End-to-End Network Service Orchestration in Heterogeneous Domains for Next-generation Mobile Networks

Ph.D Dissertation
in partial fulfilment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

by
Jorge Baranda Hortigüela

Advisor: Dr. Josep Mangues-Bafalluy

Centre Tecnològic de Telecomunicacions de Catalunya - CTTC/CERCA

Communication Networks Division

Tutor: Dr. Albert Cabellos-Aparicio

Universitat Politècnica de Catalunya

Departament d'Arquitectura de Computadors

A Dácil, Eloy, Àngel, mis padres y Tete

"Any sufficiently advanced technology is indistinguishable from magic."
-Arthur C. Clarke

Abstract

5G marks the beginning of a deep revolution in the mobile network ecosystem, transitioning to a network of services to satisfy the demands of new players, the vertical industries. This revolution implies a redesign of the overall mobile network architecture where complexity, heterogeneity, dynamicity, and flexibility will be the rule. Under such context, automation and programmability are essential to support this vision and overcome current rigid network operation processes. Software Defined Networking (SDN), Network Function Virtualization (NFV) and Network slicing are key enabling techniques to provide such capabilities. They are complementary, but they are still in its infancy and the synergies between them must be exploited to realise the mentioned vision.

The aim of this thesis is to further contribute to its development and integration in next generation mobile networks by designing an end-to-end (E2E) network service orchestration (NSO) architecture, which aligned with some guidelines and specifications provided by main standardization bodies, goes beyond current management and orchestration (MANO) platforms to fulfil network service lifetime requirements in heterogeneous multi-technology/administrative network infrastructures shared by concurrent instances of diverse network services.

Following a bottom-up approach, we start studying some SDN aspects related to the management of wireless network elements and its integration into hierarchical control architectures orchestrating networking resources in a multi-technology (wireless, optical, packet) infrastructure. Then, this work is integrated in an infrastructure manager module executing the joint resource abstraction and allocation of network and compute resources in distributed points of presence (PoPs) connected by a transport network, aspect which is not (or lightly) handled by current MANO platforms. This is the module where the integration between NFV and SDN techniques is executed. This integration is commanded by a Service Orchestrator module, in charge of automating the E2E lifecycle management of network services implementing network slices (NS) based on the vertical requirements, the available infrastructure resources, and, while fulfilling service level agreement (SLA) also during run-time operation. This architecture, focused on single administrative domain (AD) scenarios, constitutes the first group of contributions of this thesis.

The second group of contributions evolves this initial architecture to deal with the orchestration and sharing of NS and its network slice subnet instances (NSSIs) involving multiple ADs. The main differential aspect with current state-of-the-art solutions is the consideration of resource orchestration aspects during the whole orchestration process. This is fundamental to achieve the interconnection of NSSIs, hence making the E2E multi-domain orchestration and network slicing a reality in practice.

Additionally, this work also considers SLA management aspects by means of scaling actions during run-time operation in such complex scenarios.

The third group of contributions demonstrate the validity and applicability of the resulting architectures, workflows, and interfaces by implementing and evaluating them in real experimental infrastructures featuring multiple ADs and transport technologies interconnecting distributed computing PoPs. The performed experimentation considers network service definitions close to real vertical use cases, namely automotive and eHealth, which help bridging the gap between network providers and vertical industries stakeholders. Experimental results show that network service creation and scaling times in the order of minutes can be achieved for single and multi-AD scenarios, in line with 5G network targets. Moreover, these measurements serve as a reference for benchmarking the different operations involved during the network service deployment. Such analysis are limited in current literature.

Keywords: Software-Defined Networking (SDN), Network Function Virtualization (NFV), Network Slicing, End-to-End Network Service Orchestration, single/multi administrative domain, resource abstraction, interfaces, SLA Management, Experimental validation.

Resumen

5G marca el inicio de una gran revolución en las redes móviles, convirtiéndose en redes orientadas a servicios para satisfacer las demandas de nuevos actores, las industrias verticales. Esta revolución supone un rediseño total de la arquitectura de red donde la complejidad, heterogeneidad, dinamicidad y flexibilidad serán la norma. En este contexto, la automatización y programabilidad serán esenciales para superar los rígidos procesos actuales de operación de red. Las redes definidas por software (SDN), la virtualización de funciones de red (NFV) y el particionamiento de redes son técnicas clave para proporcionar dichas capacidades. Éstas son complementarias, pero aún recientes y sus sinergias se deben explotar para realizar la nueva visión.

El objetivo de esta tesis es contribuir a su desarrollo e integración en la nuevas generaciones de redes móviles mediante el diseño de una arquitectura de orquestación de servicios de red (NSO) extremo a extremo (E2E), que alineada con algunas pautas y especificaciones de los principales organismos de estandarización, va más allá de los actuales sistemas de gestión y orquestación (MANO) para instanciar y garantizar los requisitos de los diversos servicios de red desplegados concurrentemente en infraestructuras heterogéneas compartidas que combinan múltiples tecnologías y dominios administrativos (AD).

Siguiendo un enfoque ascendente, comenzamos a estudiar aspectos de SDN relacionados con la gestión de elementos de red inalámbricos y su integración en arquitecturas jerárquicas de orquestación de recursos de red en infraestructuras multi tecnología (inalámbrica, óptica, paquetes). Luego, este trabajo se integra en un módulo de administración de infraestructura que ejecuta de forma conjunta la abstracción y la asignación de recursos de red y computación en múltiples puntos de presencia (PoP) distribuidos conectados por una red de transporte, aspecto que no está (o ligeramente) considerado por los actuales sistemas MANO. Este módulo ejecuta la integración de las técnicas NFV y SDN. Esta integración está dirigida por el módulo Orquestador de Servicios, que automatiza la gestión E2E del ciclo de vida de los servicios de red implementando las diferentes particiones de red en base a los requisitos de los verticales, los recursos de infraestructura disponibles y mientras cumple los acuerdos de nivel de servicio (SLA) durante la operación del servicio. Esta arquitectura, centrada en escenarios con un único AD, forma el primer grupo de contribuciones de esta tesis.

El segundo grupo de contribuciones evoluciona esta arquitectura abordando la orquestación y compartición de particiones de red y sus componentes (NSSIs) en escenarios con múltiples AD. La consideración detallada de aspectos de orquestación de recursos es el principal aspecto diferencial con la literatura. Esto es fundamental para la interconexión de NSSIs, haciendo realidad la orquestación E2E y el particionamiento de red en escenarios con múltiples AD. Además, se considera la gestión de SLA mediante acciones de escalado durante la operación del servicio en los escenarios mencionados.

