides centralized network view and path computation to the distributed control of GMPLS. GMPLS/PCE plays an important role in the evolution of T-SDN as explained in section IX-C2.

*1) **Legacy interface towards GMPLS control plane (CP)**:*

- *ASON-UNI towards GMPLS CP* [69], [70]: it was the first proposal for interworking between GMPLS and SDN. The SDN/UNI-GMPLS architecture is depicted in fig. 15. The standard UNI provides an overlay model for requesting optical connectivity over a virtual node abstraction (big switch) of the optical domain. With SDN/UNI-GMPLS the SDN controller has no visibility over the optical domain topology.
  The experimental demonstration of the SDN/UNI-GMPLS presented in [70] comprises one extended NOX controller [153], two OpenFlow-enabled packet domains, and one GMPLS domain. An OpenFlow Gateway was implemented inside NOX to interface the GMPLS control plane via UNI.
  Following the approach proposed by [70], the authors of [75] experimentally evaluated three variations (named parallel, overlay and integrated) to interface SDN and GMPLS control planes.
- *RSVP-TE towards GMPLS CP* [75]: it is similar to the approach presented in [70], however the interface between SDN and GMPLS control planes is based on vendor's proprietary protocols and not on the ASON-UNI. Instead, the NOX controller was extended to request the optical paths to the GMPLS control plane through the RSVP-TE protocol.

*2) **Extended OpenFlow (OF+) towards GMPLS CP**:*

- *GMPLS nodes with agent-based OF+ support* [75]: this model employs an OpenFlow agent (the OpenFlow agent is explained in section V-A2) called OpenFlow Switch (OFS) on top of each GMPLS node as an interface between SDN and GMPLS control planes (Fig. 16). The OFS allows the SDN controller to obtain full topology visibility and lightpath setup capabilities via OpenFlow. Each OFS maps the requests for adding circuit-flows (extended *Flow Mod* messages) into GMPLS Label Switched Path (LSP)s by defining an Explicit Route Object (ERO) to the GMPLS control plane.
- *GMPLS nodes with native OF+ support* [75]: in this model the OFS and the GMPLS control plane are merged into a single OpenFlow-enabled GMPLS control plane (OF-GC). The procedure of end-to-end path provisioning is similar to the previous model (SDN/GMPLS using OF-agents). The main difference with the model based on OF agents is that the OF-GC is able to directly communicate with the SDN controller and internally do all the operations between OFS and GMPLS.
  Experimental results showed that integrated models achieve faster path provisioning times than parallel and overlay solutions [75]. However, the integrated model implies that vendors should modify the GMPLS control plane to offer a standard OpenFlow interface, while the overlay model can be implemented into already deployed systems by adding the OFS agent on top of the optical devices. A drawback of overlay and parallel models is that the interface between SDN controller and GMPLS devices is based on proprietary non-standard implementations that may lead to interoperability issues in a multi-vendor scenario.
  Ref. [152] presented a comparative analysis of GMPLS-PCE, SDN/UNI-GMPLS (based on [70]) and OpenFlow (based on [61]) models. By means of simulations, the authors of [152] concluded that OpenFlow can reduce the lightpath setup time of GMPLS thanks to the centralized

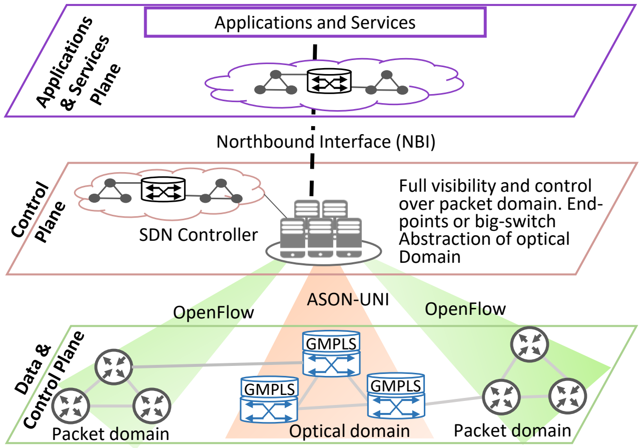

Fig. 15. First proposal of SDN and GMPLS integration via ASON-UNI (SDN/UNI-GMPLS) [70]. The UNI offers a big switch abstraction with end-point control and visibility.

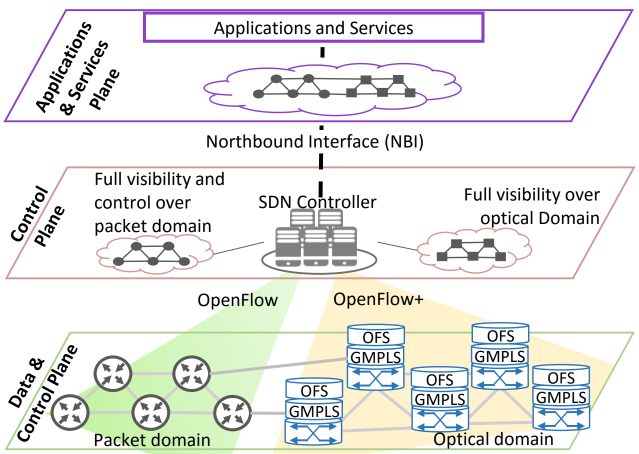

Fig. 16. GMPLS nodes with agent-based OF+ support [75], it offers full visibility of optical domain by allocating OpenFlow agents (OFS) that interact with the SDN and GMPLS control planes.

path computation and parallelized exchange of path setup messages.

*3) **Legacy interfaces (RSVP-TE and OSPF-TE) and agent-based OF+ towards GMPLS CP**:* ref. [72] proposed a hybrid GMPLS-OpenFlow model (Fig. 17): it is also based on the OpenFlow-agent, but it relies on GMPLS control plane to provide lightpath computation, establishment and verification. GMPLS provides functionalities for power equalization and switching constraints that were not available in previous OpenFlow extensions, such as [62] or [67]. On the other hand, the OF-A allows to gather full-topology and resource information at the controller, improving the visibility over optical domains of previous GMPLS-based solutions [70] (see section V-B). Two lightpath-establishment methods were proposed in [72], [76]: loose and explicit. In the former, the controller obtains an optical-domain big-switch abstraction, thus lightpaths are managed by GMPLS. In the later, the controller obtains full topology and resource information, thus it is able to compute explicit paths, while GMPLS establishes and manages the lightpaths.

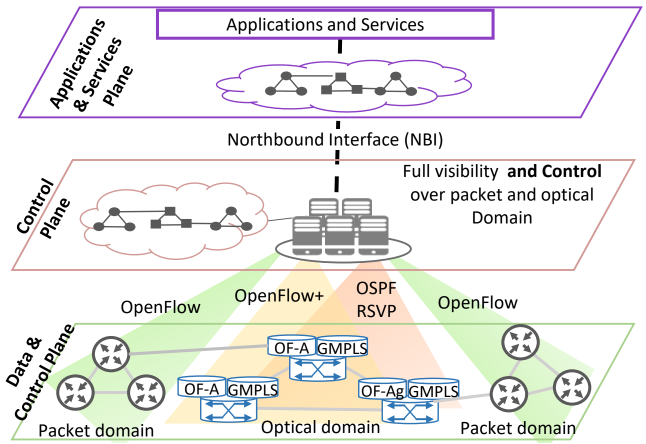

Fig. 17. Hybrid GMPLS-OpenFlow+ model. Optical specific features are managed by the GMPLS control plane, while the OpenFlow agent (OF-A) provides the visibility over the topology. OSPF-TE and RSVP-TE are protocols used by GMPLS.

VI. Research Efforts on Hierarchical Control Plane Architectures (HT-SDN)

After successful SDON proof-of-concepts (section V), the transport SDN efforts have shifted the focus towards hierarchical control plane architectures. In this work, the hierarchical T-SDN architectures are classified as HT-SDN. The transport optical network is a complex system with heterogeneous domains, comprising packets (Ethernet, MPLS and MPLS-TP), as well as circuits (SDH/SONET, OTN and WDM). It is normally composed by vendor-specific islands, each with a proprietary and centralized management plane. Each domain runs with a combination of centralized and distributed proprietary control-plane (e.g., ASON and GMPLS).

Fig. 18 depicts an example of the HT-SDN architecture. On top of the hierarchy, a parent controller or a Transport Network Orchestrator (TN-Orchestrator) application interoperates with domain controllers to provision end-to-end and inter-domain services. At the domain level, specialized controllers are in charge of intra domain services. The hierarchical architecture increases scalability and allows better integration of the heterogeneous domains. Several standardization bodies led by ONF [38] and OIF [59] support such hierarchical architecture. Through definition of proper abstractions and interfaces the HT-SDN architecture is able to control multiple vendor islands based on standard and proprietary technologies and protocols.

In HT-SDN architecture, domain controllers are in charge of intra-domain path computation and management of the optical domain complexity. Each domain controller provides an abstract representation of the network. The parent controller or a transport network orchestrator, placed above domain controllers, is in charge of inter-domain and end-to-end path computation. The TN-Orchestrator (or just orchestrator for simplicity) was defined by the OIF as an application that employs the control plane's NBIs to gather topological information and request services across the network, orchestrating the creation of multi domain services [37].

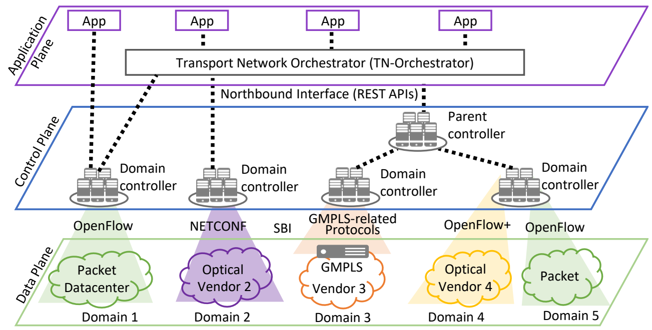

Fig. 18. Hierarchical T-SDN (HT-SDN) architecture.

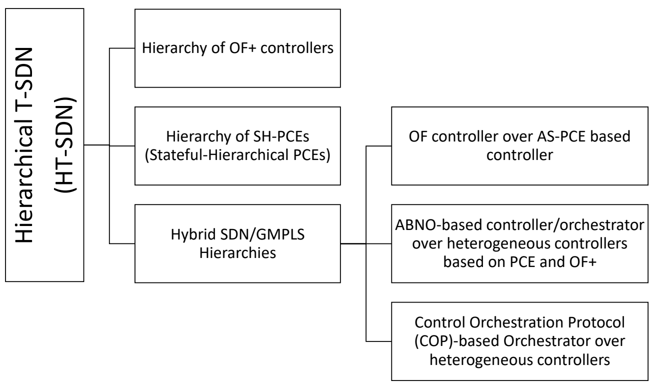

Fig. 19. Classification of HT-SDN solutions based on the SBI used towards the optical domains and the NBI used among the hierarchy of controllers

The documents published by Infonetics [154] and OIF [59] reported that for operators with complex multi-domain networks, implementing a hierarchical and multi-domain orchestration system is a necessity. Moreover, [154] predicted the use of multiple orchestration layers composed by infrastructure and service orchestrators. The former provides coordinated network operation at the physical network, while the later focuses on the service level.

Figueira et al. [155] proposed a hierarchical multi-domain SDN orchestration and control plane architecture based on a tiered framework. In this framework, datacenters, access, metro and WAN compose a regional domain, while clusters of regions create a main domain. Each domain is coordinated by a specific regional or main orchestrator (a deeper analysis of the orchestration systems and its classification is out of the scope of this survey).

Different interfaces can be used between the hierarchy of controllers and the optical domains such as: OpenFlow (II-D), Link-State Distribution Using Border Gateway Protocol (BGP-LS), PCEP (subsection IX-C2), REST APIs and NETCONF (section IX-E). In this section we provide a classification of HT-SDN solutions (summarized in Fig. 19) based on the SBIs used towards the optical domains and the NBI used throughout the hierarchy of controllers.

A. Hierarchy of OF+ controllers

Ref. [91] demonstrated the orchestration of inter and intra datacenter dynamic communications using a hierarchical architecture of pure OpenFlow controllers as depicted in Fig. 20. This approach is similar to ONF architecture described in section IX-A1.

In the assessment scenario, two datacenters were connected via optical transport network; four NOX-based domain controllers manage: the two datacenters, the IP and the optical domain of the transport network. Services across domain controllers are coordinated by another NOX-based parent controller. A global load-balancing function across datacenter, IP and optical layer was implemented at the application level to exploit the unified view and control offered by the parent controller NBI.

B. Hierarchy of SH-PCEs (Stateful-Hierarchical PCEs)

Contrary to other works, in [97] the stateful hierarchical PCE (SH-PCE) architecture was proposed as the key element to tackle transport network orchestration (Fig. 21). Orchestration across GMPLS and OpenFlow-based optical domains was demonstrated in an emulated testbed of the SH-PCE architecture. A parent PCE with orchestration capabilities coordinates inter domain path computation over three child S-PCEs (c S-PCE). One child S-PCE directly governs the GMPLS-controlled flexi-grid DWDM network. The other two child S-PCE are integrated inside OpenFlow controllers that were extended to support flexi-grid DWDM networks (following the work presented in [80]).

The main contribution of this work is to extend H-PCE with stateful capabilities, and to integrate child S-PCE with OpenFlow controllers. Multiple extensions were proposed in [97] including those to achieve: initiation, delegation and topology discovery of GMPLS and OpenFlow domains with PCEP, and support for Routing and Spectrum Assignment (RSA) of flexi-grid networks. Additional extensions to the Open Shortest Path First - Traffic Engineering (OSPF) and RSVP-TE protocols were developed to allow the control over flexi-grid Dense Wavelength Division Multiplexing (DWDM) technologies. Regarding OpenFlow, extensions (OpenFlow+) to the circuit switch addendum [62] were addressed to support flexi-grid ROADMs configuration, and the OpenFlow agent approach was used to enable optical gear.

C. Hybrid SDN/GMPLS hierarchy

*1) **SDN controller over active-stateful PCE (AS-PCE)-based controller**:* the HT-SDN architecture that is illustrated in Fig. 22 was tested in [92] using the Adrenaline testbed framework. The packet switched domains are directly controlled by OpenDaylight using OpenFlow. The two packet domains are interconnected through an optical domain with a distributed GMPLS control plane. The AS-PCE serves as domain controller, that provides a hardware abstraction layer with full visibility over the GMPLS domain. The PCEP plug-in provided by ODL was extended to support active stateful PCE [120]. Thus, the ODL controller is able to establish a PCEP session with the AS-PCE that governs the GMPLS-controlled optical domain. The AS-PCE allows the SDN controller to either request a connection between two border nodes of the

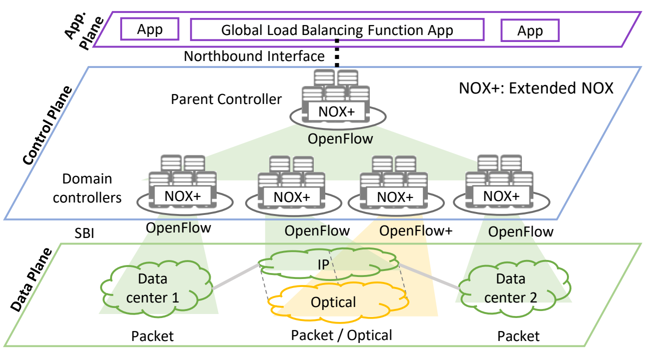

Fig. 20. Hierarchy of OF+ controllers [91].

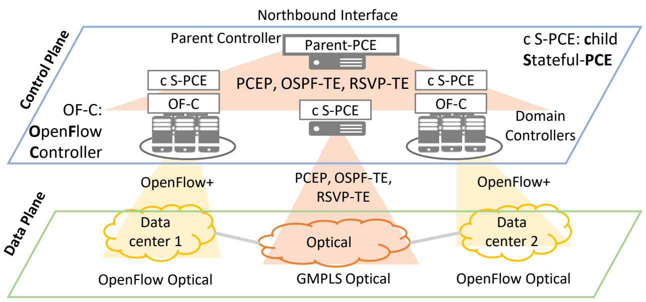

Fig. 21. Hierarchy of stateful hierarchical PCEs (SH-PCE)s [97].

optical domain or request the establishment of a specific path LSP by defining an explicit route object (ERO).

The ODL topology manager service gathers network topology by listening LLDP packets. This information is available at the Northbound interface of ODL. Orchestration applications were deployed using the REST APIs offered by ODL [156] for topology acquisition and end-to-end path computation across the multiple domains [92].

Based on the same architecture presented in [92] (see Fig. 22), the authors of [93] demonstrated integrated orchestration of network and IT resources for inter and intra datacenter dynamic control. Two remote OpenStack-controlled datacenters are interconnected through a legacy GMPLS-controlled optical domain with an AS-PCE that interoperate with the OpenDaylight controller.

In the architecture proposed by [93], an Orchestrator application with transport network and IT resource capabilities serves as a mediator for customer applications to request the provisioning of IT resources. The orchestrator requests the creation of virtual machines (VMs) to the OpenStack cloud operating system [157] using open APIs. Upon successful creation of the VMs on the hosts nodes (inside the two datacenter packet domains), the OpenStack controller sends to the orchestrator the information needed to interconnect the VMs instances (MAC, IP addresses and physical host node location). The Orchestrator sends an interconnection request through the ODL APIs. Depending on location of the VMs, the SDN parent controller configures the packet domains using OpenFlow and the optical domain using PCEP through the AS-PCE -based domain controller.

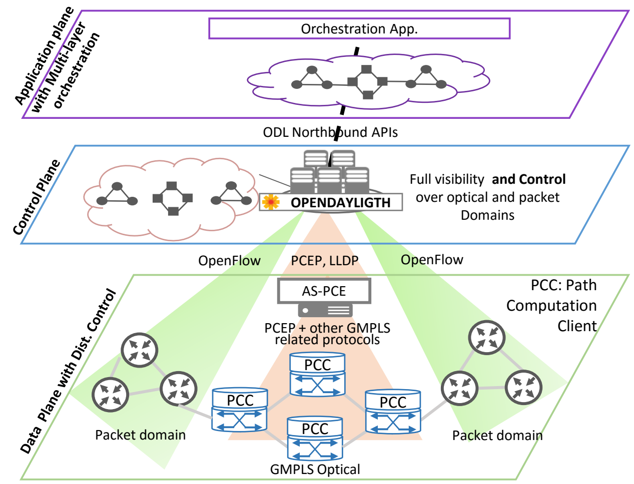

Fig. 22. Hybrid SDN and GMPLS hierarchy using OpenDaylight (parent controller) and AS-PCE (domain controller) [92].

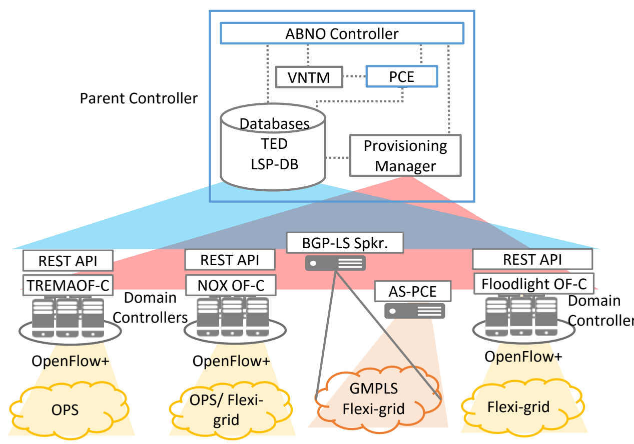

Fig. 23. Hybrid SDN and GMPLS hierarchy using ABNO (orchestrator/parent controller) over heterogeneous domains [95].

*2) **SDN controller/orchestrator over heterogeneous controllers**:* Telefonica I+D (the research and development company of the Telefonica Group) in the framework of the European projects STRAUSS and IDEALIST, built the first experimental demonstration that follows the ABNO framework [94], work that was later extended in [98]. The authors of [94] demonstrated the feasibility of ABNO to orchestrate automatic provisioning of IP connections (Juniper routers) across two optical domains, one with an emulated GMPLS control plane and other with an ADVA SDN/OpenFlow controller based on Floodlight [158]. From the pool of components of the model described in [117], the Authors of [94] only built the following modules: ABNO controller, Policy Agent (PA), topology module (or TM, keeps or maintains the TED), Virtual Network Topology Manager (VNTM), L0 PCE, Provisioning Manager (PM) and the NMS.

HTTP was used between the NMS and the ABNO controller to request/reply IP connectivity. The optical layer was configured using: the REST API provided by ADVA SDN/OpenFlow controller, and the PCEP for the GMPLS-controlled domain. Once the optical layer provides connectivity to the IP layer, the Command Line Interface (CLI) is used to setup the Juniper routers.

Due to the modularity of the ABNO framework, TM and PM are the only modules that needed vendor-specific interfaces and protocol solutions to achieve this multi-vendor and multi-layer control plane orchestration.

Also provided by Telefonica I+D, the netphony-network-protocols public repository [159] contains the implementation of four networking protocols: PCEP [30], RSVP-TE [160], OSPF-TE [161] and BGP-LS [31][162].

HT-SDN over multi-domain and multi-technology of SDN/OpenFlow-based domains was presented for the first time in [96][99] as part of the STRAUSS project: one Optical Packet Switching (OPS) domain with variable capacity controlled by a Trema-based controller [163]. The OCS domain is an EON with flexi-grid technology controlled by a NOX-based controller [153]. Topology information and end-to-end services are orchestrated through the exposed NBIs presented by each controller using the implementation of the ABNO framework done by CTTC and Telefonica [94].

In the OPS domain, two Discrete Multi-Tone (DMT) packet transmitters are directly controlled by the OpenFlow Trema-based controller, allowing to set the bit-rate according to the distances of the possible routes. The OPS router is connected to an OpenFlow agent that translates the flow tables into switching tables. The EON domain is composed by an OPS-EON interfacing node, and three EON nodes based on optical wavelength selective switches all enabled using OpenFlow agents. The SDN-enabled EON allows to provide flexible services based on signals that adapt to the requested bandwidth, link situation and the adopted Forward Error Correction (FEC) encoding.

Again in the framework of the STRAUSS project, in [95] the ABNO framework was used to orchestrate end-to-end services over: two SDN/OpenFlow controlled OPS domains, two SDN/OpenFlow controlled OPS/OCS domains as presented in [96], and a GMPLS/PCE controlled OCS domain with AS-PCE and BGP-LS speaker. Is worth noting that this experiment only involves the control planes of such domains and there is no real data plane configuration. Fig. 23 shows the international testbed including OpenFlow enabled OPS, OPS/flexi-grid and flexi-grid domains, and one GMPLS controlled flexi-grid domain.

The authors of [95] and [100] aimed at improving scalability and considering possible confidentiality issues that can arise from large and heterogeneous networks. Therefore, the topology manager of the ABNO framework was configured to work with abstracted views of the network domains based on the virtual node aggregation (also known as the big switch abstraction), instead of working with the full topology abstraction. Each domain controller is responsible for mapping the real topology into the big switch abstraction, where the edge nodes are presented as ports of the virtual node and they are connected with inter-domain links (hiding the intra-domain topology).

*3) **Control Orchestration Protocol (COP)-based Orchestrator over heterogeneous controllers**:* the COP is a solution from the STRAUSS European project allowing interoperability among heterogeneous multi-domain, multi-technology transport networks [101][102].

COP is intended for the Northbound interface of diverse control plane technologies. It provides REST APIs using RESTCONF. Technology-specific data models are defined using YANG.

COP provides the following API Calls: end-to-end connectivity provisioning service, the topology service and the path computation service.

A drawback of COP is that the data model is not derived from standard information models like in the case of ONF Transport APIs or the YANG Data Models of IETF (see section IX-E for the standardization efforts on transport APIs).

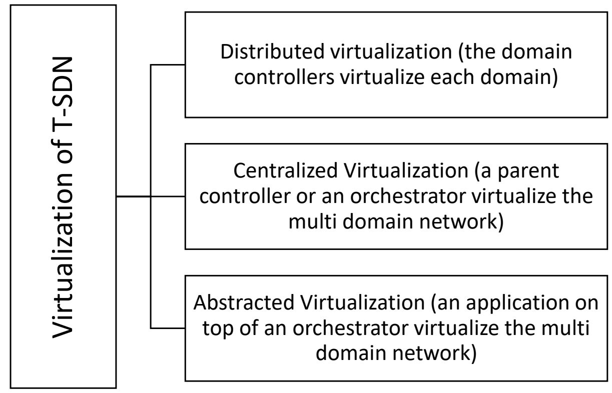

Fig. 24. Classification of T-SDN virtualization architectures

VII. Research efforts on T-SDN Virtualization

Thanks to the multiple abstraction layers provided by SDN (some of them discussed in section II-C), T-SDN enables efficient and flexible virtualization of transport networks. A Virtual Network (VN) is a logical topology composed by virtual nodes and virtual links mapped into a physical infrastructure. Multiple VNs share a common networking infrastructure, each with distinct forwarding logic, and isolated from each other. In the context of SDN, an instance of a virtual network is commonly called *slice* [164]. Each slice can be separately managed by a guest or internal SDN controller.

