Control, Management, and Orchestration of Optical Networks: Evolution, Trends, and Challenges

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

(Tutorial Review)

***Abstract*—Automating the provisioning of telecommunications services, deployed over a heterogeneous infrastructure (in terms of domains, technologies, and management platforms), remains a complex task, yet driven by the constant need to reduce costs and service deployment time. This is more so, when such services are increasingly conceived around interconnected functions and require allocation of computing, storage, and networking resources. This automation drives the development of service and resource orchestration platforms that extend, integrate, and build on top of existing approaches, macroscopically adopting software-defined networking principles, leveraging programmability, and open control in view of interoperability. Such systems are combining centralized and distributed elements, integrating platforms whose development may happen independently and parallel, and are constantly adapting to ever changing requirements, such as virtualization and slicing. Of specific interest is the (optical) transport network segment, traditionally operated independently via closed proprietary systems, and characterized by being relatively complex and hard to reach consensus regarding modeling and abstraction. In view of the targets, the transport network segment needs to be integrated into such service orchestration platforms efficiently. In this context, this paper aims at providing an introduction to control, management, and orchestration systems, of which the *network* control is a core component, along their main drivers, key benefits, and functional/protocol architectures. It covers multidomain and multilayer networks and includes complex use cases, challenges and current trends such as joint cloud/network orchestration and 5G network slicing.**

***Index Terms*—Control plane, generalized multiprotocol label switching (GMPLS), network control and management, network function virtualization (NFV), network virtualization/slicing, path computation element (PCE), service and resource orchestration, software defined networking (SDN).**

I. Introduction

The constant need to dynamically provision services in a cost effective way, within complex end-to-end scenarios, spanning multiple knowledge domains, technologies and administrative boundaries has driven the evolution of architectures and protocols for the operation of networks (more recently and generically, *telecommunication infrastructures*), referred to as their control, management and orchestration. Such *services* have grown in complexity, from conceptually simple voice and data connections in homogeneous networks within the scope and control of a single administrative entity, to services requiring the allocation of heterogeneous resources with complex placement constraints and highly dynamic usage patterns in an environment characterized by having multiple actors and stakeholders. This automation requires the development of service and resource orchestration platforms that extend, integrate and build on top of existing ones, macroscopically adopting Software Defined Networking (SDN) principles and are conceived combining centralized and distributed elements.

The paper aims at providing an overview of the current trends and challenges in such control and management. While a significant number of concepts remain valid across multiple technologies and are not restricted to optical networks, the latter still remains our reference and scope. The paper is structured as follows: in Section II we summarize the historical evolution of transport networks and the recurrent need for automating the provisioning of network connectivity services, ensuring a satisfactory level of Quality of Service with automated recovery, justifying the concept of the control plane. Section III presents the fundamentals for the control plane, describing its initial and evolving requirements and basic control models. Section IV briefly describes the ASON/GMPLS architecture, the main distributed control plane, later augmented with the Path Computation Element (PCE). Next, Section V addresses SDN key concepts and trends. Given the realistic target deployment scenarios, Section VI presents the main challenges related to the deployment of control planes in Multi-Domain and Multi-Layer networks, which are somehow generalized under the term Orchestration, detailed in Section VII. Section VIII provides the main architectural elements of Network Function Virtualization (NFV) whilst Section IX covers the related concept of network virtualization, driven by the need to partition the network and manage multiple logical networks that can be operated independently and as a whole. Section X is dedicated to a high level discussion of Network Slicing, which generalizes and establishes the relationships be-

Manuscript received November 2, 2017; revised December 19, 2017; accepted January 10, 2018. Date of publication January 14, 2018; date of current version March 1, 2018. This work was supported in part by the Spanish Ministry MINECO project DESTELLO (TEC2015-69256-R), and in part the European Comm. project METRO-HAUL (761727). This paper was presented in part at the 43rd European Conference on Optical Communication, Gothenburg, Sweden, September 2017. *(Corresponding Author: Ramon Casellas.)*

The authors are with the Centre Tecnològic de Telecomunicacions de Catalunya, Barcelona 08860, Spain (e-mail: ramon.casellas@cttc.es; ricardo.martinez@cttc.es; ricard.vilalta@cttc.es; raul.munoz@cttc.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2018.2793464

tween the previous concepts. Finally, Section XI concludes the paper.

II. Automation of Service Provisioning

A telecommunications network is composed of Network Elements (NE), interconnected by transmission links. Such elements, which may be either circuit or packet switching, switch and forward data based on a set of implicit or explicit rules: for illustration, a *programmable generic* packet switch matches incoming packets looking up for patterns in fields across multiple headers and layers, selects and ranks matches according to defined criteria (e.g., priorities, policies) and subsequently applies actions based on matches (forward, drop, transform, replicate, ...) in multiple incoming, forwarding or outgoing pipelines-chains. If this capability can be defined or configured via software and remotely, it provides new degrees of freedom in how to perform data forwarding, unconstrained by existing destination-based forwarding of e.g., IP and 802.1q networks.

Similarly, optical transceivers transmit data across optical line systems, built with optical amplifiers, and Reconfigurable Optical Add-Drop Multiplexers (ROADMs) which e.g., cross-connect a frequency slot from an input port to an output port.

In this context, to provision a network service (e.g., a data connection), a path needs to be computed, resources pre-assigned and subsequently reserved, forwarding rules defined and NEs configured. Thus, a straightforward requirement is to automate the provisioning of such services cost-effectively, ultimately allowing autonomic network operation and empowering users to efficiently control allocated resources, minimizing manual intervention. Such a process needs to be done across the whole network, with increasing traffic dynamicity requiring frequent and complex re-arrangements within multiple technological layers and in networks spanning multiple segments.

Such provisioning can be done by using the *Management Plane (MP).* That is, those elements of a system that provide management, monitoring and configuration services, affecting its operation, and that deal with Fault Management, Configuration, Accounting, Performance, Security (FCAPS) [2], for it is assumed that network elements have a dedicated management interface (e.g., a serial interface). However, using a local interface is time-consuming and not cost effective, error prone, vendor and device- oriented, so a centralized approach is justified. A separated management network enables centralized provisioning, seen as a sequence of operations for configuration and state definition. Connections set up directly via the MP are referred to as *Permanent Connections,* reflecting longer timescales and lower dynamicity.

The Simple Network Management Protocol (SNMP), for example, was defined for collecting, organizing and modifying information of managed devices on IP networks and to monitor and change a device behavior. The architecture encompasses components: i) the managed device, which implements an SNMP interface allowing read-only or read-write access to device-specific information, ii) the agent or software which runs on managed devices and iii) a Network Management Station/System (NMS). Although the architecture tried to decouple the protocol from the way management data is managed and organized (in a Management Information Base or MIB), nowadays, it is agreed that such approach is not adapted for advanced management purposes (cfr. Section V-B).

Alternatively, the automation of the provisioning can be performed using a *Control Plane (CP).* A CP is a system and set of functions specially dedicated to the dynamic and on-demand provisioning of network connectivity services between endpoints, with standard interfaces operating across domains ensuring vendor inter-operability. The CP is responsible for configuring associated switching and forwarding state at the data plane level. It is worth noting that, in transport networks, the CP was introduced as a means to ease operation (e.g., automatic discovery), off-loading the MP and simplifying the service provisioning process while, at the same time, leveraging the benefits of decentralized routing and control, such as path protection in arbitrary meshed networks and adaptive traffic engineering, including functions not originally part of a NMS, where inventory and topology are manually managed.

There is debate whether a sufficiently developed MP, with augmented interfaces can indeed provide such automation meeting all the requirements, and consensus that both planes can co-exist with a given functional split: the MP conceptually focuses on FCAPS, including the configuration of the CP itself, delegating the actual provisioning to it in a “top-down”, *separation-of-concerns* approach. However, this separation of functions remains blurred. This is illustrated by, for example, integrated solutions where a single entity may perform functions typically associated to both the CP and the MP, using either CP or MP protocols and interfaces; the adoption of concepts and architectures from other technological domains (e.g., computing) where the term CP had not been explicitly used, or the fact that a given protocol can be used as a control-plane protocol or as a management-plane protocol depending on the underlying function.

III. Optical Networks Control Plane: Fundamentals

A. Evolving Requirements for a Control Plane

As introduced, the CP supports a set of basic functions, including i) element addressing; ii) dynamic resource discovery (e.g., local interfaces and device ports and capabilities); iii) automatic topology and reachability discovery and management (by which a control plane may discover the topology without explicit pre-configuration), iv) path computation and v) actual service provisioning with recovery (protection and restoration) ensuring efficient resource usage.

