IEICE TRANS. COMMUN., VOL.E104–B, NO.3 MARCH 2021

INVITED PAPER *Special Section on Fusion of Network Virtualization/Softwarization and Artificial Intelligence towards Beyond-5G Innovative IoT Services*

End-to-End SDN/NFV Orchestration of Multi-Domain Transport Networks and Distributed Computing Infrastructure for Beyond-5G Services

Carlos MANSO[†a)], **Pol ALEMANY**[†], **Ricard VILALTA**[†], **Raul MUÑOZ**[†], **Ramon CASELLAS**[†],
and **Ricardo MARTÍNEZ**[†], *Nonmembers*

SUMMARY The need of telecommunications operators to reduce Capital and Operational Expenditures in networks which traffic is continuously growing has made them search for new alternatives to simplify and automate their procedures. Because of the different transport network segments and multiple layers, the deployment of end-to-end services is a complex task. Also, because of the multiple vendor existence, the control plane has not been fully homogenized, making end-to-end connectivity services a manual and slow process, and the allocation of computing resources across the entire network a difficult task. The new massive capacity requested by Data Centers and the new 5G connectivity services will urge for a better solution to orchestrate the transport network and the distributed computing resources. This article presents and demonstrates a Network Slicing solution together with an end-to-end service orchestration for transport networks. The Network Slicing solution permits the co-existence of virtual networks (one per service) over the same physical network to ensure the specific service requirements. The network orchestrator allows automated end-to-end services across multi-layer multi-domain network segments making use of the standard Transport API (TAPI) data model for both l0 and l2 layers. Both solutions will allow to keep up with beyond 5G services and the higher and faster demand of network and computing resources.

key words: *software-defined networking (SDN), transport networks, network function virtualization (NFV), network slicing*

1. Introduction

Network telecommunications systems are a complex set of different domains (e.g. mobile, fixed, core), each one with its own requirements. At the same time, they must interact with each other to transmit information from one point to another and offer End-to-End (E2E) network services (NSs). Nowadays the design of new networks is based on having the maximum resource flexibility and efficiency possible, evolving from a static network resource management -i.e. services being manually deployed - to a dynamic fashion [1], centralizing the transport networks control plane management and the computing resources allocation. Software-Defined Networking (SDN) allows E2E provisioning of services across multiple domains and layers (WDM, Ethernet, MPLS...) within a single request and across the entire network, enabling on-demand fast resource provisioning of NSs. Network Function Virtualization (NFV) makes use of virtualization technologies -e.g. Virtual Machines (VMs) or Containers- to efficiently manage virtualized network functions (VNFs) on computing resources.

To improve the resource efficiency, Edge architecture aims to complement Cloud architecture by placing small scale Data Centers (DCs) with computing resources (processing, memory and storage) in the access network domain instead of having all the resources in the core network (cloud). Among other benefits, traffic congestion in the transport networks is reduced and NSs with low-latency requirements can be reached. Combining NFV with edge/-cloud architectures improves the resources management and the network flexibility. Using virtualization technologies and the distributed DCs allows to deploy the NSs components -i.e. a chain of VNFs- into the best DC available. Deploying the multiple VNFs of a NS in a distributed way rises the necessity to interconnect them and create E2E NSs, this is done using SDN on transport networks interconnecting edge and cloud domains.

Networks are used by multiple verticals -i.e.sectors like automotive, media, etc.- and so, there is the necessity to ensure that each vertical's requirements are fulfilled. Network slicing allows the creation of virtual networks, each dedicated to a specific service to accomplish the expected requirements of all those verticals using the same physical network. A Network Slice (referred to as Slice) is a chain of interconnected NSs. Network Slicing was initially introduced by NGNM [2] and 3GPP became one of the first organizations to propose an architecture in [3] for Network Slicing. Finally, ETSI proposed to add the 3GPP Network Slicing architecture inside the Operation Support Services and Business Support Services (OSS/BSS) module in its NFV architecture [4].

Applying all the previous solutions (edge/cloud computing, distributed E2E NSs and Slices) on traditional networks is difficult due to the dynamic and flexible requirements. NFV allows to fulfill them through its management and orchestration (MANO) architecture [5].

Transport networks comprise multiple network segments with different technologies (e.g. WDM, MPLS-TP, Ethernet) and network layers. This makes control over the network a very hard task, as the control plane of traditional networks depends entirely on the manufacturer proprietary

---

Manuscript received May 30, 2020.
Manuscript revised July 16, 2020.
Manuscript publicized September 11, 2020.

[†]The authors are with the Optical Networks & Systems Department, Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Castelldefels (Barcelona), 08860 Spain.
a) E-mail: carlos.manso@cttc.es

DOI: 10.1587/transcom.2020NVI0001

Copyright © 2021 The Institute of Electronics, Information and Communication Engineers

implementation.This way is nearly impossible to automate the entire transport network, requiring, in most cases, direct human intervention, making it a slow and complex process.

This is becoming a bigger issue each day, as user traffic demand grows exponentially. Also, the adoption of virtualization technologies in DCs and the dynamic nature of its deployment and requirements of traffic flows, makes on-demand, E2E, connectivity services over multi-layer and multi-domain transport networks, a really useful and even necessary step towards a more efficient network [6].

SDN is a group of technologies that aims to solve these needs, enabling dynamic configuration of network devices. It separates the concepts of the data plane (forwarding process of data) from the control plane (routing process), and gives the latter an interface to be dynamically modified by external applications. Although the control of a segment of the network is centralized on the SDN controller, in order to provide the necessary orchestration between them, the authors in [7] introduced an E2E Transport SDN controller capable to orchestrate all the segments within the transport network. In this paper, we have taken this concept further relying on the SDN controller of each segment the path calculation within its domain, leaving the E2E Transport SDN controller only the responsibility of path computation between segments and orchestrating the provisioning at each segment.

The objective of this article is to present a Network Slicing solution over a NFV architecture that will allow to properly control and orchestrate distributed computing resources as well as an E2E SDN orchestrator for multi-domain multi-layer transport networks. Both solutions use multiple known interfaces and components such as Transport API (TAPI), OpenDaylight (ODL), OpenStack, Kubernetes, etc. The whole architecture is first presented and then validated in the CTTC ADRENALINE testbed, covering multiple transport network segments (edge, metro, intra-DC, and vehicular) and providing an integrated real-world solution to manage E2E network slices over multi-layer multi-domain transport networks.

The article is organized as follows: Section 2 presents the ADRENALINE testbed, where the previously introduced architecture will be assessed. Sections 3 and 4 describe in detail the implementation of Network Slicing and the SDN/NFV architecture, the standards used and the improvements made to the E2E Transport SDN controller. Afterwards, Sect. 5, shows the experimental validation of the architecture in the ADRENALINE testbed by showing an E2E Network Slice deployment process with inter-domain links requested to the E2E Transport SDN controller. Finally, in Sect. 6 the conclusions are presented.

2. ADRENALINE Testbed

The ADRENALINE testbed has been developed as a disaggregated network model. This model allows a major flexibility and efficiency when managing networks because of the separation of hardware and software. So, the system does not depend on specific components and it becomes a white box able to accept different elements for the same purpose. It is designed following the latest technological standards and architectures with optical/packet transport networks integrated with cloud/edge computing architectures (Fig. 1) or the combination of SDN and NFV presented in Fig. 2.