We refer to the *hypervisor* as the virtualization platform or layer that enables distinct slices to share a common networking infrastructure. The hypervisor introduces another abstraction layer into SDN architecture to allow the creation and management of network slices. Moreover, SDN allows to jointly optimize the virtual embedding of network and computation infrastructure.

Supporting provision of VN services is a requirement for T-SDN [40] [59]. In [59], requirement 32 specifies, in the context of multi-layer T-SDN, that data plane needs to support network slicing using: a) dedicated resources per service, and b) sharable resources among services.

In the following section we introduce a classification of virtualization architectures for T-SDN, we discuss algorithms for VN embedding in T-SDN, and present implementation strategies for network function virtualization (NFV) in T-SDN.

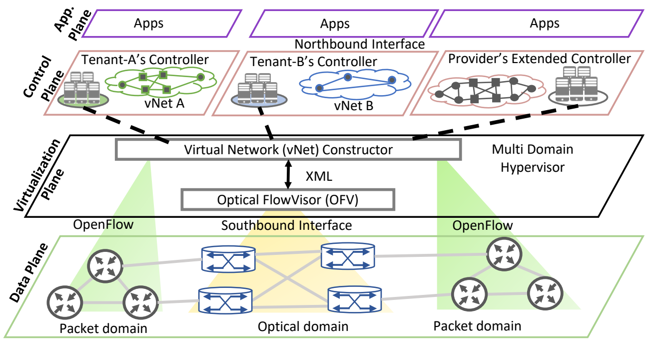

Fig. 25. Distributed (or Southbound) virtualization using a multi domain hypervisor called Optical Flow Visor [103].

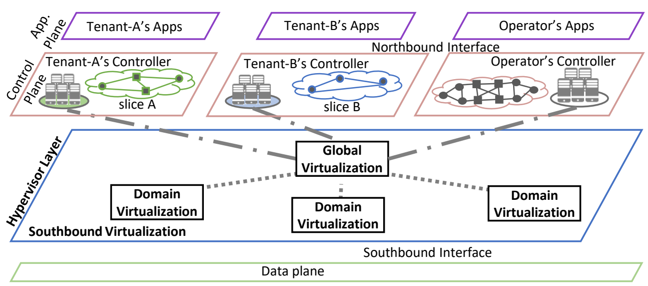

Fig. 26. Distributed (or Southbound) virtualization architecture for the multi-domain scenario, using a hierarchy of virtualization elements.

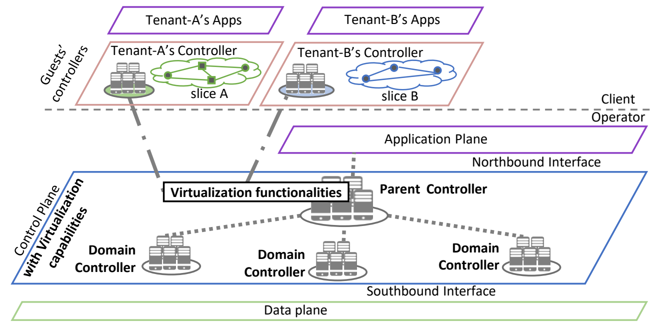

Fig. 27. Centralized virtualization architecture for the multi-domain scenario.

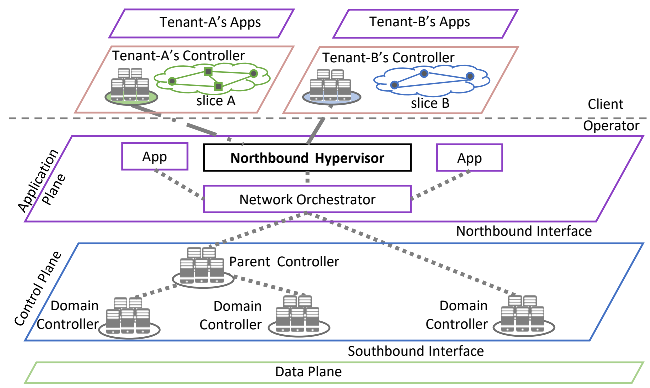

Fig. 28. Abstracted (or Northbound) virtualization architecture for the multi-domain scenario.

A. Classification of virtualization architectures for T-SDN

As in several issues related to T-SDN, there is no consensus or stable standardization about the provision of VNs in transport networks. In Fig. 24 we present a classification of VN architecture solutions for T-SDN based on the location of the virtualization platform or layer.

1) ***Distributed (or Southbound) virtualization*** *(the domain controllers virtualize each domain):* a hypervisor layer is placed between the data plane and the control plane; tenant controllers are deployed over the virtual networks provided by the virtualization platform.

The Optical Flow Visor (OFV) [103] was the first proposed for virtualization of transport networks for the monolithic T-SDN controller architecture.

The OFV proposed in [103] is based on a packet switch virtualization engine for packet-oriented SDN called FlowVisor [164]. Fig. 25 depicts the OFV architecture. The OFV manages the optical layer features, configures the optical NEs and provides impairment-aware virtual optical networks (VON)s. Inside the virtual plane, the virtual-network constructor provides converged packet- and optical-domain virtual networks. At the OFV resides the Optical Connection Controller (OCC) that configures and manages the optical devices, and serves as interface between VONs and optical data plane. The OCC of the architecture proposed in [103] can be implemented as an SDON controller plus a network management system.

Ref. [106] also considered the distributed virtualization for T-SDN, using three degrees of topology abstraction: single node (big switch), full topology (no abstraction), and abstract link model (provides an intermediate abstraction level between the previous models). Based on the analysis done in [106], the abstract link model presents the best trade off between manageability and complexity towards the tenant controllers.

In order to apply distributed virtualization in multi-domain transport networks, a hypervisor layer, composed by a hierarchy of virtualization platforms needs to be implemented. Fig. 26 depicts the hierarchical architecture of distributed (or Southbound) virtualization for multi-domain scenarios.

In [107], [108], [165], technology-specific virtualization platforms were placed on top of heterogeneous domains, and a global or parent virtualization controller created the hierarchical virtualization layer. The global virtualization has some functionalities of an SDN controller: gather a global network view, network resource assignment, VN request handling and VNs construction.

2) ***Centralized Virtualization*** *(a parent controller or an orchestrator virtualizes the multi domain network):* the virtualization platform is placed inside the SDN controller. It uses internal controller interfaces to gather and control an abstract view of the network. Using central virtualization, the SDN controller provides internal NBI for creation and management of slices.

The centralized virtualization was first used in [83], using a NETCONF-based controller, in the context of SDON architecture, discussed in section V-A3. The NETCONF-based controller was able to support heterogeneous control planes for the network slices.

Fig. 27 presents the general architecture of central virtualization for multi-domain networks, where virtualization functionalities are placed at the top of the hierarchy of controllers, i.e., in the parent or global controller of HT-SDN. By exploiting the multi-domain capabilities of the parent controller, centralized virtualization is easier to deploy in multi-domain scenarios than Southbound virtualization.

Central virtualization is supported by ONF. In the ONF OpenFlow-enabled T-SDN reference architecture [38], the virtualization platform is considered as a functionality of the Global Controller. ONF even defined a Control Virtual Network Interface (CVNI), as the interface used between controllers, including the one between global controller and virtual tenant controllers (see section IX-A1).

*3) **Abstracted (or Northbound) Virtualization** (an application on top of an orchestrator virtualizes the multi domain network):* the hypervisor is placed above the control plane, or even on top of the transport network orchestrator, as Fig. 28 depicts. Northbound virtualization, exploits the multi-domain capabilities and global abstracted view provided by the control plane or orchestrator. Thus, it can be also called abstracted virtualization.

Northbound virtualization is supported by OIF [59]. For OIF, the orchestrator must control the slicing of the transport network infrastructure. Moreover, a higher level orchestrator with management capabilities over network and IT resources must provide virtualization for transport networks (e.g., NFV) and datacenters (e.g., virtual machines).

Ref. [109] developed a Northbound network hypervisor, that exploits the APIs provided by an ABNO-based multi-domain network orchestrator. The network hypervisor allows OpenFlow-based guests' controllers to manage their own slice of the network. The ABNO orchestrator is in charge of guaranteeing end-to-end QoS for each slice, over multi-technology and multi-domain networks.

In [112], authors demonstrate a Northbound virtualization that provides dynamic VNs which are able to react upon congestion and failures. A Northbound virtualization platform exploits ABNO orchestrator capabilities for re-planning and recovery mechanisms and changes applied bellow the orchestrator are transparent for the VNs.

B. T-SDN and Network Function Virtualization (NFV)

While SDN decouples control from data-plane, NFV decouples software from hardware. NFV is a network architecture paradigm that leverages virtualization techniques to dynamically deliver Virtualized Network Functions (VNF)s. VNFs are software implementations of Physical Network Functions (PNF)s, including data, control and management functionalities, which are necessary to run a network. The VNFs can be dynamically instantiated into a cloud computing environments using Commercial Off-The-Shelf (COTS) hardware, instead of running into function- and vendor-specific hardware. NFV allows to scale up and down resources upon requirements. The freedom to place compute and storage resources in the most convenient places and hours allows to improve operational efficiencies

In this subsection we provide a short introduction to NFV and summarize the research efforts on T-SDN and NFV.

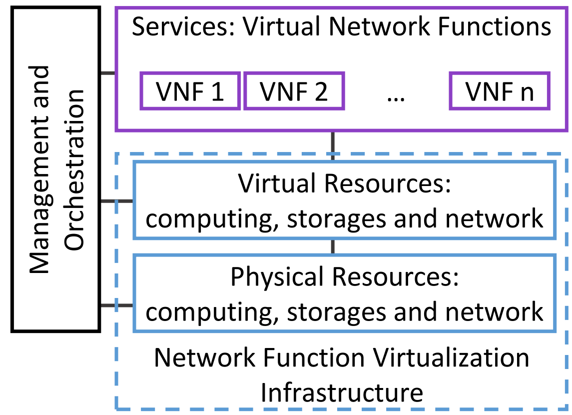

Fig. 29. Network function virtualization architecture

*1) **NFV Primer**:* according to European Telecommunications Standards Institute (ETSI), the NFV Architecture is composed of the three main elements depicted in Fig. 29: Network Function Virtualization Infrastructure (NFVI), VNFs and NFV MANagement and Orchestration (MANO) [166]. The Open Source MANO is an operator-led community that provides a MANO stack aligned with ETSi NFV architectures and information models [167].

ETSI is also addressing the problem of connectivity among VNFs and proposes the use of VNF Forwarding Graph (VNFFG) to define the chains of VNFs. The sequential concatenation of VNFs and/or PNFs to provide end-to-end services, is known as Service Function Chaining (SFC). The provisioning of SFC across WANs involves the interaction with T-SDN controllers.

Another open source effort driven towards carrier and service provider multi-domain networks is the Open Platform for NFV (OPNFV) [139]. OPNFV main goal is to facilitate the development and evolution of NFV components across various open source ecosystems: OpenDaylight, ONOS, OpenStack, Ceph, KVM, Open vSwitch, and Linux. OPNFV is developing and testing a platform to accelerate the transformation of enterprise and service provider networks.

Since the focus of this work is not on NFV, we refer the reader to [168] and [169] for state-of-the-art and research challenges on NFV and SFC, respectively.

*2) **T-SDN and NFV**:* in previous chapters of this work we have presented T-SDN architectures without NFV. On the other end, NFV can be implemented without SDN technologies. In fact, the first attempt of introducing NFV in transport networks did not use an SDN control plane, but a GMPLS/PCE control plane [170]. NFV was used to virtualize the PCE as a VNF. A PCE NFV Orchestrator creates and releases virtual PCEs (vPCE)s dynamically, adapting to demand variations of path computation requests. BGP-LS was not enabled to acquire the topology, thus all vPCEs share a static topology. A path computation entity must first consult the IP address of the vPCE to a PCE-DNS, which is responsible of vPCE load balancing.

However, providers are willing to use both technologies to boost flexibility, speed-up deployment time and reduce costs.

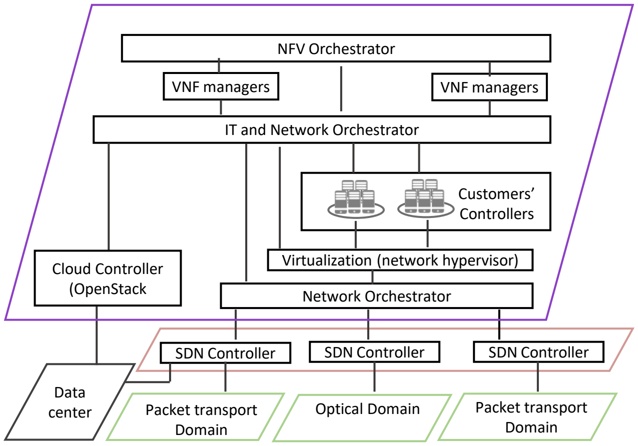

Fig. 30. Transport Network Function Virtualization architecture proposed in [115]

For instance, Verizon published the SDN-NFV Reference Architecture [114], based and co-authored with multiple vendors: Cisco, Ericsson, Hewlett Packard Enterprise, Intel, Nokia, Red Hat and Samsung.

In [111] and [115], the NFV architecture, together with the virtualization architectures of T-SDN, were used to virtualize client's SDN controllers into the cloud. By doing so, virtual T-SDN networks can be provisioned dynamically on demand, the controller can be relocated due to changes on the demands or to recover from a disaster. In sections VII-A1, VII-A2 and VII-A3 each client's controller runs in client-specific facility using dedicated hardware.

The T-SDN and NFV orchestrator for multi-tenant transport networks was first demonstrated in [111], over a single domain optical network. This orchestrator exploits NFV and Southbound virtualization provided by the optical network hypervisor of [106]. The average provisioning time of a virtual T-SDN, reported in [111], is less than 2 minutes, and involves multiple requests from the SDN NFV orchestrator: 1) creation of virtual SDN controller to a VNF manager, 2) flows setup between virtual controller, optical network hypervisor and client's network operation center (NOC), 3) creation of VON to the optical network hypervisor.

To demonstrate virtualization of tenant's controllers over heterogeneous multi-domain transport networks, the authors of [115] employed the Northbound virtualization approach together with NFV. Fig. 30 depicts the architecture proposed in [115] which exploits the multi-domain capabilities of the ABNO-based orchestrator [109] and the Northbound virtualization provided by the multi-domain network hypervisor [112] to instantiate resources across multiple domains.

In [116], the SDN-NFV architecture of [115] was extended to the mobile network. Radio Access Network (RAN) are connected to datacenter facilities through the backhaul network. The backhaul is composed of multiple domains with diverse transport network technologies. Ref. [116] demonstrated the virtualization of the backhaul network, with VNF-like SDN controllers and evolved packet core. Such virtualization can be greatly exploited in the mobile network to cope with the traffic demand variations.

C. T-SDN compatible VN embedding algorithms

Ref. [104] studied the virtual infrastructure embedding (VIE) problem for multi-domain flexi-grid networks controlled by a monolithic SDON architecture. The authors of [104] proposed a virtual link embedding algorithm to maximize the number of VONs embedded in the physical substrate, while taking into account transmission reachability and wavelength continuity constraints. The virtual link embedding in flexi-grid network involves assignment of routing, modulation format and spectrum. The research in [105] extended the one in [104] to perform both virtual link and node embedding of network and computing resources. Later in [110], the survivability against single node or link failure was introduced to the virtual embedding problem for T-SDN.

As SFC continues to gain traction in the industry, there is limited work on resilient SFC. The authors of [113] evaluated and proposed ILP models to provide resilient SFC against single-link and single-node failures.

VIII. Other Research Developments

After reviewing the research efforts on network virtualization (section VII), which is one of the main features of SDN, we now focus on other three important topics for development and successful implementation of T-SDN:

- Protection and restoration schemes, that allow transport network operators to guarantee service availability in case of failures (subsection VIII-A).
- Segment Routing, that offers the possibility to benefit from the interaction of distributed (e.g., MPLS and GMPLS) and centralized (PCE and SDN) CPs (subsection VIII-B).
- Network emulation, which supports both IP and optical network elements is a key for assessment of new T-SDN proposals (subsection VIII-C).

A. Protection and Restoration

In transport networks, failures may lead to huge amount of data loss. Thus, multiple protection and restoration schemes have been proposed in literature. In a protection scheme, the connection is provided with a primary and a protection path. The primary is used during normal operation. The protection path is used only after the primary path is affected by a failure. In a restoration scheme, the network reacts after failures to re-provision the disrupted connections. Restoration schemes avoid provisioning high amounts of bandwidth for protection. However, it does so with a penalty in longer recovery times, and lower guarantee to re-provision affected connections.

T-SDN has properties to improve restoration and protection schemes, for instance: 1) the centralized nature that allows to set up path faster by sending flow set up messages in parallel to all the nodes of a path, and increases the dynamicity of the network. 2) The network wide view gathered at the control and application plane that allows to implement optimization algorithms. 3) The multi-layer and multi-domain capabilities that can unify the protection and restoration schemes across multiple layers and domains [171], [172]. In the following, this subsection presents a list of proposals on protection and restoration for T-SDN.

- In [173] and [174] the authors presented an SDN/OpenFlow-based restoration mechanism for EONs that takes into account the physical impairments to improve the efficiency of the controller to find feasible restoration paths. The proposed extensions to the OpenFlow included those in support of EON from [81] and the definition of a new message to support the alarms (notification of link failures) in the network. The solution presented in [174] included a mechanism to determine specific single-point of failure by combining the information of the network topology and the current established lightpaths in the network. After the failure point is determined, the controller first deletes the flow entries of the working path and then set up the new path obtained with a two-phase restoration routing, spectrum, and modulation format assignment (RSMA) algorithm. The proposed restoration function was tested in the GENI (Global Environment for Network Innovations) [175] testbed using an emulated data plane with 14 BV-WXC and a NOX-based [153] controller.
- In flexi-grid networks, the spectrum selective switches (SSS) requires longer configuration times (25 milliseconds) than the OXCs of WSON. Thus, in [176], authors demonstrated that the centralized nature of SDN can improve the recovery time of flexi-grid networks, and that outperforms the GMPLS/PCE control plane restoration schemes. Giorgetti et al. prototyped, by means of simulation, an SDN-based scheme to minimize the number of node reconfigurations and contentions during recovery of flexi-grid networks. Their scheme minimizes the overall recovery time and restoration blocking probability by bundling the reconfiguration instructions for all the affected lightpaths. In [177], authors extended the simulation scenarios, showing that the contentions (spectrum and node configuration) among different recovery signaling sessions, that are very likely to happen in a distributed control plane like GMPLS, increase the recovery time.
- A multipath protection scheme for OpenFlow-based flexi-grid networks was demonstrated in [178]. The authors extended the cross-connection table proposed in [73], by adding a field to specify the Type of signal using integer positive numbers. The Type field was used to distinguish between the primary path (Type = 0) and the set of n disjoint path for protection (Type = $n > 0$). The smaller field Type is, the higher the priority of the path is. The multipath resource allocation is done upon a new connection arrives using bandwidth squeezed protection. Upon failures, the controller determines the disrupted paths. Then, it sends FLOW_MOD messages to delete flow entries at the source node of all affected paths. Thus, each connection is forwarded along the remained protection path with highest priority. Results on a simulated environment demonstrated a reduction on blocking probability when using multipath protection against no protection. A drawback of this scheme is that it provisions a large amount of bandwidth for the multiple protection path.
- The authors of [179] proposed a Backup Reprovisioning with Partial Protection (BRPP) scheme for WDM networks, for disaster survivability. Thanks to SDN, the BRPP can use a mixture of logical protection and physical restoration. After every network state change, BRPP runs a global backup reprovisioning heuristic in an abstracted view of the network (at application plane) to calculate the protection paths. Only after failures, BRPP establishes the protection path in the data plane. Thus, avoiding extra delays due to contentions in the data plane. BRPP considers resource reallocation of disrupted and non-disrupted paths based on degraded service tolerance, to reduce blocking probability of the backups.

B. *Segment Routing*

The Source Packet Routing in Networking (SPRING), also called Segment Routing (SR) is a new source routing paradigm. It was announced by Cisco in 2013. SR provides traffic engineering (TE) solutions, while addressing several control plane drawbacks of legacy IP/MPLS networks e.g., improve scalability, simplicity, and ease of operation. SR is being standardized by the IETF SPRING Working Group [180]. In SR, Segment Identifiers (SID)s are labels, encoded in 32 bits MPLS labels, that represent intermediate path points. A path is specified at the source node using an ordered list of SIDs, compatible with an MPLS label stack.

SR was built for centralized control plane architectures, for instance SDN. SR offers the possibility to combine the advantages of distributed (e.g., MPLS and GMPLS) and centralized (PCE and SDN) control planes. Moreover, using SDN-based SR, the controller reduces the signaling, as it does not need to configure every node belonging to a flow. The controller need just to send configuration packets to the source node of the flow.

It is not surprising that there are works that already target SDN-based SR for multi-layer and multi-domain networks. Sgambelluri, *et al* [181] proposed the first SDN/OpenFlow-based SR implementation for multi-layer packet-optical network. In [181], dynamic packet rerouting with optical bypass capabilities was demonstrated. Later, Sgamberulli, *et al* [182], extended their work for multi-domain and multi-layer scenarios, using a mesh control plane architecture. Using non-standard east/west interfaces, the authors proposed a methodology to exchange intra-domain SID information. Kukreja *et al* [183], implemented a hierarchical SDN control plane, using standard Northbound (APIs) and Southbound (BGP-LS and PCEP) interfaces, to allow orchestration of multi-domain SR networks.

We expect to see more efforts in SDN-based SR for carrier networks.

C. Emulation

Among the SDN network emulation platforms, Mininet is the most popular open source solution [184] [185]. Mininet uses lightweight-virtualization mechanisms such as processes and virtual Ethernet pairs in network namespaces. The lightweight-virtualization based emulation allows to test large network instances in a laptop, which is not possible using a full-system emulation that uses one virtual machine per network element.

However, Mininet or any other open available SDN emulators do not support optical network elements.

To the best of our knowledge, SONEP (Software-Defined optical Network emulation platform) was the first SDN optical network emulator [186]. SONEP is a container based emulator composed by: virtual OTS from Infinera, virtual links (WDM and ethernet), virtual hosts and OpenFlow switches. Even though it is a promising solution for fast prototyping of T-SDN, SONEP is not available and cannot be used by the T-SDN research community.

More recently, LINC-OE (LINC-Switch for optical emulation) was developed by Infoblox/Flow-Forwarding community in collaboration with ON.Lab for the inclusion into the ONOS packet/optical convergence use case [187]. LINC is a software switch that supports OpenFlow and OF-config [188]. When the LINC-Switch is configured with a backend to emulate a ROADM, it becomes an optical emulator, where each ROADM runs as logical switch within LINC-Switch container [189]. LINC-OE supports OpenFlow, and uses proprietary extensions based on the "experimental" capability, in line (but not interoperable) with the ones presented in [39] LINC-OE allows to use Mininet for configuration, and support failures at links, ports and ROADMs.

A public repository from Telefonica I+D provides a Java based Emulator of a Transport Node (L1/L0) with GMPLS control plane [190].

IX. Standardization efforts on T-SDN

Multiple standardization bodies are working on defining standards for T-SDN including ONF, OIF, IETF and ITU-T. Until now ONF and IETF are the main organization for T-SDN standardization. The main efforts are based on OpenFlow (ONF) and GMPLS (IETF), and recently the work on the transport APIs is gaining momentum. However there is a long run to have stable standards on T-SDN.

Fig. 31 presents a classification of the standardization efforts on T-SDN. This section starts introducing the main SDOs involved in T-SDN, then it summarizes the efforts following the classification of Fig. 31, and then it presents a Global T-SDN demonstration from OIF and ONF using HT-SDN architecture.