That said, the CP requirements are constantly evolving, including enabling end user control (e.g., User-Network-Interface services) and extending its applicability to multi-domain and multi-layer networks, notably in view of an ever increasing IP over optical convergence. In our context, it is worth noting that the specifics of the optical technology add additional complexity, where the CP must account for Dense Wavelength Division (DWDM) multiplexing hundreds of nominal central frequencies, with rates ranging 10,100 and higher Gb/s, covering both all-optical and opaque switching, while enabling allocation of variable sized optical spectrum (flexi-grid) and requiring the configuration of multiple devices in a line system (e.g., optical amplifiers, filters, tunable lasers and programmable bandwidth variable transceivers), while taking into account the fact that not all NE are contentionless, directionless or may have restrictions in how data can be switched. Provisioned optical channels must ensure a quality of transmission, accounting for effects of physical impairments, power levels, as well as specific technology constraints such as spectrum or wavelength continuity/contiguity.

More recently, CP requirements relate to the increasing potential of hardware for programmability, building on open and standard solutions to avoid vendor lock-in and favor interoperability. New emerging use cases are to support *network virtualization* while empowering users with finer control, including end-to-end applicability, within the so-called 5G and IoT networks and over-arching control (covering all network segments down to the data-center).

B. Control Plane Design and Control Models

The design of a control plane must address the aforementioned requirements, yet it is in part subject to market pressure to consider and build on top of existing mechanisms (e.g., IP/MPLS networks), extending a well-tested, deployed and mature protocol for a new function instead of designing a new one, aiming at low risk, fast adoption and reduced time to market. Additionally, concurrent efforts in different Standards Defining Organizations (SDOs) often result in multiple choices and deployment models. In any case, the design of a CP involves a set of *entities that inter-communicate*, with their functions and responsibilities, defined within functional and protocol architecture(s). In simple terms, control plane models can be distributed or centralized, although common deployments will be hybrid combining both elements.

In the *distributed control model* each node has a controller component (a control plane entity) which communicates with other controllers.[1] The controller itself may be divided into specific controller functions (e.g., routing controller, signaling controller, etc.). CP functions are distributed: each routing controller is responsible for the dissemination of resources under its control (e.g., its own links) so the network view is built in a cooperative way. For a given connection, the signaling controller of the ingress node is typically responsible for the path computation function based on the topology obtained and for triggering the signaling process by which resources are reserved for the connection and forwarding / switching is configured. The signaling process is also distributed across ingress, intermediate and egress nodes. Common distributed models assume that there is IP connectivity between controllers, supporting IP-based control channels, although how this IP connectivity is provided is not specified. Such models have their roots in the design of IP dynamic routing and later on the IP/MPLS control plane, assuming administrative regions loosely tied with changing interconnections as traffic fluctuates and failures occur, and exemplified by the Automatically Switched Optical Network (ASON) [3] and Generalized Multiprotocol Label Switching (GMPLS) architectures. There is no central authority that coordinates the network operation. On the other hand, in *centralized models* a controller interacts with CP agents located in the NE, and CP logic remains in the controller, justified in part by their (relative) simplicity, addressing the shortcomings of distributed control planes.

Both models have their strengths and weaknesses: a central control is conceptually simpler, a single point of deployment of policies and business logic, easier to deploy Application Programming Interfaces (APIs), and requires less state synchronization. It may also present a bottleneck or single point of failure, with potential fault-tolerance issues. On the contrary, some functions (dynamic restoration, fast rerouting) are difficult to achieve in a centralized model, and a distributed CP is more robust and mature, although implementations usually need to conform to a wider set of protocols. It may also operate independently of the NMS, although it is not the default mode of operation. Realistic deployments will nonetheless be hybrid, combining elements from both models.

IV. ASON and GMPLS

The most common distributed control plane follows the ASON/GMPLS architecture and protocols (Fig. 1(a), see [4], [5] and references within). GMPLS extends and adapts the MPLS control plane applying the label swapping paradigm to control any packet or circuit-switched network. It maintains the basic operating principles and procedures while: *i)* differentiating *switching capabilities* and adapting the meaning of label swapping paradigm (that is, forwarding a packet involves a composition of label pop, swap and push operations) for a given technology (e.g., in wavelength switching the label is the central frequency, which remains constant in transparent domains), *ii)* supporting out-of-band signaling with a clear data plane and control plane separation, *iii*) taking into account technology specific constraints such as discrete bandwidth, or *lightpath* continuity.

A. GMPLS Core Components

The Link Management Protocol allows neighboring nodes part of a control plane adjacency to unambiguously associate data plane adjacencies (e.g., fiber links), correlate identifiers and assure compatible capabilities.

The Open Shortest Path First with Traffic Engineering extensions (OSPF-TE) protocol describes the characteristics of nodes and links, so the state and capabilities of the resources are distributed and updated to all of the nodes; The Routing Controller component uses Link State Advertisement messages, thus reusing the basics of IP dynamic routing. Convergence is determined when all controllers in the network have an updated and common view of the data plane topology (stored in the Traffic Engineering Database or TED), subsequently used in path computation (or Routing and Wavelength/Spectrum Assignment in an optical context). In simple terms, OSPF has been extended to piggy-back Traffic Engineering data about data plane links and nodes, while relying on the existing database synchronization mechanism that uses the actual control plane topology. OSPF-TE must be seen as a database synchronization protocol between control plane entities in which such data base includes (but decouples) the topology of the control plane IP network (in terms

[1] Strictly, there needs not be a 1-to-1 relationship between a control node and transport node, although this is a common deployment.

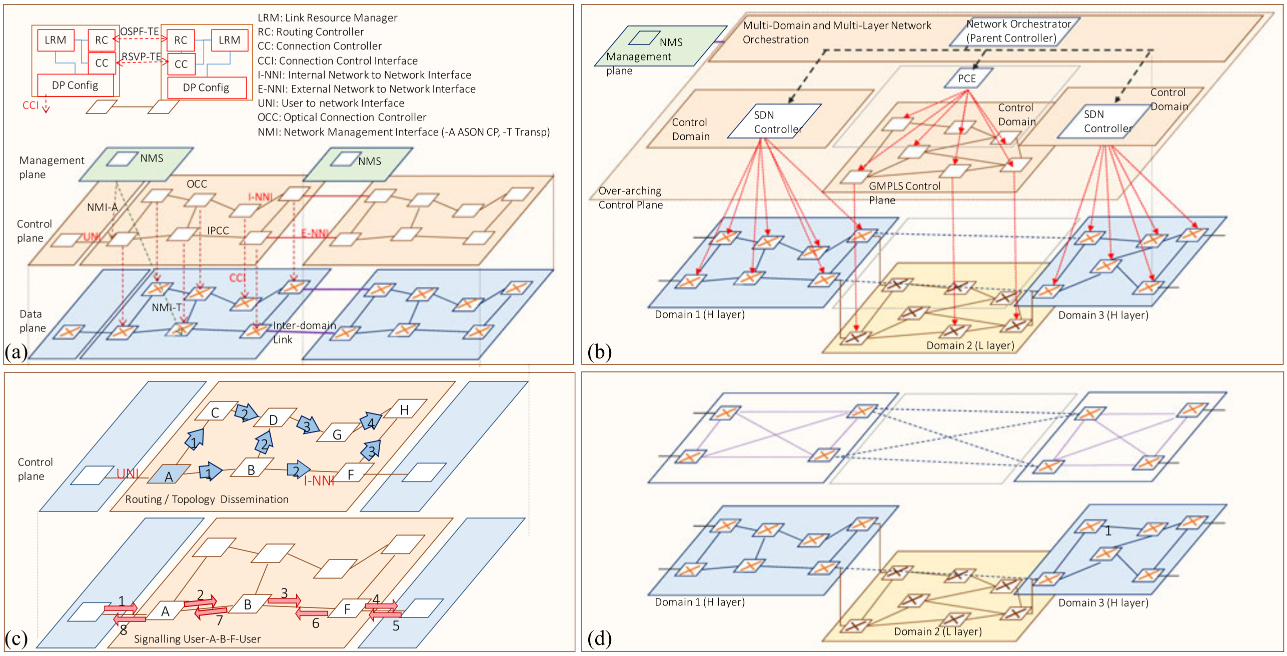

LRM
RC
OSPF-TE
RC
LRM
CC
RSVP-TE
CC
DP Config
DP Config
CCI
LRM: Link Resource Manager
RC: Routing Controller
CC: Connection Controller
CCI: Connection Control Interface
I-NNI: Internal Network to Network Interface
E-NNI: External Network to Network Interface
UNI: User to network Interface
OCC: Optical Connection Controller
NMI: Network Management Interface (-A ASON CP, -T Transp)
Management plane
NMS
NMS
OCC
NMI-A
I-NNI
Control plane
UNI
IPCC
E-NNI
CCI
NMI-T
Data plane
Inter-domain Link
(a)
NMS
Management plane
Multi-Domain and Multi-Layer Network Orchestration
Network Orchestrator (Parent Controller)
PCE
Control Domain
SDN Controller
Control Domain
SDN Controller
Control Domain
Over-arching Control Plane
GMPLS Control Plane
Domain 1 (H layer)
Domain 3 (H layer)
Domain 2 (L layer)
(b)
Control plane
C
D
G
H
1
2
3
4
2
3
UNI
A
1
B
2
I-NNI
F
Routing / Topology Dissemination
1
2
3
4
A
7
B
6
F
5
8
Signalling User-A-B-F-User
(c)
1
Domain 1 (H layer)
Domain 3 (H layer)
Domain 2 (L layer)
(d)

Fig. 1. a) Fully distributed ASON/GMPLS architecture reference; b) Topology dissemination and signaling procedures for a distributed CP; c) Multi-layer and Multi-domain network orchestration across SDN and GMPLS/PCE domains; d) Sample abstracted topology using virtual link meshes models.

of IP routers or controllers participating in the topology) and the topology of the data plane [see Fig. 1(b)].