El tercer grupo de contribuciones valida las arquitecturas, procedimientos e interfaces resultantes pues se han implementado y evaluado sobre infraestructuras experimentales reales que presentan múltiples AD y tecnologías de transporte interconectando PoP distribuidos. Esta experimentación considera definiciones de servicios de red cercanos a casos de uso de verticales reales, como automoción y eHealth, ayudando a cubrir la brecha entre los proveedores de red y los verticales. Los resultados experimentales muestran que la creación y el escalado de servicios de red se pueden realizar en pocos minutos en escenarios con un único o múltiples ADs, en línea con los indicadores de red objetivos de 5G. Estas medidas, escasas en la literatura actual, sirven como referencia para caracterizar las diferentes operaciones involucradas durante el despliegue de servicios.

Palabras clave: Redes Definidas en Software (SDN), Virtualización de Funciones de Red (NFV), Particionamiento de Red (Network Slicing), Orquestación de servicios de red extremo a extremo, único/múltiple dominio/s administrativos, abstracción de recursos, interfaces, gestión de acuerdos de nivel de servicio (SLA), validación experimental.

Acknowledgements

The development and writing of this thesis could not have been possible without the help of a large number of people that – directly or indirectly – supported the different stages of this work. These words are only a humble attempt to show my deep gratitude to them for advising, supporting me, working with me, or listening to me when I needed them.

First and foremost, I would like to thank my advisor Dr. Josep Mangues-Bafalluy for his exceptional guidance and the passion he dedicates to the research process. Your motivation and rigour are an example to me to keep working and loving this job. You gave me the opportunity to pursue this goal in a first-class research environment and you have demonstrated your trust in me in all the meetings and discussions we have shared through all these years to shape this work. Thanks for your patience, your clever advice and clear vision, which have been essential to develop this thesis. I am very proud and grateful for having received your guidance. These words are also extensible to Dr. Albert Cabellos-Aparicio for his help and guidance in Universitat Politècnica de Catalunya (UPC). Just, *Thank you*!.

I want to continue showing my gratitude to the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA). Here, I have found a really special environment to learn, think out of the box and unleash my intuitions without restrictions, where developing a PhD is an unforgettable experience. I am really grateful for all what I have learned from all the people there. Not only from the other researchers of the Mobile Networks department, where I develop my duties, but also from the researchers of other departments and divisions and former researchers that I had the opportunity to meet. Sharing work and coffee/beer/lunch conversations with you is always a pleasure and an inspiration. I apologize in advance if I forget anyone of you. Thanks Luca, Ricardo, Luis, Miguel Angel, Pol, Selva, Jordi, Vílchez, Engin, Manuel, José, Iñaki, JuanLu, Arturo, Ciprian. Thanks a lot from the bottom of my heart.

During the elaboration of this thesis, I had the opportunity to meet many outstanding researchers around the world in the framework of different European research projects like 5G-Crosshaul, 5G-TRANSFORMER that allowed me to develop most of the work included in this thesis. The technical discussions I shared with them broaden my mind and their push and useful suggestions allowed me to reach some of the results I am presenting here. Thanks to Kiril, Carlos Jesus, Carla, Giada, Juan, Giuseppe, Konstantin, Xi.

I wouldn't want to finish these lines without expressing all my gratitude to my family: my wife and my two sons, my parents and my brother for their unconditional and endless support, to stay next

to me and never let me give it up. A big part of this work is yours and this small homage is totally deserved. You also made a lot of effort to push this forward.

Barcelona, Spain - 23rd June, 2021 Jorge Baranda

Contents

List of Figures

List of Tables

Abbreviations

3GPP	3rd Generation Partnership Project
5G-PPP	5G Infrastructure Association Public Private Partnership
ABNO	Application-Based Network Operation
A-CPI	Application-Controller Plane Interface
AD	Administrative Domain
AF	Application Functions
AI	Artificial Intelligence
AM	Abstraction Manager
AMF	Access and Mobility Management Function
API	Application Programming Interface
ARPU	Average Revenue Per User
AS-PCE	Active Stateful Path Computation Element
AUSF	Authentication Server Function
BBU	Baseband Unit
BFD	Bidirectional Forwarding Detection
BH	Backhaul
CAM	Cooperative Awareness Message
CAPEX	Capital Expenditures
CCI	Co-Channel Interference
CD	Consumer Domain
CIDR	Classless InterDomain Routing
CIM	Cooperative Infrastructure Manager
CLI	Command Line Interface
CN	Core Network
COP	Control Orchestration Protocol
cRAN	Cloud RAN
CROOE	Composite Resource Orchestrator Engine
CSMF	Communication Service Management Function
CU	Centralized Unit
CUPS	Control and User Plane Separation
DB	Database
DC	Data centre
DF	Deployment Flavour

D-CPI	Data-Controller plane interface
DENM	Decentralized Environmental Notification Messages
DLT	Distributed Ledger Technologies
DN	Data Network
DSL	Domain Specific Language
DU	Decentralized Unit
E2E	End-to-End
EE	Execution Engine
eMBB	enhance Mobile Broadband
EM	Element Manager
EMS	Element Management System
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
EVS	Extended Virtual Sensing
EXTREME	Experimental Testbed for Research on 5G/4G Mobile nEtworks
FF	Fast Failover
FH	Fronthaul
GMPLS	Generalized Multi-Protocol Label Switching
GUI	Graphical User Interface
HPC	High Performance Computing
HSS	Home Subscriber Server
ICT	Information and Communication Technologies
ID	Identifier
IE	Information Element
IFA	Interfaces and Architecture
IL	Instantiation Level
ILP	Integer Linear Programming
ISG	Industry Specification Group
IT	Information Technologies
KPI	Key Performance Indicator
LL	Logical Link
LSO	Lifecycle Service Orchestration
MANO	Management and Orchestration
MCS	Modulation and Coding Scheme
MEF	Metro Ethernet Forum
ML	Machine Learning
MME	Mobility Management Entity
mMTC	Massive machine-type communications
mmWave	millimeter Wave
MNO	Mobile Network Operator
MON	Monitoring
MTP	Mobile and Compute Transport Platform
MTU	Maximum Transmission Unit
NBI	Northbound Interface
NE	Network Element
NEF	Network Exposure Function
NETCONF	Network Configuration
NF	Network Functions
NFV	Network Function Virtualization
NFV-NS	NFV Network Service