A. Main Standardization bodies (SDO)s working on T-SDN

*1) **Open Networking Foundation (ONF):*** the ONF [1] is a young organization (2011) dedicated to the promotion and adoption of SDN and OpenFlow through open standards development. The ONF Architecture Framework Working Group developed a specification of the SDN architecture [24]. Such architecture supports the hierarchy of controllers, abstraction, virtualization and orchestration. Among other tasks related to transport SDN, ONF is also working on the definition of use cases, OpenFlow protocol extensions, NBIs and an information model for SDN.

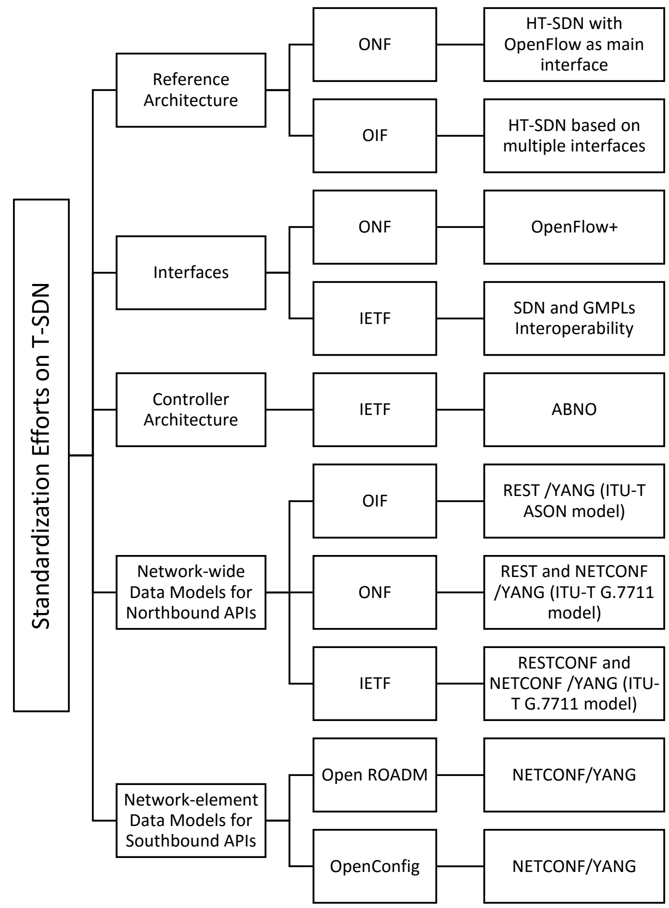

Fig. 31. Classification of the main standardization efforts of T-SDN

In 2013 ONF chartered the Optical Transport Working Group (OTWG) (renamed as Open Transport WG) to develop extension for OpenFlow-Switch protocol in order to support optical transport networks [1]. The first document from the OTWG was the OpenFlow-enabled Transport SDN [38], released in May 2014, where they proposed a target reference architecture and presented four T-SDN use cases, including bandwidth on demand, private optical networks, optical virtual private networks (O-VPN)s, and IP/MPLS plus transport optimization.

*2) **Optical Internetworking Forum (OIF):*** the OIF is an organization with more than 100 members, including service providers and optical vendors. It is dedicated to facilitate and improve: interoperability, cost-efficiency and robustness of optical internetworks. Optical internetworks are data networks composed of routers and data switches interconnected by optical networking elements [37].

Regarding T-SDN, in 2003 the OIF Carrier Working Group summarized high level requirements for deployment of transport SDN architectures, applications and services [59].

*3) **Internet Engineering Task Force (IETF)**:* the IETF is an open international community of network designers, operators, vendors, and researchers concerned with the evolution of Internet architecture and its smooth operation. IETF has developed the Generalized Multiprotocol Label Switching (GMPLS) [48], that use RSVP-TE [160] and OSPF [161] for distributed control of optical networks, that became the de-facto control plane for transport domains.

There are IETF Working Groups to cover several areas of SDN, for instance PCE Working Group (PCE-WG), Traffic Engineering Architecture and Signalling Working Group (TEAS-WG), Interface-to-the-Routing-System Working Group (I2RS-WG), Common Control and Measurement Plane Working Group (CCAMP-WG), and SDN Research Group (SDNRG).

- The PCE-WG is very active in providing remote control of PCE-based architectures using PCEP [30] and BGP-LS [31] (sections IX-C2).
- I2RS-WG is an IETF project with a different view of SDN, where a centralized control plane or an application layer optimizes routing decisions of the traditional routing system distributed control plane [191]. I2RS defines a standard, programmable and asynchronous interface to the state of Internet routing infrastructure (e.g. OSPF, IS-IS, and BGP), for application-based operations. Thus, the application layer can take some routing decisions for specific flows, while leaving traditional routing protocols to manage the rules for other flows. However, the I2RS is still in a early-stage, with a slow development pace.
- Abstraction and Control of TE Networks (ACTN) is a technology under development by the IETF TEAS-WG for virtualization of multi-layer and multi-domain transport networks [192]–[194]. ACTN refers to the set of virtual network operations needed to orchestrate, control and manage large-scale multi-domain TE networks [195]. ACTN reuses GMPLS/ASON and PCE architectures to provide virtualization of transport networks.
- The SDN research group proposed a modular and multi-domain SDN orchestration architecture called the PCE-Based Architecture for Application-Based Network Operations (ABNO) RFC 7491 [117] (section IX-D). ABNO is based on a pool of existing IETF standard blocks.

*4) **Standardization sector of International Telecommunication Union (ITU-T)**:* ITU-T is the standardization sector of International Telecommunication Union (ITU) [150]. The ITU-T Joint Coordination Activity on SDN (JCA-SDN) published a roadmap to keep an up-to-date information on all standardization activities on SDN [196].

The ITU-T Study Group 15 (SG15) is studying Transport aspects of SDN in close alignment with the ONF. The SG15 is working on two drafts Recommendations: "Architecture for SDN control of Transport Networks", and "Common Control Aspects" of the interaction between the ASON control plane, SDN control plane, management plane and transport data plane [196].

B. Standardization of T-SDN Reference Architectures

*1) **ONF Reference Architecture**:* Fig. 32 depicts the ONF reference architecture, which is based on a hierarchy of controllers that communicates between each other by means of the OpenFlow protocol. The ONF reference architecture includes two Southbound interfaces both using OpenFlow:

- Control Data Plane Interface (CDPI): it can be based either on OpenFlow or on other protocols. Like the SBI, the CDPI is used between controllers and network elements.
- Control Virtual Network Interface (CVNI): it is based on OpenFlow, and is used for interaction among controllers in a hierarchical control plane architecture. The CVNI is also intended to interface virtual controllers.

The CVNI allows, for instance a client or Global controller to interact with a virtual representation/abstraction of the network provided by a domain (child) controllers. REST-based NBIs are envisioned for communication with the application plane and the control plane (see section IX-E2 for more details on the ONF transport APIs). However, the global controller also provides a virtualization layer for client controllers that communicates with it using the CVNI (OpenFlow). The domain controller can be a legacy control plane over network elements that do not support OpenFlow. There are two approaches to represent the network: the abstract switch and the abstract link. The abstract switch (or big switch) is simpler but offers less visibility and control over the domain. The abstract link provides a topology with virtual nodes and links, rising the complexity but increasing visibility and control over the low level domain to the parent controller.

*2) **OIF Reference Architecture**:* in 2015 the OIF published a framework document [37] that identifies critical interfaces and components for transport SDN. The OIF framework is based on the hierarchical architecture (HT-SDN) and on the ITU-T ASON control plane model. Fig. 33 depicts the reference architecture envisioned by the OIF. From the Fig. 33, it is clear that OIF (as well as ONF) supports a hierarchical control plane architecture for transport SDN. The hierarchical architecture is more suitable for the multi-domain and multi-technology nature of carrier networks [37].

Even though the OIF framework for the implementation of transport SDN is based on ONF's reference architecture, there are some differences. For instance, on the application layer, a network orchestrator is the main application to serve internal purposes of the operator. While, the network orchestration of the ONF's reference architecture is managed by a Global (or parent) controller (see Fig. 32).

Contrary to ONF reference architecture, the OpenFlow is not the main protocol to use. In [37] are identified already existing protocols that can be reused at the SBI and NBI of transport SDN. For instance, the SBI of domain controllers should provide a variety of protocols to interact with the infrastructure layer (or data plane). In a transport network, the infrastructure layer may include brownfield domains that use a distributed control plane (GMPLS or ASON) or a centralized network management systems. Thus, the provisioning

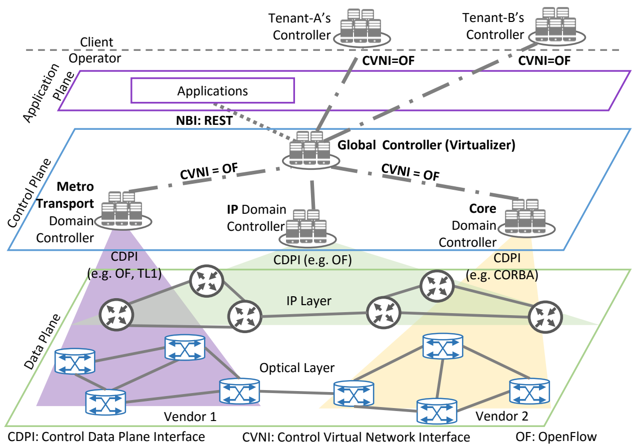

Fig. 32. Reference Hierarchical Control Architecture proposed by ONF [38].

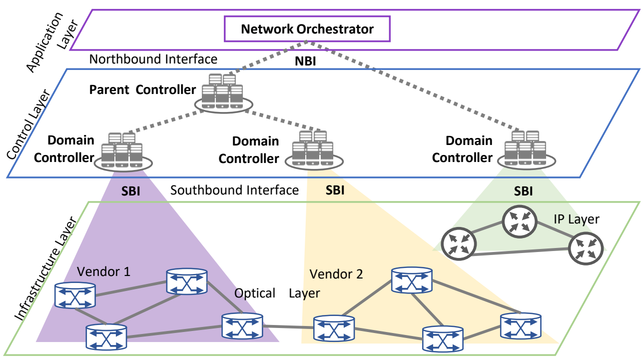

Fig. 33. HT-SDN architecture proposed by OIF [37].

of diverse SBI protocols allows to interact with: network elements, distributed control planes, and centralized network management systems.

C. Standardization of Southbound Interfaces (SBI)s

*1) **ONF OpenFlow Extensions for transport networks (OF+)**:* at the end of 2013 the ONF published the OpenFlow Switch Specification v1.4.0 [35] that introduced for the first time support for optical ports. A set of optical port properties allow the configuration and monitoring of frequency and power of transmitted and received optical signals. However, the optical extensions presented in [35] are very limited, and a large set of features and constraints of circuit-oriented and optical networks are expected to be included in OpenFlow specifications beyond v1.5 [197].

Based on the requirements analysis for Transport SDN/OpenFlow [40] and OIF carrier requirements on Transport SDN [59], the ONF-OTWG published a set of recommendation to support the control of optical transport networks [39]:

- Match and Actions extensions:
  – Match extensions for identifying signals at layer 0, using attributes of an OCh: Grid, Channel Spacing, center frequency, channel mask.
  – Match extensions for identifying signals at layer 1, using attributes of an ODUj/k: Optical channel Data Unit (ODU) type, ODU Tributary Slot, ODU Tributary Port Number signals.
  – No Action extension: the *SET_FIELD* mechanism is used to specify the attributes of the egress signals, thus without incurring into OpenFlow Action extension.
- Port attributes extension to identify port types at L0 and L1 of OTN standard interfaces.
- Adjacency discovery for OTN transport networks based on the in-band exchange of identifier information as defined in ITU-T G.7714.1 [150].

Future extensions considered by the ONF-OTGW includes: Operations Administration and Management (OAM)/Monitoring of optical network links, connection protection, multilayer connections and the use of OpenFlow among controllers in the CVNI.

2) IETF Standardization of SDN/GMPLS Interfaces: GMPLS is the most popular control plane for optical transport networks. Several demonstrations of dynamic optical transport networks used open source interfaces for centralized SDN/OpenFlow+ and proprietary interfaces and protocols for distributed GMPLS [198]. Multiple architectures and protocols have been proposed, each with different degree of integration and flexibility between the two control planes. In section V-B, we presented the first proposals of SDN and GMPLS control plane interoperation. For an overview of the interworking between GMPLS and OpenFlow we refer the reader to [199]. The PCE is the key element for centralization of path computation tasks over a GMPLS domain, while PCEP and BGP (BGP-LS) allows GMPLS and SDN interoperability. The SDN controller can use PCEP for provisioning of LSPs, and BGP-LS to get topology visibility.

- *Active Stateful PCE (AS-PCE) & PCE protocol (PCEP):* The PCE described in [49] was conceived to decouple and centralize the path computation from the distributed control plane of MPLS and GMPLS. The PCEP presented in the RFC 5440 [30] describes an open interface to communicate and manage LSPs remotely within MPLS and GMPLS control planes. The IETF PCE Working Group is very active with 26 RFCs and several drafts are under revision.
  According to RFC 5440 [30], the PCE can only compute a path upon receiving a request from a Path Computation Client (PCC) or other PCE, which is not compatible with the SDN paradigm. If the PCE is not able to trigger the creation of LSPs on demand, then it is not possible to achieve software-driven network control and operation.
  Therefore, the IETF-PCE Working Group is being developing PCEP extensions to boosts control, visibility and scalability of the PCE [200][120]. Such work is contributing to standardize the interoperability between SDN and GMPLS. The proposed PCEP extensions added three main capabilities to the PCE: stateful, active and hierarchical.
  – Stateful PCE (S-PCE) [200]: this extension allow the PCE to access the LSP-DB, that have the information of established LSPs. Thus, the PCE have knowledge of network state (stateful) and is able to improve path

computation tasks.

  - Active-Stateful PCE (AS-PCE) [120]: using the knowledge of network state, a PCE can control when and where to set up or modify the LSPs (active). Thus, the PCE can optimize the path computation of new and existent connections (active) based on the knowledge of current LSPs (stateful).
  - Hierarchical Active-Stateful PCE (HAS-PCE): the extensions of draft [120] allow a parent PCE or a network controller to compute paths across multiple domains (hierarchical). Each domain may be controlled by child PCEs.

  For instance, in [201] the AS-PCE was defined as the key element to allow: dynamic configuration of EONs, and standardization of SDN and GMPLS control planes interoperation. To broaden the concepts of AS-PCE for flexi-grid networks refer to [201].

- *North-Bound Distribution of Link-State and TE Information using BGP messages (BGP-LS):* In a hierarchical PCE architecture the RFC 5440 [30] did not define a procedure to gather topological information of multiple autonomous domains at the parent PCE. The extensions to the border gateway protocol (BGP) proposed in [31] allow a PCE (or a network controller) to access the TE databases (TED) of IGP area(s) or autonomous system(s). The BGP-LS allows to collect, filter (based on policies) and distribute with a PCE the LSDB and TED from IGP areas. Therefore, BGP-LS becomes part of the SBI offered by SDN controllers in order to get the topology of legacy domains.

D. Standardization of Controller Architecture (ABNO)

The RFC 7491 [117] describes an architecture to bring together many existing standard blocks defined by IETF to provide application-based network operations. ABNO framework represents an attempt to standardize the building blocks and internal interfaces of multi-vendor, multi-domain and multi-technology SDN controller by reusing IETF components. ABNO was conceived to allow the interoperability between legacy (e.g., IP/MPLS, GMPLS) and OpenFlow domains. It avoids vendor lock-in and provides support for the NMS and OSS.

Fig. 34 illustrates the generic ABNO architecture as proposed by RFC 7491 [117]. Depending on the use case, real implementation can have other interfaces or connections between different components. Components like the PCE, TED and LSP-DB can be replicated e.g., specific per-domain TEDs and coordinated PCEs for cross domain path computation.

The NMS and the OSS can introduce high level control, operation and management of the network. While the applications (e.g.,the transport network Orchestration) can directly trigger network operations. As depicted in Fig. 34 the NMS, OSS and the set of applications that are called the Application Service Coordinator (ASC), communicates with the Northbound interface (NBI) of the ABNO framework. It is worth noting that there is no standard definition for the NBI of the ABNO.

The ABNO controller (also called orchestration controller in [100]) is the central component of the architecture and provides the interfaces to the NMS, OSS and applications towards the network. The ABNO controller coordinates the work-flow among other ABNO blocks in alignment with NMS, OSS, and applications requirements and the current network conditions.

There are two main databases that are accessible for all the other blocks:

- The Traffic Engineering Database (TED): stores the topology information and can alternatively include capacity and status of the elements. The TED is mainly used for path computation.
- The Label Switch Path Database (LSP-DB): includes the paths and resources assigned to LSPs, that are currently or to be established in the network. The LSP-DB is mainly used for planning and optimization of LSPs.

Other components can also store critical information in databases e.g., policies, services, Shared Risk Link Group (SRLG), among others.

The PCE described in [49] handles constrained path computation over a network graph provided by the TED, and it is one of the main components of the ABNO framework.

The Virtual Topology Network Manager (VTNM) is defined in RFC 5212 [118], and it is in charge of multi-layer path provisioning. The VTNM manages the LSPs establishment at (possibly multiple) low-layer networks and feed logical links to higher-layer connections. The VTNM must establish connections in the server layer (e.g. Optical) to support connectivity in the client layer (e.g. IP). Thus, creating a virtual network topology for the client layer [202]. The VNTM can provide traffic demands adaptation capabilities, so that just enough capacity is created or released to the higher-layer network as needed.

The Provisioning Manager (PM) provides the appropriate interfaces for the establishment of LSPs in the network. In a hierarchy of controllers, the PM is able to interact with the control plane of the network domains (domain controllers e.g., GMPLS, AS-PCE, SDN) using the NBI of such control planes (PCEP [30], NETCONF RFC6241 [32], REST APIs). Additionally in a UCP approach, the PM is able to directly use the proper interfaces (ForCES RFC5810 [27], NETCONF RFC6241, OpenFlow) to interact with individual network devices.

Other blocks are described in RFC 7491 [117] like the Policy Agent that governs the communication of policies among the framework. Another important component described in [117] is the Operations, Administration and Maintenance (OAM) handler, in charge of the overall state of the network, reacting to alarms and potential faults to trigger the proper actions in other components for recovery, maintenance and elaboration of the reports.

The ABNO framework includes the Interface to the Routing System (I2RS), that is a work in progress described in draft [191].

The Application-Layer Traffic Optimization (ALTO) server can be also part of the ABNO framework, it provides abstract representations of the network to applications on top of

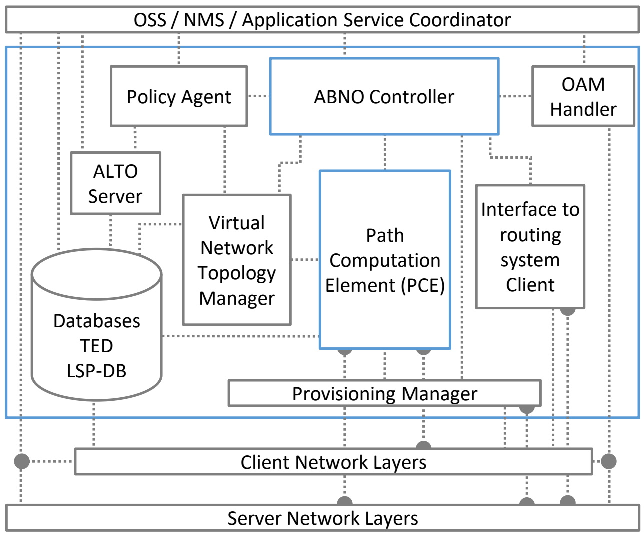

Fig. 34. Generic ABNO architecture from RFC 7491 [117].

ABNO. The abstractions are computed from the information stored in the TED, LSP-DB, policies and paths computation from the PCE, to simplify the route selection for the application layer traffic. The ALTO protocol is described in RFC 7285 [119].

E. Standardization of Network-wide Models for Northbound APIs

Open application programming interfaces (API)s to control and manage transport networks are a major topic of interest for network providers in order to foster programmability to lower CAPEX and OPEX of their multi-layer and multi-vendor transport infrastructure.

This subsection summarizes the standardization efforts on data models and transport APIs for network-wide services used at the Northbound interface of the T-SDN control plane.

The process to define the APIs starts with the definition of a UML information model, from which can be created the data model that will be supported by a protocol like NETCONF or using REST-like protocol running over HTTP to provide the APIs. Thus, we first define information model and the difference with data model:

- **Information model:** describes the managed objects (network device or system) at a conceptual level, including the relationships among the objects. The information model is implementation and transport protocol independent. For instance, the Unified Modeling Language (UML) is very common to create an information model.
- **Data model:** defines explicitly and precisely the structure, syntax and semantics of the managed objects's information model data. The data model should be complete and consistent. The data model includes protocol-specific rules that explain how to map managed objects onto lower-level protocol constructs.

Programming interfaces towards network elements have been used for a long time by network management systems. The Network management system allows to configure network elements from a centralized entity, similar to the centralized control proposed by SDN. For instance the Simple Network Management Protocol (SNMP) was developed to provide a programmatic interface towards network devices in order to build smart management applications. However, back in 2002 it was already accepted that SNMP had failed as a network management protocol [203]. The Transaction Language 1 (TL1) and the Common Object Request Broker (CORBA) are two widely used management protocols in telecommunications Networks.

*1) **OIF Northbound TAPIs (based on ITU-ASON model)**:*

the OIF framework [37] employed the functional elements of the ITU-T ASON model to define the APIs. By offering an open API to the call and connection control, as well as the routing control (that provides access to network topology information and path computation) of the ASON model, it is possible to foster: acquisition of topology information, request of path computation and provision of new services. The OIF APIs are based on REST-like protocol and JSON encoding [37].

The following APIs were defined by the OIF framework for T-SDN:

- Interface to the Call Control or Service request: enables to retrieve connectivity services from the network, such as: creation, deletion, listing and query.
- Interface to Connection control: typically an internal interface used by the service interface to setup the connectivity, however external API can be added for the Connection control.
- Route query or Path computation interface: allows to request path computation and optimization prior to request establishment of connectivity service. Together with topology interface conform the interface to routing control.
- Network topology interface: enables listing and reading of topology objects directly from the CP, such as: vertex, edge end, edge, and edge end resource.
- Abstraction control APIs: support of virtualization and abstraction of network resources for specific services. Network abstraction is a representation where some of the topology details are not visible. While network virtualization means a subset of the network resources.
- Notification APIs: retrieves information (or reports) of events, such as alarms, performance monitoring threshold crossing, object creation/deletion, state change, attribute value.

A simplified version of the OIF APIs was implemented for the OIF-ONF Global Transport SDN Prototype Demonstration [8] (see section IX-A2). In the demo the following three APIs were defined and tested:

- Interface to the Call Control or Service request interface.
- Route query or Path computation interface.
- Network topology interface.

The service API allowed the same application to be tested across heterogeneous domains. It also allows multiple orchestrators to access the same set of controllers where each orchestrator have access to a subset of resources (virtual slice of the network). However, two key functionalities were not implemented: the virtualization network service and the inter domain link discovery function. For the demo, virtualization and inter domain link discovery were manually coordinated, and represent future works.

The OIF already identified that the key to boost inter-operability and programmability of transport SDN is the definition of standard Northbound APIs. The Common Information Model (CIM) is a necessity to improve information mapping, to allow consistent behavior and to unify the access to functions across protocols. A CIM for packet and circuit switched networks is being developed by the ONF Information Modeling project [33].

*2) **ONF Northbound TAPIs (based on CIM)**:* the definition of a CIM is the foundation to create standard APIs. IETF, ITU-T and ONF adopted the same core information model, ITU-T Recommendation G.7711 [122] and ONF-CIM [33] to boost convergence, interoperability and efficiency of models. The ONF-CIM provides a representation of data plane resources for the purpose of management and control.

The ONF-CIM has been developed through collaboration among ITU-T, TeleManagement Forum (TMF) and ONF, and it was published as ITU-T Recommendation G.7711 [122]. The ONF-CIM is based on the TMF Multi-Technology Operations Systems (MTOSI) [204]. MTOSI is an XML-based Operations System (OS)-to-OS open standards-based interface suite.

The ONF OTWG Transport API (TAPI) project is working in the specification of standard transport APIs. The ONF-TAPI maps to the objects described by the ONF Core Information Model (ONF-CIM) [33].