Signaling is the process to set up (and subsequently release) a connection, known as Label Switched Path, after having completed the path computation function. It takes as input an ordered sequence of nodes and links along with specific resources (e.g., wavelength, frequency slot), in order to actually reserve the resources and configure the hardware (e.g., cross-connections). It involves control plane nodes along the path exchanging messages within the Reservation protocol with Traffic Engineering extensions (RSVP-TE) protocol. The *Path* message is used to "create state", from the ingress node to the egress node, indicating the required bandwidth, frequency slot, modulation format, etc., followed by a backwards *Resv* message to actually "signal" reservation, and allocation of resources [see Fig. 1(b)].

B. Path Computation Element (PCE)

Advanced path computation mechanisms are needed in specific scenarios: controllers in multi-domain networks do not have full topology visibility or, in optical networks, controllers may lack the computational resources to perform path computation subject to constraints such as discrete wavelength availability with wavelength or spectrum continuity constraints, optical impairments, restrictions regarding node internal connectivity, availability of regenerator pools, etc. Additionally, some data useful for path computation may not be available within the control plane.

This results in a clear driver to formally decouple the path computation function from the rest of the control plane, out of the closed and highly integrated nodes within the so called vendor domains, and rendered accessible via a standard, open protocol enabling its use and deployment in other control plane models. A PCE [6] is an entity (component, application or network node) that is capable of computing a network path or route based on a network graph or TED and applying computational constraints. A path computation client may request a path using the PCE protocol (PCEP). A key aspect of the PCE is that it became a step towards decoupling of functions, enabling programmability and operator policy enforcement, allowing independent software upgrades. The PCE has equally been extended to manage connections, suggest optimum connections and, at a later stage, trigger their establishment actually becoming a form of an SDN controller as it will be detailed later, abstracting a GMPLS network and becoming a single point of entry.

V. Software Defined Networking

Software Defined Networking (SDN) is simplistically defined as a centralized control model architecture and protocols, highlighting the CP and DP separation, and enabling an application layer. The simplest architectures encompass a single, logically centralized *Controller* (control layer) on top of the data plane NE or devices (infrastructure or data plane layer), with the control logic placed within the controller. The interface and (associated protocol) by which a controller communicates with devices is referred to as the *South Bound Interface* (SBI), while the set of Application Programming Interfaces (API) offered to applications is named the *North Bound Interface* (NBI). Much has been written about SDN during the past years (see, for example [7]), applied not only to packet switched networks within a campus or intra-DC but also to transport networks, Wide-Area Networks (WAN) and, in particular, optical networks.

A finer characterization of SDN involves identifying opportunities for a better integration with Operators' Business and

Support Systems (OSS-BSS) and planning tools [8], easing implementation of network-wide policies having an open and single point of definition and enforcement, while enabling new business models. The uses of SDN go well beyond re-implementing distributed control plane logic in a central location, and are more related to developing a systematic approach to resource management in heterogeneous contexts with: *i)* interfaces definition around standardized data models, *ii)* the use of unified protocol frameworks overcoming known limitations supporting network-wide transactions & rollback, *iii)* the availability of open systems and open source software covering most key aspects of system development, *iv)* allowing more flexibility in the configuration of network behavior and facilitating innovation. Last, this decoupling of hardware and software is allowing vendor-neutral deployments of hardware that is *disaggregated* and modular (commonly referred to as white boxes), as presented later in Section V-E, exploiting the capabilities of hardware to be programmed, while enabling an application ecosystem.

A. Multiple Choices for Controller SBI

A design item of interest within SDN is the SBI, and different protocols and architectures are available depending on the use case. There are several issues with the selection of the protocol, depending on the needs to interact with infrastructure elements. While it is desirable to have a common and unified protocol, in practice it may be need to interact with multiple elements using multiple protocols. Yet, an ideal protocol for SBI should be flexible, extensible, supporting aforementioned CP functions while, quite importantly, remaining future-proof allowing generic configuration for yet-to-be-invented devices. This last requirement often addressed by decoupling the protocols to *transport information* between entities, from the *way the information is structured*. Other relevant aspects that affect the design and choice of such a protocol are the possible encodings (byte encoded for efficiency or text based for ease of use and debugging) or the availability of frameworks (i.e., languages, libraries, software development kits), their feature set and maturity, along with actual device vendor support or the auxiliary (often open source) software tools for generic processing (parsers) and global established knowledge-base.

OpenFlow [9] is a particular case of standard interface and protocol leveraging programmability and exploiting the fact that most modern NEs can be abstracted identifying a common set of functions, around the concept of flows, matches and action tables. Main achievements of OpenFlow, a part from significantly helping kick-starting the SDN adoption, are the conception of a generic packet switch model. However, the OpenFlow protocol, remains a low level, byte oriented protocol and, although considered stable, deployed, and fit-for-purpose for packet switched networks. It is complex to extend and its applicability to optical networks is not straightforward. In the packet switching domain, the P4 language [10] is trying to overcome some of the perceived limitations of OpenFlow, providing a high –level language so the forwarding behavior of a switch can be programmed and deployed on a wide range of hardware. Finally, although research and standardization efforts have produced documents extending OpenFlow for optical transport networks [11], other alternatives exist due to factor such as the complexity of the optical hardware, the initial focus of OpenFlow to packet switches, the arguable need to support vendor extensions, and the need for better control and management frameworks fulfilling operators requirements.

B. Towards Better Control and Management Frameworks

From the perspective of an operator, the configuration of a control plane (e.g., definition of routing policies, configuration of routing peers) remains a management task. On the other hand, some deployments of optical transport networks are purely managed, without a dedicated control plane. As such, in a related work, the need of better management frameworks and protocols has long been established.

Legacy protocols such as SNMP have a strong coupling between the device data model(s) and the underlying transport protocol. SNMP as protocol is low level, lacks desired flexibility, expressiveness and does not support advanced functions such as Remote Procedure Calls (RPC), so a logical operation can turn into a sequence of interactions keeping state until the operation is complete, and if error, needing to roll the device back into a consistent state. There is a semantic mismatch between the task-oriented view preferred by operators and the data-centric view provided by SNMP.

Additionally, there is a need to have better configuration management, a clear separation of configuration and operational data, while enabling high level constructs more adapted to operators' workflows supporting network-wide transactions, rollback capabilities and transactional semantics. The Internet Architecture Board held a workshop on network management [12], considering existing solutions, requirements, and gap analysis, resulting in the creation of new working groups. In short, the limitations of existing solutions drove activities within unified information and data modeling, to a large extent regardless of whether the actual modeling and configuration applies to management operation or a control one.

While such frameworks are initially focused on management tasks, it is reasonable to adopt them holistically, covering most aspects related to device and network control.

A device *Information Model* macroscopically describes the device capabilities, in terms of operations and configurable parameters, using high level abstractions without specific details on aspects such as a particular syntax or encoding. A *Data Model* determines the structure, syntax and semantics of the data that is externally visible.

C. The YANG Modeling Language

YANG [13] is a data modeling language, where a model includes a header, imports and include statements, type definitions, configurations and operational data declarations as well as actions (RPC) and notifications. The language is expressive enough to structure data into data trees within the so called *datastores*, by means of encapsulation of containers and lists, and to define constrained data types (e.g., following a given textual pattern); to condition the presence of specific data to the support of optional features and to allow the refinement of models by extending and constraining existing models (by inheritance/augmentation), resulting in a hierarchy of models.

Although initially conceived to model configuration and state data for network devices, YANG has become the data modeling language of choice for multiple network control and management aspects (covering devices, networks, and services, even pre-existing protocols) due in part, for its features and flexibility and the availability of tools [14]. For example, an SDN controller may export the underlying optical topology in a format that is unambiguously determined by its associated YANG schema, or a high-level service may be described so that an SDN controller is responsible for mediating and associating high-level service operations to per-device configuration operations. This middleware is often referred to as *Service Orchestrator* (see Section VII).