2.1 Optical/Packet Transport Network and Cloud/Edge Computing Architecture

Figure 1 presents the ADRENALINE physical architecture. It is composed of the packet and optical transport networks and the distributed computing resources.

The transport network of the testbed has a multi-layer multi-domain architecture, composed of 4 packet transport networks (PTN) and 1 photonic mesh network (PMN). The packet domains cover the edge, metro, intra-DC and vehicular segments. There are 13 OpenFlow switches, deployed on Commercial off-the-shelf (COTS) hardware controlled by Open vSwitch (OVS). To provide connectivity between the optical and packet domains, five switches are equipped with a tunable 10 Gb/s 10 gigabit small Form-factor Pluggable (XFP) to access the PMN. A fix Small Form-factor Pluggable (SFP+) is used in the vehicular PTN to provide connectivity to the 5GBARCELONA network, an open, global, and neutral project for the validation and adopting of 5G technologies and applications through tests in the real environment of the city of Barcelona. A programmable SDN-enabled Sliceable Bandwidth/bitrate variable transceiver (S-BVT) [8] is also used to transmit multiple flows at variable data rate, and different multicarrier technologies, OFDM or discrete multi tone (DMT).

The optical domain is composed by a PMN and a passive optical network (PON). The PMN is equipped with 2 Reconfigurable optical add-drop multiplexer (ROADMs) and 2 Optical cross connect (OXCs) nodes. They have 5 bidirectional flexi/fix grid Dense wavelength-division multiplexing (DWDM) amplified optical links up to 150 km, deploying a total of 610 km of optical fiber. The PON is equipped with a 19-core 25 km multi core fiber (MCF) and a bundle of single mode fiber (SMF). The PMN is managed by an Open Line System (OLS) controller, with an SDN controller orchestrating both the OLS and the S-BVT, following a partial optical disaggregated architecture.

Regarding the computing resources placed alongside the whole testbed network, Fig. 1 presents a combination of Cloud/Edge computing nodes placed in different network domains. In total there are seven Points of Presence (PoPs) or DCs: a) Edge-DC1, Edge-DC3 and the Cloud-DC use OpenStack installations, the first two are single-node (controller and compute in one node) while the last one is a multi-node (one controller and two compute nodes). b) Edge-DC2, Edge-DC4, MEC-1 and MEC-2 use Kubernetes installations, and in particular for the last two DCs, they also have an OpenFlow switch as they designed to be used as edge domain for a vehicular scenarios.

When using multiple and different technologies, there

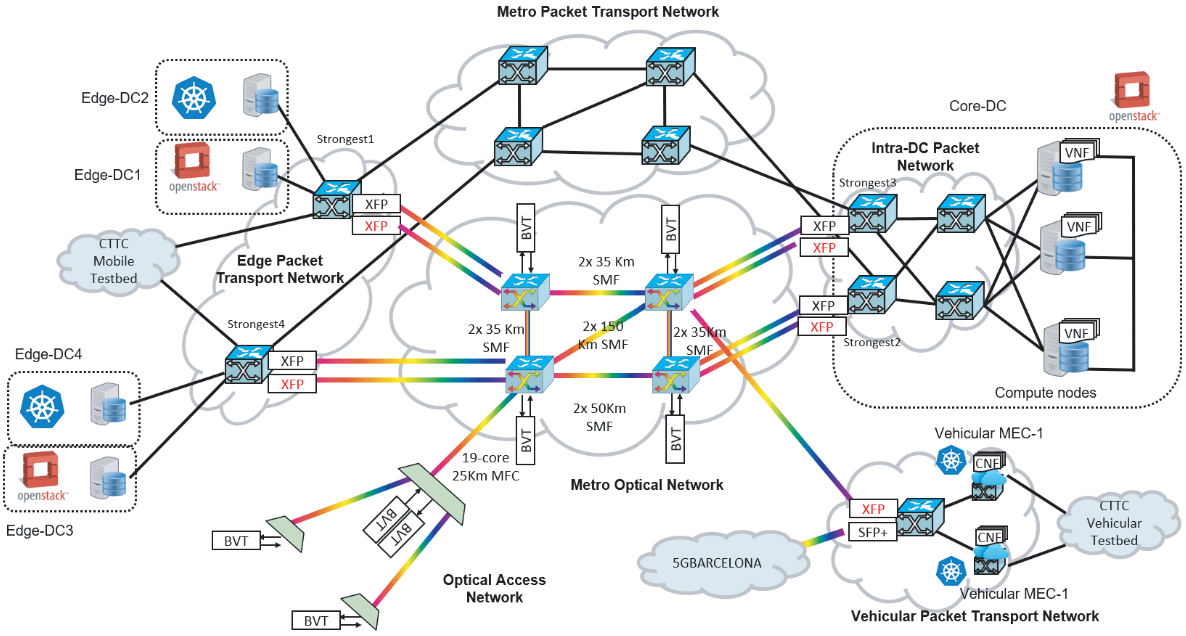


Fig. 1 ADRENALINE testbed physical network architecture.

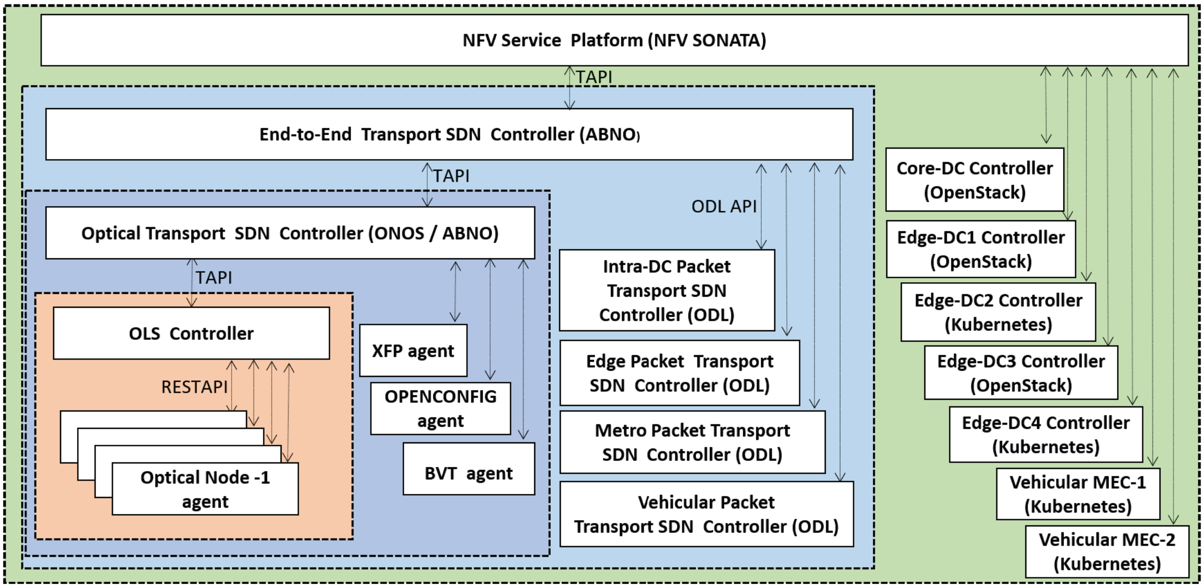


Fig. 2 ADRENALINE testbed E2E transport SDN/NFV architecture.

is always a risk on whether they may conflict with each other, but as the software used is quite well known and already used in deployment scenarios, the main difficulties were related to the software compatibilities and the integration with the Service Platform to manage them.