The ONF-TAPI project defined the functional requirements for the development of TAPIs [123]. Its main target is to drive the detailed UML information model specifications, from which YANG and JSON data models can be defined to generate SWAGGER APIs. SWAGGER is one of the most popular specifications for describing, producing, consuming, and visualizing REST APIs [205].

Apart from nodes and links the CIM models the network with the following 3 logical termination points:

- Node-end-points: they are related with the forwarding capabilities of the nodes, and represent the input/output points of the nodes;
- Service-end-points: they provide abstracted views towards the clients (e.g. an application). A single service-end-point can be mapped to multiple node-end-points;
- Connection-end-points: they define the enabled forwarding capabilities inside a node and are mapped to node-end-points.

Based on the OIF framework [37] the ONF-TAPI proposed a set of services to abstract the transport network control plane functions:

- Topology Service: is used to retrieve information about topologies, nodes, links and node-edge-point details.
- Connectivity Service: concerned with the creation and management of connections between service-end-points. Allows the creation of point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-multipoint (MP2MP) connectivity at layers 2, 1 and 0.
- Path Computation Service: request for computation and optimization of paths.
- Virtual Network Service: create, update, delete virtual network topologies between pairs of service-end-points.
- Notification Service: retrieves information (or reports) of events.

ONF has open sourced the following Open Source SDN (OSSDN) Repositories [206]:

- SNOWMASS project: contains models and code for TAPI, including the TAPI information model in UML, its mapping into YANG and JSON data models/schema, and SWAGGER REST APIs.
- Englewood project: aims at developing a set of software modules to prototype, test, validate and facilitate the deployment of ONF-TAPIs, in heterogeneous T-SDN environments over open-source controllers (ONOS or ODL), or proprietary platforms (vendor specific SDN controller or legacy NMSes).
- EAGLE project: maintains documents and code of the ONF Information Model Project. Provides open source code to auto-generate YANG model code from a UML code.

*3) **IETF Network-Services Models (based on CIM) and NETCONF/YANG**:* from the set of requirements defined in the workshop held by the Internet Architecture Board (IAB) on Network Management [203], the IETF developed the Network Configuration protocol (NETCONF) [32]. Later, in 2014, the Internet Engineering Steering Group (IESG), the responsible for technical management of IETF activities, recommended to no longer use SNMP and to use NETCONF/YANG for network reconfiguration. [5]

NETCONF provides a standard framework and a set of standard Remote Procedure Call (RPC) methods to manipulate the configuration of network devices. In NETCONF, the devices's configuration data, and the protocol data, are encoded with the Extensible Markup Language (XML). NETCONF is primarily transported over the Secure Shell Transport Layer Protocol (SSH).

NETCONF allows a network element to expose a standard API, which is very suitable for SDN environments. Nevertheless, NETCONF do not defines the way to express its payload. The YANG data modeling language is standardized in RFC 6020 [136]. YANG provides the means to define the content (both data and operations) carried via NETCONF [136]. YANG is a data modeling language used to model configuration and state data manipulated by the NETCONF: protocol, remote procedure calls, and notifications. YANG uses XML to represent the contents of the data stores.

While NETCONF allows a network element to expose a standard API, YANG represents the basis of SDN pro-

[5] 1)"IETF Working Groups are therefore encouraged to use the NETCONF/YANG standards for configuration, especially in new charters", 2) "SNMP MIB modules creating and modifying configuration state should only be produced by Working Groups in cases of clear utility and consensus to use SNMP write operations for configuration, and in consultation with the OPS ADs/MIB doctors." March 2, 2014. Writable MIB Module IESG Statement.

grammatic APIs implementation. YANG is becoming relevant beyond NETCONF, today YANG Data Models are also used for REST-based interfaces by encoding the YANG model using JavaScript Object Notation (JSON) text. JSON is a lightweight data-interchange format based on dictionary-like data structure. Draft [207] defines the encoding rules for representing a YANG Data Model as JSON text. Thus, YANG Data Models can be converted to JSON or XML to provide REST or NETCONF APIs.

REST is based on popular technologies such as HTML, JSON and XML that enables straightforward development with programming languages as Python, Java and C. Interaction is done through HTTP basic operations GET, POST, PUT and DELETE. REST is the most used paradigm for definition of APIs, it allows to reduce development times and provides multiple debugging tools. Thus, when compared to traditional bit-oriented protocol stacks, REST-like protocols improve development time and offer better debugging capabilities.

After the IESG recommendation to use NETCONF/YANG in 2014, there has been an explosion of YANG Data Models development in IETF [208], as a consequence, YANG Data Models are now the basis of APIs implementation (sections IX-E3 and IX-F).

The IETF NETCONF Data Modeling Language Working Group (netmod-WG) recognized the benefits of using a CIM as the foundation to develop purpose and protocol specific interfaces [209].

In the following we list some T-SDN related YANG Data Models proposed by different IETF Working Groups:

- TEAS-WG: representation and manipulation of technology agnostic Traffic Engineered (TE) Topologies [124], configuration and management of TE interfaces, tunnels and LSPs [125]. The TE Topology allows to define physical impairments, abstract topologies (virtual) and it is protocol independent. YANG data model for the Abstraction and Control of TE networks (ACTN) and Virtual Network (VN) operation [126].
- CCAMP-WG is working on the definition of the transport network YANG model to facilitate the deployment and operation of transport network open interfaces. An overview-like draft [127] presents YANG Models for the Northbound interface of a transport network controller including: requirements, functions, and a list of related YANG models. Draft [130] provides the model for Layer 0 WSONs for impairment-unaware routing and wavelength assignment (RWA), and [131] focuses on flexi-grid technology.
- PCE Working Group is also working in the definition of a YANG model for the management of PCEP [128].
- I2RS-WG: [129] describes the YANG model to manipulate Layer 1 network topologies (e.g. OTN).

YANG has been already adopted by industry-wide open management and control initiatives e.g., OpenDaylight (ODL) and Open Network Operating System (ONOS) (see section X).

F. Standardization of Network-element Models for Southbound APIs

When it comes to the Southbound, the definition of APIs focus on the creation of standard models to represent the heterogeneous network elements in the transport network. YANG provides a standard way for writing data models. In order to achieve open network automation, service providers are forming Working Groups to create standard YANG data models. In the following sub-sections we present the main efforts in T-SDN for definition of YANG data models to offer open APIs at the optical network elements level.

1) ***Open ROADM:*** is a recent initiative by AT&T to define open standards for a disaggregated *white-box* Reconfigurable Optical Add/Drop Multiplexers (ROADM) [210], so it can be dynamically managed by an SDN control plane. A *white-box* ROADM provides open APIs towards a transport SDN controller.

The disaggregated ROADM proposed in [132] is conformed by three optical functions: the ROADM switch (optical amplifiers, couplers, and wavelength selective switch), transponders and pluggable optics. Each functional element provides open standard-based APIs towards the T-SDN controller. The data models are written in YANG, thus transponders and ROADMs need to support NETCONF/YANG. OPEN ROADM defined the following three different models:

- Device model (vendor specific): provides a detailed view of the devices using a generic representation of transponders and ROADMs. ROADMs can be colorless-directionless or colorless-directionless-contentionless. The Device Model allows optical equipment vendors to fill a template to describe their devices. Failure identification is based on data collected from Device Models.
- Network model (vendor neutral): provides a generic and vendor independent representation of the network. It abstracts the Device Model (vendor-specific) to a generic representation. Path computation functions performed by an SDN controller is based on data from the Network Model.
- Service model: service representation based either on Network or Device Models.

The idea behind Device Models is to abstract the hardware internals to the SDN controller [133]. In the ROADM Network Model the mapping of physical to logical ports are identified by the Device Model and the controller only has read permission of the mapping information.

In the Device Model services are represented by connection objects that specifies the following information:

- ports of transponders and ROADMs traversed by the service;
- wavelengths and power targets for each of those ports.

2) ***OpenConfig:*** the OpenConfig Working Group [134] is an initiative from Google composed by technical contributors from a variety of network operators. By 2016 OpenConfig cover a broad set of companies: Google, AT&T, Microsoft, British Telecom, Facebook, Comcast, Verizon, Level3, Cox, Yahoo, Apple, Jive, Deutsche Telekom, Bell Canada, SK Telecom, Bloomberg and Netflix. OpenConfig focus on the development of vendor-neutral YANG data models for model-driven network management.

OpenConfig focuses on configuration, state parameters and performance monitoring at the network elements level. Some of the models already defined by OpenConfig are: BGP, BGP routing information base MPLS, Optical-Transport, Routing Policies, Telemetry, and VLAN [135].

Some of the modules defined by OpenConfig regarding optical devices are:

- OpenConfig-terminal-device: describes the terminal optics device model for management of terminal systems such as physical, pluggable client port and logical grooming elements;
- OpenConfig-wavelength-router: defines operational and state data for an optical transport line system node, or ROADM;
- OpenConfig-optical-amplifier: describes configuration and operational state data for optical amplifiers.

G. Global T-SDN Demonstration

1) Interoperability demonstration of ONF OpenFlow+ and OIF Northbound TAPI: in 2014 OIF and ONF joint efforts to test prototype ONF OpenFlow extensions for the SBI and prototype Northbound APIs for T-SDN [8]. The demonstrations included:

- nine vendors: ADVA, Alcatel-Lucent, Ciena, Coriant, FiberHome, Fujitsu, Huawei, NEC and ZTE;
- five Carriers: TELUS, Verizon (North America), China Mobile, China Telecom (Asia) and Deutsche Telecom (Europe);
- one consulting carrier: Orange;
- two consulting research institutions: KDDI R&D Lab, China Academy of Telecommunications research.

The infrastructure layer consisted of multi-domain, multi-vendor and multi-carrier scenario, composed by Ethernet and OTN (ODU and OCH) switches. The prototype implementation was based on the hierarchical reference architecture proposed by ONF [38] (see Fig. 32).

The OpenfFlow testing included two types of extensions:

- CDPI test case: interface between domain controllers and network elements. Successfully tested optical extensions to OpenFlow 1.3 (proposed by ONF [39]), that allow matching of optical fields and description of optical ports (for retrieval of switch capabilities). This extensions allowed to install and delete match tables for cross-connection in the switches, to establish connections across a single-domain.
- CVNI test case: interface between global controllers and domain controllers, were the last provides a virtualized network representation to the former (abstract switch or abstract link representation). The abstract switch (big switch representation) was successfully tested, while for the abstract link a scalability issue arose. Using abstract link case (topology abstraction using multiple virtual nodes and links) requires the parent controller to maintain OpenFlow sessions with each virtual switch, while the domain controller must translate the match entries in each virtual switch to corresponding actions in the network. This may produce a massive overhead in the presence of large networks.

Apart from OpenFlow, the domain controllers supported vendor-specific SBIs. The demonstration proved to be an effective approach to have domain controllers that provide diverse SBIs towards heterogeneous optical elements.

2) Interoperability demonstration of ONF TAPI: in 2016 OIF and ONF joined efforts again to perform a multi-vendor interoperability test of the ONF TAPIS [121]. The contributors where mainly vendors: Nokia, Juniper, Adva, Telefonica, Ciena, Corian, NEC; a carrier (Verizon), an orchestrator system vendor (Sedona) and a research institution (CTTC). This interoperability test demonstrated the potential of ONF TAPIs to enable orchestration of on-demand connectivity setup, control and monitoring across diverse multi-layer, multi-vendor, multi-carrier networks.

Only the Topology, connectivity and notification services of the ONF TAPI [123] where tested, using the following use cases: Multi-domain service provisioning, low latency L0 path for inter-data center connectivity, variable bandwidth paths and the mapping of NFV onto multi-vendor, multi-domain T-SDN.

In section XII-D we present the open issues that where identified after testing the ONF TAPI.

X. Main Open Source T-SDN -related projects

Open source has powered the innovation across many technology fields, and SDN is not the exception. SDN opened the door for Open Source projects in networking, fostering the innovation through experimentation and contribution of a growing SDN community. Such projects are filling the gap of slow standardization process on SDN technologies. There are over 30 SDN controllers on the market today, from open source projects to vendor proprietary platforms. However, in this paper we focused on the main open source SDN platforms that can support a control plane capable of managing large service provider networks.

- *OpenDaylight* [156]: designed to serve a broad set of use cases and end user types, but with a main focus on service provider, enterprise, and academic networks. ODL is already a common platform for vendors' solutions, and used in pioneering demonstrations.
- *Open Network Operating System (ONOS)* [211]: a relatively newer player that specifically focuses on carrier networks. ONOS is gaining large momentum among service providers and academy.

ODL and ONOS are both based on Java programming language and the Open Services Gateway initiative (OSGI) that provides high modularity and allows loading service specific bundles at runtime. They both support distributed architectures for improvement of scalability and reliability, and provide the largest set of features among the SDN controllers. Both ODL and ONOS are supported by the Linux Foundation (nonprofit organization enabling mass innovation through open source), that also hosts other open source networking efforts that are part of the software-defined transformation:

- Open Sourced Enhanced Control, Orchestration, Management and Policy (ECOMP) architecture [138]. Its goal is to support full automation and incrementally reduce dependencies on the Legacy OSS;
- Open Network Automation Platform (ONAP) Project: Merger of Open Source ECOMP [138] and OPEN-O [137];
- Open Platform for NFV (OPNFV): development and evolution of NFV components across various open source ecosystems [139];
- Open source Platform for Network Data Analytics (PNDA) a big data analytics platform for networks and services [140].

A. Carrier-grade T-SDN Controller Platforms

1) ***OpenDaylight (ODL):*** is an Open Source Software project under the Linux Foundation founded by industry leaders in 2013 [156]. The initial aim of ODL is to accelerate SDN development and industry adoption, through the creation of a common industry supported controller platform. Some of the companies contributing to ODL development are: Cisco, Juniper Networks, VMware, Microsoft, and Ericsson. At the moment four releases are available: Hydrogen (Feb, 2014), Helium (Oct, 2014), Lithium (June 2015) and Beryllium that was released in February 2016.

ODL was the first controller that provided a framework to implement control and management services for heterogeneous multi-vendor networks.

OpenDaylight is becoming a de facto standard for SDN controllers with a growing support from the vendor industry that present OpenDaylight-based commercial products (e.g., ADVA, Brocade, Calient, Cisco, Ciena, Corian, Cyan, Ericcsson, HPE, Infinera, NEC, among others that are listed in the Solutions Provider Directory of ODL project [156]).

- *Control Layer:* the main components of ODL are service abstraction layer (SAL), the basic network functions, the enhanced network services, and the network abstraction (Policy/intent) service functions and pluggable modules. Service Abstraction Layer (SAL) represents a key bundle between service producers and consumers. Modules that provide services have to register their APIs to the SAL registry. Whenever a request from service consumer comes, SAL binds them into “contract”. There are two SAL architecture: application driven SAL and module driven SAL. ODL uses a model-driven SAL (MD-SAL) framework that maintains YANG data structures in a common data store and provides a messaging infrastructure (notifications and RPCs) that facilitates the incorporation of new applications and protocols.
  The following basic network functions are preconfigured with the controller: topology processing, OpenFlow statistics manager, OpenFlow switch manager, OpenFlow forwarding rules services, Layer 2 switch an host tracker. The enhanced network services are platform, protocol and vendor -specific services that provides ODL. ODL supports the largest amount of features among all controllers. Some of the enhanced network services ares: BGP-LS/PCEP, VTN (Virtual Tenant Network) and service function chaining.
  ODL supports four methods for configuration of policies and intents: Application Layer Traffic Optimization (ALTO), Group Based Policy (GBP) and Network Intent Composition (NIC).
- *Southbound interface:* ODL’s MD-SAL and the plug-in model allows to incrementally support multiple Southbound interfaces and protocols (vendor-specific and standard). For instance ODL supports OpenFlow, SNMP, NETCONF, OVSDB, BGP, PCEP, LISP and other vendor specific interfaces such as TL1 and CORBA. Commonly, one plugin includes connection, session and state managers, error and packet handler mechanism and set of basic services. Supported protocols communicate to the SAL.
- *Northbound interface:* ODL Controller exposes Northbound APIs to the upper layer applications using OSGi framework or bidirectional REST APIs. OpenDaylight APIs can be REST, RESTCONF, NETCONF and AMQP (Advanced Message Queuing Protocol). Through them, ODL exposes multiple functionalities to external applications. Each internal service in ODL can expose its own NBI using the REST API. On top of ODL APIs, there is a framework for Authentication, Authorization and Accounting (AAA).

2) ***Open Network Operating System (ONOS):*** it started in 2014 by the ON.Lab [212], is the first Open Source SDN controller to focus on service provider networks [211]. In 2015, ONOS joined the Linux Foundation to realize its full potential as an open source SDN and NFV project for service providers. The goals of ONOS as presented in the ONOS whitepaper [213] are: *1)* A control plane that ensures carrier grade features, i.e., scalable, high performance and five nines availability. *2)* Enable Web style agility. *3)* Help service providers migrate their existing networks to *white-boxes*. *4)* Lower service provider CapEx and OpEx. Such goals are tackled by the following set of features.

- *Distributed Core:* The adoption of a distributed core architecture is the key to meet carrier grade requirements, and the main difference with ODL (before the fourth version ODL did not provide clustering capabilities).
  ONOS maintains the centralized logical control of SDN, while running as a service on a cluster of servers, following the approach presented by Koponen et al. in [214].
  A cluster of controllers allows scalability by instantiating control capacity as needed. High availability is provided by a fast failover upon an ONOS server instance failure. An important difference between ODL and ONOS control layer architecture is at the SAL, i.e., the way they connect the protocol plugins with the network-specific functions: while ODL uses the MD-SAL, ONOS uses an architecture that is more AD-SAL. ONOS defines a set of subsystems providing several primary services available for the network applications [215]. For instance, while

the MD-SAL of ODL can store data for the YANG data models defined by the plugins, the AD-SAL is stateless.

- *Northbound APIs:* The ONOS Northbound APIs hide the complexity of the network and the distributed core. It provides two main abstractions to foster web style agility:
  - the intents framework, that allows to request services from the network such as policy statements and connectivity requirements, with high-level intent-based queries, without dealing with implementation details. ONOS intents can be by: network resource, constraints (bandwidth, optical frequency, and link type), criteria (packet header fields or patterns to describe a slice of traffic), and instructions (header field modifications, or output port);
  - the Network view, that is a consistent view of the elements in the network and their related states such as utilization and established connections. A specific API of this abstraction provides a graph representation.
- *Southbound APIs:* tt the ONOS core, network elements are described with generic objects. Using, device specific plugins (or Southbound providers), that adapt protocols to the Southbound API, ONOS can communicate with OpenFlow, NETCONF or other legacy-based protocols like PCEP and TL1. The Southbound API isolates ONOS distributed core from protocols and interface -specific plugins [215].
  ONOS is devoted to the use and creation of commodity hardware and *white-box* devices that can be fully controlled by open and standard Southbound APIs.

ONOS vision of the network follows the datacenter approach of using commodity hardware to bring economy and agility to carrier networks, while avoiding vendor lock-in. In consequence, packet and optical network elements are replaced with low cost *white-box* components, and the central offices are rearchitected as datacenters.

While *white-box* packet switches are already standardized and commercialized, the *white-box* optical network elements are in early-stages of development.

Among the optical network elements, the ROADM is the key component. ONOS foundation built the first disaggregated *white-box* ROADM using open source software and commodity hardware from Fujitsu, Ciena, Lumentum, Oplink, and Calient.

The ONOS *white-box* ROADM was disaggregated into three main functional components: transponders, Wavelength Selective Switch (WSS) and backplane. The ONOS *white-box* ROADM is based on NETCONF/YANG, and a list of APIs were defined for each component, in order to allow:

- device discovery;
- device capabilities detection (ports);
- device configuration (power, alarms, and transmission values);
- cross-connection provisioning (configuration of OXC matrix).

As future work, ONOS intent to follow the standardization efforts, that were recently started by the OPEN ROADM project [210] (section IX-F1) and the OpenConfig Working Group [134] (section IX-F2).

The ONOS project has work in progress with optical equipment vendors (Ciena, NEC, Huawei) and service providers (AT&T and SK Telecom). Among the Operator's use cases being developed for ONOS, the following two are of great importance for T-SDN.

1) *Operator Use Case: Packet Optical Convergence*
   As we have exposed in section V, SDN allows to gather and manage a converged packet/optical topology. This use case identified the need for providing multi-layer native support in ONOS.
   An overview of their achievement at OFC 2015 [187], included: converged packet/optical network graph abstraction, multi-layer PCE, restoration and protection mechanisms, and development of vendor-specific Southbound plugins to enable T-SDN in legacy equipment (providers). An important feature missing in [187] is the discovery of optical layer topology, thus topological information was manually configured.
   As part of this use case, an optical emulator platform called LINC-OE was developed [189]. LINC-OE a software switch that emulates *white-box* ROADMs with extended OpenFlow 1.3 (OpenFlow 1.3+) support. See section (VIII-C) for some extra details on LINC-OE. The multi-layer network emulation is composed using Mininet with OVS and LINC-OE elements.
   In 2015 a demonstration included Ciena and Fujitsu TL1 providers, and Huawei PCEP provider [211], for a real multi-vendor and multi-layer scenario.
2) *Operator Use Case: Central Office Re-architected as a Datacenter (CORD)*
   In order to bring datacenter economies and cloud agility in carrier networks, while avoiding vendor lock-in, CORD combines NFV, SDN and Cloud using commodity IT and network infrastructure. This use case elaborates in transforming the Point of Presence (POP) and Central Office (CO)s of operators into mini datacenters. Commodity servers are interconnected by a fabric constructed from *white-box* packet switches.
   CORD started as an ONOS use case, however it become a full open source project [216], which goal is to create: a reference open source architecture from commodity servers, *white-box* switches, disaggregated access technologies (e.g., vOLT, vBBU, vSG, vRouter, vPGW [6]), and open source software (e.g., OpenStack, Docker, ONOS, Extensible Cloud Operating System (XOS)). ONOS/CORD project cover residential, mobile and enterprise domains, each with specific features and configuration.

B. Carrier-grade SDN and NFV Orchestration Platforms

Network orchestration establishes the connectivity for a service, which might be: within a server, a data center or span multiple data centers (even in Points of presence or central

[6]Virtual Optical line termination (vOLT), Virtual Baseband Unit (vBBU), Virtual Serving Gateway (vSG), Virtual Packet Gateway (vPGW)

offices as with CORD [216]) and geographically separated customer premises. Additionally, with the advent of NFV and SFC, network orchestration is also required to interconnect functions within a service (see section VII-B1).

Service providers and carriers are looking to obtain system-wide service automation and are even looking forward to have a closed-loop automation that uses monitoring to provide dynamic resource allocation and failure management when and where needed. System-wide service automation involves not just services but also network and cloud orchestration: end-to-end service orchestration, network orchestration and cloud/resource (compute and storage) orchestration. Thus, service orchestration should be seen at a holistic system level and several technologies need to be coordinated: SDN[7] (including SDN for data centers and T-SDN for the transport network.), NFV, Cloud management software (OpenStack or VMware), APIs, and Big data.

This subsection briefly describes the main carrier-grade and Open Source platforms for system-wide service automation: Open-O [137] and Open Source Ecomp [217]. At the beginning of 2017, the Linux Foundation announced merger of Open Source ECOMP and OPEN-O to form new Open Network Automation Platform (ONAP) Project [218]. ONAP will allow to automate, design, orchestrate, and manage SDN and NFV services and virtual functions. Backed by leading operators and vendors [8] and the Linux Foundation, we believe that ONAP will shape the future of life-cycle SDN and NFV services orchestration. As in the case of OpenDaylight and ONOS; OPEN-O, ECOMP and ONAP will accelerate the implementation of SDN and NFV orchestration engines. These engines will incrementally automate the services and replace the legacy OSS systems.

*1) **OPEN-Orchestrator (Open-O):*** in early 2016, the Linux Foundation formed the OPEN-Orchestrator Project (OPEN-O) to develop the first open source software framework and orchestrator for agile operations of SDN and NFV [137]. By the beginning of 2017 the members of Open-O are: China Mobile, China Telecom, Ericsson, GigaSpaces, Huawei, Infoblox, Intel, HKT, Red Hat, Raisecom, Boco Inter-Telecom, ZTE, VMware, Canonical and CloudBase Solutions.

The mission of OPEN-O project is to: enable end-to-end service agility across SDN, NFV, and legacy networks via a unified orchestration platform supporting NFV orchestration (NFVO) and SDN orchestration [137]. The idea behind open-O is to allow *Any Service on Any Network*.