D. NETCONF and RESTCONF

An *associated protocol* offers primitives to view and manipulate the data, providing a suitable encoding as defined by the data-model. For YANG, the NETCONF protocol [15], enables remote access to a device, and provides the set of rules by which multiple clients may access and modify a *datastore* within a NETCONF server (e.g., device). It is based on the exchange of XML-encoded RPC messages over a secure (commonly Secure Shell, SSH) connection.

NETCONF enabled devices include a NETCONF server, Management applications include a NETCONF client and device Command Line Interfaces (CLIs) can be a wrapped around a NETCONF client. The layering mode relies on having configuration or notification data (Content Layer) that is exchanged between a client and a server, with a set of well-defined operations (e.g., <get-config>, <edit-config>, within the Operations Layer) encapsulated in RPC messages or notifications (Message Layer) and using a Secure Transport. The data is arranged into one or multiple configuration datastores. A configuration datastore is the complete set of configuration information that is required to get a device from its initial default state into a desired operational state. After establishing a session over a secure transport, both entities send a hello message to announce their protocol capabilities, the supported data models, and the server's session identifier. When accessing configuration or state data, with NETCONF operations, subtree filter expressions can select subtrees, providing a great degree of flexibility.

Alternatively, RESTCONF (an effort to map NETCONF operations to REST operations over HTTP following REST model) can also be applied, arguably simpler but less complete.

A significant number of initiatives are defined around the use of YANG models yet the number of different, often partially overlapping, models is increasing, and this is likely to remain an issue for the foreseeable future. There is little experience effectively using such models and the underlying complexity needs to be managed.

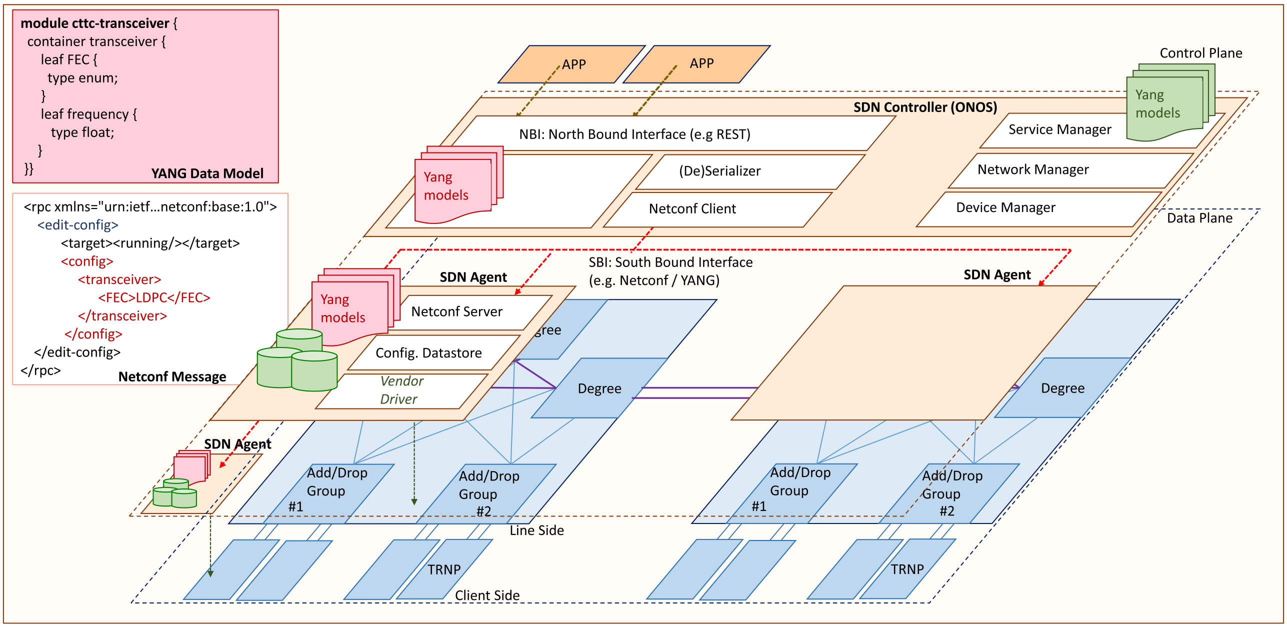
E. SDN Control of Disaggregated Optical Networks

Traditional optical transport networks are proprietary, integrated and closed, where the entire transport network acts as a single vendor managed domain. It can export high-level interfaces and open NBI, yet the internal details and interfaces are hidden from the operator.

Disaggregation of optical networks refers to a deployment model of optical systems, by composing and assembling open, available components, devices and sub-systems. This disaggregation can be partial or total (down to each of the optical components) and is driven by multiple factors, notably, the mismatch between the needs of operators and the ability to deliver adapted solutions by vendors; the increase in hardware commoditization; the different rate of innovation for different components; the promised acceleration on the deployment of services and the consequent reduction in operational and capacity expenses.

Disaggregation aims at providing a new degree of flexibility, allowing component migration and upgrades without vendor lock-in. On the other hand, it can be argued that disaggregated optical nodes may not have the same level of integration and performance that integrated systems. In short, disaggregated optical networks imply a trade-off in terms of current and potential performance, vendor support and cost. It is expected that short term disaggregation will involve common functions adhering to open standards and interfaces, yet allowing vendor specific extensions and high-performance solutions with added value. Full disaggregated optical transmission systems are not considered to be a short-term opportunity for highly efficient optical layers.

Disaggregation imposes a new set of challenges in its control and management. It is clearly a use case for open interfaces exporting programmability, and the increase of unified and systematic information and data modelling activities is a crucial step in this regard. However, optical networks are particularly challenging to model due to the lack of agreed-upon hardware models, and this is critical for the development of an interoperable ecosystem around disaggregated hardware. Research work commonly uses the NETCONF/YANG approach to control optical elements (e.g., see [16], [17] for recent use cases). Additionally, there are several cross-vendor initiatives such as the OpenROADM multi-source agreement [18], which focuses on functional disaggregation and, for the first release, covers pluggable optics, transponders and ROADMs. OpenROADM has released a set of YANG models covering aspects such as devices and networks. Likewise, OpenConfig [19], a collaborative effort by network operators, has published a set of models providing a configuration and state model for terminal optical devices within a DWDM system, including both client- and line-side parameters. It remains to be seen which models are significantly adopted. SDN Control of disaggregated networks is shown in Fig. 2, where a SDN controller enables the provisioning of dynamic optical services (stored within the service manager) while maintaining a coherent view of the network and its abstraction (within the network manager) and configures one or more optical devices using Netconf/YANG as SBI. The controller acts as a Netconf client acting on the data stores located within the device, whose SDN agent (a common name to generically refer to the device component that communicates with the controller) implements a Netconf server. YANG models are predefined or dynamically discovered and commonly stored both at the controller and device level. Applications can be developed on top of the controller.



```
module cttc-transceiver {
 container transceiver {
   leaf FEC {
    type enum;
   }
   leaf frequency {
     type float;
   }
}}
```

```
<rpc xmlns="urn:ietf…netconf:base:1.0">
   <edit-config>
         <target><running/></target>
         <config>
              <transceiver>
                  <FEC>LDPC</FEC>
              </transceiver>
          </config>
  </edit-config>
</rpc>
```

Fig. 2. Centralized SDN control of disaggregated optical networks, using Netconf/YANG. Simplified Yang module showing a transceiver data model and the corresponding Netconf edit-config message to configure the relevant parameter.

F. Telemetry and Data Analytics

In our context, telemetry refers to streaming data relating to underlying characteristics of a given device - either operational state or configuration – and is often relying on monitoring infrastructure, either embedded in the underlying device, or by dedicated equipment. There is a clear trend of integrating network telemetry, and subsequent data analytics into the SDN control, in view of implementing operator-defined and adaptive policies for multi-fold purposes: traffic off-loading, efficient resource usage, dynamic link resizing, automatic traffic engineering, validation of Service Level Agreements (SLA) related performance parameters and network optimization.

A unified approach for data collection and processing facilitates shifting the focus to the actual processing of information, towards more autonomous networks. A first key requirement is the efficient collection of data from multiple sources. This requires the development and adoption of advanced interfaces overcoming the limitations of existing ones to maximize efficiency and minimize latency, bandwidth usage and data processing requirements. These interfaces have stringent functional requirements, such as i) to monitor the status of hundreds (or thousands) of entities in a large scale network; ii) to configure programmable pipe-lines in terms of asynchronous events and flexible filters, including expressive and domain embedded languages in a publisher-subscriber pattern; iii) to enable the automatic discovery and monitoring of key parameters, commonly known as performance monitoring, and iv) to do this in a context defined by the use of Open Source projects and open and standard interfaces. A second requirement is to adapt existing control plane architectures to make use of this telemetry data. Ongoing IETF drafts are being produced adapting and extending existing YANG notification mechanisms [20] allowing a network operator to subscribe notifications on a per client basis; configure what parameters to apply filtering and selective collection at the point of origin of the notification, and to request whether notifications are periodic, event-driven, etc.