2.2 SDN/NFV Control Orchestration Architecture

Figure 2 shows the SDN/NFV architecture used in the ADRENALINE testbed following the NFV reference defined in [9]. Based on this, the SDN/NFV architecture is structured in two main blocks: the E2E Transport SDN Transport controller to manage connectivity services across packet/optical domains, and the NFV Service Platform (SP) to manage network services and Slices.

Regarding the implementation of the NFV architecture within the ADRENALINE testbed, on top of Fig. 2 there is the NFV SP acting as the NFV-O and Virtual Network Function Manager (VNFM). The NFV SP is connected to other two elements: the End-to-End Transport SDN Controller previously presented acting as the WAN Infrastructure Manager (WIM), and a set of Virtual Infrastructure Managers (VIMs) to control the DC’s computing resources. With these two elements, the selected NFV SP main features are: a)

NFV elements management to control any related action during the life cycle of either Slices, NSs, VNFs or Cloud-native Network Functions (CNFs), b) Multi-VIM management as the testbed has multiple PoPs based on two different technologies -i.e. OpenStack for VMs and Kubernetes for containers-, and c) End-to-end connectivity to create the interconnection among different NFV elements based either on VMs or containers deployed in different VIMs through the E2E Transport SDN Controller.

The E2E Transport SDN controller is the entity responsible of the life cycle management of the E2E transport connectivity services, the composition and retrieval of the overall topology and the path computation services. It is a proprietary implementation developed by CTTC based on the IETF Application-Based Network Operations (ABNO) architecture presented in [10]. It is based on a hierarchical approach where the E2E Transport SDN controller orchestrates the ODL controllers from the packet domains using its own Application Programming Interface (API). It also orchestrates the optical SDN controller, which at the same time is the optical domain orchestrator.

The optical SDN controller (ABNO or ONOS) orchestrates the optical domain managing the OLS, a proprietary SDN optical controller, presented in [11] and [12], as well as the agents for the optical lasers, following a optical partially disaggregated model. The TAPI is used as the South-Bound Interface (SBI) for the OLS controller and ABNO/ONOS, as well as for the North-Bound Interface (NBI) of the E2E transport SDN controller for the NFV SP.

3. End-to-End Network Slicing Orchestration for Distributed Cloud/Edge Computing Infrastructures

This section is divided into two main parts: firstly, the internal NFV SP elements are presented and secondly, the management and orchestration of the VIMs resources placed in the multiple domains.

3.1 NFV Service Platform Architecture

Currently there are multiple NFV-O options available such as as Open Source MANO (OSM), Open Automation Network Platform (ONAP), NFV SONATA, etc. NFV SONATA was selected as the testbed NFV-O due to the following reasons: 1) it can orchestrate multi-domain (edge/-cloud) networks, 2) it is compatible with the E2E Transport SDN Controller and 3) it has multiple features available -e.g. network slicing-.

The SONATA SP architecture is detailed in [13], this subsection focuses on two of its internal components. First, the Network Slice Manager (NSM) and its internal architecture [14], the data objects to manage Slices and how this component is integrated within the NFV SONATA SP. Second, the WIM TAPI plugin for the IA component to be integrated into the network architecture presented in Sect. 2.1 and communicate with the E2E Transport SDN Controller.

3.1.1 Network Slice Manager

The NSM is in charge to manage any action related to Slices. Its main features are:

- Shared NS: A 5G networks objective is an efficient resource management of the existing resources. Even though each Slice is designed to be deployed for a specific service, there might be situations in which different Slices might share NSs. When deploying a Slice with a shared NS, the NSM must check if there are deployed Slices with that NS. If so, the new Slice is attached to the existing NS.
- Multi-VIM Deployment: A network is not anymore a set of nodes interconnecting the user to the backbone where there is a massive DC. Nowadays, networks tend to distribute DCs from the access to the core domains and all of them with an NFV/SDN architecture. The Network Slice Manage must be able to deploy in different parts of a network [15] to create E2E Slices in a multi-domain network.
- Hybrid Slices: Currently there are different technologies to create virtual network elements (e.g. OpenStack, Kubernetes). For this reason, the NSM must manage the instantiation of NSs based on different virtualization technologies [16].
- NS composition: Interconnecting NSs is the most essential and complex feature as its evolution depends on the three previous features. The NSM must consider multiple factors to interconnect NSs: whether the NS is shared, the virtualization technology (e.g. Hybrid Slices) used, where to deploy each NS element (e.g. Multi-VIM deployment), etc.

In order to manage the Slices life cycle, it is necessary to define what is possible to deploy (available options) and to register what is deployed (used options). To keep control of these two questions, Network Slicing makes use of two data objects:

- Network Slice Template (NST): Originally defined by the GSMA [17] and later on adopted by the ETSI, it contains the NSs list and how they are interconnected among other information.
- Network Slice Instance (NSI): The deployments records, each based on a NST. As defined in [18], it keeps the information to manage the virtual resources deployed in each VIM composing a Slice.

The NSM architecture has evolved in order to accomplish a correct integration with the other NFV SONATA SP components. First, the local Data Bases (DBs) were removed in order to push all the data safeness into the Catalogue and Repositories modules, so the NSM could be re-started without losing information if necessary. Furthermore, the NSM had to be integrated according to how NFV SONATA SP works: any request and process must pass through the Gatekeeper component which keeps control of all the asynchronous processes and adds certain security level. Finally, to have the previously described features, it was necessary to adapt the way a Slice was created depending on the virtualization technology involved. Always through the Gatekeeper module, the NSM had to: a) request to the IA component the virtual links (VLs) creation in each VIM to connect the NS instances, b) request, once the VLs are ready, to the MANO Framework component to instantiate the NSs composing the Slice and c) request to the IA component (through the WIM Transport plugin) the creation of the inter-domain VL (a service connectivity) to interconnect the NSs to complete the E2E Slice. This last action happens only when the NSs are distributed in multiple VIMS.

3.1.2 WIM Transport API Plugin

Due to its NFV-O and VNFM functionalities, the NFV SONATA SP has to interact with different VIMs and WIMs. These elements will probably be based on different technologies, that is the reason for the IA module to exist. Located in the lowest NFV SONATA SP architecture layer, it has a SBI implementing different APIs to communicate with different VIMs and WIMs. On the VIMs side, the IA generates HEAT, Neutron and Keystone templates to orchestrate OpenStack resources, for DC networking and authentication respectively. On the WIMs side, the IA makes use of the WIM TAPI plugin to request the E2E Transport SDN Controller the service connectivity creation to connect the NSs placed in the different VIMs.

The WIM TAPI makes use of the Open Networking Foundation (ONF) TAPI specification to create a Wide Area Network (WAN) connection by focusing on the enforcement of the E2E connectivity between PoPs [19]. To do so, it is necessary to setup the networking connectivity between VNFs and WAN terminations by configuring the correct data flow with a VL over the physical infrastructure.