NFVI	NFV Infrastructure
NFVIaaS	NFVI as a Service
NFVO	NFV Orchestrator
NPN	Non-Public Network
NR	New Radio
NRF	Network Repository Function
NSA	Non Standalone
NSC	Network Service Composition
NSD	NFV-NS Descriptor
NSF	Network Service Federation
NSI	Network Slice Instance
NSMF	Network Slice Management Function
NSO	Network Service Orchestration
NSSI	Network Slice Subnet Instance
NSSMF	Network Slice Subnet Management Function
OBU	On-Board Unit
OE	Orchestration Engine
OF	Open Flow
ONF	Open Networking Foundation
OPEX	Operational Expenditures
OSM	Open Source Mano
OSS/BSS	Operations Support Systems/ Business Support Systems
OXC	Optical Cross-Connect
PCF	Policy Control Function
PCP	Personal Basic Service Set Control Point
PCRF	Policy Control and Charging Rules Function
PD	Provider Domain
PDCP	Packet Data Convergence Protocol
PNF	Physical Network Function
PoC	Proof of Concept
QoS	Quality of Service
RA	Resource Allocation
RAN	Radio Access Network
RESTCONF	Representational State Transfer Configuration
RLC	Radio Link Control
RMA	Resource Management Application
RNIS	Radio Network Information Service
RO	Resource Orchestrator
ROADM	Reconfigurable Optical Add/Drop Multiplexer
ROOE	Resource Orchestration Engine
RRH	Remote Radio Head
RSU	Roadside Unit
SAP	Service Access Point
SBA	Service Based Architecture
SBI	Southbound Interface
SCT	Service Creation Time
SDN	Software Defined Networking
SDK	Service Development Kit
SDO	Standard Development Organization
S-GW	Serving Gateway

SM	Service Manager
SMF	Session Management Function
SNMP	Simple Networking Management Protocol
SO	Service Orchestrator
SOE	Service Orchestrator Engine
SOL	Solutions
T-API	Transport API
UDM	Unified Data Management
UE	User Equipment
UP	User Plane
UPF	User Plane Function
URL	Universal Resource Locator
URLLC	Ultra-reliable low-latency communications
VCA	VNF Configuration Agent
VIM	Virtualised Infrastructure Manager
VL	Virtual Link
VM	Virtual Machine
VNF	Virtual Network Function
VNFD	Virtual Network Function Descriptor
VNFFG	VNF Forwarding Graph
VNFM	VNF Manager
VPN	Virtual Private Network
VS	Vertical Slicer
VStr	Video Streaming
VStrC	Video Streaming Controller
VStrS	Video Streaming Server
WAN	Wide Area Network
WIA	Wireless Interface Agent
WICM	WAN Infrastructure Connection Manager
WIM	Wide Area Network Infrastructure Manager
YAML	Yet Another Markup Language
YANG	Yet Another Next Generation

Chapter 1

Introduction

This chapter introduces in Section 1.1 the context of this thesis and its main motivations. This section also presents a general introduction to the tackled research problem, as well as the expected impacts of the proposed architectures and solutions in academia and industry. Section 1.2 provides a brief description of the structure and the content of this thesis. And in Section 1.3, we provide the main contributions coming out from this dissertation.

1.1 Motivation

Currently, mobile networks are living a revolution that is transforming the telecom network ecosystem. Unlike previous generations of mobile networks, which mostly focused on the provision of high-speed mobile broadband access, 5G networks entail the beginning of a new telecommunications era, creating the base for upcoming *"XG"* networks. In this era, as mentioned, the focus is not only on higher capacity but also on greater reliability and availability, massive amount of communications and lower latencies. These expected capabilities will allow the introduction and support by the mobile network market of new innovative services requested by vertical industries, such as eHealth, automotive, media or Industry 4.0. This will lead to new business opportunities for operators and enterprises.

5G means the beginning of a complete redesign of the overall network architecture, a disruption not only from an architectural perspective, but also from the control and management perspective. In this *new* network, complexity and heterogeneity will be the rule, hence requiring increasing automation and programmability to support the mentioned range of future service opportunities in a flexible and agile manner. Furthermore, the borders between traditional network segments (access, transport and core) will likely to be blurred due to the deployment of distributed computing resources across the mobile network infrastructure continuum. Under such context, the transport (plus distributed computing) network segment will gain importance, increasing its design complexity. Availability and capacity requirements will give special relevance to the deployment of dense wireless networks at the edge of the network, acting as a first level of aggregation. At higher levels of aggregation, the optical technology will continue to be the most relevant solution to transport the expected amount of data. The use of wireless technologies is seen as a cost-effective solution to meet the required network performance in scenarios like densely populated cities or suburban areas, where the deployment of optical fibre is not suitable from a business perspective due to different reasons, e.g., cost of deployment or investment return. In such scenarios, communications at millimetre wave (mmWave) bands (28-100 GHz frequency range) will likely to be an enabler to achieve the aforementioned objectives due to their multi-gigabit per second capacity. Hence, the combination of wireless domains (and other technologies) with classical solutions like optical transport networks to create heterogeneous deployments managed as a whole is of research interest.

This overall redesign will also change the ecosystem of stakeholders of mobile networks, where the roles of infrastructure provider, connectivity provider and service provider can be separated. Under such context, the Software Defined Networking (SDN), Network Function Virtualization (NFV) and network slicing paradigms are seen as fundamental key-enabling concepts to realize the vision of 5G and upcoming generations of mobile networks and integrate the multiple technology shifts occurring alongside, like the rise of open radio access networks (RAN), complete network virtualization, the emergence of cloud computing in telecom networks' back-office systems, mobile edge computing, and network functions.

At a high-level, the research problem that this thesis addresses is *how to provide reduced end-to-end (E2E) service provisioning times in the envisaged heterogeneous and multi-provider shared scenario of 5G networks and ensure the fulfilment of service requirements during its lifetime.* This thesis answers this question by designing and developing an E2E network service orchestration architecture that, based on the synergies of previously mentioned key-enabling concepts, evolves current orchestration frameworks to avoid reinventing the wheel. These frameworks mainly deal with the instantiation of resources associated to network services with a limited understanding of their requirements but their architectural design present high-level considerations that do not delve into the challenges associated to the distribution and interconnection of the network functions forming the network services. This enables mechanisms to make an efficient use of the heterogeneous set of shared abstracted resources not only just at its local domain but possibly implying other administrative domains. Another remarkable aspect of this thesis is the validation and benchmarking of the proposed architecture, interfaces and procedures including real experimental infrastructures covering the proposed scenario and applied to vertical use cases, namely automotive, eHealth or Industry 4.0.

The research question motivated and introduced above represents a big contribution for both the research and the industry communities. The challenges associated to the management of network slices not only from the service perspective but also from the associated resource orchestration logic to ensure exclusive E2E control over a shared infrastructure spanning different multiple administrative domains owned by different organizations, are still under discussion from a theoretical and from a practical point of view. These aspects together with the defined architecture, allow the development of specialised stakeholders focusing on different aspects of the network service deployment chain and the creation of new business models based on the exchange/sharing of network services between multiple administrative domains. All these considerations contribute to continue the development to automate the deployment and management of the multiple use cases integrating the new players in the telecom ecosystem, the vertical industries, which are expected to bring social gains.