However, the limitations are being identified and alternatives proposed. The gRPC protocol [21] can be used for the modification and retrieval of configuration from a NE, as well as the control and generation of telemetry streams to a data collection system, so a single gRPC service definition can cover both configuration and telemetry, while still relying on payloads containing data instances of YANG schemas or with more efficient serializations and encodings (e.g., binary format compression).

In short, enabling telemetry and data analytics requires both efficient data collection that scales and tailored control plane architectures. The centralized model of common SDN architectures is, at the same time, an opportunity and a threat; it is more straightforward to make efficient use of collected data and enable application ecosystems in a centralized model, but such systems scale less than distributed ones.

G. Machine Learning Assisted Network Operation

The increase in network telemetry and data analytics is favoring the design of control support systems that enable a more efficient, performing and autonomous network operation. The concept is not new (cfr. *knowledge plane*) [22] yet the development of the underlying, supporting technologies is still ongoing. The first control supporting systems have been using rule-based decisions, in which policies are described in terms of rules and actions.

Machine Learning (ML) can be loosely defined as field of computer science and an application of artificial intelligence that provides to systems the ability to automatically learn and improve from experience without being explicitly programmed. It covers algorithms that can learn from and make predictions on data, building a model from sample inputs.

There is a clear trend to adopt ML techniques to aid in the decisions involving network control and management. In simple terms, a system is trained with prepared data-sets so the performance of the system improves in view of future actions. The applications of ML are diverse, from fault detection, to traffic matrix estimation, recognizing traffic and network behavior patterns. This topic is expected to increase significantly in the upcoming years.

VI. Multi-Domain and Multi-Layer Networks

Transport networks are increasingly segmented in domains, e.g., to enhance scalability, due to confidentiality reasons or by virtue of having non-interoperable vendor islands. Regardless of the looseness of the term *Domain* (admitting multiple definitions, such as the set of elements as defined by management boundaries, vendor or technology islands, topology visibility or path computational responsibility), CP entities in multi-domain networks have inherent limited topology visibility outside a given domain and interoperability issues for cross-domain signaling. Exchange of topological information between domains is limited to the dissemination of reachability yielding sub-optimal choices and domain local optimality does not imply end-to-end optimality. In fully distributed models, such exchange of information takes place between inter-domain neighbors, which, in turn, inject part of the information to their respective domains. In hybrid and centralized deployments, amongst different interconnection options, a common trade-off is to rely on a hierarchical arrangement of controllers [see Fig. 1(c)), along with some degree of topology abstraction and aggregation (see Fig. 1(d)], minimizing interoperability and ownership issues. Common abstraction models aggregate in terms of virtual links [23].

Multi-layer networks involve multiple technologies (e.g., a packet switched layer and a circuit switched layer) or multiple levels within a given technology. Services are understood within a client-server model where a lower layer connection supports multiple higher layer connections, enabling grooming and multiplexing. A multi-layer CP implies being able to provision services across multiple layers.

Regarding CP design, a basic model can be defined where each layer has its own CP instance, with little to no interaction (*Overlay Model*), justified in practice by current market and vendor segmentation. Such an approach lacks a joint control of the involved layers, thus limiting efficient resource usage (i.e., having topology visibility of all the layers is required to attain optimal path computation). Conceptually opposite, other interconnection models rely on full topology visibility and are significantly more complex. For example, the GMPLS peer model within the so-called Multi-Layer and Multi-Region Networks (MLN/MRN) relies on having a common OSPF-TE protocol instance disseminating link attributes for the involved layers, tagging each link with its *Switching Capability* and technology-dependent attributes (e.g., available nominal central frequencies) in order to be used by the controller during path computation and ensuring the coordinated establishment of connections supporting the technology-implicit hierarchy (e.g., IP/MPLS packets over an Optical Data Unit (ODU) over an optical channel). Similarly, a single instance (e.g., SDN controller) may take responsibility for controlling all switching layers, assuming full visibility of the regions, operating as a single control domain, locally separating the technology domains for provisioning purposes and using dedicated provisioning interfaces at defined demarcation points. While this approach may be suitable for small domains with a reduced set of layers (e.g., IP over optical), it does not scale and the current trend is to roll out hybrid models in which each layer operates independently to a large extent, yet there is some abstracted information exchanged and inter-layer coordination ensures efficient resource usage.

Macroscopically, the issues of multi-layer networks are similar to multi-domain networks, with the added complexity of dealing with different data plane technologies. However, the same trend applies: abstract network details, provide interoperability layers with standard and uniform models and deploy controllers arranged in particular settings. As we will see next, this is known as network orchestration.

VII. Orchestration

The term Orchestration often appears when referring to control and management architectures, but there is only a rough consensus on its actual meaning and scope. The Open Networking Foundation (ONF) defines orchestration as *the selection of resources to satisfy service demands in an optimal way, where the available resources, the service demands and the optimization criteria are all subject to change*. The orchestration function adjusts the state of the resources under its control to move toward that optimum [24]. Within Networks Function Virtualization (NFV, presented next), the term refers to the *coordination of the resources and networks needed to set up cloud-based services and applications*, a process using a variety of virtualization software and industry standard hardware [25].

In particular, *Network Orchestration* addresses the overarching control across multiple heterogeneous domains (both in terms of control and data plane domains). It commonly relies on hierarchical control architectures, where the parent controller is referred to as the *Orchestrator* ensuring over-arching control relying on multi-domain abstracted topology and service databases, with each specific domain performs its own topology abstraction and control adaptation (as shown in Fig. 3, dark grey area). *Joint IT/Cloud and Network Orchestration* is used to refer to the coordination of resources to deploy services and applications that require storage, computing and networking resources The former is exemplified by the need to provision network connectivity services across heterogeneous domains (e.g., OpenFlow islands inter-connected by a GMPLS/PCE optical network) [26], [27]. The latter, by the provisioning of cloud services that require end-to-end Intra/Inter data center (DC) control and inter-connection of Virtual Machines (VMs) located in distributed data centers, in multiple geographically disperse sites or locations, where cloud locations are optimized for applications

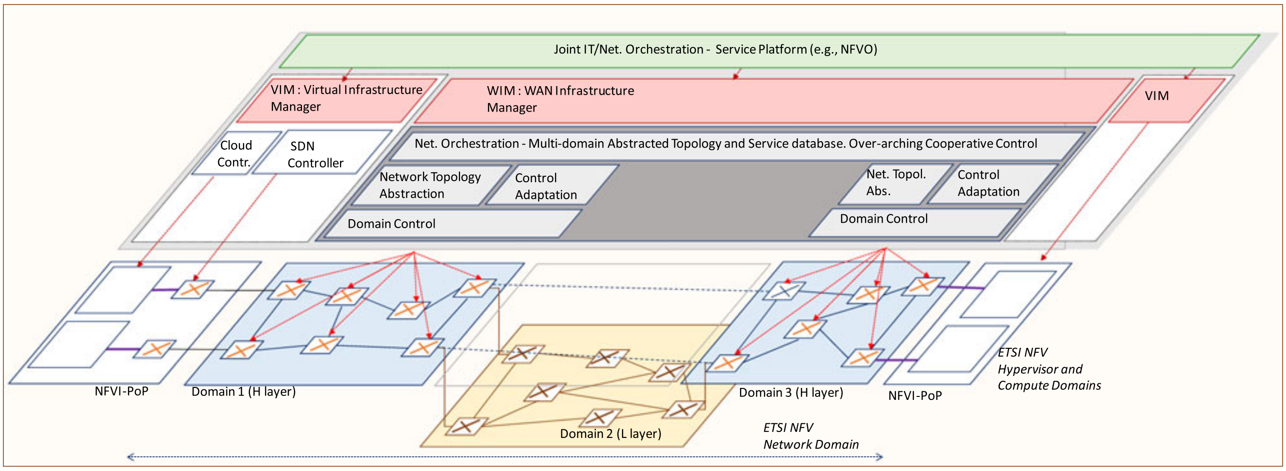

Joint IT/Net. Orchestration - Service Platform (e.g., NFVO)
VIM : Virtual Infrastructure Manager
WIM : WAN Infrastructure Manager
VIM
Cloud Contr.
SDN Controller
Net. Orchestration - Multi-domain Abstracted Topology and Service database. Over-arching Cooperative Control
Network Topology Abstraction
Control Adaptation
Net. Topol. Abs.
Control Adaptation
Domain Control
Domain Control
NFVI-PoP
Domain 1 (H layer)
Domain 3 (H layer)
NFVI-PoP
ETSI NFV Hypervisor and Compute Domains
Domain 2 (L layer)
ETSI NFV Network Domain

Fig. 3. A generalized view on Network and Joint IT/Network Orchestration, aligned with the ETSI/NFV framework and architecture.

delivery [28]. Such orchestration is driven by the increased use of virtualized servers, the need to interconnect the supporting VMs and/or containers and fundamentally the fact that the service provisioning process no longer stops at the physical network node and needs to interact with whatever mechanism the hosting nodes (and virtualization hypervisor) offers, commonly requiring instantiating one or multiple software switches within the host(s) and associate virtual and physical interfaces to software switch instances.