With its first Release (Sun), OPEN-O proposes an architecture based on microservices, that allows operators to incrementally transform their networks (including OSS/BSS) [219]. OPEN-O Sun Release is built around 5 main projects providing the main functionalities:

- Global Service Orchestrator (GS-O): responsible for providing end-to-end services;
- SDN-Orchestrator (SDN-O): responsible for connectivity services across SDN and legacy networks;
- NFV-Orchestrator (NFV-O): provides an ETSI MANO-aligned NFV orchestrator;
- Common Services: provides common services for other OPEN-O components: microservice bus, high availability services, driver manager, Log, authentication and authorization, and Protocol Stack (RESTCONF and NETCONF);
- Common Topology and Orchestration Specification for Cloud Applications (TOSCA): provides a TOSCA[9] parser, execution engine and model designer. The Open-O TOSCA services allows for on-boarding and orchestration of TOSCA descriptors. Such services are consumed by GS-O and NFV-O for orchestration of VNFs, network services and SFCs.

*2) **Open Sourced Enhanced Control, Orchestration, Management and Policy (ECOMP):*** AT&T, is perhaps the Carrier leading the implementation of T-SDN and NFV. In early 2017 AT&T announced that 50% of their network is already software-defined enabled, and are planning to virtualize 75% of its network by 2020. The convergence of SDN, NFV and cloud technology led AT&T (within AT&T's Domain 2.0 (D2) program) to the creation of Enhanced Control-Orchestration-Management and Policy (ECOMP) software. AT&T identified over 200 network functions that will be virtualized and controlled by ECOMP by 2020. ECOMP allows AT&T to deliver network on demand services, its goal is to support full automation and incrementally reduce dependencies on the Legacy OSS [217]. In late 2016 AT&T handed the ECOMP project to the Linux Foundation [138].

ECOMP represents the intelligence of how network functions are on-boarded and lifecycle managed on carrier optimized cloud infrastructure.

ECOMP uses two major architectural frameworks: *1)* Design Time Framework: provides a catalog-driven visual design & simulation tools, templates and catalogs, that allows to define resources, services and products. ECOMP relies on several domain-specific languages to design and create the models: YANG, TOSCA, and OpenStack Heat templates. *2)* Runtime Execution Framework: executes the logic programmed in the design time framework. Distribute policy enforcement and service templates among the ECOMP subsystems. There are five major software subsystems in ECOMP:

- Master Service Orchestrator (MSO): provides orchestration at a very high level, with an end to end view of the infrastructure, network, and application scopes.
- Controllers: multiple controller dedicated to specific resource domain: Cloud Infrastructure Controller, Network Controller and Applciation Controller.
- Data Collection, Analytics and Events (DCAE): supports closed loop control, trouble-shooting, and higher-level

[7]In section X-B we refer to SDN because for carriers and service providers, the end-to-end service orchestration spans data centers and transport networks and involves both SDN and T-SDN

[8]Founding Platinum members of ONAP include Amdocs, AT&T, Bell Canada, China Mobile, China Telecom, Cisco, Ericsson, GigaSpaces, Huawei, IBM, Intel, Nokia, Orange, Tech Mahindra, VMware and ZTE. Silver members of ONAP are ARM, BOCO Inter-Telecom, Canonical, China Unicom, Cloudbase Solutions, Metaswitch and Raisecom.

[9]TOSCA is an OASIS standard language that targets orchestration of cloud applications. TOSCA abstracts configuration data from hardware or services to make cloud services more interoperable and portable. Moreover carriers and service providers are using TOSCA to configure NFV services.

correlation for business and operations activities by monitoring Key Performance Indicators (KPIs).

- Active and Available Inventory (A&AI): collects real time data of cloud infrastructure and VNFs resources, services. It even collects data of the products customers buy. A&AI is implemented as a geo-redundant data base, updated by the controllers.
- Policy platform: allows to define and constrain, via high level abstractions, the behavior of ECOMP's modules and systems.

XI. Vendor Solutions

By the time of writing this survey, transport optical network vendors are at the inflexion point to change from a closed and lock-in prone solutions (*black-box* approach) towards more standard SDN and NFV -based solutions, that should open up control and visibility of their equipment. On the other hand, packet switched network vendors are ahead of SDN implementation and some of them are even offering commercial *white-box* products, mainly for the datacenter use case (e.g. Accton, Celestica, and Quanta Computer). *White-box* networking allows to use standard, off-the-shelf switches and routers, and to give full control of such devices to an SDN controller via OpenFlow or other standard SBIs.

The slower penetration of SDN into transport networks was expected. While in packet switched networks a standard data plane allows to easily embrace SDN and *white-box* approach, the heterogeneous optical transport networks represent a bigger challenge (as described in section III-C). However, commercial advances on SDN and NFV for transport networks are already making possible to disaggregate optical network functions that were normally integrated into a single *black-box*.

In the transport optical network industry, several vendors already have experimental OpenFlow extensions to support optical domains (mainly for Ethernet and OTN switches). Their main apparent focus is the interoperation between centralized SDN (domain) controller and a distributed GMPLS control plane, or a centralized NMS. The GMPLS control plane and/or NMS serves as the SDN enabler, while on top of it, the SDN domain controller provides NBI APIs to access service requests and topology functionalities, mainly down to the OTN layer. Such interoperation is possible through the protocols and techniques presented in sections IX-C2 and V-B. Thus, GMPLS control plane and NMSs are playing an important role for the T-SDN solutions proposed by vendors, following a *black-box* approach.

T-SDN must support multi-vendor interoperability in hierarchical architectures. Optical network equipment industry already offers orchestration platforms for network and service orchestration. For instance Ciena Blue Planet [220], Juniper NorthStar [221], Cisco Evolved Programmable Network (EPN) and Cisco Network Serivices Orchestration (NSO) [222]. A vendor-specific transport network orchestrator could lead to vendor lock-in, which carriers expect to avoid with SDN. A healthy hierarchical architecture should have a vendor-neutral Orchestrator, which can be developed at the application layer, or it can also be based on IETF ABNO [117]. The network orchestration market is expected to grow fast, the first vendor neutral transport network orchestration is the multi-layer and multi-vendor solution from Sedona [223], which is still tied to a list of vendors.

This section presents first the optical *black-box* solutions where the optical layer is under the control of a vendor-specific controller, NMS or abstraction layer. Then, are listed some commercial *white-box* solutions for fiber switches devices (e.g. Optical cross connects). Finally, Table VII and Table VIII gives a comparison of *black-box* and *white-box* optical solutions, respectively.

A. Optical Black-Box solutions

*1) **Nokia (Alcatel-Lucent)**:* the company's Network Services Platform includes NOKIA-ALU's SDN-based software with its 5260 service-aware management system, Service Router operating system and the 1830 Photonic Service Switch GMPLS routing engine algorithms from Bell Labs.

The Network Services Platform (NSP) combines an OpenDaylight-based controller, the Alcatel-Lucent 5620 Service Aware Manager (SAM), a service router operating system and photonic GMPLS service switch with routing engine algorithms [224].

The NSP integrates multi-layer automation and visibility from Layer 0 to Layer 3, through a simplified network abstraction.

NSP is composed of the Network Services Director (NSD) and the Network Resource Controller (NRC). NSP maintains a multilayer network abstraction model that provides information on topology, state, capacity, and utilization that is available for both the NSD and NRC.

The NRC provides centralized control capabilities over multiple layers and domains. The NRC is composed (together with the key performance indicator & analytics) by an optical-PCE, an IP/MPLS-PCE and a multi-layer and multi-domain PCE, each with specialized algorithms and functionalities for the specific domain.

NSP can work together with the Nuage Networks (Alcatel-Lucent subsidiary) solutions for virtualized networking in datacenter and remote enterprise as well as with the Alcatel-Lucent's CloudBand NFV orchestration.

NSP should support and allow multi-vendor operation by offering open Northbound Transport APIs (NB-TAPIs), definition of network abstraction model using standardized data modeling as well as services definition using common templates. The SBIs offered by NSP are PCEP, NETCONF, BGP-LS, OSPF, IS-IS/TE and OpenFlow.

To support optical transport domains (not a market ready solution) NSP relies on the vendor proprietary management system SAM, that provides a control interface towards the optical devices. The SAM exhibits a big switch abstraction that contains only the involved endpoints and services requirements, and perform the necessary routing and resource allocation algorithms.

The NSP reduces the service delivery complexity. However, route and/or resources cannot be controlled by the NSP and therefore neither by applications or orchestration engines on top of it.

More recently, NOKIA is participating into the OPEN-ROADM MSA [225], providing disaggregated *white-box* ROADMs and transponders that provide a NETCONF interface towards the SDN controller.

2) ***Ciena:*** in 2015 the SDN portfolio of Ciena was called Agility which targets a highly programmable and vendor-neutral network architecture [226]. Agility is composed by the Multilayer WAN Controller (MLWC) and a pool of network applications. The MLWC is based on OpenDaylight to foster openness and flexibility. MLWC added functionalities to target service provider networks and Ciena specific data plane. MLWC provides TL1 and CORBA SBIs for communication with Ciena optical infrastructure.

In the summer of 2015 Ciena acquired Cyan and together represent an important player in the SDN market with the Blue Planet orchestration suite. Blue Planet provides multi-layer and multi-vendor network virtualization, orchestration and management (even third-party OSS, NMS/EMS) software for end-to-end lifecycle service orchestration. It is built with a modular and programmable structure based on the concept of micro-services, that supports the control of multiple technologies and domains. Blue Planet integrates with third-party SDN controllers, for instance ONOS. Through ONOS Blue planet provides service orchestration capabilities to control and orchestrate *white-box* -based switching fabrics in central offices (as part of the ONOS CORD use case) [220] (see section IX-F1). Blue Planet also integrates with open stack and NFV MANO for datacenter and NFV service orchestration.

3) ***Cisco:*** Cisco SDN solution for service providers is called Cisco Open Network Architecture, which was composed in 2016 by [222]:

- Evolved Programmable Network (EPN), a multilayer control architecture that provides a unified view of physical and virtual device. It supports SDN, NFV, and open source technologies.
- Evolved Service Platform (ESP), serves as a modular orchestration engine for end-to-end service orchestration.
- and Application Centric Infrastructure (ACI), it is a policy-based architecture to optimize the application deployment lifecycle.
- Application Policy Infrastructure Controller (APIC): is the single point of automation and fabric element management for the ACI.

For T-SDN Cisco offers the nLight, a multi-layer controller for IP and optical convergence, that is part of the EPN. The nLight controller, built with the Cisco Open Networking Environment (ONE), is based on a client-server model that reuse GMPLS UNI [149]. The Cisco nLight Control Plane provides an intermediate solution between the overlay model (limited information exchange) and the peer model (large information exchange) by abstracting the information that each layer shares with the other.

The protection scheme exploits the visibility in both layers (IP and optical) to provide protection at the IP and restoration at the optical layer. This approach leads to decrease protection bandwidth at the optical layer and increase utilization of router interfaces.

More recently, Cisco presented the Network Services Orchestration (NSO), a NETCONF/YANG model-based Orchestration architecture that have some similarities with the SDN and NFV Orchestration Platforms described in section X-B. NSO main blocks are: Service manager, Device manager and Network Element Drivers. The model-based architecture allows NSO to be deployed in multi-vendor networks. YANG data models are used to define services, topologies and devices. Allowing to support specific SBIs and NBIs by defining specific data models.

4) ***Coriant:*** in [227] a multi-layer hierarchical control architecture with an extended OpenDaylight-based parent controller that orchestrates a Coriant proprietary transport controller (domain controller) was demonstrated by Coriant. The Coriant transport controller manage the optical domain and provides a big switch abstraction that hides the complexity of optical transport domains. It also provides a Northbound interface based on proprietary extension of OpenFlow v1.4 (OpenFlow+) [35] that adds capabilities to handle circuit-switching and to establish constraints such as latency. OpenFlow+ can be used by a parent controller or orchestration engine to interact with the abstract representation of the optical domain.

Using the REST APIs provided by OpenDaylight the authors of [227] demonstrated applications for establishment of end-to-end Ethernet services, Optical VPN and network optimization (bandwidth optimization and congestion control of packet optical services). The extension to the OpenDaylight included circuit and service managers, the REST APIs to access the new managers and the OpenFlow+ plugin. However, the big switch abstraction provided by the transport controller does not provide enough information to the parent/orchestration layer to perform multi-layer optimization.

The Authors of [227] presented the initial stage in the development of Coriant Transcend SDN solution portfolio, that includes: an orchestrator based on OpenDaylight, a Transport Controller that provides OpenFlow+, a suite of applications that exploit the data plane and also provides the integration with NFV [228]. The Transcend Transport controller provides NETCONF, SNMP and TL1 as Southbound protocols to interact with the optical domain.

The Coriant Transcend SDN Transport Controller spans optical DWDM layers, electrical ODU switching layers, or Carrier Ethernet (CE) and MPLS-TP based packet layers, to provide end-to-end service control. It offers an open and OIF standard-based REST NBI. It also supports a RESTCONF/YANG based topology and service APIs according to the IETF standard for east/west interfaces with other SDN controllers on the same control layer [228].

A Multi-layer Network Optimization and Migration Planning Service is offered by Coriant for multi-layer (Layer 0-3) optimization of network resources. However, for Coriant the orchestration and other business applications are customer specific.

*5) **Huawei**:* the Smart Network Controller (SNC) is the main component of Huawei end-to-end SDN solutions [229]. SNC was launched in 2013 as a carrier-class SDN controller for the IP layer. Huawei participated in multiple proof of concepts of multi-layer converged control of IP and optical domain in collaboration with Telefonica, China Telecom and ONOS [229].

In OFC 2015 the SNC was integrated with ONOS using the Northbound APIs provided by ONOS. With the Huawei implementation of SNC-ONOS controller was demonstrated a converged management of IP (Huawei NE series routers) and Optical (Huawei OptiX OSN series) layers of a real network. SNC-ONOS provided bandwidth-on-demand, transport network virtualization services and the provisioning of network level services that can be customized by tenants.

Huawei is getting ready for commercial deployment of T-SDN. In collaboration with China Telecom, Huawei announced in 2015 the consummation of what they called the first T-SDN deployment [229].

Huawei SNC-Transport controller (SNC-T) uses PCEP and OSPF as the SBI to communicate with a GMPLS-controlled transport devices. The SNC-T abstracts and manages transport devices and provides a NBI RESTful API for the application layer and the orchestration solution called NetMatrix. Netmatrix is responsible for multi-domain network synergy, it allows provisioning of services across multiple controllers including third-party controllers, datacenter and IP controllers. Netmatrix includes a MTOSI interface towards Huawei OSS for management functions. Huawei completed T-SDN multi-vendor interoperability tests organized by the OIF and ONF in September 2014. The T-SDN architecture of Huawei also provides applications for bandwidth on demand (BoD), IP + Optical optimization and optical VPNs.

To the best of our knowledge Huawei is fostering the integration of heterogeneous data plane with legacy and open interfaces using ONOS and SNC. Huawei, together with ONOS and ONF, are developing an open carrier-grade SDN ecosystem to foster the service providers SDN commercialization.

*6) **Infinera**:* among the optical equipment companies, Infinera made a step forward by opening up their Intelligent Transport Network products by means of the Open Transport Switch (OTS) software [230]. Thanks to the use of a OTS-agent, Infinera is the only company with optical solutions that provides APIs without their old NMS.

OTS was proposed in 2013 as an OpenFlow-enabled lightweight, virtual switch for abstraction and virtualization of the optical data plane [231]. It is composed by three modules: 1) a discovery agent for discovery and registration of SDN-controlled resources, 2) a control agent for monitoring and propagation of notifications and alarms to the Controller, and 3) dataplane agent for programming the NE datapaths. OTS is an open architecture that eliminates the relationship between optical bandwidth services and physical network constraints. The OTS approach allows the providers to use their own Control/Orchestration system after adding a specific OTS plugin.

The Infinera OTS was used in multiple demonstrations. A single multi-Layer SDN controller based on Floodlight and ESnet's On-Demand Secure Circuits and Advance Reservation System (OSCARS) application, exploits the OTS OpenFlow support for the optical layer and Brocade and NEC for the IP layer [230]. Telefonica and Infinera presented a proof-of-concept of the ABNO architecture for allowing Network-as-a-Service (NaaS) in a single-carrier, multi-vendor and multi-layer environment that includes Infinera Intelligent Transport Network and IP/MPLS layer [230].

The experience gained with the previous demonstrations led OTS to evolve from OpenFlow to pure Web 2.0 API. Thus, a controller or an orchestrator can use the Web 2.0 API for provisioning, discovery, monitoring, management and configuration functions of the transport layer data plane. OTS uses YANG modeling language to represent the network topology and resources.

More recently Infinera presented the Xceed Software Suite [230], that is based on OpenDaylight controller and provides a set of applications for multi-layer path computation and bandwidth calendaring. It reuses the OTS technology for network abstraction but includes YANG modeling and other open APIs. Infinera's Xceed is the first optical transport vendor solution to join the "Powered by OpenDaylight" program, that indicates the compliance with technical standards and quality for commercial products or services based on OpenDaylight.

*7) **Juniper**:* the Juniper's Northstar controller is an example of how the programmability and centralized SDN intelligence can be provided with legacy technologies to enable granular visibility and control of IP/MPLS flows in carrier networks [221]. This controller is based on an active stateful PCE architecture as defined in draft [200]. The Southbound interfaces included in the NorthStar are IS-IS, OSPF and BGP-LS for topology learning, and REST APIs for discovery of the optical topology. PCEP is used for installing or modifying paths and NETCONF/Yang is used as a management interface.

The Juniper NorthStar supports multi-vendor equipment through the cited standard Southbound interfaces, and it is specially compliant with Coriant and ADVA optical network elements. The controller provides Northbound REST APIs that allows additional centralized network infrastructure services and multi-domain orchestration.

Table VII presents a comparison of commercial T-SDN solutions.

B. Optical White-Box solutions

Apart from offering a standard open interface, a *white-box* component differs from a traditional one in its lack of intelligence, they completely depend on a centralized entity to create forwarding and routing tables via the SBI offered by the device. Thus, a standard open SBI must be carefully defined to have optical *white-box* network elements. Such SBI is under definition by ONF.

The optical extensions to OpenFlow defined in [39] allow the control of OTN switches and passive ROADMs (do not require any optical-to-electrical conversions) made by WSS and fiber switches. The management of fiber switches only

TABLE VII
VENDOR TRANSPORT SDN SOLUTIONS - OPTICAL *black-box*

Optical Vendor	Orchestration	Controller or Virtualization engine	SBI	NBI
ADVA	Ensemble Orch. ETSI MANO-compliant. End-to-end VNF and network-service-lifecycle management	Esemble Controller (ODL-based). RAYcontrol (GMPLS-based Controller). Multi-layer	GMPLS protocols, SNMP, TL1, OF, NETCONF	REST and Neutron APIs
Alcatel-Lucent	CloudBand NFV Orch.	NSP Controller. NMS + ODL-based. Layer 0-3 support. The NMS configures and manages the optical domain	NETCONF/Yang, PCEP, BGP-LS, OSPF, IS-IS/TE, OpenFlow, SNMP	NETCONF /YANG, REST APIs
Ciena	Blue Planet: Multi-Layer and Multi-vendor Orch., virtualization and management. End-to-end LSO	Multi-layer WAN Controller (MLWC). ODL-based. Layer 0-2 support	TL1 and CORBA (Optical devices). OF, NETCONF, SNMP, PCEP, BGP, OVSDB - Open SBI	REST APIs
Cisco	Network Service Orchestration (NSO)- multi-vendor support	nLight Controller (IP + Optical). GMPLS-based	PCEP, BGP-LS, RSVP (Optical devices). OF, OpFlex, NETCONF	REST APIs
Coriant	None (Orch. and other business apps. should be build by customers)	Transcend Transport Controller	NETCONF, SNMP, TL1	REST APIs, OpenFlow+
Ericsson	Management and Orchestration	Ericsson SDN Controller. ODL-based. Offers a Transport SDN domain controller Apps.	OF, OVSDB, BGP, NETCONF, PCEP, BGP-LS	REST APIs
Fujitsu	Virtuora Orch.	Virtuora network Controller. ODL-based. Multi-layer and Multi-vendor	NETCONF, TL1 and SNMP	REST APIs
Huawei	NetMatrix Orch.	Smart Network-Transport Controller (SNC-T) ONOS-based (IP + Optical)	PCEP and OSPF for GMPLS domain	REST APIs
Infinera	None	Xceed controller (ODl-based) and applications. Open Transport Switch (OTS) Abstraction and Virtualization Engine, compatible with third-party Controllers/ Orchestrators.	NETCONF, OF, XML, REST, OVSDB and Vendor specific interfaces	YANG and REST APIs
Juniper	Contrail Service Orch.	NorthStart Controller: L1-3 PCE-based controller	PCEP, NETCONF, OSPF-TE, BGP, BGP-LS, ISIS-TE, XMPP	REST APIs

involves configuration of cross connection matrices. Thus, since 2015 there are commercial *white-box* fiber switches for intra datacenter networks from Calient [232], Lumen [233] and Polatis [234].

The Open ROADM MSA, launched by AT&T, Ciena, Fujitsu, and Alcatel-Lucent, is working on the definition of interoperability specifications for ROADMs. The specifications consist of both Optical interoperability as well as YANG data models for disaggregated ROADMS composed by: switch (WSS, amplifiers and couplers), transponder and pluggable optics [210]. Each functional component provides an open standard API based on NETCONF (section IX-F1).

The ON.Lab ONOS foundation is working on the control plane interoperability for the Open ROADM [211]. ON.Lab built the first disaggregated ROADM using open source software and commodity hardware from Fujitsu, Ciena, Lumentum, Oplink, and Calient (section X-A2). The ROADM was disaggregated into three main functions: transponders, WSS and backplane.

In early 2016, Lumentum [235] showcased disaggregated *white-box* optical building blocks that includes: terminal amplifier, line amplifier, mux/demux, and ROADM-WSS for datacenter and metro edge networks.

Table VIII compares some of the available commercial *white-box* optical solutions, and the list of vendors participating in the OPEN-ROADM MSA.

XII. T-SDN ARCHITECTURE OPEN ISSUES

Control, management and orchestration of transport networks is a challenging multi-layer, multi-domain and multi-vendor problem. Such a wide problem led to multiple solutions; yet it seems that there may not be a single solution to fit all the scenarios. T-SDN is an open subject with many open issues to be debated within the research community and a very fast innovation pace. This section provides a list of areas in T-SDN architecture that are expected to need significant future work.

A. Control plane architecture

The architecture of the control plane for heterogeneous multi-domain transport networks is a major concern. Controllers are in continuous evolution to meet the requirements of service providers on availability, scalability and high performance. The Hierarchical control plane architecture (HT-SDN) seems to be the best choice for T-SDN.

The main open source controllers are based on similar internal architectures, but they continue to evolve. In the initial phases of SDN implementation it was important to support multiple Southbound interfaces to control green-field and brown-field domains.

For the future, the NBI has become more important. Precisely in order to enable HT-SDN, controllers should become interoperable at the NBI, so that different controllers can speak

TABLE VIII
VENDOR TRANSPORT SDN SOLUTIONS - OPTICAL *white-box*

Optical Vendor	Controller	SBI offered by the device	Type of device
Calient	ODL-based Optical Topology Management Controller	OpenFlow V1.3 and V1.4, TL1, SNMP and CORBA	Optical switch [232]
Ciena, Fujitsu & Nokia	Virtuora NC	NETCONF - OpenROADM YANG data models	WSS and Transponders [225]
Lumen	None*	OpenFlow V1.4, NETCONF, REST API, SNMP	Optical switch for datacenter networks [233]
Lumentum	None*	TL1, SNMP	Disaggregated *white-boxes* Terminal amplifier, line amplifier, mux/demux, and ROADM-WSS for datacenter and metro edge networks [235]
Polatis	None*	OpenFlow, NETCONF, SNMP, TL1, and SCPI (Standard Commands for Programmable Instruments)	Optical Switch for datacenter networks [234]
Fujitsu	Virtuora NC	NETCONF, SNMP, TL1, and SCPI	1FINITY Metro Data Center Interconnect [236]

None*: At the moment the vendor does not provide any controller

to the same orchestrator or parent controller. So, some level of agreement must be reached between controller makers about compatible network abstractions, common information models and interoperable APIs exposed at the NBI. Such requirements lead us to the next open issues.