1.2 Outline of the dissertation

The work in this dissertation is organized into twelve chapters. Ten out these chapters constitute the conducted research work, which are structured into three main parts. Part I contains chapters 2, 3, 4, and 5. It focus on providing the required background context, a review of related research work, the statement of the specific research problem this thesis tackles and a high-level overview of the proposed network service orchestration architecture developed in the described experimental framework to provide answers to the posed research question. Part II explains the design and concepts used in the orchestration architecture to handle the lifecycle management of network slices in the form of network services distributively deployed within a single administrative domain including heterogeneous (transport) network and compute resources. Part III evolves the work of Part II to solve the problem of orchestrating, sharing and scaling network services in a scenario considering multiple administrative domains. A more detailed explanation of all the chapters follows.

Chapter 2 provides some basic background on the main technological paradigms that will be employed to propose a network service orchestration architecture contributing to realise the view and requirements of next-generation mobile networks.

Chapter 3 presents related research work on the application of the previous paradigms into the mobile network ecosystem identifying the gaps covered by this Thesis work.

Chapter 4 describes the research problem, which has been derived from the lacks detected in the literature presented in the previous chapter.

Chapter 5 provides a high-level description of the proposed end-to-end network service orchestration architecture and the experimental framework used to demonstrate its feasibility.

Chapter 6 starts the second part of this dissertation, focusing on the problem, proposing an evaluating a solution for the dynamic interconnection of network endpoints in a transport network featuring multiple control and data plane technologies typical of next-generation mobile networks scenarios.

Chapter 7 describes the architecture of the building block in charge of carrying out the management and allocation of computing resources in distributed data centres and the associated transport network resources needed to interconnect the different network functions while satisfying the demanded requirements. A key feature of this block is the application of resource abstraction mechanisms to facilitate the network service orchestration process.

Chapter 8 presents the design of the brain of the proposed orchestration architecture. This building block is in charge of managing the network slice instance requests and the lifecycle operations of the associated network service implementing them. This chapter presents the evaluation of this building block coordinating the outcomes of the work of chapters 6, 7. This shows the validity of the proposed end-to-end network service orchestration architecture, hence providing an initial set of answers to the proposed research question within a single administrative domain scenario.

Chapter 9 starts the third part of this dissertation by describing the concept of network service composition and how this concept is the foundation for the dynamic deployment and sharing of network slice (subnet) instances in the form of composite networks services in multiple administrative domains.

Chapter 10 is devoted to answer a specific aspect of the formulated research question in Chapter 4 based on the foundations set in the previous chapter. This chapter also includes a performance evaluation of the proposed solution using the experimental framework explained in Chapter 5.

Chapter 11 extends the work in Chapters 9 and 10 to handle the problem of (auto-)scaling (shared) network slice (subnet) instances deployed in multiple administrative domains without service disruption from an E2E perspective.

Chapter 12 summarises the work carried out in this thesis, states its main conclusions and presents some future lines of work.

Additionally, this Thesis presents three appendices with complementary information. Appendix A describes the workflow followed by the brain of the proposed orchestration architecture for other relevant network service lifecycle management operations, like service onboarding and termination. Appendix B presents some initial extensions of the proposed orchestration architecture considering some aspects described in the future work section of Chapter 12. Finally, Appendix C presents a comprehensive list with all the publications derived from the work presented in this thesis.

1.3 Contributions

Next, we provide a birds-eye view of the key contributions of this thesis, which are ordered from a bottom-up perspective with respect to the proposed network service orchestration architecture.

1. Design of SDN-based management and configuration procedures to automatically set up and perform flexible recovery of wireless resources in a multi-technology transport network based on a hierarchical SDN transport orchestration architecture.

2. Design of a modular architecture integrating the management of transport network resources with computing, networking and storage resources available at distributed cloud locations. This module is basic to execute the integration of NFV and SDN paradigms while providing proper resource abstraction and allocation mechanisms for the automatic orchestration of network services.

3. Design of a modular NFV/SDN-based orchestration architecture performing the automatic, flexible and dynamic end-to-end (E2E) lifecycle management of network services implementing network slice instances based on the vertical requirements, the available abstracted resources at the underlying heterogeneous transport and cloud infrastructure, while fulfilling service level agreements (SLA). The resulting orchestration architecture integrates the results of items 1 and 2 as well.

4. We propose a solution to the problem of orchestrating and sharing of network slices constituted by multiple network slice subnet instances in single administrative domain and multiple administrative domain scenarios in chapters 9 and 10.

5. We mix the problem of multi-administrative domain orchestration with SLA assurance in chapter 11 to derive operational workflows including the support of (auto-)scaling procedures of network services spanning multiple domains based on monitored SLA violations.

6. We implement and demonstrate the different researched concepts in a real experimental infrastructure considering multiple control and transmission technologies in single and multi-administrative domain scenarios. The performed evaluations analyse the time impact of the different operations contributing to the instantiation and scaling of different network services when considering the contributions presented in the previous points. These procedures can be completed in the order of few minutes, in line with expected Key Performance Indicator (KPI) for such systems, as the ones defined by the 5G Infrastructure Association Public Private Partnership (5G-PPP).

7. We apply the researched concepts and the resulting end-to-end network service orchestration architecture in Proof of Concepts featuring close to real vertical use cases, namely automotive ([10], [11]) and eHealth ([12]).

Part I

Background and Problem Statement

Chapter 2

Background

This chapter presents the required background concepts supporting the work performed in this thesis. First, we present the key technological paradigms enabling the transformation of next-generation mobile network architecture. Namely, these concepts are Software Defined Networking (SDN), Network Function Virtualization (NFV) and Network Slicing. Then, this chapter finishes with a description of the 5G network architectural principles as a natural way to understand why the integration and development of previous concepts are needed to realise the proposed view for upcoming and future next-generation mobile networks.

2.1 Software Defined Networking

Traditional technology used in the Internet to reliably deliver data traffic is based on a distributed control and transport network protocols running inside switches and routers. This distributed approach implies the reconfiguration of every forwarding device through a set of supported protocols to modify its policies, hence leading to a huge effort by network operators to configure these devices when they want to create or modify an end-to-end network service. Furthermore, these devices often use vendor-specific command line interface, thus putting further complexity into this process and making networks complex and hard to operate.

Software Defined Networking (SDN) [13] is a recent networking technique that has attracted the attention of network operators thanks to its promise of providing simplified programmability of network resources by applying the flexibility of software. SDN bases on the idea of decoupling the control plane logic from the data plane forwarding infrastructure. The control plane is moved to a logically centralized software element referred to as the *SDN controller*, which interacts with the underlying network elements (NEs) to perform not only traditional routing functions but also network-wise decisions. Since it has a broader perspective of the resources under its control, the SDN controller can potentially make better decisions than current distributed control schemes. For instance, an SDN controller could decide to concentrate the traffic through a set of NEs during light traffic conditions to switch off non-required NEs, thus achieving energy savings. Hence, this decoupling allows transforming network resources into programmable and flexible resources for a highly scalable and automated network control. Since NEs, such as switches and routers, become less complex and programmable, they can implement faster forwarding mechanisms even in general-purpose hardware, which increases the data plane performance and reduces capital expenditures (CAPEX). Moreover, SDN proposes open standard interfaces that can reduce the traditional vendor-dependency of network operations while increasing automation and manageability. Other benefits derived from the use of common interfaces and the control plane logical centralization are: increased inter-operability between transmission technologies, faster end-to-end network service creation and operational expenditures (OPEX) reductions.