Synthetically, orchestration refers to the coherent coordination of heterogeneous systems, allocating diverse resources and composing functions to offer end-user and operational services and applications, automating processes and using or invoking the programming interfaces of subordinate or external systems, platforms and infrastructures, often with transactional semantics and using high-level frameworks, constructs and languages.

VIII. Network Function Virtualization (*NFV*)

NFV can be initially defined as an architecture and deployment model around the idea of replacing dedicated network appliances — such as routers and firewalls — with software implementations (*guests*) running on common shared hardware (*hosts*), becoming Virtualized Network Functions (VNFs). NFV relies and builds on top the of state of art and advances regarding servers “virtualization” and cloud computing and management, i.e., the ability to allocate VM, or containers over a common, shared infrastructure by means of a hypervisor, with direct control over the hardware resources. It is important to highlight that it is the function that is virtualized, keeping the same function logic but executed in a virtualized environment. The benefits have been well established, including lower costs, replacing dedicated appliances with shared servers; use capacity on demand and efficient resource usage, reduce operational costs with fewer appliances to deploy and maintain, enable e.g., migrations, support on-demand, and pay-as-you-go deployment models and enable innovation by making it easier to develop and deploy network functions.

The ETSI NFV architecture defines the NFV Infrastructure (NFVI) deployed across multiple points of presence (NFVI-PoP) for supporting the instantiation of VMs, along with the *Management and Orchestration* (MANO) subsystem, which deals with the orchestration of VNFs and how to deploy them as components of the so called *Network Services*. The MANO includes the Virtualized Infrastructure Manager(s) VIM. VIMs, manage and provide access to storage, network and computer resources, one or more VNF managers and the NFV Orchestrator (NFVO) functional component. The NFVO performs *Service Orchestration*, that is, the part of service instantiation involving the functional split of the service into/amongst different VNFs – that may be managed by different managers (VNFMs) by different vendors – and their logical interconnection (called VNF forwarding graphs) and *Resource Orchestration* that deals with the allocation of resources to support the VNFs and the logical links. Resource orchestration is important to ensure there are adequate compute, storage, and network resources available to provide a network service. To meet that objective, the NFVO can work either with the VIM or directly with NFV infrastructure (NFVI) resources, depending on the requirements It has the ability to coordinate, authorize, release, and engage NFVI resources independently of any specific VIM. It also provides governance of VNF instances sharing resources of the NFVI.

The NFV architecture has been used to deploy reliable instances of the control plane, in which SDN controllers are instantiated as VNFs [29], [30]. NFV provides the ability to orchestrate interconnected and virtualized functions that can be tailored and deployed on demand. Open challenges for NFV involve its relationship and integration with SDN (and in particular with transport SDN) and its applicability in multiple administrative domains. The use of the SDN/NFV framework is a possible implementation of the joint IT/Network orchestration, as shown in Fig. 3, where VNF can be instantiated in Points of Presence (PoP) that are interconnected by transport networks. The NFV architecture accommodates this by defining a specialized VIM (referred to as WAN Infrastructure Manager, or WIM) which delegates its functions or is implemented in terms of a SDN controller or network orchestrator.

IX. (Optical) Network Virtualization

As transport networks evolve, the need to provide network abstraction and virtualization has emerged as a key requirement for operators. *Network virtualization* refers to the process by which multiple logical (virtual) networks are supported over a common, shared physical network infrastructure [31] (note that the physical aspect is understood at the lowest level. Since network virtualization can become recursive, a logical virtual network may also be virtualized). A common research problem is e.g., virtual optical network embedding (see [32]).

Network virtualization is an enabler for *multi-tenancy* [33], an ownership concept in which tenants are given a different partial and abstracted topology view, and are allowed to utilize and independently control allocated virtual network resources as if resources were real. The granularity level of control given to tenants can vary, depending on the involved new business models.

The mechanisms to actually support network virtualization are diverse, and strongly depend on the uses cases and associated requirements, notably in terms of traffic isolation, service level agreements and performance guarantees. In most cases, such mechanisms rely on a combination of i) actual hardware device support for multi-user and virtualization ensuring resource and traffic isolation and ii) software layers and middleware that perform the necessary control functions.

For example, an optical flexi-grid network can be partitioned, based on a selection of NE or ROADMs, physical ports or link fibers and nominal central frequencies of the DWDM grid (hard partitioning) so a virtual network is thus a subgraph of the underlying network topology graph. A Bandwidth Variable Transceiver (BVT) can tune its bit-rate and bandwidth dynamically with a trade-off between reach and capacity. A BVT may, in turn, be composed of multiple transceivers, each one of such sub-transceivers being configured independently. Such sliceable BVT (S-BVT) enables transmitting from one point to multiple destinations, changing the traffic rate to each destination and the number of destinations on demand. Consequently, a set of such sub-transponders can be assigned to support one or more logical links of the virtual network.

In another scenario, a virtual network can be an *L2/L3 Overlay Network*, i.e., an arbitrary L2/L3 network in which software based switches and routers are instantiated in specific hosting nodes and virtual network links are supported over physical network paths provisioned with actual resource reservation or relying on overprovisioning and statistical multiplexing.

Control plane architectures, which initially assumed direct control over the physical resources need to be extended to explicitly support multitenancy and control of virtualized resources (e.g., an SDN controller for a transport network may need to implement a dedicated interface towards a network hypervisor instead of the SBI to the network element).

X. 5G-Network Slicing

Generalizing the concept of network virtualization, and driven by recent standardization work at SDOs such as 3GPP, IETF, ETSI, the term *Network Slicing* has appeared as an emerging requirement for future 5G networks. While the roots of the concept are related to network virtualization, including the partitioning (slicing) of a single (commonly physical) infrastructure in order to construct multiple (logical) infrastructures, there are important differences that are worth highlighting. In particular, more emphasis is given to the actual network functions and how they are arranged and configured, forming a complete logical construct or network, tailored, customized and optimized for a given service or service set, or to support a given actor or customer (e.g., vertical industry). Second, a given slice can combine both data and control plane functions and functional elements, which are inherent part of the slice. In this context, concepts such as traditional data connectivity services such as Virtual Private Networks, Network Virtualization or NFV Network Services become specific cases of this generic construct.

From the automation perspective, a challenge is to conceive not only systems able to allocate, manage and deallocate a given slice during its life-time (as in a Slice-as-a-Service or SlaaS), but also to be able to provision, potentially dynamically, control plane instances for the specific control of the allocated slice, supporting a wide range of control models, i.e., from basic monitoring of the slice operation to a full control on the slice down to the constituting elements of the slice. For example, a tailored ETSI NFV MANO system can be instantiated associated to the slice lifetime for the instantiation of Network Services over the actual elements of the slice. Another relevant challenge is to support this concept across multiple (federated) domains across administrative boundaries. Fig. 4 illustrates the concept of having a physical infrastructure (composed of network, computing and storage resources) that can be virtualized, resulting in multiple *virtual infrastructures*. Such virtual infrastructures are composed of virtual inks that interconnect VMs, the latter supporting generic functions (F1, F2 . . .) forming a logical network or construct (slice). Resources within this slice can also be orchestrated, with systems dedicated to manage the composing functions or instantiated SDN control planes for the virtualized network elements.

A. A Suitable Framework: ETSI NFV

The ETSI NFV framework can be used as a starting point for a concrete implementation of a generic slicing architecture, in which network slice instances are NFV Network Services (NS), encompassing NS endpoints and one or more VNFs interconnected by logical links, forming VNF Forwarding Graphs (VNFFGs). Logical links are thus mapped to supporting network connectivity services which may, in turn, span multiple network segments. This is shown in Fig. 5, where multiple NFVO (green, blue), potentially managed by different users or operators can have shared access to a common NFVI managed by their respective VIM/WIMs, and each NFVO instantiated network service (with its corresponding VNFs) is a slice instance.