In the testbed network presented in Fig. 1, different PoPs are connected to a fabric of SDN-controlled virtual switches combined with several optical switches and managed using a hierarchical SDN control plane with the E2E Transport SDN Controller on the top. When the creation of a connection between a VNF and a WAN termination is requested through the WIM TAPI, the E2E Transport SDN Controller receives the request with the Quality of Service (QoS) requirements and establishes the path either on the packet or the optical domain. As each WAN termination and VNF connection is composed by two unidirectional flows, it is possible to perform load-balancing or priority routing.

3.2 Management of Distributed Cloud/Edge Infrastructure Combining VMs and Containers

Based on the available market options, the ADRENALINE testbed has two different VIM software types installed in the multiple PoPs presented in Sect. 2.1:

- OpenStack manages cloud computing resources in a flexible and efficient way through the use of VMs, which in the NFV environment are mapped as VNFs.
- Kubernetes makes use of containers to deploy the network functions of a NS. The use of containers technology instead of VMs lead to the concept of CNFs.

The NFV SP, is able to manage both VIMs types by requesting (through the Gatekeeper) to the MANO Framework and the IA components introduced in Sect. 3.1). These two components contain a set of different plugins to communicate with the different VIMs and the WIM (3.1.2).

3.3 End-to-End Network Slicing Management Composing Multiple NFV Network Services

The E2E Slice deployment follows the same actions either if it is an Hybrid Slice [16] or not. Figure 3 shows the steps to deploy an E2E Slice in a multi-VIM scenario: First, the vertical requests the NSM within the NFV SP an E2E Slice based on a NST (Fig. 3-(1)). Once the NSM receives the request, it creates the NSI record and and completes it with the NSs placement information (Fig. 3-(2)). With the NSI ready, the NSM requests the SONATA SP MANO (MANO) module the deployment of each NS (Fig. 3-(3)) specifying where each VNF must be deployed -i.e. OpenStack VIM (Fig. 3-(4/5)) or Kubernetes VIM (Fig. 3-(6/7))-. When a NS is requested (Fig. 3-(8)), the NSM updates the NSI record and follows with the missing NSs (Fig. 3-(3)) or it waits until all are deployed (Fig. 3-(9)) to update the NSI record. If the NSs are placed in different VIMs, then the NSM re-

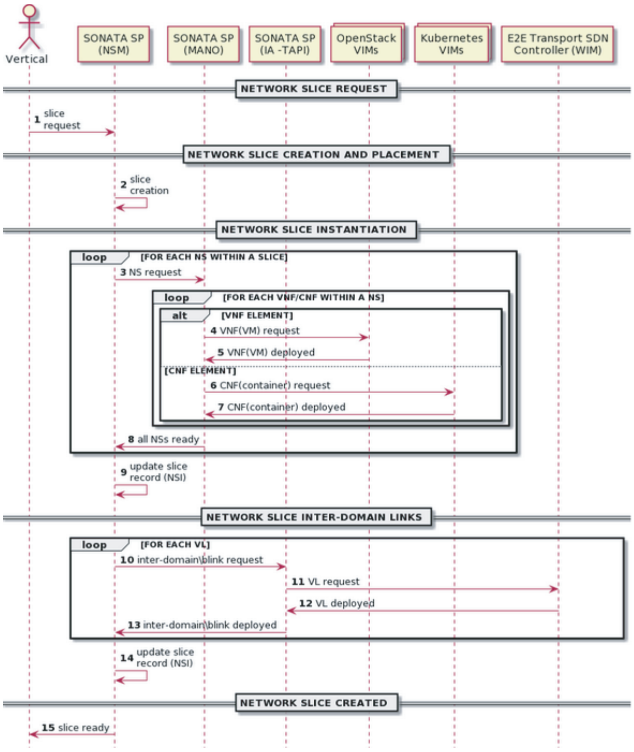


Fig. 3 Network slice deployment.

quests the SONATA SP IA and its internal WIM TAPI plugin (IA-TAPI) to create the inter-domain links to connect the NSs (Fig. 3-(10)), then the IA-TAPI requests (Fig. 3-(11)) the E2E Transport SDN Controller (WIM) to configure the virtual links. The WIM, then follows the process described in Sect. 4.4 and once they are done, the WIM answers back to the IA-TAPI (Fig. 3-(12)) and this to the NSM (Fig. 3-(13)). Finally the NSM, updates the NSI record (Fig. 3-(14)) and informs the vertical (Fig. 3-(15)).

The previous process involves a Slice with a VNF-based NSs. When deploying an Hybrid Slice (VNFs or CNFs composing NSs) and once VNFs and CNFs become VMs and containers respectively, it must be considered how each technology works in order to interconnect them. As described in Sect. 3.2, a VM is independent from other VMs (e.g. each VM has its own IP address) but containers no (i.e. containers are identified by port). Because of these reasons, when deploying an Hybrid Slice it is important that all VNFs are deployed before the CNFs in order to pass the correct VMs information to the containers as done in [16].

4. End-to-End Transport SDN Orchestration for Multi-Domain Transport Networks

The E2E Transport SDN controller provides E2E connectivity to support the NFV SP services, among others.

4.1 End-to-End Transport SDN Controller Architecture

The TAPI-enabled SDN controller architecture is organized in different components, depicted in Fig. 4, which are the following:

- Connectivity Service Orchestrator (CSO): It handles the requests from the NBI coming from the NFV-O. It process the solicited TAPI service and manages the required workflow making requests to the other components.
- Connection Manager: It handles the SBI requests the controllers at his charge. It also keeps a database with the state of the services and connections instantiated.
- Topology Manager: It handles the retrieval and storage of all the information about the topology from SDN controllers. It provides all the data available from the domain controllers, e.g. nodes and ports available and the technology provided (Ethernet, WDM...). The internal topology stored is based on the TAPI model.
- Virtual Network Topology Manager (VNTM): It handles the creation of virtual links across the optical segments that connects packet networks.
- Path Computation Element (PCE): It handles the process of finding the best multi-layer path across the different domains, taking into account the available bandwidth at the inter-domain links. It performs domain selection, as the actual computation and allocation of the path within the domain are performed by the domain controller. It is integrated as a TAPI-enabled component, where the CSO gives the topology, and then asks for a path, both as TAPI objects.

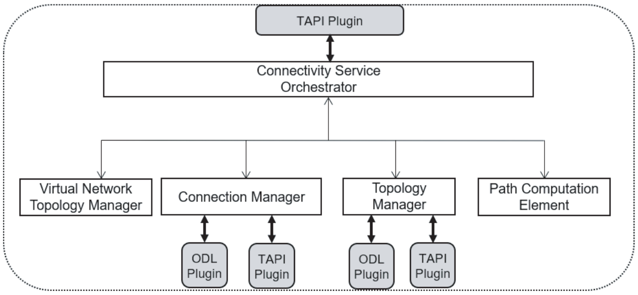


Fig. 4 E2E transport SDN controller architecture.

The Connection and Topology managers are extensible modules using a plugin architecture. They are capable of supporting multiple controllers like ODL, TAPI, and more.

4.2 Transport API (TAPI)

The ONF TAPI is a technology-independent information model that enables interoperability with other SDN components. It provides a common interface for SDN control on layers 0, 1 and 2, defining a transport network common model.