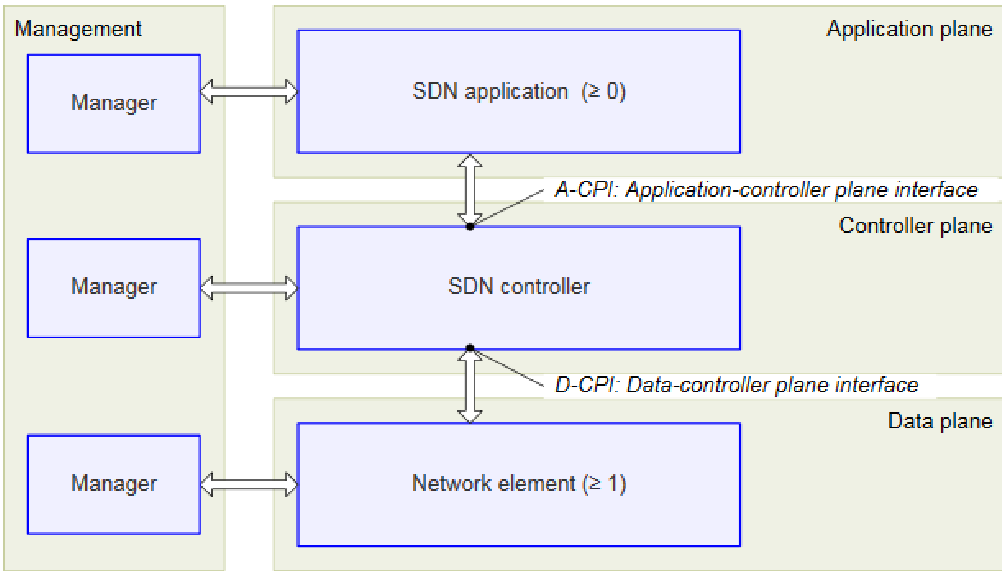


Figure 2.1: SDN architecture overview (extracted from [1])

Figure 2.1 captures the essence of the SDN concept from an architectural perspective. This view has been provided by the Open Networking Foundation (ONF) [14], a non-profit organization funded by companies such as Samsung, Deutsche Telekom, Telefonica, Google, Microsoft, Verizon, or AT&T aimed at promoting networking through SDN and standardizing related technologies. Specifically, Figure 2.1 has been extracted from [1], the Technical Recommendation of the ONF released in June 2014 that included the first version of the SDN architecture. Afterwards, in 2016, the ONF released an update of this architecture [15], which clarifies and extends the previous release, in an attempt to unify the discussion across the different ONF working groups, standards development organizations (SDOs), and open source communities.

As it can be seen in Figure 2.1, the SDN architecture defines three layers: the application plane, the controller plane and the data plane. The application plane is constituted by zero or more applications, which interact with the SDN controller through the Application-Controller plane interface (A-CPI) to perform decisions over a set of resources exposed by one or more SDN controllers. The A-CPI is also called the Northbound interface (NBI). The controller plane contains one or more SDN controllers, which is the central entity of the architecture. As such, the SDN controller satisfies the client requests arriving from the NBI by orchestrating its underlying resources and exposing the suitable abstractions to ease programmability. The SDN controller is also in charge of updating network and service state and coordinating the required actions to maintain the requested policy. The latter is done through the Data-Controller plane interface (D-CPI), also called Southbound Interface (SBI).

A review of different alternative protocols for the D-CPI or SBI can be found in [16]. These protocols deal not only with the modification of the forwarding table of the network element but also to manage the configuration of the network element itself.

Out of these protocols, this thesis work considers the OpenFlow (OF) protocol [17] as a protocol for controlling the forwarding behaviour of network elements. The OF protocol has been standardized by the ONF and has been used as the de-facto standard to perform such operations due to its support from the industry, research and open-source communities. SDN controllers like ONOS [18], OpenDayLight [19], Ryu [20], software switches like OpenvSwitch [21] or data centre orchestration networking solutions like Openstack [22] use this protocol. OF protocol bases on the flow abstraction concept, where the forwarding behaviour is configured based on the combination of L2-L4 packet header fields stored in the flow forwarding table. The SDN controller exchanges TCP messages with an OF agent residing in the NE through a secure channel to configure such forwarding table. It is worth mentioning that, recently, the Programming Protocol-independent Packet Processors (P4) language is gaining more attention from the research community for its promise of increased level of abstraction to enable software switches to support flexible mechanisms for parsing packets and remove the burdens of repeatedly extending the OF specification [23].

For the case of NE management protocols in SDN, multiple alternative protocols are still under discussion. In the case of the A-CPI, there is more progress towards a desired consensus for the used protocol. This aspect is further developed in Chapter 3. However, as discussed there, it is clear that the adoption of open interfaces is essential to enable the automation of network resource orchestration in complex and heterogeneous setups involving multiple transmission and control technologies.

2.2 Network Function Virtualization

Network Function Virtualization (NFV) is a network architecture concept that is based on the application of standard Information Technologies (IT) virtualization techniques as a new way to design, deploy and manage networking services. Network functions (NFs) traditionally running in dedicated hardware middleboxes such as a firewall, or a intrusion detection system, to name a few, can now be dynamically deployed to be run as software in all-purpose processors, hence decoupling functionality from the hardware and removing the burdens introduced by proprietary hardware appliances. In this way, NFV allows network operators to reduce their CAPEX and OPEX while accelerating the required time-to-market to deploy new networking services, reducing the risks associated to such deployments and delivering agility and flexibility to address changing demands.

Due to the potentiality of the NFV [24] concept, operators created in November 2012 the NFV Industry Specification Group (ISG) [2] under the umbrella of the European Telecommunications Standards Institutes (ETSI). This group is in charge of developing the requirements and architectures to accelerate the progress of NFV. As explained in [2], since their creation, the ETSI ISG NFV community has generated over 100 publications, and has evolved through several phases, moving from pre-standardization studies to detailed specifications (Release 2, Release 3, Release 4) and the initial Proof of Concepts (PoCs) efforts have evolved and led to interoperability events (Plugtests). Nowadays, this large community (300+ companies including 38 of the world's major service providers) is still working intensely to develop the required standards for NFV as well as sharing their experiences of NFV implementation and testing.