B. Abstractions

To choose the right level of abstraction to understand and fully optimize the transport network resource utilization is the key to future T-SDN success. A good abstraction layer should have the right balance between: amount of information (complexity) and degree of provided control (flexibility). The optical layer features and impairments must be carefully considered when defining the level of abstraction that will be provided by APIs of T-SDN.

An interesting open issue is to analyze the trade-off between scalability and flexibility of provided abstractions (given by the visibility into the optical domain). In T-SDN the definition of a standard abstraction of the optical layer topology, impairments and complexity remains an open debate. These abstractions can happen at the Northbound and Southbound interfaces of the T-SDN control plane. Today many vendor solutions provide programmability of the optical data plane, however most of them provide an abstraction view that hides layer 0. Is this the right compromise between flexibility and complexity?

C. Common Information model

Definition of a common information model (CIM) is the base to build proper technology-agnostic standard abstractions at the Northbound and Southbound of SDN architecture. ONF is working on the development of a CIM [33].

From a technology-agnostic CIM, a protocol or technology-specific data model can be built. YANG is becoming the de-facto data-modeling language. The APIs are then provided by the data model using a specific format and protocol. The format for Northbound APIs is already accepted to be RESTful, however the data model is still under debate with ongoing research and standardization efforts. Currently, ONF is leading the definition of specifications for T-SDN related CIM [33].

D. Northbound Transport APIs

To ease the achievement of multi-vendor, multi-layer and multi-technology T-SDN implementation, the Northbound APIs should be based on a common abstraction model that support optical and IP network devices. The definition of open and well-defined Transport-APIS (TAPIs) is one of the foundation to enable end-to-end programmability of transport networks. So that, vendor-agnostic applications and network Orchestration systems, on top of the SDN hierarchical architecture can consume those APIs to deploy full service programmability across heterogeneous domains and layers. The domain controllers can be legacy, vendor-specific, or OpenFlow -based, and the SBIs can differ among the domains, however they should provide standard APIs either directly or using adaptation layers towards an application engine that run network Orchestration services.

Since 2016, ONF is working in the standardization of Open TAPIs [123]. The multi-vendor interoperability test [121] managed by OIF and ONF demonstrated the potential of ONF TAPIs to enable orchestration of on-demand connectivity setup, control and monitoring across diverse multi-layer, multi-vendor and multi-carrier networks. In the following are presented some of the issues identified in [121]:

- domain controllers tend to provide different abstractions and models of the network, so the parent (or multi-domain) controller or network orchestrator must perform appropriate mapping and service decomposition;
- domain controllers may differ in the API styles, so the parent controller needs to implement diverse API mechanisms based either on RPC or SCRUD (SCRUD Standard CRUD: Search, Create, Read, Update and Delete) envelopes;

- the TAPI need to allow the definition of node rules and constraints to enable standard definition of connectivity constraints;
- there should be a clear way to define the responsibilities among the hierarchy of controllers, for instance regarding the path computation services in multi-domain and multi-layer network;
- heterogeneity of transport networks lead to synchronization and verification issues of connectivity services due to differences in the vendor-specific implementation of optical nodes.

E. Intent-based networking

A hot topic in SDN is the Intent networking. There are different types of intents, but the main idea is to define a need instead of how to implement such need. ONOS [211], ODL [156], ONF-Boulder [206], and Openstack [157] have already intent-based NBIs. The intent is another level of abstraction that can be provided by SDN. Network operators can benefit from intent-based NBI in order to simplify: the development of network applications, the management and the control of their networks. An intent-based interface can be provided either by a T-SDN controller or by a layer between the T-SDN control plane and the application level.

For instance, the definition of an intent interface for creating virtual topologies and/or slices is still an open issue for transport networks.

An example of success is the Intelligent Network Deployment Intent Renderer Application (INDIRA), that provides an interface to the network based on natural language queries [237]. Such natural-language queries are then translated into more specific network actions that consume the TAPIs of a particular SDN controller. INDIRA goes beyond the intent-based NBI to provide an interactive network assistant to ease the configuration of network connections [237].

Intent-based networking and applications like INDIRA interactive assistant might be the starting point for networks-bots. In a very near future network-bots will interactively assist network operators to manage their networks, later the same bots will be able to provide self-healing and closed-loop automation capabilities to continuously optimize the network resources.

F. Southbound Interface (SBI)

The standardization of proper SBIs is the foundation of SDN. While in data center and campus networks OpenFlow is the main SBI, standard OpenFlow does not cover yet the full range of properties of optical layers, and there are many non-stable extensions (OpenFlow+ or OF+).

In T-SDN multiple SBI protocols coexist: NETCONF, PCEP, BGP-LS, SNMP, OpenFlow+, among others. Among them NETCONF/YANG has gained more traction due to its intrinsic adaptability and capacity to control the heterogeneous IP and optical network devices of transport networks. The technology-specific data models for using NETCONF and REST based protocols are still under development mainly by IETF and ONF. Some consortia like OpenConfig [134], and Open ROADM (with main focus on *white-box* ROADMs) are leading the definition of vendor-agnostic standard YANG models to benefit interoperability and programmability of transport networks devices. A very important missing component in today's controllers is generic support for NETCONF.

One issue is that the optical layer is in continuous evolution, and there are many different vendor-specific implementations. For instance, flexi-grid is a well standardized technology that lacks standard implementations by vendors and neither OpenConfig nor OpenROADM are yet capable of representing flexi-grid networks. For protocols like BGP-LS and PCEP there are still multiple IETF drafts to cope with the new advances at the optical layer and in the PCE architecture. Thus, either the protocol implementation is not updated or the controller does not support the new features of the protocol.

In consequence, there is a lot of work to be developed in order to have stable standard SBIs for transport networks.

G. Scalability and reliability of control plane

The concept of centralization of the control plane is at the basis of the SDN approach. But it should not be forgotten that controllers and orchestrators are software processes running inside a computer platform that have obviously limited resources. Therefore, as the controlled data-plane network gets larger and/or as the number of controlled flows increases, the SDN control plane starts facing the issue of scalability [238]. Moreover, centralization also may imply a reduction of the reliability, whenever single points of failures are generated in the system.

HT-SDN has proven to scale well with the use of recursive hierarchical controller architecture [121]. Apart from the hierarchical architecture, in order to further improve scalability and enhance reliability in large networks, the controllers are not implemented by a software running on a single machine, but by a cluster of machines (e.g. in the Cloud or hosted in the datacenter of the network operator) running a distributed version of the controller [214], [239]. Since the cluster has to behave in all circumstances as a single logical entity, some techniques allowing to preserve a constant synchronization between the multiple instances of the controller must be adopted. Usually, such techniques imply the use of some consensus protocol (e.g. the RAFT [240]), supported by an exchange of messages between the remote processes. The logical ports through which the peer instances exchange consensus messages are called West interfaces (to distinguish them from the NBI and the SBI). Some well-known controllers, such as ONOS [211], were developed to support a distributed architecture since the beginning, some others, such as ODL, are under improvement to achieve or consolidate cluster capability [156].

A more ambitious target, but an interesting opportunity for the future, would be to make different controllers to interoperate exploiting each its East/West interface [241]. Such a solution may be of help in the multi-carrier scenario, where controllers belonging to different network operators should communicate between each other, but on the other hand, disclosing a minimum set of information (e.g. and abstracted topology) controlled according to the inter-carrier policies.

The distributed implementation and its impact on control-plane performance [242], [243] is still a widely open topic, deserving a lot of additional research, especially focusing on the problem of delays (both in instance synchronization and from controllers to devices) and controller placement.

H. Transport Network Orchestration

The exact definition of Network Orchestration and the definition of the precise roles of a transport network Orchestrator is still an open issue. Also regarding this topic, we are in a context largely still uncovered by standardization. For instance the parent or multi-domain controller can provide network orchestration functionalities; as well, another application on top of the parent controller can provide transport network orchestration.

The T-SDN use cases are relevant when considering large services and user ecosystems across the WAN. The orchestration of network services across the WAN is an open issue, that has been approached faster by the industry than the academia. Today carriers and service providers need to orchestrate SDN and NFV systems to provide end-to-end services (End-to-End Orchestration). This problem involves network slicing and the mapping of Network Function Chaining (NFC) across multi-domain, multi-vendor and multi-layer networks.

Some optical network equipment vendors already offer network orchestration platforms, such as Ciena Blue Planet [220], Juniper NorthStar [221], Cisco Network Service Orchestration (NSO) [222]. However, carriers and service providers expect to have vendor-neutral Orchestration, which can be developed at the application layer.

Initial demonstrations of Northbound APIs-based orchestration across multi-vendor IP and Optical domains were presented in [102], [223], [244]. In [223] vendors created an adaptation layer to populate a common API defined by SEDONA systems. In both demonstrations the orchestrator did not have control over the physical layer.

A few Communication Service Providers are already building their own network Orchestration solutions. AT&T already created the Enhanced Control, Orchestration, Management and Policy (ECOMP) architecture [217]. ECOMP allows AT&T to deliver network on demand services, its goal is to support full automation and incrementally reduce dependencies on the Legacy OSS. A version of ECOMP was later open sourced by Linux Foundation [138].

In early 2016, the Linux Foundation formed the OPEN-Orchestrator Project (OPEN-O) to develop the first open source software framework and orchestrator for agile operations of SDN and NFV [137]. ONOS is also developing an orchestration platform for the CORD project to provide everything as a service (XaaS) exploiting SDN, micro-services and disaggregation using open source software and commodity hardware [216]. Later, the Linux foundation announced merger of Open Source ECOMP and OPEN-O to form new Open Network Automation Platform (ONAP) Project [218]. ONAP will allow to automate, design, orchestrate, and manage SDN and NFV services.

This context of uncertainty leave a degree of freedom on where to implement the control plane intelligence whether in the controller or in the orchestrator. The decision between controller or orchestrator is not irrelevant, especially as it may greatly influence inter-domain interoperability in the HT-SDN. The more functions are implemented in the controller, the more the entire control plane becomes locked into a specific controller, and the orchestrator will probably have hard time to make it interoperate with others. On the other hand, the simpler are the controllers, the more complex becomes the orchestrator, with the risk of being severely limited in scalability. This trade-off is somehow similar to what happens with controllers and data equipment, but at a higher abstraction layer. Up to date, the impression is that even very evolved and complex controllers such as ONOS and OpenDaylight have not reached yet a very stable stage of development, and thus a lot has to be implemented in the network orchestrator or application level to customize it to the operators needs.

I. Algorithms

The graph-based view of the network gathered at the control plane or at the orchestrator, allows the deployment of applications that run optimization algorithms for multidomain and multi-layer networks. Such algorithms, previously hidden inside vendor-specific solutions, can now be proposed by third-party entities. This will foster the formation of a scientific ecosystem, which surely benefits from academia especially in developing and demonstrating Operational research algorithms for: resource allocation, restoration, resiliency, disaster recovery, virtual network embedding, using a wide variety of architectures, for instance HT-SDN with and without NFV.

In this context there are elements of change compared to the past that will generate innovation also on the scientific ad mathematical side. In fact, a lot of work has been done in the past to develop algorithms tailored to the distributed control planes, such as GMPLS. Now, this previous work has to be retuned to the SDN centralized conception, however taking into account that centralization is at a logical level, while it has to be backed by distribution of process in a cluster for scalability and reliability (section XII-G). Algorithmically, it is surely an interesting challenge.

The *white-box* approach followed by the OPEN ROADM MSA and other players such as ONOS/CORD, moves L0 complexity management from the devices to the SDN control plane. Therefore, even at the lower control layers (domain controllers), some development of standard algorithms to cope with the analog nature of the data plane is necessary.

J. T-SDN, NFV and security

Service providers are implementing and testing NFV before T-SDN. However, the virtualization of network functions in a T-SDN enabled network is a promising area in very early-stage of development. T-SDN and NFV can be used to foster flexibility, agility and resiliency. For instance, a virtual controller can be scaled up/down or even relocated based on network conditions, and upon failures and disasters. The CORD project, is a powerful general-purpose platform to leverage T-SDN and NFV innovation [216].

There is a big need for Service Function Chaining (SFC) across transport networks, however SFC in T-SDN remains an open issue with few research contributions and a big need from industry. This includes the proposal of optimization problems for VNFs placement with VNF SFC across transport networks, and the deployment of SFC.

Security is a big issue in T-SDN. Multiple tenants can share the same infrastructure at different levels. It is still an open area to analyze how SDN principles can be applied to improve security of the tenants. For instance: network slices must provide isolation degrees, based on service level agreements, and authentication services to the clients. SDN architecture introduces both threads and solution capabilities on security due to the centralized control plane [7], [245]. Transport networks manage high capacity and cover long distances, thus are target of large-scale attacks. However, security for T-SDN scenarios has not been studied.

SDN, NFV and security are the basis of software defined WAN (SD-WAN), a technology that creates programmable overlay networks among enterprise customer premise entities (CPE). Ahead of service providers adoption of T-SDN and NFV, SD-WAN brings agility and programmability for enterprise networks. SD-WAN adopts SDN and NFV principles at the edge of transport networks, creating agile and programmable overlay networks. The adoption of T-SDN and NFV in telecommunication service providers networks will play a very important role towards the necessary evolution of their services.

K. Migration Path towards T-SDN

In principle, the SDN-architecture allows the control and the data planes to evolve separately. However, as we have shown in the previous sections, the two main obstacles that may jeopardize the fast deployment of T-SDN may be summarized as follows:

- the adoption of SDN requires data-plane devices to be SDN-enabled. Therefore, it represents an important investment for the operators that need to update their equipment. Such investment, in some cases may be hard to sustain;
- the heterogeneity and the complexity of equipment, and in particular of the photonic switches, makes it difficult to develop a once-for-all controller and obliges to implement expensive ad-hoc developments on the SBI.

These two issues are accompanied to the lack of standardization on the NBI that we have already discussed in the previous sections.

The migration from legacy transport networks to T-SDN is still an open issue that today faces telecommunication providers [**?**]. Hybrid T-SDN deployment is expected to be the most promising approach, as it allows to exploit legacy control-plane solutions (such as GMPLS/ASON and PCE architectures) without replacing the old equipments, and avoiding fully greenfield domains. In sections V and VI the control plane was able to use OpenFlow for isolated locations at the edge of the transport network (e.g. datacenter and campus), while OpenFlow+ and other SBIs (NETCONF, BGP-LS, PCEP) were used to interface the transport network elements. Only in sections V-A and VI-A, the control plane was based on a single SBI.

Segment routing represents another interesting technology ensuring a migration step towards T-SDN. It allows implementing SDN concepts into MPLS-based networks with multi-layer and multi-domain capabilities (see section VIII-B).

On the side of SBI and NBI standardization, the new concept gaining attention and popularity is the *white-box* optical device. The *white-box* approach is based on disaggregation of optical devices for leveraging modular building blocks with open interfaces (section XI-B). For instance, vendors like Lumentum already have commercial disaggregated *white-box* building blocks, including: terminal amplifier, line amplifier, mux/demux, and ROADM-WSS for datacenter and metro edge networks [235]. As consequence, standardization of disaggregated *white-box* ROADMs has just begun with the OPEN ROADM MSA activities [235]. Service providers are willing to deploy *white-box* devices into their optical transport networks, so the full capabilities of SDN and NFV can be exploited.

L. Big data and Machine learning for network analytics

Big data analytics first leverages analytical methods to obtain insights from traffic data. Second, gives guidance to traffic engineering applications to set the network policies [246]. Apart from traffic, other data can be analyzed, such as alarms, trouble tickets, and even the combination of non-network related information e.g., climate and social events.

There are some works that proposed the application of big data technologies to improve network control and management processes in datacenter environments [246]–[248]. However, the use of big data analytics in T-SDN remains an open issue. Sharing and analyzing information across different layers and domains of a transport network can improve the performance. The T-SDN control plane enables the collection of big data from all the different layers and domains. Nonetheless, the amount of data makes optimization and decision making so complex that traditional approaches are inadequate. This is surely another interesting challenge.

The industry seems to be moving faster regarding network analytics and artificial intelligence than academy. It was recently announced that one of Chinas leading telecom supplier of IT services plans to incorporate a machine learning engine to predict traffic patterns to improve automation of network resource allocations [249].

XIII. Concluding Remarks

A. Summary

This paper provides a comprehensive survey on Transport SDN. To recapitulate, the main points of T-SDN evolution can be summarized as follows.

To offer transport connectivity, a transport-network provider must control multi-domain, multi-layer and multi-vendor network in some cases composed by diverse optical technologies. This complexity seriously challenged the model of SDN as it was conceived for purely-packet and datacenter networks.

SDN had to be reviewed before extending it to transport networks.

Since most of the challenges to SDN derived from the optical technology, enabling SDN into optical networks (SDON) was the first step towards T-SDN. The process of effectively matching SDN and optical networks is still ongoing.

SDN principle of separating control plane from the forwarding devices relies on the creation of standard interfaces between these two planes, *i.e.*, standard Southbound interfaces. However, realizing standard SBIs means to uniformly abstract the heterogeneity of implementations. Transport-network equipment manufacturers (especially, optical-equipment vendors) have added value to their solutions by introducing innovative features and device capacities that differentiate their products from other vendors: that partially clashes with the concept of uniform abstraction.

A solution to this contradiction is the hierarchical control plane (HT-SDN) paradigm. An operator can manage multiple domains, where domain-specific controllers provide abstracted views towards higher order controllers or network orchestrator, using Northbound interfaces. HT-SDN is well supported by standardization bodies (e.g., ONF and OIF) and vendors, and has been the architectural choice for T-SDN research efforts. The separation of control and data plane allows the orchestration of end-to-end services across domains, using abstracted views provided by the Northbound APIs of the control plane. HT-SDN is also a promising solution to the co-existence of SDN with other legacy but widespread control-plane implementations, such as GMPLS. So, it is also the key to accelerate T-SDN deployment.

Transport network orchestration must be ideally vendor agnostic, avoiding vendor lock-in supported by standard NBIs. Thus, HT-SDN moves the issue of standardization from SBI to NBI: standard Northbound APIs (also called transport API or TAPI) are needed to leverage multi-domain and multi-vendor interoperability and independence of orchestrators from controllers. The TAPI must provide the right compromise between complexity and flexibility in the transport network. This approach can generate the ecosystem of network-software developers, perhaps led by open-source communities and projects, independent from vendors and network operators, competing to offer orchestration and application at lower prices than traditional vendors. At that point, the promise of T-SDN will honor the promise of being a cost-saving technology, as it happened in the datacenter world.

Another important aspect is the definition of common data models of transport network devices (optical and IP), that today is led by IETF and consortia from carriers and service providers called OpenConfig. Such common data models are defined using YANG and are focused on creating vendor-agnostic data models to provide a common Southbound transport API towards an even more open T-SDN architecture, where an Open Source controller can easily control devices form different vendors.

The last evolutionary step is the integration of SDN and NFV to foster the deployment of control plane and virtual data-plane network functionalities, adding all features (e.g. security) to provide services to final customers (business, residential, mobile, etc.), as we have explained in sections VII-B and XII.

B. The future of T-SDN

The future of T-SDN will be defined by a hierarchical and heterogeneous architecture, where data plane and control plane elements use and expose well-defined common interfaces. In the data plane of transport networks, legacy devices will last longer than in other segments, but a new wave of *white-box* devices will be increasingly introduced. The control plane will incrementally stitch together all the domains (access, metro, core) in the network including software-defined 5G system. The legacy OSS will be incrementally replaced by new NFV-SDN orchestration and automation systems based on Open Source projects such as ONAP. NFV and T-SDN will continue to foster dynamic and scalable deployment of services, with further integration of cloud and network infrastructure. More Open Networking projects will be adopted by vendors, carriers and service providers to build controllers, applications, orchestrators, and even the APIs. Last, but not least, big data and machine learning will incrementally provide network analytics and intelligence at the application plane (or at the orchestration level) to enhance the decision making of transport-network and service automation.

C. Final Comments

At the conclusion of this long journey through the early history of T-SDN, we can conclude by underlining a remarkable aspect of the adventure of this new technology. That is the extreme dynamism of the SDN concept that is able to quickly reshape attitudes of scientific and industrial world that previously seemed to be consolidated from ages.

Let us mention for instance the attitude towards standardization. In pre-SDN age, the concept of openness of a system was strictly related to coding everything into an official standard, possibly also with legal implications. With SDN, everybody started drifting from this attitude, to exploit quick-and-dirty solutions backed by a release of some widespread software (such as is, for instance, OpenFlow). But with T-SDN, early deployments soon revealed that the lack of standard was not appropriate to control multiple domains and so T-SDN is getting back to standardization. Also ONF is now supporting standard development by its transport Working Group, oriented to develop a common information model and standard interfaces (e.g. the standard TAPI effort, jointly with OIF). Today SDN is starting to boost the speed of standardization process, and SDOs are making strong commitments with Open Source communities in order to make this possible.

Similar comments can be made about central vs. distributed control: we started from the all-distributed paradigm of the Internet protocols, to move to the absolute centralization of SDN; but with T-SDN (and SDON) again there is a drift back to distribution (at least, per-domain), as testified by the HT-SDN architecture, and by coexistence with some distributed control plane such as MPLS and GMPLS/ASON through protocol extensions for PCEP, BGP-LS and Segment Routing.

This swinging between attempts to escape ossification and roadmap corrections after reality checks indicate that SDN can indeed prove to be a disruptive technology also for transport networks.

T-SDN is a reality: it cannot be clearer that there is a huge demand for T-SDN. It gained big momentum in the last years, with big efforts from academia, industry, standardization and open source communities. However, T-SDN is just in an initial stage, and there are many open issues to be solved. There is a long path before stable standards will rule the implementation of T-SDN. Therefore, it is really fascinating and exciting for researchers to be part of this evolution, but - more important - economically vital for transport-network operators.

ACKNOWLEDGMENT

The scouting activities that allowed the preparation of this paper have been conducted in the framework of a research contract between TIM S.p.A. and Politecnico di Milano concerning Transport SDN solutions for long-distance and optical transport networks.