Technically, it is a model defined in UML/YANG for multi-layer transport networks, with an API specified using OpenAPI Specification (OAS) which provides HTTP endpoints for requests using the RESTCONF protocol. It provide consumers with the following features:

- Topology services: It allows the retrieval of the topology information from the SDN controllers and export to upper components, in terms of nodes, ports and links.
- Connectivity services: It allows the request, update, delete and retrieval of connectivity services over the transport network.
- Notification services: It allows subscription to controller notifications such as state changes or failures.
- Path computation services: It allows path computation services across the transport network.
- Virtual network services: It allows the creation of virtual network topologies inside the transport network.

Although the Notification and Virtual network services are not implemented in this architecture, the most important features of the other services are implemented and sufficiently support the workflows needed. It is also extensible, as augmentations can be defined to add more information to any part of the topology (node, link, port... etc).

4.3 Topology Retrieval, Abstraction and Exposure

Figure 5 shows the composition of the internal topologies. At start (step 1), the Topology Manager initializes the controller plugins (step 2), get the topologies from the SDN

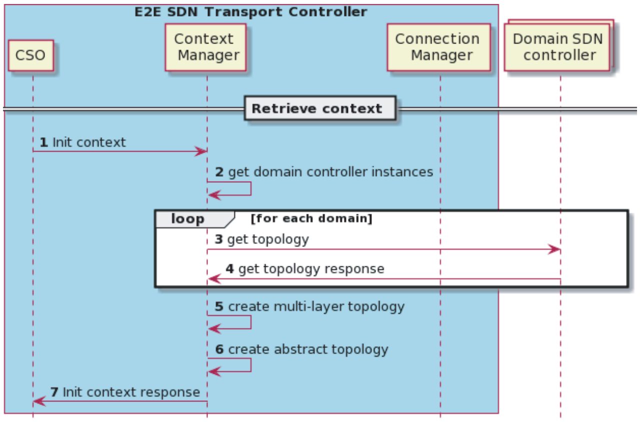


Fig. 5 Initialization workflow.

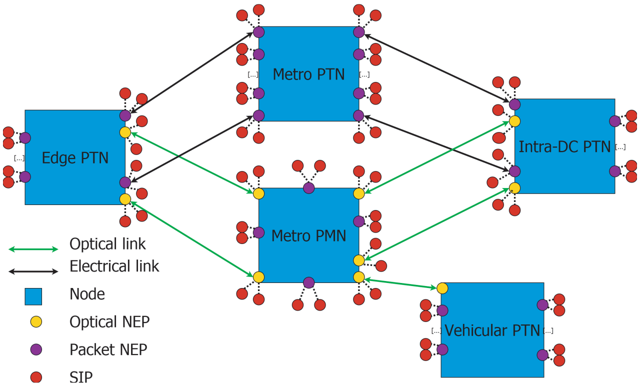


Fig. 6 Adrenaline abstract multi-layer and multi-domain TAPI topology.

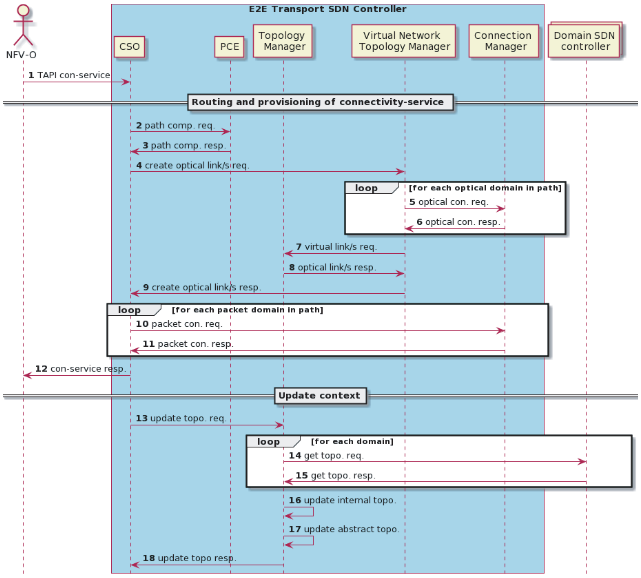


Fig. 7 Connectivity service provisioning workflow.

controllers and stores them in the internal database of the Topology Manager (steps 3, 4). These plugins act as an interface to the SDN controllers that the Topology Manager can interact with, translating its data model to the TAPI model of our controller and decoupling both systems. In the ADRENALINE, two plugins have been developed: one for the packet domain supporting the ODL API, a Remote Procedure Call (RPC) based on HTTP protocol; another one to interact with the optical SDN controller based on TAPI. The internal topology based on the TAPI model is expressed in terms of nodes and links. Nodes aggregate Node Edge Points (NEPs) acting as node ports. They contain information about whether it has Service Interface Points (SIPs) mapped, and information about the protocol they support. In the case of DWDM, they can be augmented with a standard extension specifying the upper and lower frequency of each slot, the grid type and the granularity. Links interconnect two nodes and terminate on NEPs, which may be mapped to one or more SIPs at edge of network. The SIPs are the endpoints where a connectivity service can be requested from, one for input and output directions of the NEP. As SIPs are a TAPI concept, when retrieving the topology from SDN controllers with other data models (as with ODL domains), they have to be generated from the NEPs/ports retrieved.

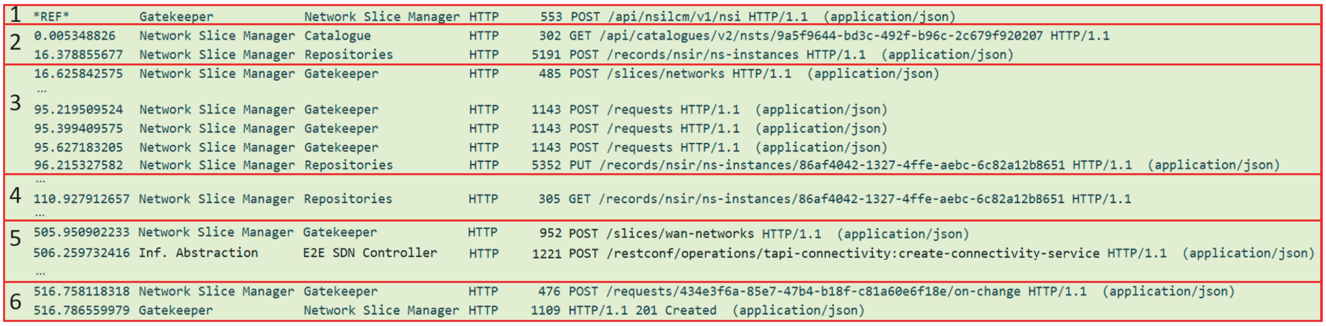
After the domain topologies have been retrieved and translated into the internal TAPI model, two additional topologies are created. One multi-domain topology containing all the lower domains and the inter-domain links (step 5), creating a global view of the packet and optical transport network. Inter-domain links are defined by means of a static configuration file. These inter-domain links can be of different priorities, to prioritize the use of some in detriment of others when calculating inter-domain routes. Also, an abstract topology is composed (step 6) by all the lower-domain SDN controllers and the inter-domain links, but where each domain is seen as a single node, allowing to an easier and faster path computation between the endpoints.

Figure 6 shows how the abstract topology is seen in ADRENALINE. It is composed by each of the domains seen as a single node. This node-domain contains all nodes, NEPs and SIPs of the same domain aggregated into it.