Figure 2.2 presents the reference architecture framework proposed by ETSI NFV ISG group in ETSI-NFV GS002 [25] specification, which was published in October 2013 and has become the reference architectural design since then. On top of Figure 2.2, we can observe the set of semantic level specifications (procedures, information models) covering each of the reference points identified in the basic architecture, and syntactically precise specifications for some of the interfaces. The semantic level specifications have been produced by the ETSI NFV Interfaces and Architecture (IFA) working group and the syntactic specifications have been produced by the ETSI NFV Solutions (SOL) working group.

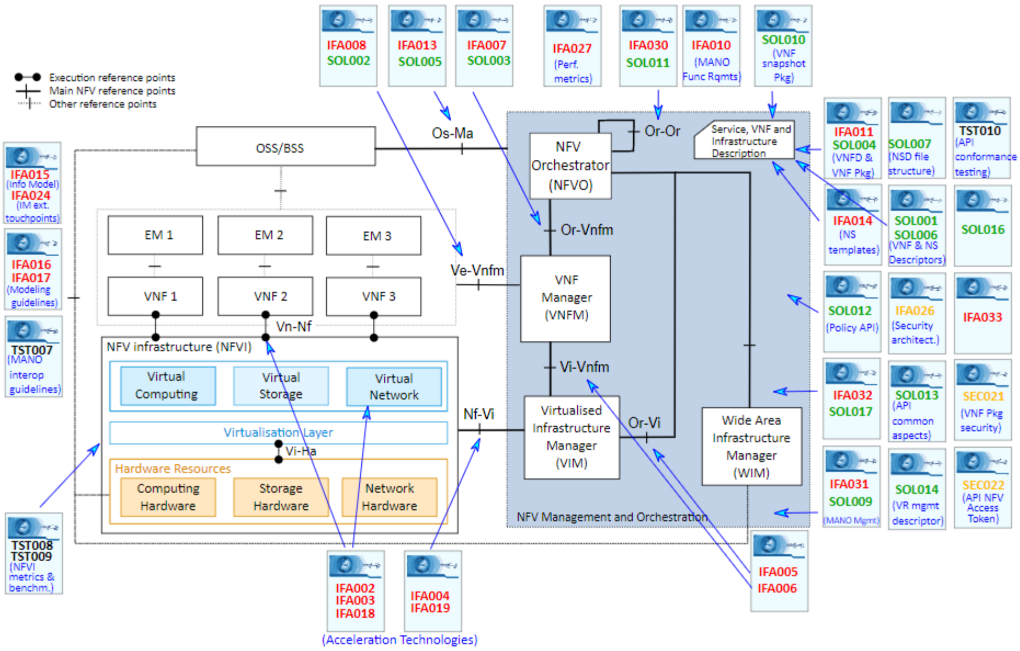


Figure 2.2: NFV architecture overview (extracted from [2])

The main building blocks of the NFV architecture are:

- *NFV Infrastructure (NFVI)*, which is the set of hardware resources providing processing, storage and connectivity to VNFs, and the Virtualization Layer which abstracts these hardware resources and decouples the VNF software from the underlying hardware.

- *NFV Management and Orchestration (NFV MANO)*, which is the control layer in charge of performing the lifecycle management of VNFs and their associated physical and/or software resources supported by the NFVI. At the same time, the NFV MANO component consists of three elements. The Virtualised Infrastructure Manager (VIM) is the entity responsible for controlling and managing the interaction of a VNF with the computing, storage and networking resources provided by the NFVI. The VNF Manager (VNFM) is responsible for the lifecycle management of VNFs under the control of the NFV Orchestrator (NFVO), which is achieved by instructing the VIM. Finally, the NFVO is the entity responsible for binding together different VNFs to create an end-to-end, resource-coordinated NFV network service (NFV-NS).

- *Virtualised Network Functions (VNFs) and Element Managers (EMs)*. The VNFs are the software implementation of diverse network functions capable of running at the NFVI infrastructure according to the characteristics determined by the MANO component and the EMs are the entities in charge of performing the management over one or several VNFs.

Finally, there is the Operations Support Systems and Business Support Systems (OSS/BSS). This entity interacts with the NFV architectural framework and it is present in current operator network deployments, even not including the NFV stack. This entity is responsible for supporting operations, administration and maintenance functions such as network inventory, service provisioning, network configuration, fault management, billing or customer management, between others. This entity is relevant in the whole picture of mobile networking, because SDN and NFV will not achieve their full potential until OSS/BSS systems will not be aligned with them. The dynamicity and flexibility of novel network services requires of automation and intelligence at the time to perform provisioning, configuration, billing or fault management.

As it can be seen, the ETSI NFV working groups have provided a broad knowledge base in the form of reports and specifications, but the continuous update and generation of new elements in this knowledge base show the need from further research to achieve the holistic vision proposed by the NFV concept.

2.3 Network Slicing

Network slicing is an emerging networking concept [3] that considers the segmentation of physical network infrastructures to allow the dynamic and concurrent deployment of multiple logical (temporarily) self-contained networks orchestrated in different ways according to their specific service requirements. This is possible thanks to a converged cloud and network infrastructure, with common control and operation capabilities introduced by the programmability of SDN and NFV paradigms and leveraging on the separation between hardware and software.

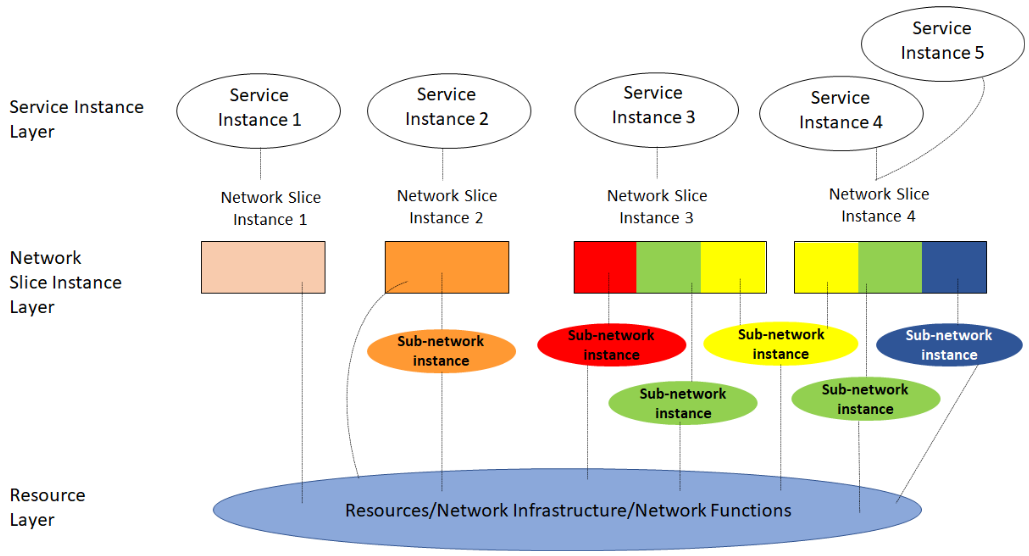


Figure 2.3: Network slicing concept (adapted from [3])