In this context, there are still a few challenges, some expected to be addressed in successive refinements of the architecture. For example, the focus on the Network Service may not cover all use cases, and additional functions are required to render the NFVO/VIM a full featured service platform for

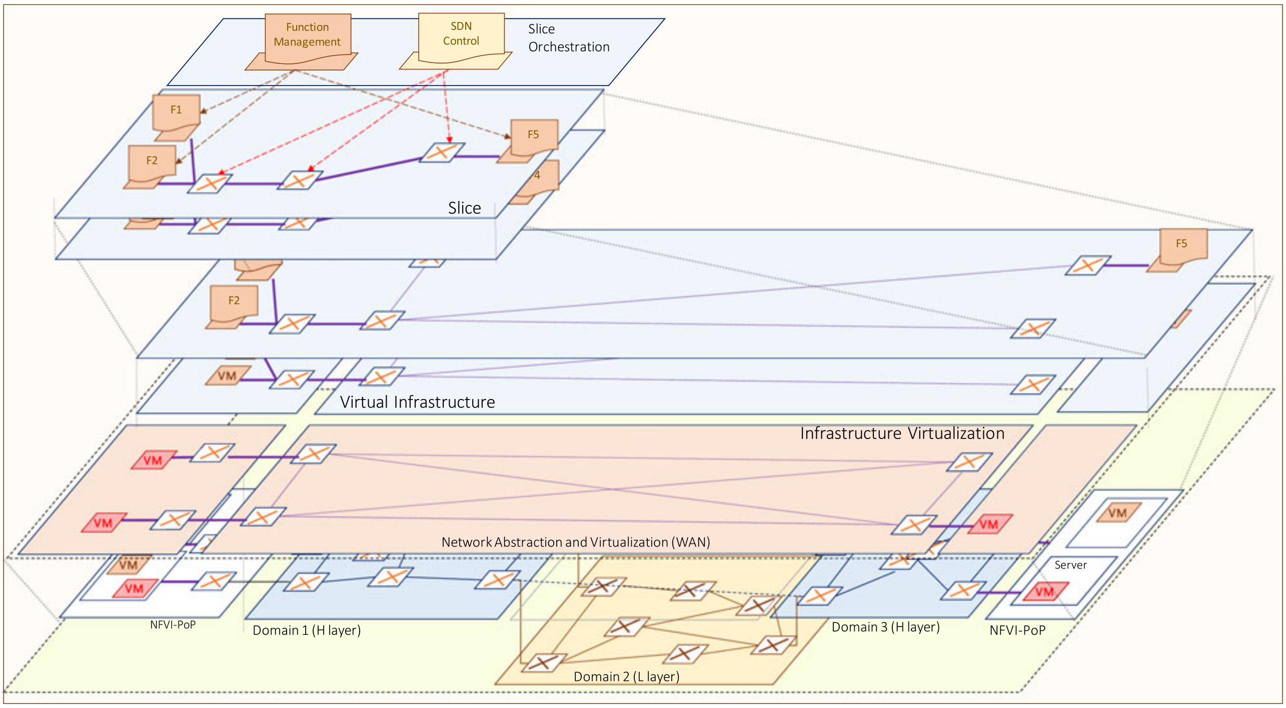


Fig. 4. Network Slicing Concept: Virtualize an infrastructure encompassing network, computing and storage resources, so virtual infrastructures can support interconnected functions tailored for a service or customer, with dedicated control, management and orchestration.

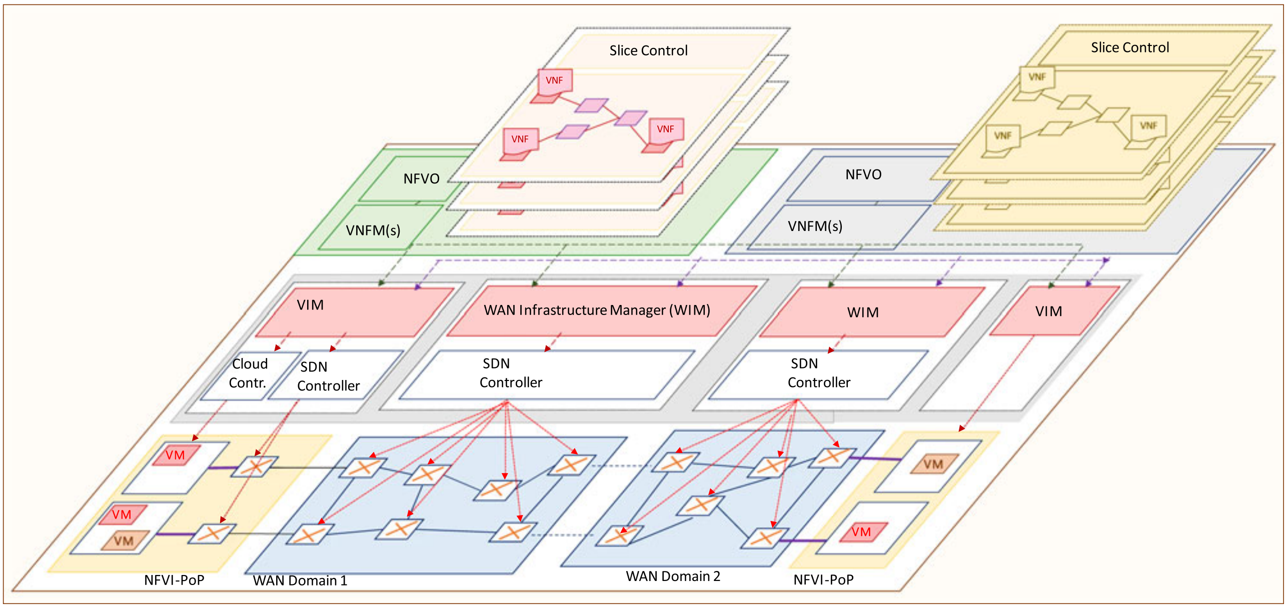


Fig. 5. Network Slicing using the integrated SDN/NFV framework. Different Tenants (e.g., Green/Blue) manage their NFVO to deploy Network Services and Slices over a set of shared VIM/WIM spanning multiple PoP and domains. Each slice has a dedicated Control Plane instance.

SlaaS, which requires specific support for slicing and multi-tenancy. Reference implementations of the architecture have been focused on centralized deployments until recently, where the use of multiple VIMs is considered and the integration with transport networks is still an ongoing debate; current specifications mention the need to provision paths between VNFs in VNFFG and such paths have oftentimes been assumed to be L2 and L3 tunneling technologies. The current trend is reflected in considering multiple VIMs interconnected by a WIM which delegates its functions or is implemented in terms of a SDN controller or orchestrator. For a detailed analysis, see, for example [34]. This commonly assumes a single administrative domain, for the interfaces between NFVOs are at this time unspecified.

Such SDN controllers may, in turn, have specific capabilities supporting network virtualization, although overlapping functionalities are not uncommon. For example, OpenDayLight supports Virtual Tenant Networks (VTN), which allow users to define the network with a look and feel of conventional L2/L3 network. Once the network is designed on VTN, it will automatically be mapped into underlying physical network. The implementation also supports a limited form of control over the logical elements of the VTN. The ONOS controller, in turn, implements the Abstraction and Control of Traffic Engineered Networks (ACTN) architecture where Virtual Networks may be used to support VNF logical links across multiple domains [35].

XI. Conclusion

The provisioning of services (network connectivity, services involving heterogeneous resources) needs to be automated, accounting for the stringent requirements in terms of quality of service, latency, bandwidth, enabling automatic recovery (protection and restoration). While the initial consensus on the functional split between Management and Control Planes still applies, the separation is becoming diffuse and both layers are adopting a common approach for data and information models, and exploiting the benefits of the increasing programmability of devices and systems.

The challenges stem from the fact that this automation needs to happen in a heterogeneous environment across multiple technological and administrative domains, spanning multiple network segments growing complexity, involving hybrid deployments with centralized and distributed elements. Although ASON/GMPLS and more recently SDN/OpenFlow are the main technologies behind the concepts of distributed and centralized control architectures. Each solution has its own applicability domain, and they are both possible components of a wider overarching control and orchestration, where a hierarchy of functions and roles work in a coordinated way. There is a need for open, standard interfaces covering devices, networks, and service models while adopting a unified approach for modelling across technologies and SDOs.

SDN core principles, the trend of data modelling and open systems and interfaces, the increase device programmability and general softwarization can be broadly applied, with specific subsystems becoming part of a wider SDN-based service and resource orchestration system.

Let us note that, although it is reasonable to present the concepts of Orchestration, NFV and Network Virtualization isolated, it is important to understand their inter-dependencies, for NFV services are orchestrated using multiple NFVI points of presence across multiple network domains that are in turn, orchestrated by a SDN controller and where network virtualization is used to support the required connectivity between functions, while meeting isolation and service level agreements.