4.4 End-to-End Transport Connectivity Service Management

Figure 7 shows how a typical connectivity service is provisioned along the transport network.

A request for the provisioning, between two SIPs, of a connectivity service coming from a user (NFV-O or another TAPI-capable user) is received through the NBI (step 1). The CSO, the main module, receives the request and calls the other modules as needed. First, the CSO requests the PCE to compute a multi-domain multi-layer path between the SIPs that is capable of supporting the bandwidth requested, using the “ tapi −path−computation:input” object (step 2). The PCE processes the request using the abstract topology given by the CSO and returns a path between the domains (step 3). Then, the CSO segments the computed multi-layer path into domain segments, specifying for each segment (packet or optical) the associated SIPs. If the route



```
1 *REF*          Gatekeeper           Network Slice Manager HTTP      553 POST /api/nsilcm/v1/nsi HTTP/1.1  (application/json)
2 0.005348826    Network Slice Manager Catalogue            HTTP      302 GET /api/catalogues/v2/nsts/9a5f9644-bd3c-492f-b96c-2c679f920207 HTTP/1.1
  16.378855677   Network Slice Manager Repositories         HTTP     5191 POST /records/nsir/ns-instances HTTP/1.1  (application/json)
  16.625842575   Network Slice Manager Gatekeeper           HTTP      485 POST /slices/networks HTTP/1.1  (application/json)
  …
3 95.219509524   Network Slice Manager Gatekeeper           HTTP     1143 POST /requests HTTP/1.1  (application/json)
  95.399409575   Network Slice Manager Gatekeeper           HTTP     1143 POST /requests HTTP/1.1  (application/json)
  95.627183205   Network Slice Manager Gatekeeper           HTTP     1143 POST /requests HTTP/1.1  (application/json)
  96.215327582   Network Slice Manager Repositories         HTTP     5352 PUT /records/nsir/ns-instances/86af4042-1327-4ffe-aebc-6c82a12b8651 HTTP/1.1  (application/json)
  …
4 110.927912657 Network Slice Manager Repositories          HTTP      305 GET /records/nsir/ns-instances/86af4042-1327-4ffe-aebc-6c82a12b8651 HTTP/1.1
  …
5 505.950902233 Network Slice Manager Gatekeeper            HTTP      952 POST /slices/wan-networks HTTP/1.1  (application/json)
  506.259732416 Inf. Abstraction      E2E SDN Controller    HTTP     1221 POST /restconf/operations/tapi-connectivity:create-connectivity-service HTTP/1.1  (application/json)
  …
6 516.758118318 Network Slice Manager Gatekeeper            HTTP      476 POST /requests/434e3f6a-85e7-47b4-b18f-c81a60e6f18e/on-change HTTP/1.1  (application/json)
  516.786559979 Gatekeeper            Network Slice Manager HTTP     1109 HTTP/1.1 201 Created  (application/json)
```

Fig. 8 Network slice deployment.

returned by the PCE has an optical segment, the CSO then asks the VNTM to provision the virtual link (step 4). First, the VNTM calls the Connection Manager to request the provisioning of the optical connection, specifying the associated SIPs (step 5). Then, the VNTM calls the Toplogy Manager to create a new virtual link between the NEPs of the packet domains connected by the provisioned optical connection, containing the bandwidth needed for the virtual link (steps 7, 8). Then, the CSO, asks the Connection Manager again to provision the rest of packet connections to the involved SDN controllers along the calculated route, specifying for each packet segment the associated SIPs (steps 10, 11). Those SIPs need to be translated into the format the SDN controller data model understands, in case they are not TAPI-enabled. Finally, if all the packet connections were properly provisioned at their respective domains, the response is given to the NFV-O, the topologies are again retrieved and the internal and abstract multi-layer multi-domain topologies are updated (steps 12-18).

In this way, a multi-domain E2E service can be provisioned, relying on the SDN controllers responsibility to effectively provision the connection at its segment. The E2E Transport SDN controller is then accountable for path computation and connectivity between domains. The E2E service can also be multi-layer, as each SDN controller is responsible of the specifics of the connection on its domain and optical links are abstracted to appear as packet links.

5. Experimental Validation

Having described the ADRENALINE architecture in Sect. 2, this section presents the experimental validations for the processes in Figs. 3 and 7.

The specifications of each node in the architectures of Sects. 2.1 and 2.2 are: the NFV SONATA SP has Ubuntu 16.04.5 LTS with an Intel(R) Xeon(R) CPU E5-2603 v4 @ 1.70 GHz, 32 GB of RAM and 1TB of storage disk. All the nodes with computing resources placed in the Core or Edge DCs have Centos 7 with an Intel(R) Xeon(R) CPU E5-2420 0 @ 1.90 GHz CPU, 64 GB of RAM and 2 TB of disk storage. The OpenStack version is Pike while the Kubernetes version is the v1.16.1.

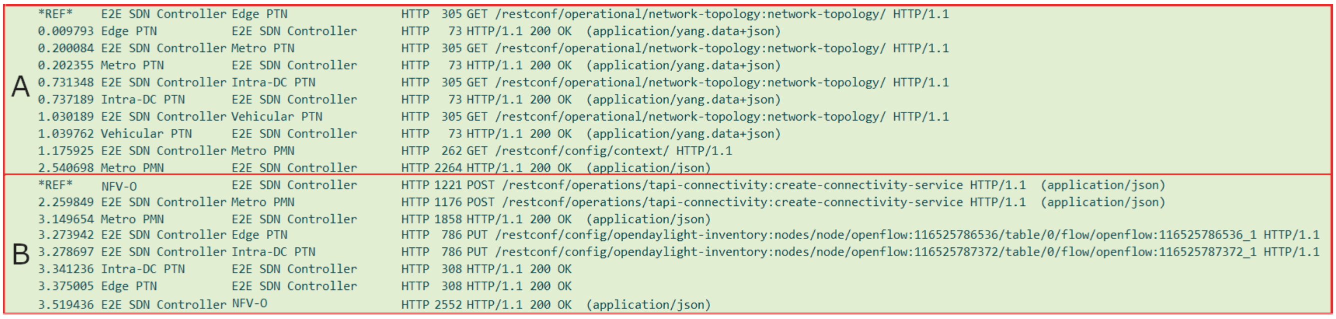
5.1 Network Slicing Service Validation

Figure 8 presents the deployment procedure of a Slice with three NSs placed across the Edge-DC1 and the Cloud DC described in Sect. 2.1. The vertical requests the NSM (using the Portal and through the Gatekeeper) an E2E Slice based on a NST (Fig. 8(1)). The NSM gets the NST descriptor, processes the request to create the NSI record and saves it into the Repositories in less than 17s (Fig. 8(2)). With the NSI record ready, the NSM takes around 80s (Fig. 8(3)) to request: the VLs creation (/slices/networks) to connect the VMs; the NSs deployment (/requests) to create the VMs in the correct VIM; and finally, the NSI record (/records/nsir/ns-instances/...) update. While deploying, the NSM keeps checking that all the NSs are ready (Fig. 8(4)). Once ready, after around 409s, the NSM requests the IA (through the Gatekeeper) all the necessary inter-domain links creation. Then, the IA uses the WIM TAPI Plugin to request the E2E SDN Controller to create the inter-domain links over the physical network, requiring around 11s to do it (Fig. 8(5)). Finally, when the E2E Slice is ready, the NSM updates the NSI information and informs the Gatekeeper that the whole process is done (Fig. 8(6)). The whole process took 516 seconds -i.e. 8 min and 36 seconds- to deploy 3 NSs composed by 2VNFs each -i.e. 6 VMs- placed in different VIMs and to interconnect them.