As explained by the NGMN Alliance in [3] and represented in Figure 2.3, the network slicing concept consists of three layers: 1) service instance layer, 2) network slice instance layer, and 3) resource layer. The service instance layer represents the end-user and/or business services, which are provided by the service providers. Each of these services are represented by a service instance and are supported by the network slice instance layer. This is in turn supported by the resource layer, which may consist of physical resources such as compute, network, memory, storage, etc, used to implement and interconnect more complex entities such as network functions. A network slice instance (NSI) provides the characteristics which are required by a Service Instance. These characteristics are the set of network functions and their required resources, which are arranged and configured, forming a complete logical network to meet certain network characteristics. Additionally, an NSI may be composed by one or more sub-network instances, the so-called network slice subnet instance (NSSI), which may be shared among NSIs.

Based on this concept, several standardization bodies have started to work to adapt and include it in their ecosystem. In TR28.801 [26], the 3rd Generation Partnership Project (3GPP) completed a study on management and orchestration on network slicing which is the basis for the normative specification work in 3GPP releases starting from Release 15. In this study, 3GPP includes the components of an NSI, the lifecycle management phases (preparation, instantiation, configuration and activation, run-time operation and decommissioning) and the three logical management functions in charge of executing it. These functions are: (i) the Communication Service Management Function (CSMF), in charge of translating communication service requirements to network slice requirements, (ii) the Network Slice Management Function (NSMF), responsible for the management and orchestration of NSIs, and (iii) the Network Slice Subnet Management Function (NSSMF), responsible for the management and orchestration of NSSIs in collaboration with the NSMF. ETSI NFV has carried out activities to map the NFV architecture and concepts with the 3GPP network slicing work in the ETSI-NFV EVE012 report [4]. This report establishes the correspondence between an NSI (as defined by 3GPP) with an ETSI-NFV network service (NFV-NS) and the NSSI with the concept of nested element as constituent part of the NFV-NS description, which may be shared by different NSIs. Additionally, this report proposes the interaction between the network slice management functions proposed by 3GPP and the ETSI-NFV architecture through the use of the *Os-Ma-Nfvo* reference point. Figure 2.4 presents this mapping proposed by ETSI-NFV EVE012 between ETSI-NFV ecosystem and 3GPP network slicing work.

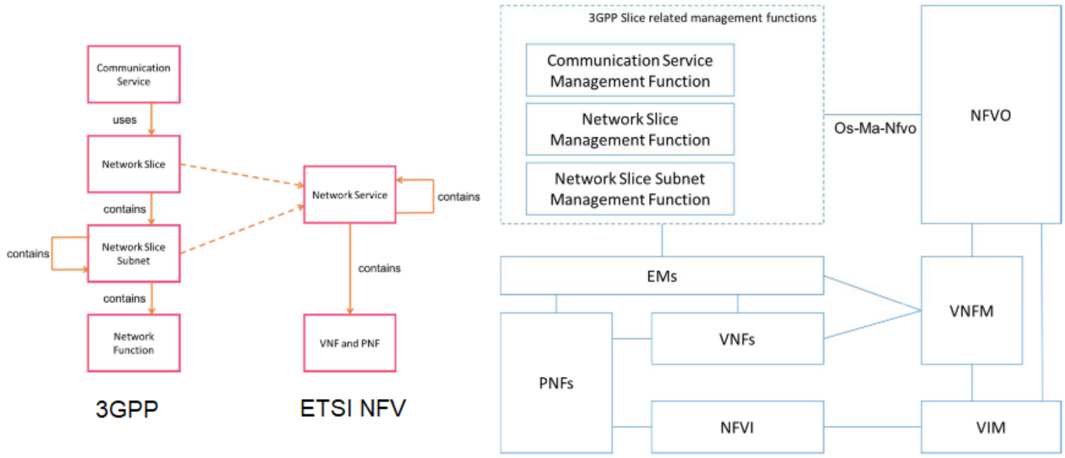


Figure 2.4: Mapping between 3GPP and ETSI-NFV (left) information models and (right) framework (extracted from [4]).

2.4 Next-generation mobile network transformation

While 4G mobile network generation was focused on providing enhanced (evolved) performance in terms of higher peak throughput, there is a broad consensus that 5G and beyond will not just be an evolution of 4G networks dealing only with new radio interfaces or waveforms to cope with the expected 1000x increase and low latency requirements in mobile traffic [27].

5G mobile networks consider as main types of services enhanced mobile broadband (eMBB), ultra-reliable and low-latency communications (URLLC) and massive machine-type communications (mMTC) as an attempt to target new innovative services and business models to open the telecom ecosystem to new players, like the vertical industries (e.g., automotive, eHealth, media and entertainment, energy, and factories of the future).

The different characteristics and requirements of each type of previously mentioned services make challenging the definition of a suitable architecture with current management schemes, even if they were considered independently. However, to have an economically and environmentally sustainable solution, only a common design considering all three service types is being considered. These considerations imply a radical redesign of the mobile network architecture to allow the automatic, dynamic and flexible orchestration of novel network services over a heterogeneous set of abstracted resources to satisfy the requirements imposed by such type of services while safely coexisting in a shared infrastructure.

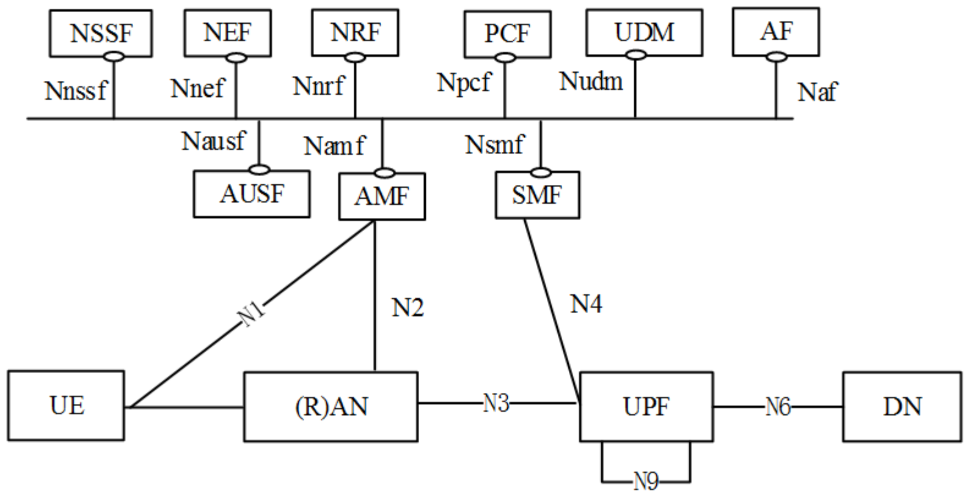


Figure 2.5: 5G System Architecture (extracted from [5])