REFERENCES

[1] (2015) Open Networking Foundation. [Online]. Available: http://www.opennetworking.org
[2] Open Networking Foundation, "Software-defined networking: The new norm for networks," *ONF White Paper*, Apr 2012.
[3] N. McKeown *et al.*, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.
[4] N. Gude *et al.*, "NOX: Towards an Operating System for Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, Jul 2008.
[5] Clean Slate Program, Standford University, "OpenFlow Switch Specification. Version 1.0.0 (Wire Protocol 0x01)," ONF Technical Specification 001, Dec 2009. [Online]. Available: http://archive.openflow.org/
[6] A. Lara, A. Kolasani, and B. Ramamurthy, "Network Innovation using OpenFlow: A Survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 493–512, First 2014.
[7] D. Kreutz *et al.*, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.
[8] Optical Internetworking Forum and Open Networking Foundation, "Global Transport SDN Prototype Demonstration," OIF/ONF White Paper, Oct 2014. [Online]. Available: http://www.oiforum.com/
[9] S. Jain *et al.*, "B4: Experience with a Globally-deployed Software Defined Wan," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, Aug 2013.
[10] F. Hu, Q. Hao, and K. Bao, "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 2181–2206, Fourthquarter 2014.
[11] H. Farhady, H. Lee, and A. Nakao, "Software-Defined Networking," *Comput. Netw.*, vol. 81, no. C, pp. 79–95, Apr 2015.
[12] A. Blenk *et al.*, "Survey on Network Virtualization Hypervisors for Software Defined Networking," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 655–685, Firstquarter 2016.
[13] S. M. *et al.*, "OpenFlow and Multi-layer Extensions: Overview and Next Steps," in *Software Defined Networking (EWSDN), 2012 European Workshop on*, Oct 2012, pp. 13–17.
[14] J.-P. Elbers and A. Autenrieth, "From static to software-defined optical networks," in *Optical Network Design and Modeling (ONDM), 2012 16th International Conference on*, April 2012, pp. 1–4.
[15] R. Alvizu and G. Maier, "Can open flow make transport networks smarter and dynamic? An overview on transport SDN," in *Smart Communications in Network Technologies (SaCoNeT), 2014 International Conference on*, June 2014, pp. 1–6.
[16] J. R. de Almeida Amazonas, G. Santos-Boada, and J. Sole-Pareta, "A critical review of OpenFlow/SDN-based networks," in *Transparent Optical Networks (ICTON), 2014 16th International Conference on*, July 2014, pp. 1–5.
[17] G. Zhang *et al.*, "A survey on ofdm-based elastic core optical networking," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 65–87, First 2013.
[18] N. Cvijetic, "SDN for Optical Access Networks," in *Advanced Photonics for Communications*. Optical Society of America, 2014, p. PM3C.4.
[19] N. Cvijetic *et al.*, "SDN and OpenFlow for Dynamic Flex-Grid Optical Access and Aggregation Networks," *Journal of Lightwave Technology*, vol. 32, no. 4, pp. 864–870, Feb 2014.
[20] A. Thyagaturu *et al.*, "Software Defined Optical Networks (SDONs): A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2016.
[21] L. Liu, "Sdn orchestration for dynamic end-to-end control of data center multi-domain optical networking," *China Communications*, vol. 12, no. 8, pp. 10–21, August 2015.
[22] A. Martinez and M. Yannuzzi and V. Lpez and D. Lpez and W. Ramrez and R. Serral-Graci and X. Masip-Bruin and M. Maciejewski and J. Altmann, "Network Management Challenges and Trends in Multi-Layer and Multi-Vendor Settings for Carrier-Grade Networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 2207–2230, Fourthquarter 2014.
[23] B. Nunes *et al.*, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp. 1617–1634, Third 2014.
[24] ONF Architecture Framework Working Group. (2014, Jun) SDN Architecture. SDN ARCH 1.0. [Online]. Available: https://www.opennetworking.org
[25] E. Haleplidis *et al.*, "Software-Defined Networking (SDN): Layers and Architecture Terminology," RFC 7426, Jan. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7426.txt
[26] B. Pfaff and B. Davie, "The Open vSwitch Database Management Protocol," RFC 7047, Dec. 2013. [Online]. Available: https://rfc-editor.org/rfc/rfc7047.txt
[27] A. Doria *et al.*, "Forwarding and Control Element Separation (ForCES) Protocol Specification," RFC 5810, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc5810.txt
[28] H. Song, "Protocol-oblivious Forwarding: Unleash the Power of SDN Through a Future-proof Forwarding Plane," in *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13, 2013, pp. 127–132.
[29] G. Bianchi *et al.*, "OpenState: Programming Platform-independent Stateful Openflow Applications Inside the Switch," *SIGCOMM Computer Communication Review*, vol. 44, no. 2, pp. 44–51, Apr 2014.
[30] A. Ayyangar *et al.*, "Path Computation Element (PCE) Communication Protocol (PCEP)," RFC 5440, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc5440.txt
[31] J. Medved *et al.*, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP," RFC 7752, mar 2016. [Online]. Available: https://rfc-editor.org/rfc/rfc7752.txt
[32] R. Enns *et al.*, "Network Configuration Protocol (NETCONF)," RFC 6241, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc6241.txt
[33] Open Networking Foundation. (2015, Nov) Common Information Model Overview. V1.1. [Online]. Available: https://www.opennetworking.org
[34] M. Casado, N. Foster, and A. Guha, "Abstractions for Software-defined Networks," *Communications of the ACM*, vol. 57, no. 10, pp. 86–95, Sep. 2014.
[35] Open Networking Foundation, "OpenFlow Switch Specification. Version 1.4.0 (Wire Protocol 0x05)," ONF Technical Specification-012, Oct 2013. [Online]. Available: https://www.opennetworking.org/
[36] H. Huang *et al.*, "Green datapath for tcam-based software-defined networks," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 194–201, November 2016.
[37] Optical Internetworking Forum, "Framework for Transport SDN: Components and APIs," OIF-FD-TRANSPORT-SDN-01.0, May 2015. [Online]. Available: http://www.oiforum.com/documents/framework-for-transport-sdn-components-and-apis/
[38] ONF-OTWG, "OpenFlow-enabled Transport SDN," ONF Solution Brief, May 2014. [Online]. Available: https://www.opennetworking.org
[39] ONF-OTWG. (2015, Mar) Optical Transport Protocol Extensions. V1.0. [Online]. Available: https://www.opennetworking.org
[40] ONF-OTWG. (2014, Aug) Requirements Analysis for Transport OpenFlow/SDN. V1.0. [Online]. Available: https://www.opennetworking.org
[41] ONF-OTWG. (2015, Oct) Wireless Transport SDN Proof of Concept White Paper. V1.0. [Online]. Available: https://www.opennetworking.org

[42] E. Rosen and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Mar. 2013. [Online]. Available: https://rfc-editor.org/rfc/rfc3031.txt

[43] D. Frost, L. Levrau, and M. Bocci, "MPLS Transport Profile User-to-Network and Network-to-Network Interfaces," RFC 6215, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc6215.txt

[44] *Interfaces for Optical Transport Network*, Recommendation G.709, ITU-T Std., Feb 2012.

[45] *Spectral grids for WDM applications: DWDM frequency grid*, Recommendation G.694.1, ITU-T Std., Feb 2012.

[46] I. Tomkos *et al.*, "A tutorial on the flexible optical networking paradigm: State of the art, trends, and research challenges," *Proceedings of the IEEE*, vol. 102, no. 9, pp. 1317–1337, Sept 2014.

[47] O. Gerstel *et al.*, "Elastic optical networking: a new dawn for the optical layer?" *Communications Magazine, IEEE*, vol. 50, no. 2, pp. s12–s20, Feb 2012.

[48] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," RFC 3945, Mar. 2013. [Online]. Available: https://rfc-editor.org/rfc/rfc3945.txt

[49] A. Farrel and G. R. Ash, "A Path Computation Element (PCE)-Based Architecture," RFC 4655, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc4655.txt

[50] R. Doverspike and J. Yates, "Optical network management and control," *Proceedings of the IEEE*, vol. 100, no. 5, pp. 1092–1104, May 2012.

[51] Open Networking Foundation, "SDN/OpenFlow Products," Open Networking Foundation, 2015. [Online]. Available: https://www.opennetworking.org

[52] S. Das, G. Parulkar, and N. McKeown, "Simple unified control for packet and circuit networks," in *IEEE/LEOS Summer Topical Meeting (LEOSST)*, July 2009, pp. 147–148.

[53] B. Mukherjee, *Optical WDM Networks*. Springer Science+Business Media, Inc., 2006.

[54] B. Collings, "New devices enabling software-defined optical networks," *IEEE Communications Magazine*, vol. 51, no. 3, pp. 66–71, March 2013.

[55] R. Younce, J. Larikova, and Y. Wang, "Engineering 400g for colorless-directionless-contentionless architecture in metro/regional networks [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 10, pp. A267–A273, Oct 2013.

[56] N. Amaya, G. Zervas, and D. Simeonidou, "Introducing node architecture flexibility for elastic optical networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 6, pp. 593–608, June 2013.

[57] M. Garrich *et al.*, "Experimental demonstration of function programmable add/drop architecture for roadms [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 2, pp. A335–A343, February 2015.

[58] P. Bhaumik *et al.*, "Software-defined optical networks (sdons): a survey," *Photonic Network Communications*, vol. 28, no. 1, pp. 4–18, 2014.

[59] Optical Internetworking Forum, "OIF Carrier WG Requirements on Transport Networks in SDN Architectures," OIF Technical Document, Sep 2013. [Online]. Available: http://www.oiforum.com/

[60] Z. Zhu *et al.*, "Demonstration of cooperative resource allocation in an openflow-controlled multidomain and multinational sd-eon testbed," *J. Lightwave Technol.*, vol. 33, no. 8, pp. 1508–1514, Apr 2015.

[61] S. Das, G. Parulkar, and N. McKeown, "Unifying Packet and Circuit Switched Networks," in *IEEE GLOBECOM Workshops*, Nov 2009, pp. 1–6.

[62] S. Das, "Extensions to the OpenFlow protocol in support of circuit switching," Addendum to OpenFlow protocol specification (v1. 0) - Circuit Switch Addendum v0. [Online]. Available: http://www.openflow.org/wk/images/8/81/OpenFlow_Circuit_Switch_Specification_v0.3.pdf

[63] V. R. Gudla *et al.*, "Experimental Demonstration of OpenFlow Control of Packet and Circuit Switches," in *Optical Fiber Communication Conference*. Optical Society of America, 2010, p. OTuG2.

[64] S. Das *et al.*, "Packet and Circuit Network Convergence with OpenFlow," in *Optical Fiber Communication Conference*. Optical Society of America, 2010, p. OTuG1.

[65] S. Das *et al.*, "Application-Aware Aggregation and Traffic Engineering in a Converged Packet-Circuit Network," in *Optical Fiber Communication Conference*. Optical Society of America, 2011, p. NThD3.

[66] S. Das. (2011) Aggregation on a Converged Packet-Circuit Network. [Online]. Available: http://openflowswitch.org/wk/index.php/Aggregation_on_a_Converged_Packet-Circuit_Network

[67] L. Liu *et al.*, "OpenFlow-based Wavelength Path Control in Transparent Optical Networks: a Proof-of-Concept Demonstration," in *European Conference and Exposition on Optical Communications (ECOC)*. Optical Society of America, 2011, p. Tu.5.K.2.

[68] L. Liu *et al.*, "Experimental validation and performance evaluation of OpenFlow-based wavelength path control in transparent optical networks," *Opt. Express*, vol. 19, no. 27, pp. 26 578–26 593, Dec 2011.

[69] D. Simeonidou, R. Nejabati, and S. Azodolmolky, "Enabling the future optical Internet with OpenFlow: A paradigm shift in providing intelligent optical network services," in *International Conference on Transparent Optical Networks (ICTON)*, June 2011, pp. 1–4.

[70] S. Azodolmolky *et al.*, "Integrated OpenFlow-GMPLS control plane: an overlay model for software defined packet over optical networks," *Opt. Express*, vol. 19, no. 26, pp. B421–B428, Dec 2011.

[71] L. Liu *et al.*, "First Field Trial of an OpenFlow-based Unified Control Plane for Multi-layer Multi-granularity Optical Networks," in *Optical Fiber Communication Conference (OFC)*, Mar 2012, p. PDP5D.2.

[72] M. Channegowda *et al.*, "Experimental Evaluation of Extended OpenFlow Deployment for High-Performance Optical Networks," in *European Conference and Exhibition on Optical Communication*. Optical Society of America, 2012, p. Tu.1.D.2.

[73] L. Liu *et al.*, "OpenSlice: An OpenFlow-based control plane for spectrum sliced elastic optical path networks," in *European Conference and Exhibition on Optical Communications (ECOC)*, Sept 2012, pp. 1–3.

[74] L. Liu *et al.*, "Interworking between OpenFlow and PCE for dynamic wavelength path control in multi-domain WSON," in *Optical Fiber Communication Conference (OFC)*, Mar 2012, pp. 1–3.

[75] L. Liu, T. Tsuritani, and I. Morita, "Experimental demonstration of OpenFlow/GMPLS interworking control plane for IP/DWDM multi-layer optical networks," in *International Conference on Transparent Optical Networks (ICTON)*, July 2012, pp. 1–4.

[76] M. Channegowda *et al.*, "Experimental demonstration of an OpenFlow based software-defined optical network employing packet, fixed and flexible DWDM grid technologies on an international multi-domain testbed," *Opt. Express*, vol. 21, no. 5, pp. 5487–5498, Mar 2013.

[77] L. Liu *et al.*, "Demonstration of a Dynamic Transparent Optical Network Employing Flexible Transmitters/Receivers Controlled by an OpenFlow-Stateless PCE Integrated Control Plane [Invited]," *J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A66–A75, Oct 2013.

[78] H. Y. Choi *et al.*, "Demonstration of BER-Adaptive WSON Employing Flexible Transmitter/Receiver With an Extended OpenFlow-Based Control Plane," *IEEE Photonics Technology Letters*, vol. 25, no. 2, pp. 119–121, Jan 2013.

[79] R. Casellas *et al.*, "An integrated stateful PCE/OpenFlow controller for the control and management of flexi-grid optical networks," in *Optical Fiber Communication Conference (OFC)*, Mar 2013, pp. 1–3.

[80] R. Casellas *et al.*, "Control and management of flexi-grid optical networks with an integrated stateful path computation element and OpenFlow controller [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 10, pp. A57–A65, Oct 2013.

[81] L. Liu *et al.*, "OpenSlice: an OpenFlow-based control plane for spectrum sliced elastic optical path networks," *Opt. Express*, vol. 21, no. 4, pp. 4194–4204, Feb 2013.

[82] J. Zhang *et al.*, "First demonstration of enhanced software defined networking (eSDN) over elastic grid (eGrid) optical networks for data center service migration," in *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, March 2013, pp. 1–3.

[83] M. Siqueira *et al.*, "An optical SDN Controller for Transport Network virtualization and autonomic operation," in *IEEE Globecom Workshops*, Dec 2013, pp. 1198–1203.

[84] J. Oliveira *et al.*, "Experimental testbed of reconfigurable flexgrid optical network with virtualized GMPLS control plane and autonomic controls towards SDN," in *SBMO/IEEE MTT-S International Microwave Optoelectronics Conference (IMOC)*, Aug 2013, pp. 1–5.

[85] M. Bahnasy, K. Idoudi, and H. Elbiaze, "Software-defined DWDM optical networks: OpenFlow and GMPLS experimental study," in *Global Communications Conference (GLOBECOM)*, Dec 2014, pp. 2173–2179.

[86] E. Magalhes *et al.*, "Global roadm-based spectrum equalizer in sdn architecture for qot optimization at dwdm networks," in *Optical Fiber Communications Conference and Exhibition (OFC), 2014*, March 2014, pp. 1–3.

[87] G. C. Santos et al, "Employment of ia-rwa in virtual optical networks using a pce implemented as a sdn application," in *Computer Networks*

and Distributed Systems (SBRC), 2014 Brazilian Symposium on, May 2014, pp. 207–213.
[88] M. Bahnasy, K. Idoudi and H. Elbiaze, “OpenFlow and GMPLS Unified Control Planes: Testbed Implementation and Comparative Study,” *J. Opt. Commun. Netw.*, vol. 7, no. 4, pp. 301–313, Apr 2015.
[89] J. Oliveira, et al, “Toward terabit autonomic optical networks based on a software defined adaptive/cognitive approach [invited],” *J. Opt. Commun. Netw.*, vol. 7, no. 3, pp. A421–A431, Mar 2015.
[90] M. Siqueira *et al.*, “Providing Optical Network as a Service with Policy-based Transport SDN,” *Journal of Network and Systems Management*, vol. 23, no. 2, pp. 360–373, 2015.
[91] Y. Zhao *et al.*, “Unified control system for heterogeneous networks with Software Defined Networking (SDN),” in *International ICST Conference on Communications and Networking in China (CHINACOM)*, Aug 2013, pp. 781–784.
[92] A. Mayoral *et al.*, “Experimental validation of automatic lightpath establishment integrating OpenDayLight SDN controller and Active Stateful PCE within the ADRENALINE testbed,” in *International Conference on Transparent Optical Networks (ICTON)*, July 2014, pp. 1–4.
[93] A. Mayoral *et al.*, “Integrated IT and network orchestration using OpenStack, OpenDaylight and active stateful PCE for intra and inter data center connectivity,” in *European Conference on Optical Communication (ECOC)*, Sept 2014, pp. 1–3.
[94] A. Aguado *et al.*, “ABNO: a feasible SDN approach for multi-vendor IP and optical networks,” in *Optical Fiber Communication Conference*, 2014, p. Th3I.5.
[95] R. Munoz *et al.*, “Experimental assessment of ABNO-based network orchestration of end-to-end multi-layer (OPS/OCS) provisioning across SDN/OpenFlow and GMPLS/PCE control domains,” in *European Conference on Optical Communication (ECOC)*, Sept 2014, pp. 1–3.
[96] Y. Yoshida *et al.*, “First international SDN-based Network Orchestration of Variable-capacity OPS over Programmable Flexi-grid EON,” in *Optical Fiber Communication Conference: Postdeadline Papers*. Optical Society of America, 2014, p. Th5A.2.
[97] R. Casellas *et al.*, “SDN orchestration of OpenFlow and GMPLS flexi-grid networks with a stateful hierarchical PCE [invited],” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 1, pp. A106–A117, Jan 2015.
[98] A. Aguado *et al.*, “ABNO: A Feasible SDN Approach for Multivendor IP and Optical Networks [Invited],” *J. Opt. Commun. Netw.*, vol. 7, no. 2, pp. A356–A362, Feb 2015.
[99] Y. Yoshida *et al.*, “SDN-Based Network Orchestration of Variable-Capacity Optical Packet Switching Network Over Programmable Flexi-Grid Elastic Optical Path Network,” *J. Lightwave Technol.*, vol. 33, no. 3, pp. 609–617, Feb 2015.
[100] R. M. noz *et al.*, “Transport Network Orchestration for End-to-End Multilayer Provisioning Across Heterogeneous SDN/OpenFlow and GMPLS/PCE Control Domains,” *J. Lightwave Technol.*, vol. 33, no. 8, pp. 1540–1548, Apr 2015.
[101] R. Vilalta, et al., “The need for a Control Orchestration Protocol in research projects on optical networking,” in *European Conference on Networks and Communications (EuCNC)*, Jun 2015, pp. 340–344.
[102] A. M. L. de Lerma et al., “First experimental demonstration of distributed cloud and heterogeneous network orchestration with a common transport api for e2e services with qos,” in *Optical Fiber Communication Conference*. Optical Society of America, 2016, p. Th1A.2.
[103] S. Azodolmolky *et al.*, “Optical FlowVisor: An OpenFlow-based Optical Network Virtualization Approach,” in *National Fiber Optic Engineers Conference*. Optical Society of America, Mar 2012, p. JTh2A.41.
[104] A. N. Patel *et al.*, “Distance-adaptive virtual network embedding in software-defined optical networks,” in *OptoElectronics and Communications Conference*, June 2013, pp. 1–2.
[105] Z. Ye *et al.*, “Virtual infrastructure embedding over software-defined flex-grid optical networks,” in *IEEE Globecom Workshops*, Dec 2013, pp. 1204–1209.
[106] A. Autenrieth *et al.*, “Evaluation of virtualization models for optical connectivity service providers,” in *International Conference on Optical Network Design and Modeling*, May 2014, pp. 264–268.
[107] R. Vilalta, et al, “Dynamic Multi-domain Virtual Optical Networks Deployment with Heterogeneous Control Domains,” in *Optical Fiber Communication Conference*. Optical Society of America, 2014, p. M3H.4.
[108] J. Zhang, Y. Zhao, and H. Yang, “Software Defined Networking: From dynamic lightpath provisioning to optical virtualization,” in *OptoElectronics and Communication Conference*, July 2014, pp. 691–693.
[109] R. Vilalta *et al.*, “Network virtualization controller for abstraction and control of openflow-enabled multi-tenant multi-technology transport networks,” in *Optical Fiber Communications Conference*, March 2015, pp. 1–3.
[110] Z. Ye *et al.*, “Survivable virtual infrastructure mapping with dedicated protection in transport software-defined networks [invited],” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 2, pp. A183–A189, February 2015.
[111] R. Muoz *et al.*, “SDN/NFV orchestration for dynamic deployment of virtual SDN controllers as VNF for multi-tenant optical networks,” in *Optical Fiber Communications Conference (OFC)*, March 2015, pp. 1–3.
[112] A. Aguado *et al.*, “Dynamic virtual network reconfiguration over sdn orchestrated multitechnology optical transport domains,” *J. Lightwave Technol.*, vol. 34, no. 8, pp. 1933–1938, April 2016.
[113] A. Hmaity *et al.*, “Virtual network function placement for resilient service chain provisioning,” in *International Workshop on Resilient Networks Design and Modeling (RNDM)*, Sept 2016, pp. 245–252.
[114] K. Bogineni et al, “SDN-NFV Reference Architecture,” Version 1.0, feb 2016. [Online]. Available: http://innovation.verizon.com/content/dam/vic/PDF/Verizon_SDN-NFV_Reference_Architecture.pdf
[115] R. Vilalta *et al.*, “Multitenant transport networks with sdn/nfv,” *J. Lightwave Technol.*, vol. 34, no. 6, pp. 1509–1515, March 2016.
[116] R. Martínez *et al.*, “Integrated SDN/NFV Orchestration for the Dynamic Deployment of Mobile Virtual Backhaul Networks over a Multilayer (Packet/Optical) Aggregation Infrastructure,” in *Optical Fiber Communication Conference*. Optical Society of America, 2016, p. Th1A.1.
[117] D. King and A. Farrel, “A PCE-Based Architecture for Application-Based Network Operations,” RFC 7491, Mar. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7491.txt
[118] K. Shiomoto *et al.*, “Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN),” RFC 5212, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc5212.txt
[119] R. Alimi, Y. Yang, and R. Penno, “Application-Layer Traffic Optimization (ALTO) Protocol,” RFC 7285, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7285.txt
[120] S. Sivabalan *et al.*, “PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model,” Internet Engineering Task Force, Internet-Draft draft-ietf-pce-pce-initiated-lsp-05, Apr. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-pce-pce-initiated-lsp-05
[121] Optical Internetworking Forum and Open Networking Foundation, “SDN Transport API Interoperability Demonstration,” OIF/ONF White Paper, Feb 2017. [Online]. Available: http://www.oiforum.com/
[122] *Generic protocol-neutral information model for transport resources*, Recommendation G.7711/Y.17022, ITU-T Std., Aug 2015.
[123] ONF-OTWG, “Functional Requirements for Transport API,” ONF Technical report: ONF TR-527, Jun 2016. [Online]. Available: https://www.opennetworking.org
[124] V. P. Beeram *et al.*, “YANG Data Model for TE Topologies,” Internet Engineering Task Force, Internet-Draft draft-ietf-teas-yang-te-topo-04, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-teas-yang-te-topo-04
[125] T. Saad *et al.*, “A YANG Data Model for Traffic Engineering Tunnels and Interfaces,” Internet Engineering Task Force, Internet-Draft draft-ietf-teas-yang-te-03, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-teas-yang-te-03
[126] Y. Lee *et al.*, “A Yang Data Model for ACTN VN Operation,” Internet Engineering Task Force, Internet-Draft draft-lee-teas-actn-vn-yang-00, Jul. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-lee-teas-actn-vn-yang-00
[127] X. Zhang *et al.*, “YANG Models for the Northbound Interface of a Transport Network Controller: Requirements, Functions, and a List of YANG Models,” Internet Engineering Task Force, Internet-Draft draft-zhang-ccamp-transport-ctrlnorth-yang-00, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-zhang-ccamp-transport-ctrlnorth-yang-00
[128] D. Dhody *et al.*, “A YANG Data Model for Path Computation Element Communications Protocol (PCEP),” Internet Engineering Task Force, Internet-Draft draft-pkd-pce-pcep-yang-05, Jan. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-pkd-pce-pcep-yang-05
[129] X. Zhang, B. Rao, and X. Liu, “A YANG Data Model for Layer 1 Network Topology,” Internet Engineering Task Force,

Internet-Draft draft-zhang-i2rs-l1-topo-yang-model-01, Oct. 2015, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-zhang-i2rs-l1-topo-yang-model-01

[130] Y. Lee *et al.*, "A Yang Data Model for WSON Optical Networks," Internet Engineering Task Force, Internet-Draft draft-ietf-ccamp-wson-yang-01, Apr. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-ccamp-wson-yang-01

[131] U. A. de Madrid *et al.*, "YANG data model for Flexi-Grid Optical Networks," Internet Engineering Task Force, Internet-Draft draft-vergara-ccamp-flexigrid-yang-02, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-vergara-ccamp-flexigrid-yang-02

[132] Open ROADM MSA. (2016, May) Open ROADM Overview. [Online]. Available: http://openroadm.org/download.html

[133] Open ROADM MSA. (2016, Jun) ROADM Network Model and Device Model. [Online]. Available: http://openroadm.org/download.html

[134] (2016) OpenConfig working Group. [Online]. Available: http://www.openconfig.net/

[135] (2016) GIT repositories of OpenConfig working Group. [Online]. Available: https://github.com/openconfig

[136] M. Bjorklund, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)," RFC 6020, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc6020.txt

[137] (2016) OPEN-Orchestrator Project. [Online]. Available: http://www.open-o.org

[138] (2016) Open ECOMP. [Online]. Available: http://www.openecomp.org/

[139] (2016) Open Platform for NFV (OPNFV). [Online]. Available: https://www.opnfv.org/

[140] (2016) Open source Platform for Network Data Analytics (PNDA). [Online]. Available: http://pnda.io/

[141] S. Das, G. Parulkar, and N. McKeown, "Rethinking IP Core Networks," *J. Opt. Commun. Netw.*, vol. 5, no. 12, pp. 1431–1442, Dec 2013.