5.2 Transport Connectivity Service Validation

The establishment of a connectivity service between the intra-DC and the edge PTNs is going to be validated. At the initialization of the E2E Transport SDN controller, it retrieves the topologies configured to be responsible of. In Fig. 9(A) (corresponds to Fig. 5) the wireshark capture of the message exchange between the SDN controllers and the E2E Transport SDN controller is showed. It can be observed that initialization requires around 2.5 seconds in order to retrieve all the topologies.

Figure 10 shows the connectivity service requested by the NFV SONATA SP. The service provided should be capable of supporting 1 GBPS. It shows the extension “cttc-ext:match”, introduced in Sect. 4.2, to be compatible with the ODL API. It allows to describe a flow between two IP



```
*REF*    E2E SDN Controller Edge PTN               HTTP  305 GET /restconf/operational/network-topology:network-topology/ HTTP/1.1
0.009793 Edge PTN           E2E SDN Controller     HTTP   73 HTTP/1.1 200 OK  (application/yang.data+json)
0.200084 E2E SDN Controller Metro PTN              HTTP  305 GET /restconf/operational/network-topology:network-topology/ HTTP/1.1
0.202355 Metro PTN          E2E SDN Controller     HTTP   73 HTTP/1.1 200 OK  (application/yang.data+json)
0.731348 E2E SDN Controller Intra-DC PTN           HTTP  305 GET /restconf/operational/network-topology:network-topology/ HTTP/1.1
0.737189 Intra-DC PTN       E2E SDN Controller     HTTP   73 HTTP/1.1 200 OK  (application/yang.data+json)
1.030189 E2E SDN Controller Vehicular PTN          HTTP  305 GET /restconf/operational/network-topology:network-topology/ HTTP/1.1
1.039762 Vehicular PTN      E2E SDN Controller     HTTP   73 HTTP/1.1 200 OK  (application/yang.data+json)
1.175925 E2E SDN Controller Metro PMN              HTTP  262 GET /restconf/config/context/ HTTP/1.1
2.540698 Metro PMN          E2E SDN Controller     HTTP 2264 HTTP/1.1 200 OK  (application/json)
*REF*    NFV-O              E2E SDN Controller     HTTP 1221 POST /restconf/operations/tapi-connectivity:create-connectivity-service HTTP/1.1  (application/json)
2.259849 E2E SDN Controller Metro PMN              HTTP 1176 POST /restconf/operations/tapi-connectivity:create-connectivity-service HTTP/1.1  (application/json)
3.149654 Metro PMN          E2E SDN Controller     HTTP 1858 HTTP/1.1 200 OK  (application/json)
3.273942 E2E SDN Controller Edge PTN               HTTP  786 PUT /restconf/config/opendaylight-inventory:nodes/node/openflow:116525786536/table/0/flow/openflow:116525786536_1 HTTP/1.1
3.278697 E2E SDN Controller Intra-DC PTN           HTTP  786 PUT /restconf/config/opendaylight-inventory:nodes/node/openflow:116525787372/table/0/flow/openflow:116525787372_1 HTTP/1.1
3.341236 Intra-DC PTN       E2E SDN Controller     HTTP  308 HTTP/1.1 200 OK
3.375005 Edge PTN           E2E SDN Controller     HTTP  308 HTTP/1.1 200 OK
3.519436 E2E SDN Controller NFV-O                  HTTP 2552 HTTP/1.1 200 OK  (application/json)
```

Fig. 9 Topology retrieval (A) and connectivity service (B)

```
{
  "connectivity-constraint": {
    "requested-capacity": {
      "total-size": {
        "value": 1,
        "unit": "GBPS"
      }
    },
    "connectivity-direction": "BIDIRECTIONAL",
    "cttc-ext:match": {
      "ipv4-target": "76.76.87.98",
      "link-layer-type": "2048",
      "ipv4-source": "12.12.13.14"
    }
  },
  "uuid": "877a2b6d-34ba-4754-b4eb-289324f69783",
  "end-point": [
    {
      "layer-protocol-name": "ETH",
      "direction": "BIDIRECTIONAL",
      "service-interface-point": {
        "service-interface-point-uuid": "da24fd5c-cc77-4e61-9409-0a3c935606ed"
      }
    },
    {
      "layer-protocol-name": "ETH",
      "direction": "BIDIRECTIONAL",
      "service-interface-point": {
        "service-interface-point-uuid": "eb05bdf8-14a4-485f-9eb4-10896ecf6a9b"
      }
    }
  ]
}
```

Fig. 10 TAPI connectivity service request.

client addresses, thus making both endpoints visible to each other. The direction of the service should be bidirectional and within Ethernet standard. The workflow expected is shown in Fig. 7.

The NFV SONATA SP requests a TAPI connectivity service using the "restconf/operations/tapi-connectivity:create-connectivity-service" HTTP URI. Once received, the CSO first asks the PCE to compute a path between the SIPs, using the abstract topology, ensuring the inter-domain links are capable of supporting the bandwidth requested (1 GBPS in this case). A inter-domain path is feasible, going through the Metro PMN.

The CSO requests the VNTM to provision the optical link (by means of the Connection Manager), shown in Fig. 9(B) (corresponds to Fig. 7), and also to create a virtual link that goes through the Metro PMN. The CSO then asks the Connection Manager to provision the connection at each of the packet domains the path computed traverse. This way, a multi-layer, and multi-domain connectivity service is finally provisioned. As seen in Fig. 9(B), all three connections are requested and performed in 3.5 seconds; 0.9 for the Metro PMN, 0.1 for the Edge PTN and 0.07 for the Intra-DC PTN, while the rest 2.4 seconds are spent on internal processing.

6. Conclusions

We have introduced a solution to orchestrate edge/cloud computing resources capable of services using NFV orchestration and network slicing technologies. First an E2E Slice deployment process has been presented as an efficient network and computing resources management solution, taking less than 10 minutes to have all network functionalities deployed and ready to be used, using its own virtual network. To be able to deploy these E2E services on multi-layer and multi-domain networks, an E2E Transport SDN controller has also been demonstrated, deploying the connectivity service in less than 4 seconds.

To assess the reliability of these technologies, this article has presented the ADRENALINE testbed, a multi-layer SDN transport network to connect a multi-domain edge/-cloud computing architecture. The testbed also contains a Network Slicing solution over an NFV architecture to manage and orchestrate E2E Network Slices with virtual services distributed in multiple DCs and using different virtualization technologies. Both solutions have been proved useful and feasible on a multi-layer and multi-domain transport network, ready to be deployed to support the new 5G services and architecture.

Acknowledgments

Work partially funded by the EC through the 5GPPP INSPIRE-5GPlus (871808), GENCAT FemIoT and MINECO AURORAS (RTI2018-099178-B-I00) projects.