Significant research work is needed in having a complete integration in which constrained Network Slice Instances are allocated in a context spanning multiple administrative domains, supported by multiple physical infrastructures with heterogeneous control and data planes, while ultimately requiring flexible control and monitoring by the instance owner. Advances related to efficient telemetry, data analytics, and machine learning assisted network control and management, which are being initially conceived for physical networks are expected to apply to individual slice instances, involving a trade-off on the volume of generated data, the scalability of the solution and the inherent abstraction associated to hierarchical systems.

Finally, we have covered the trends and challenges affecting transport networks in general, and the extension of the underlying principles to also cover all the network segments including not only the wired-access and aggregation, metro and long-haul, but also Radio Access Networks (RAN) and Evolved Packet Core (EPC) need also to be addressed in an end-to-end 5G-management and orchestration platform.

References

[1] R. Casellas, R. Martínez, R. Vilalta, and R. Muñoz, "Control, management and orchestration of optical networks: An introduction, challenges and current trends," in *Proc. 43rd Eur. Conf. Opt. Commun.*, Gothenburg, Sweden, Sep. 2017, Paper Tu.1.G.1.

[2] Information Technology—Open Systems Interconnection—Systems Management Overview, ISO/IEC 10040, 1998.

[3] "Architecture for the automatically switched optical network (ASON)," ITU-T Recommendation G.8080, Nov. 2001.

[4] R. Muñoz, R. Martínez, and R. Casellas, "Challenges for GMPLS lightpath provisioning in transparent optical networks: Wavelength constraints in routing and signalling," *IEEE Commun. Mag.*, vol. 47, no. 8, pp. 26–34, Aug. 2009.

[5] N. Sambo *et al.*, "GMPLS-controlled dynamic translucent optical networks," *IEEE Netw.*, vol. 23, no. 3, pp. 34–40, May–Jun. 2009.

[6] F. Paolucci, F. Cugini, A. Giorgetti, N. Sambo, and P. Castoldi, "A survey on the path computation element (PCE) architecture," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1819–1841, Oct.–Nov. 2013.

[7] 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, 2013.

[8] L. Velasco, D. King, O. Gerstel, R. Casellas, A. Castro, and V. Lopez, "In-operation network planning," *IEEE Comm. Mag.*, vol. 52, no. 1, pp. 52–60, Jan. 2014.

[9] OpenFlow Switch Specification v1.5.1, ONF TS-025, March 26, 2015.

[10] The P4 Language Consortium, P4 Language Specification 2016 revision, May 2017. [Online]. Available: http://p4,org/spec

[11] "Optical transport protocol extensions version1.0," ONF TS-022, March 15, 2015.

[12] J. Schoenwaelder, "Overview of the 2002 IAB network management workshop," IETF Request for Comments 3535, May 2003.

[13] M. Bjorklund, "The YANG 1.1 data modeling language," IETF Request for Comments 7950, Aug. 2016.

[14] Yang Central tools, retrieved 2017-12-14, last modified 2016-10-28, [Online]. Available: http://www.yang-central.org/twiki/bin/view/Main/YangTools

[15] R. Enns, "Network configuration protocol NETCONF," IETF Request for Comments 6241, Jun. 2011.

[16] M. Dallaglio, N. Sambo, F. Cugini, and P. Castoldi, "Control and management of transponders with NETCONF and YANG," *J. Opt. Commun. Netw.*, vol. 9, no. 3, pp. B43–B52, 2017.

[17] M. Dallaglio, N. Sambo, F. Cugini, and P. Castoldi, "YANG models for vendor-neutral optical networks, reconfigurable through state machine," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 170–178, Aug. 2017.

[18] The Open ROADM Multi-Source Agreement (MSA), Dec. 2017. [Online]. Available: http://www.openroadm.org

[19] The OpenConfig project, Dec. 2017. [Online]. Available: http://openconfig.net/

[20] A. Clemm, A. González Prieto, E. Nilsen-Nygaard, A. Triàthy, S. Chisholm, and H. Trevino, "Subscribing to event notifications," draft-ietf-netconf-rfc5277bis-01, Oct. 27, 2016.

[21] "A high performance, open-source universal RPC framework," Dec. 2017. [Online]. Available: https://grpc.io/

[22] D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A knowledge plane for the internet," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, New York, NY, USA, 2003, pp. 3–10.

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

[24] "SDN Architecture, Issue 1.1," ONF ONF TR-521, 2016.

[25] "Network functions virtualisation (NFV); Architectural framework," ETSI GS NFV 002 (V1.2.1), Dec. 2014.

[26] Y. Yoshida *et al.*, "SDN-based network orchestration of variable-capacity optical packet switching network over programmable flexi-grid elastic optical path network," *J. Lightw. Technol.*, vol. 33, no. 3, pp. 609–617, Feb. 2015.

[27] V. López *et al.*, "Demonstration of SDN orchestration in optical multi-vendor scenarios," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, 2015, Paper Th2A.41.

[28] A. Mayoral, R. Vilalta, R. Muñoz, R. Casellas, and R. Martínez, "SDN orchestration architectures and their integration with Cloud Computing Applications," *Elsevier Opt. Switching Netw.*, vol. 26, pp. 2–13, Nov. 2016.

[29] R. Casellas, R. Vilalta, R. Martínez, and R. Muñoz, "Highly-available SDN control of flexi-grid networks with network function virtualization enabled replication," *J. Opt. Commun. Netw.*, vol. 9, no. 2, pp. A207–A215, Feb. 2017.

[30] R. Muñoz *et al.*, "Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks," *J. Opt. Commun. Netw.*, vol. 7, no. 11, pp. B62–B70, Nov. 2015.

[31] R. Nejabati, E. Escalona, S. Peng, and D. Simeonidou, "Optical network virtualization," *IEEE 15th Opt. Netw. Des. Model.*, Bologna, Italy, pp. 8–10, 2011.

[32] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, no. 3, pp. 450–460, Feb. 2014.

[33] X. Li *et al.*, 5G-Crosshaul Network Slicing: Enabling multi-tenancy in mobile transport networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 128–137, Aug. 2017.

[34] B. Chatras, S. Tsang Kwong, and U. N. Bihannic, "NFV Enabling Network Slicing for 5G," in *Proc. 20th Conf. Innov. Clouds, Internet Netw.*, Mar. 2017, pp. 219–225.

[35] R. Casellas, R. Vilalta, R. Martínez, R. Muñoz, H. Zheng, and Y. Lee, "Experimental Validation of the ACTN architecture for flexi-grid optical networks using Active Stateful Hierarchical PCEs," in *Proc. 19th Int. Conf. Transp. Opt. Netw.*, Girona, Spain, Jul. 2–6, 2017.

Ramon Casellas (SM'12) received the Graduated degree in telecommunications engineering from the UPC-BarcelonaTech, Barcelona, Spain and ENST Telecom Paristech, Paris, France, in 1999, and the Ph.D. degree from ENST, Paris, France, in 2002. He worked as an undergraduate researcher with France Telecom R&D and British Telecom Labs, and is currently working as an Associate Professor at ENST. He joined CTTC in 2006, working in international and technology transfer research projects. His research interests include network control and management, traffic engineering, GMPLS/PCE, SDN, and NFV. He has published more than 180 papers, 4 IETF RFCs, and 4 book chapters.

Ricardo Martínez (SM'14) received the M.Sc. and Ph.D. degrees, both in telecommunications engineering, from the UPC–BarcelonaTech University, Barcelona, Spain, in 2002 and 2007, respectively. He has been actively involved in several EU public-funded and industrial technology transfer projects. Since 2013, he is a Senior Researcher at CTTC in Castelldefels, Spain. His research interests include control and orchestration architectures for heterogeneous and integrated network, and cloud infrastructures along with advanced mechanisms for provisioning/recovering quality-enabled services.

Ricard Vilalta (SM'17) received the Telecommunications Engineering and Ph.D. degrees from UPC, Barcelona, Spain, in 2007 and 2013, respectively. He is a Senior Researcher with the Communication Networks Division, CTTC. He has been involved in international, EU, national and industrial research projects, and published more than 170 journals, conference papers, and invited talks. He is also involved in standardization activities in ONF, IETF, and ETSI. His research interest includes SDN/NFV, network virtualization, and network orchestration.

Raül Muñoz (SM'12) received the Graduated degree in telecommunications engineering and Ph.D. degree in telecommunications, both from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 2001 and 2005, respectively. He is the Head of the Optical Networks and Systems Department. Since 2000, he has participated in more than 40 R&D projects funded by the EC's Framework Programmes, the Spanish Ministries, and the industry. He has coordinated EU-Japan FP7 project STRAUSS. He has published more than 200 journal and international conference papers.