[142] "OFELIA (OpenFlow in Europe: Linking Infrastructure and Applications)," FP7 EU project, 2013. [Online]. Available: http://www.fp7-ofelia.eu/

[143] X. Hesselbach, M. Huerta, and R. Alvizu, "Strategy for ip traffic support with qos in obs networks," in *International Conference on Transparent Optical Networks (ICTON)*, vol. 3, Jun 2008, pp. 126–129.

[144] L. Liu *et al.*, "Field Trial of an OpenFlow-Based Unified Control Plane for Multilayer Multigranularity Optical Switching Networks," *Journal of Lightwave Technology*, vol. 31, no. 4, pp. 506–514, Feb 2013.

[145] L. Liu *et al.*, "Software-defined fragmentation-aware elastic optical networks enabled by OpenFlow," in *European Conference and Exhibition on Optical Communication (ECOC)*, Sept 2013, pp. 1–3.

[146] M. Channegowda, R. Nejabati, and D. Simeonidou, "Software-Defined Optical Networks Technology and Infrastructure: Enabling Software-Defined Optical Network Operations [Invited]," *J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A274–A282, Oct 2013.

[147] L. Liu, R. Casellas, T. Tsuritani, I. Morita, R. Martinez and Raul Munoz, "Experimental demonstration of an OpenFlow/PCE integrated control plane for IP over translucent WSON with the assistance of a per-request-based dynamic topology server," in *European Conference and Exhibition on Optical Communications (ECOC)*, Sept 2012, pp. 1–3.

[148] L. Liu *et al.*, "Experimental demonstration of an OpenFlow/PCE integrated control plane for IP over translucent WSON with the assistance of a per-request-based dynamic topology server," *Opt. Express*, vol. 21, no. 4, pp. 4183–4193, Feb 2013.

[149] Y. Rekhter *et al.*, "Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model," RFC 4208, Mar. 2013. [Online]. Available: https://rfc-editor.org/rfc/rfc4208.txt

[150] (2015) Standardization sector of International Telecommunication Union (ITU-T). [Online]. Available: http://www.itu.int/ITU-T/

[151] S. Das, G. Parulkar, and N. McKeown, "Why OpenFlow/SDN Can Succeed Where GMPLS Failed," in *European Conference and Exhibition on Optical Communication (ECOC)*. Optical Society of America, 2012, p. Tu.1.D.1.

[152] A. Giorgetti *et al.*, "OpenFlow and PCE architectures in Wavelength Switched Optical Networks," in *International Conference on Optical Network Design and Modeling (ONDM)*, April 2012, pp. 1–6.

[153] (2008) NOX. [Online]. Available: http://www.noxrepo.org/

[154] M. Howard. (2015, Feb) The Evolution of SDN and NFV Orchestration. [Online]. Available: http://investor.juniper.net/files/doc_downloads/2015/May/Latest%20Resources/2000604-en_v001_h27j2q.pdf

[155] N. Figueira and R. Krishnan, "SDN Multi-Domain Orchestration and Control: Challenges and innovative future directions," in *International Conference on Computing, Networking and Communications (ICNC)*, Feb 2015, pp. 406–412.

[156] (2015) OpenDaylight Project. [Online]. Available: http://www.opendaylight.org

[157] (2015) OpenStack Foundation. [Online]. Available: http://www.openstack.org

[158] (2015) Project Floodlight. [Online]. Available: http://www.projectfloodlight.org

[159] (2016) Netphony Network Protocols Repository. [Online]. Available: https://github.com/telefonicaid/netphony-network-protocols

[160] L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions," RFC 3473, Mar. 2013. [Online]. Available: https://rfc-editor.org/rfc/rfc3473.txt

[161] Y. Rekhter and K. Kompella, "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)," RFC 4203, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc4203.txt

[162] Q. Wu *et al.*, "BGP-LS Traffic Engineering (TE) Metric Extensions," Internet Engineering Task Force, Internet-Draft draft-previdi-idr-bgpls-te-metric-extensions-01, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-previdi-idr-bgpls-te-metric-extensions-01

[163] (2015) TREMA. [Online]. Available: http://trema.github.io/trema/

[164] R. Sherwood *et al.*, "Flowvisor: A network virtualization layer," OpenFlow Technical Report TR-2009-1, Oct 2009. [Online]. Available: http://archive.openflow.org/

[165] R. Vilalta, et al, "Dynamic Multi-Domain Virtual Optical Network Deployment With Heterogeneous Control Domains [Invited]," *J. Opt. Commun. Netw.*, vol. 7, no. 1, pp. A135–A141, Jan 2015.

[166] Optical Internetworking Forum, "GS NFV 003-V1. 2.1-Network Function Virtualisation (NFV): Terminology for Main concepts in NFV," ETSI GS NFV 003 v1.2.1, Dec 2014. [Online]. Available: http://www.etsi.org/

[167] (2016) Open Source Mano (OSM). [Online]. Available: https://osm.etsi.org/

[168] R. Mijumbi *et al.*, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 236–262, Firstquarter 2016.

[169] D. Bhamare *et al.*, "A survey on service function chaining," *Journal of Network and Computer Applications*, vol. 75, pp. 138 – 155, 2016.

[170] R. Vilalta *et al.*, "Transport PCE network function virtualization," in *2014 The European Conference on Optical Communication (ECOC)*, Sept 2014, pp. 1–3.

[171] D. Zhang *et al.*, "Highly survivable software defined synergistic ip+optical transport networks," in *Optical Fiber Communication Conference*. Optical Society of America, 2014, p. Th3B.6.

[172] H. Yang *et al.*, "Performance evaluation of multi-stratum resources integrated resilience for software defined inter-data center interconnect," *Opt. Express*, vol. 23, no. 10, pp. 13 384–13 398, May 2015.

[173] L. Liu *et al.*, "Experimental demonstration of OpenFlow-based dynamic restoration in elastic optical networks on GENI testbed," in *European Conference on Optical Communication (ECOC)*, Sept 2014, pp. 1–3.

[174] L. Liu *et al.*, "Dynamic OpenFlow-Based Lightpath Restoration in Elastic Optical Networks on the GENI Testbed," *J. Lightwave Technol.*, vol. 33, no. 8, pp. 1531–1539, Apr 2015.

[175] (2015) GENI (Global Environment for Network Innovations). [Online]. Available: https://www.geni.net

[176] A. Giorgetti *et al.*, "Fast restoration in SDN-based flexible optical networks," in *Optical Fiber Communications Conference and Exhibition (OFC)*, March 2014, pp. 1–3.

[177] A. Giorgetti, F. Paolucci, F. Cugini and P. Castoldi, "Dynamic restoration with GMPLS and SDN control plane in elastic optical networks [Invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 2, pp. A174–A182, February 2015.

[178] H. Yang *et al.*, "Multipath protection for data center services in OpenFlow-based software defined elastic optical networks," *Optical Fiber Technology*, vol. 23, pp. 108 – 115, 2015.

[179] S. Savas *et al.*, "Backup reprovisioning with partial protection for disaster-survivable software-defined optical networks," *Photonic Network Communications*, pp. 1–10, 2015.

[180] C. Filsfils *et al.*, "Segment Routing Architecture," Internet Engineering Task Force, Internet-Draft draft-ietf-spring-segment-routing-07, Dec. 2015, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-spring-segment-routing-07

[181] A. Sgambelluri *et al.*, "First Demonstration of SDN-based Segment Routing in Multi-layer Networks," in *OFC*. Optical Society of America, 2015, p. Th1A.5.

[182] A. Sgambelluri *et al.*, "Experimental demonstration of multi-domain segment routing," in *ECOC*, Sept 2015, pp. 1–3.

[183] N. kukreja *et al.*, "Demonstration Of SDN-Based Orchestration For Multi-Domain Segment Routing Networks," in *ICTON*, Jul 2016.

[184] B. Lantz, B. Heller, and N. McKeown, "A Network in a Laptop: Rapid Prototyping for Software-defined Networks," in *ACM SIGCOMM Workshop on Hot Topics in Networks*, ser. Hotnets-IX. ACM, 2010, pp. 19:1–19:6.

[185] N. Handigol *et al.*, "Reproducible network experiments using container-based emulation," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '12. ACM, 2012, pp. 253–264.

[186] S. Azodolmolky *et al.*, "SONEP: A Software-Defined optical Network emulation platform," in *International Conference on Optical Network Design and Modeling*, May 2014, pp. 216–221.

[187] G. Parulkar, T. Tofigh, and M. D. Leenheer, "Sdn control of packet over optical networks," in *Optical Fiber Communications Conference and Exhibition (OFC)*, Mar 2015, pp. 1–27.

[188] (2016) Infoblox. [Online]. Available: https://www.infoblox.com/resources

[189] (2016) LINC-OE: LINC-Switch for optical emulation. [Online]. Available: https://github.com/FlowForwarding/LINC-Switch/blob/master/docs/optical_extension.md

[190] (2016) Netphony GMPLS Emulator Repository. [Online]. Available: https://github.com/telefonicaid/netphony-gmpls-emulator

[191] T. Nadeau *et al.*, "An Architecture for the Interface to the Routing System," Internet Engineering Task Force, Internet-Draft draft-ietf-i2rs-architecture-15, Apr. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-i2rs-architecture-15

[192] D. Ceccarelli and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks," Internet Engineering Task Force, Internet-Draft draft-ceccarelli-teas-actn-framework-02, Apr. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ceccarelli-teas-actn-framework-02

[193] D. Dhody *et al.*, "Packet Optical Integration (POI) Use Cases for Abstraction and Control of TE Networks (ACTN)," Internet Engineering Task Force, Internet-Draft draft-dhody-actn-poi-use-case-06, Apr. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-dhody-actn-poi-use-case-06

[194] Y. Lee *et al.*, "Information Model for Abstraction and Control of TE Networks (ACTN)," Internet Engineering Task Force, Internet-Draft draft-leebelotti-teas-actn-info-02, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-leebelotti-teas-actn-info-02

[195] Y. Lee *et al.*, "Requirements for Abstraction and Control of TE Networks," Internet Engineering Task Force, Internet-Draft draft-ietf-teas-actn-requirements-02, Apr. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-teas-actn-requirements-02

[196] *SDN standardization activity roadmap*, JCA-SDN-D-001 Rev.2, ITU-T Std., July 2015. [Online]. Available: http://www.itu.int/en/ITU-T/jca/sdn/

[197] Open Networking Foundation, "OpenFlow Switch Specification. Version 1.5.1 (Wire Protocol 0x06)," ONF Technical Specification-025, March 2015. [Online]. Available: https://www.opennetworking.org/

[198] R. Casellas *et al.*, "Interworking of GMPLS and OpenFlow domains: Overarching control of flexi grid optical networks," in *European Conference on Optical Communication (ECOC)*, Sept 2014, pp. 1–3.

[199] R. Casellas *et al.*, "Overarching Control of Flexi Grid Optical Networks: Interworking of GMPLS and OpenFlow Domains," *Journal of Lightwave Technology*, vol. 33, no. 5, pp. 1054–1062, March 2015.

[200] J. Medved *et al.*, "PCEP Extensions for Stateful PCE," Internet Engineering Task Force, Internet-Draft draft-ietf-pce-stateful-pce-14, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-pce-stateful-pce-14

[201] R. Munoz *et al.*, "Dynamic and Adaptive Control Plane Solutions for Flexi-Grid Optical Networks Based on Stateful PCE," *J. Lightwave Technol.*, vol. 32, no. 16, pp. 2703–2715, Aug 2014.

[202] T. Takeda, E. Oki, and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering," RFC 5623, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc5623.txt

[203] J. Schnwlder, "Overview of the 2002 IAB Network Management Workshop," RFC 3535, Mar. 2013. [Online]. Available: https://rfc-editor.org/rfc/rfc3535.txt

[204] (2016) TeleManagement Forum's (TMF) Multi-Technology Operations Systems (MTOSI). [Online]. Available: https://www.tmforum.org/mtosi/

[205] (2016) SWAGGER. [Online]. Available: http://swagger.io/

[206] (2016) Open Source SDN. [Online]. Available: http://opensourcesdn.org/

[207] L. Lhotka, "JSON Encoding of Data Modeled with YANG," Internet Engineering Task Force, Internet-Draft draft-ietf-netmod-yang-json-10, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-netmod-yang-json-10

[208] H.-K. Lam *et al.*, "Usage of IM for network topology to support TE Topology YANG Module Development," Internet Engineering Task Force, Internet-Draft draft-lam-teas-usage-info-model-net-topology-03, May 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-lam-teas-usage-info-model-net-topology-03

[209] M. Betts *et al.*, "Framework for Deriving Interface Data Schema from UML Information Models," Internet Engineering Task Force, Internet-Draft draft-betts-netmod-framework-data-schema-uml-03, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-betts-netmod-framework-data-schema-uml-03

[210] (2016) Open ROADM MSA. [Online]. Available: http://openroadm.org/home.html

[211] (2015) Open Network Operating System Foundation. [Online]. Available: http://onosproject.org/

[212] (2014) Open Networking Lab (ON.Lab). [Online]. Available: http://onlab.us/

[213] K. Bogineni et al, "Introducing ONOS - a SDN network operating system for Service Providers," Whitepaper, 2014. [Online]. Available: http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOS-final.pdf

[214] K. Teemu *et al.*, "Onix: A Distributed Control Platform for Large-scale Production Networks." in *OSDI*, vol. 10, 2010, pp. 1–6.

[215] P. Berde *et al.*, "ONOS: Towards an Open, Distributed SDN OS," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '14, 2014, pp. 1–6.

[216] (2016) CORD: Central Office Re-architected as a Datacenter. [Online]. Available: ttp://opencord.org/about/

[217] AT&T Inc., "Enhanced Control, Orchestration, Management, and Policy," 2016. [Online]. Available: http://about.att.com

[218] (2016) Open Network Automation Platform (ONAP) Project. [Online]. Available: https://www.onap.org/

[219] (2017) OPEN-Orchestrator Project. [Online]. Available: https://wiki.open-o.org/

[220] (2016) Ciena's Blue Planet division. [Online]. Available: http://www.blueplanet.com/

[221] (2016) Juniper Networks. [Online]. Available: http://www.juniper.net/

[222] (2016) Cisco Systems. [Online]. Available: http://www.cisco.com/

[223] (2016) Sedona Systems. [Online]. Available: http://sedonasys.com/

[224] (2016) Alcatel-Lucent. [Online]. Available: https://www.alcatel-lucent.com

[225] (2016) Open ROADM MSA. [Online]. Available: http://www.openroadm.org/

[226] (2016) Ciena Corporation. [Online]. Available: http://www.ciena.com/

[227] A. Felix *et al.*, "Multi-layer SDN on a commercial network control platform for packet optical networks," in *Optical Fiber Communications Conference and Exhibition (OFC)*, March 2014, pp. 1–3.

[228] (2016) Coriant. [Online]. Available: http://www.coriant.com/

[229] (2016) Huawei. [Online]. Available: http://www.huawei.com/en/

[230] (2016) Infinera Corporation. [Online]. Available: http://www.infinera.com/

[231] A. Sadasivarao *et al.*, "Open transport switch: A software defined networking architecture for transport networks," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13, 2013, pp. 115–120.

[232] (2016) Calient Technologies. [Online]. Available: http://www.calient.net/

[233] (2016) Lumen Networks. [Online]. Available: http://www.lumenetworks.com/

[234] (2016) Polatis. [Online]. Available: http://www.polatis.com/index.asp

[235] (2016) Lumentum. [Online]. Available: https://www.lumentum.com

[236] (2016) Fujitsu. [Online]. Available: http://www.fujitsu.com/

[237] A. Mercian *et al.*, “INDIRA: ‘Application Intent’ network assistant to configure SDN-based High Performance Scientific Networks,” in *Optical Fiber Communication Conference*, 2017, p. Tu3L.6. [Online]. Available: http://www.osapublishing.org/abstract.cfm?URI=OFC-2017-Tu3L.6
[238] R. Veisllari *et al.*, “Scalability analysis of SDN-controlled optical ring MAN with hybrid traffic,” in *EEE International Conference on Communications (ICC)*, June 2014, pp. 3283–3288.
[239] K. Phemius, M. Bouet, and J. Leguay, “Disco: Distributed multi-domain sdn controllers,” in *IEEE Network Operations and Management Symposium (NOMS)*, May 2014, pp. 1–4.
[240] D. Ongaro and J. Ousterhout, “In search of an understandable consensus algorithm,” in *USENIX Annual Technical Conference*, 2014, pp. 305–320.
[241] H. Yin *et al.*, “SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains,” Internet Engineering Task Force, Internet-Draft draft-yin-sdn-sdni-00, Dec. 2012, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-yin-sdn-sdni-00
[242] F. Benamrane, F. J. Ros, and M. B. Mamoun, “Synchronisation cost of multi-controller deployments in software-defined networks,” *International Journal of High Performance Computing and Networking*, vol. 9, no. 4, pp. 291–298, 2016.
[243] L. Schiff, S. Schmid, and P. Kuznetsov, “In-Band Synchronization for Distributed SDN Control Planes,” *ACM SIGCOMM Computer Communication Review*, vol. 46, no. 1, pp. 37–43, 2016.
[244] H. Ding *et al.*, “Experimental demonstration and assessment of multi-domain sdtn orchestration based on northbound api,” in *Asia Communications and Photonics Conference*. Optical Society of America, 2015, p. ASu4F.1.
[245] S. Scott-Hayward, G. O’Callaghan, and S. Sezer, “Sdn security: A survey,” in *IEEE SDN for Future Networks and Services (SDN4FNS)*, Nov 2013, pp. 1–7.
[246] L. Cui, F. R. Yu, and Q. Yan, “When big data meets software-defined networking: Sdn for big data and big data for sdn,” *IEEE Network*, vol. 30, no. 1, pp. 58–65, Jan 2016.
[247] G. Wang, T. Eugene, and A. Shaikh, “Programming Your Network at Run-time for Big Data Applications,” in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN ’12. ACM, 2012, pp. 103–108. [Online]. Available: http://doi.acm.org/10.1145/2342441.2342462
[248] J. Wu *et al.*, “Big data meet green challenges: Greening big data,” *IEEE Systems Journal*, vol. 10, no. 3, pp. 873–887, Sept 2016.
[249] (2017) BOCO Inter-Telecom. [Online]. Available: www.boco.com

Appendix

Acronyms

ABNO Application-Based Network Operations
ACI Application Centric Infrastructure
ACTN Abstraction and Control of TE Networks
AD-SAL API-Driven Service Abstraction Layer
ALTO Application-Layer Traffic Optimization
API Application Programming Interfaces
APIC Application Policy Infrastructure Controller
ASC Application Service Coordinator
ASE Amplified Spontaneous Emission
ASON Automatic Switched Optical Networks
BER Bit Error Rate
BGP-LS Link-State Distribution Using Border Gateway Protocol
BRPP Backup Reprovisioning with Partial Protection
BV-OXC Bandwidth Variable Optical cross-connect
BVT Bandwidth Variable optical Transponder
CapEx Capital Expenditure
CCAMP-WG Common Control and Measurement Plane Working Group
CDPI Control Data Plane Interface
CE Carrier Ethernet
CF Central Frequency
CIM Common Information Model
CLI Command Line Interface
CO Central Office
COP Control Orchestration Protocol
CORBA Common Object Request Broker Architecture
CORD Central Office Re-architected as a Datacenter
COST Ultra-High Capacity Optical Transmission Networks
COTS Commercial off-the-shelf
CP Control Plane
CTTC Centre Tecnol`ogic Telecomunicacions Catalunya
CVNI Control Virtual Network Interface
DAL Device and resource Abstraction Layer
DMT Discrete Multi-Tone
DP Data Plane
DWDM Dense Wavelength Division Multiplexing
ECOMP Enhanced Control-Orchestration-Management and Policy
EMS Element Management System
EON Elastic Optical Network
EPN Evolved Programmable Network
ERO Explicit Route Object
ESP Evolved Service Platform
ETSI European Telecommunications Standards Institute
FEC Forward Error Correction
FT-SDN flat/mesh T-SDN
GMPLS Generalized Multi-Protocol Label Switching **HAS-PCE** Hierarchical Active-Stateful PCE
HTTP HyperText Transfer Protocol
I2RS-WG Interface-to-the-Routing-System Working Group
IA-R Impairment-Aware Routing
IA-RWA Impairment-Aware Routing and Wavelength Assignment
IETF Internet Engineering Task Force
IS-IS Intermediate System to Intermediate System
ITU International Telecommunication Union
LLDP Link Layer Discovery Protocol
LSP Label Switched Path
LSP-DB Label Switched Path Database
MANO MANagement and Orchestration
MD-SAL Model-Driven Service Abstraction Layer
MLWC Multilayer WAN Controller
MPLS Multi-Protocol Label Switching
MTOSI Multi-Technology Operations System Interface
NaaS Network-as-a-Service
NB-API Northbound Application Programming Interfaces
NBI Northbound Interface
NE Network Element
NETCONF Network Configuration Protocol
NFV Network Function Virtualization
NFVI Network Function Virtualization Infrastructure
NMS Network Management System
NOS Network Operating System
NRC Network Resource Controller
NSD Network Services Director
NSFNet National Science Foundation Network
NSP Network Services Platform
OAM Operations Administration and Management

OBS Optical Burst Switching
OCC Optical Connection Controller
OCh OTN Optical Channel
OCS Optical Circuit Switching
ODL Opendaylight controller
ODU Optical channel Data Unit
OF OpenFlow
OF+ Extended OpenFlow
OF-GC OpenFlow-enabled GMPLS control plane
OFS OpenFlow Switch
OFV Optical Flow Visor
OIF Optical Internetworking Forum
ONF Open Networking Foundation
ONF-OTWG ONF Optical Transport Working Group
ONOS Open Network Operating System
OpEx Operational Expenses
OPS Optical Packet Switching
OSC Optical Supervisory Channel
OSCARS On-Demand Secure Circuits and Advance Reservation System
OSGI Open Services Gateway initiative
OSNR Optical Signal to Noise Ratio
OSPF Open Shortest Path First
OSPF Open Shortest Path First - Traffic Engineering
OSS Operations Support System
OSSDN Open Source SDN
OTN Optical Transport Network
OTS Open Transport Switch
OVS Open Virtual Switch
PA Policy Agent
PAC.C Packet and Circuit Convergence
PBB Provider Backbone Bridge
PCC Path Computation Client
PCE Path Computation Element
PCEP Path Computation Element Protocol
PM Provisioning Manager
PNF Physical Network Functions
POF Protocol-Oblivious Forwarding
POP Point of Presence
PXC Photonic cross-connects
QoT Quality of Transmission
REST Representational State Transfer
ROADM Reconfigurable Optical Add-drop Multiplexers
RPC Remote Procedure Call
RSA Routing and Spectrum Assignment
RSVP-TE Resource Reservation Protocol - Traffic Engineering
S-PCE Stateful Path Computation Element
SAL Service Abstraction Layer
SB-API Southbound Application Programming Interfaces
SBI Southbound Interface
SDH Synchronous Digital Hierarchy
SDN Software-Defined Networking
SDNRG SDN Research Group
SDO Standards Developing Organization
SDON Software-defined Optical Networks
SFC Service Function Chaining
SH-PCE Stateful-Hierarchical PCE
SID Segment Identifiers
SNC Smart Network Controller
SNMP Simple Network Management Protocol
SONET Synchronous Optical Network
SPRING Source Packet Routing in Networking
SR Segment Routing
SRLG Shared Risk Link Group
T-SDN Transport Software-Defined Networking
TAPI Transport APIs
TDM Time Division Multiplexing
TEAS-WG Traffic Engineering Architecture and Signalling Working Group
TED Traffic Engineering Database
TL1 Transaction Language 1
TMF TeleManagement Forum
TN-Orchestrator Transport Network Orchestrator
TOSCA Topology and Orchestration Specification for Cloud Applications
UCP Unified Control Plane
UML Unified Modeling Language
UNI User Network Interface
vBBU Virtual Baseband Unit
VCG Virtual Concatenation Group
VM Virtual Machine
VN Virtual Network
VNE Virtual Network Embedding
VNF Virtualized Network Functions
VNTM Virtual Network Topology Manager
VOFS Virtual OpenFlow Switch
vOLT Virtual Optical line termination
VON Virtual Optical Network
vPGW Virtual Packet Gateway
vSG Virtual Serving Gateway
WDM Wavelength Division Multiplexing
WSS Wavelength Selective Switch
WXC Wavelength cross-connects
XML Extensible Markup Language
XOS Extensible Cloud Operating System