References

[1] V. Lopez, J.M. Gran Josa, V. Uceda, F. Slyne, M. Ruffini, R. Vilalta, A. Mayoral, R. Munoz, R. Casellas, and R. Martinez, "End-to-end service orchestration from access to backbone," IEEE/OSA J. Opt. Commun. Netw., vol.9, no.6, pp.B137–B147, 2017.

[2] NGNM, "Description of network slicing concept," Technical Report, NGMN Alliance, Jan. 2016.

[3] 3GPP, "Study on management and orchestration of network slicing for next generation network," standard, 3rd Generation Partnership Project (3GPP), Jan. 2018.

[4] ETSI, "Network functions virtualisation (NFV) release 3; evolution and ecosystem; report on network slicing support with etsi NFV

architecture framework," standard, European Telecommunications Standards Institute (ETSI), Sophia Antípolis, Fr, Dec. 2017.

[5] ETSI, "Network functions virtualisation (NFV); management and orchestration," standard, European Telecommunications Standards Institute (ETSI), Sophia Antípolis, Fr, Dec. 2014.

[6] A. Mayoral López de Lerma, Integrated IT and SDN Orchestration of multi-domain multi-layer transport networks, Ph.D. thesis, Departament de Teoria del Senyal i Comunicacions, Universitat Politècnica de Catalunya, 2019. An optional note.

[7] R. Vilalta, A. Mayoral, R. Muñoz, R. Casellas, and R. Martinez, "Hierarchical SDN orchestration for multi-technology multi-domain networks with hierarchical ABNO," 2015 European Conference on Optical Communication (ECOC), pp.1–3, IEEE, 2015.

[8] M.S. Moreolo, L. Nadal, J. Fabrega, F. Vilchez, R. Casellas, R. Muñoz, C. Neumeyr, A. Gatto, P. Parolari, and P. Boffi, "Modular SDN-enabled S-BVT adopting widely tunable MEMS VCSEL for flexible/elastic optical metro networks," 2018 Optical Fiber Communications Conference and Exposition (OFC), pp.1–3, IEEE, 2018.

[9] ETSI, "Network functions virtualisation (NFV); architectural framework," standard, European Telecommunications Standards Institute (ETSI), Sophia Antípolis, Fr, Dec. 2014.

[10] D. King, A. Farrel, Q. Zhao, V. Lopez, R. Casellas, Y. Kamite, Y. Tanaka, Y. Lee, and Y. Yang, "Rfc7491: A PCE-based architecture for application-based network operations," Internet Engineering Task Force (IETF), March 2015.

[11] R. Casellas, R. Martínez, R. Vilalta, and R. Muñoz, "Abstraction and control of multi-domain disaggregated optical networks with openroadm device models," J. Lightwave Technol., vol.38, no.9, pp.2606–2615, 2020.

[12] R. Casellas, F.J. Vílchez, L. Rodríguez, R. Vilalta, J.M. Fàbrega, R. Martínez, L. Nadal, M.S. Moreolo, and R. Muñoz, "An OLS controller for hybrid fixed/flexi grid disaggregated networks with open interfaces," 2020 Optical Fiber Communications Conference and Exhibition (OFC), pp.1–3, IEEE, 2020.

[13] J. Bonet, C. Marques, F. Vicens, I. Dominguez, E. Fotopoulou, A. Zafeiropoulos, P. Alemany, R. Casellas, R. Vilalta, R. Muñoz, J. de la Cruz, E. Kapassa, M. Touloupou, K. Lambrinoudakis, G. Xilouris, S. Kolometsos, P. Tradakas, P. Karkazis, T. Soenen, A. Román, and A. Pol, D5.2 Service Platform Final Release, 1st ed., H2020 EUC 5GTANGO, June 2019.

[14] R. Vilalta, P. Alemany, R. Casellas, R. Martínez, C. Parada, J. Bonet, F. Vicens, and R. Muñoz, "Zero-touch network slicing through multi-domain transport networks," 2018 20th International Conference on Transparent Optical Networks (ICTON), pp.1–4, 2018.

[15] P. Alemany, J. de la Cruz, R. Vilalta, R. Casellas, R. Martínez, R. Muñoz, A. Pol, and A. Román, "Comparison of real-time communications service KPI in edge and cloud domains through multi-domain transport networks," 2020 24th International Conference on Optical Network Design and Modelling (ONDM), 2020.

[16] P. Alemany, R. Vilalta, F. Vicens, I. Dominguez Gómez, R. Casellas, R. Martínez, S. Castro, J. Martrat, and R. Muñoz, "Hybrid network slicing: Composing network slices based on VNFs, CNFs network services," 2020 IEEE Conference on Network Softwarization (NetSoft), 2020.

[17] GSMA, "Generic network slice template (version 2.0)," Technical Report, GSM Association, London, UK, Oct. 2019.

[18] ETSI, "Next generation protocols (NGP); E2E network slicing reference framework and information model," standard, European Telecommunications Standards Institute (ETSI), Sophia Antípolis, Fr, Sept. 2018.

[19] V. Lopez, R. Vilalta, V. Uceda, A. Mayoral, R. Casellas, R. Martinez, R. Muñoz, and J.P. Fernandez Palacios, "Transport API: A solution for SDN in carriers networks," ECOC 2016; 42nd European Conference on Optical Communication, pp.1–3, 2016.

Carlos Manso received the B.S. and M.S. degrees in Telecommunications Engineering from Universitat Politècnica de Catalunya (UPC) in Barcelona (Spain) in 2016 and 2018, respectively. He has been working in CTTC as a research assistant since 2018. Since 2020 he is also a Ph.D. candidate in SDN networks, focusing on cloud native transport networks orchestration and machine-learning techniques for SDN networks.

Pol Alemany received the B.S. and M.S. degrees in Telecommunication both from the UPC in Barcelona (Spain) in 2013 and 2017, respectively. Since 2017, he works as assistant researched in Centre Tecnològic de Telecomunicacions de Catalunya (CERCA - CTTC). He is also working on his PhD Research focused on SND/NFV networks, network slicing and security.

Ricard Vilalta graduated in 2007 and received a Ph.D. degree in telecommunications in 2013, both from the UPC. He is a senior researcher at CTTC, in the Optical Networks and Systems Department. He is an active contributor in several standardization bodies such as ONF, ETSI and IETF. He is also a member of the technical steering team of Open Transport Configuration & Control in ONF.

Raul Muñoz graduated in 2001 and received a Ph.D. degree 2005, both from UPC. He is Head of the Optical Networks and Systems Department and Senior Researcher. Since 2000, he has participated in 36 R&D projects funded by the European Commission's Framework Programmes (H2020, FP7, FP6, FP5, CELTIC and ITEA), the Spanish research programmes, and the industry.

Ramon Casellas graduated by both the UPC- university and ENST Paris. After working as an undergraduate researcher at France Telecom R&D and British Telecom Labs, he completed a Ph.D. degree in 2002. He worked as an associate professor at the ENST and joined the CTTC in 2006. He currently is a senior research associate, involved in several international R&D and technology transfer projects.

Ricardo Martínez graduated and PhD by the UPC-BarcelonaTech university in 2002 and 2007, respectively. He has been actively involved in several public-funded (national and EU) R&D as well as industrial technology transfer projects. Since 2013, he is Senior Researcher at CTTC. His research interests include control and network management architectures, protocols and traffic engineering mechanisms for next-generation packet and optical transport networks.