This redesign starts with the definition of a new core network (CN) architecture based on softwarization principles. Figure 2.5 shows the general 5G reference architecture defined by 3GPP in document TS23.501 [5]. This architecture has its foundations on the modularization and decomposition of the mobile network entities into network functions (NFs). In comparison to 4G networks, monolithic NFs are split into basic modules for the control plane (CP) and the user plane (UP), also known as Control and User Plane Separation (CUPS). In this process of decomposition into basic modules, the distinction between NFs relating to the Radio Access Network (RAN) and the CN can minimize the inter-dependency between segments (i.e., coupling between core and access technology) and blur its borders since they can be implemented in any part of the mobile network where resources to deploy such NFs are available while satisfying the required constraints [6]. This, in conjunction with the flexibility offered by the New Radio (NR) air interface and the envisaged support of multiple access technologies, enables the provision of different logical realizations of the mobile network architecture based upon the performance and functional requirements of the requested services. Next, a brief description of the different functional entities presented in Figure 2.5.

- *User Equipment (UE)*: this entity defines a mobile device capable of establishing a connection with a 5G and an 4G network.
- *(Radio) Access Network (RAN)*: this functional entity consists of NR Gigabit Nodes (gNBs) as base stations providing the UP and CP protocol termination for the radio interfaces towards the UE. However, one key difference between 5G and earlier mobile generations is that a 5G system will consist of multiple highly integrated air interface variants with tailored radio functionality to support the different needs of the diverse service types.
- *Access and Mobility Management Function (AMF)*: it is the entity in charge of managing the termination of RAN CP functionality through the N2 interface and performing mobility and registration, connection management procedures with UEs through the interface N1, between other tasks.

- *Authentication Server Function (AUSF)*: it provides access authentication and authorization functionalities.
- *Session Management Function (SMF)*: it includes, between others, session management and establishment procedures, UE IP address allocation, selection and control of UP functions and configuration of traffic steering at UPF through the N4 interface.
- *Network Exposure Function (NEF)*: it is the entity which receives information from other network functions (based on exposed capabilities of other network functions) and stores and provides secure access to the received information to other network functions and Application Functions (AFs).
- *Network Repository Function (NRF)*: it supports the discovery, profile maintenance and provision of information of available NF instances and their supported services.
- *Network Slice Selection Function (NSSF)*: this entity selects the set of network slice instances serving the UE and determines the AMF (set) that serves the UE, possibly by querying the NRF.
- *Policy Control Function (PCF)*: it is the entity that supports a unified policy framework to govern network behaviour, providing rules to CP function(s) to enforce them.
- *Unified Data Management (UDM)*: it provides the support for the user identification, the management registration of serving NFs and access authorization based on subscription data.
- *Application Function (AF)*: they are provided by third parties and, depending on the operator deployment, interact with other 3GPP CN functions to provide services such as traffic routing.
- *User Plane Function (UPF)*: this entity provides the following functionalities: anchor point for intra-and-inter-RAT mobility, packet routing and forwarding, UP quality of Service (QoS) handling, packet inspection and traffic accounting and reporting.

Finally we have the block that represents the *Data Network (DN)*, which is composed by the external networks, i.e., Internet providing services to mobile network UE's.

As a consequence of the modularization of the architecture, the number of interfaces to manage the amount of possible inter-NF and CP-UP relationships has exploded. Following the defined service based architecture (SBA), each CP NF is connected to a message bus that exposes its functionality to authorized and subscribed NFs over service based interfaces, hence providing higher flexibility and customization to define the required logical architectures satisfying the different requirements. For the purpose of exposure of functionality to third parties as well as among 5G NFs, Representational State Transfer (REST) based Application Programming Interfaces (APIs) [28] are the alternative considered to implement such interfaces.

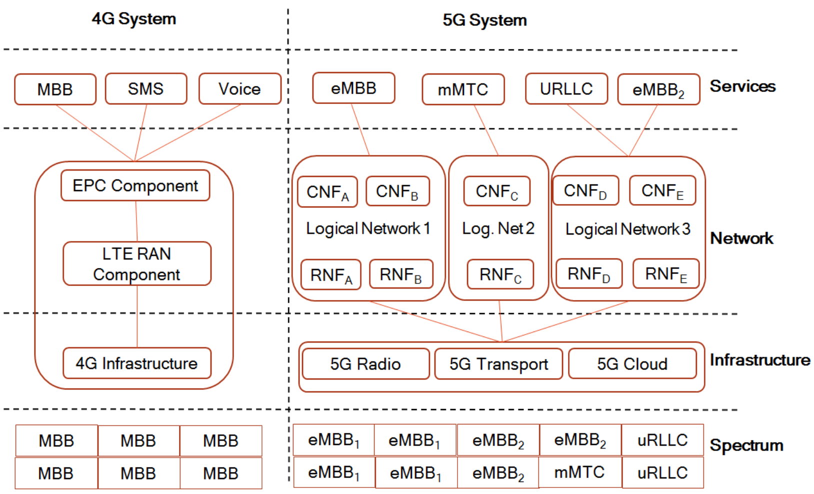


Figure 2.6: 4G vs 5G System Architecture (adapted from [6])

Figure 2.6 shows the conceptual difference between 4G and 5G and beyond mobile networks.

This figure, adapted from [6], shows the multiple different logical realizations that a 5G network may orchestrate in the available infrastructure to satisfy the requirements imposed by the different types and multiple instances of the network services running on top. This makes clear how the concepts explained in previous sections are essential to realize this vision.

Besides the changes introduced at the RAN and at the CN level, the architecture of the transport network is also going to be deeply affected and adapted accordingly. The mobile transport network is the segment of the network connecting the RAN with the CN, which carries the so-called *backhaul* (BH) traffic. Traditionally, this segment has been seen as a static *commodity or a dumb pipe* where the BH traffic flows.

Currently, 3GPP and other bodies like CPRI [29] and O-RAN [30] consider also a flexible split of RAN functions, similar to the split between CP/UP presented previously. In this case, this split is done into centralized units (CUs) and distributed units (DUs), defining the so-called cloud RAN (cRAN) concept. The traffic between DUs and CUs is the so-called *fronthaul* (FH) traffic.

The possibility of virtualizing NFs on top of commodity hardware has made possible the cRAN concept. Under cRAN, cellular base station functions are hosted in cloud computing facilities which can be distributed across the transport network, the CUs or baseband units (BBU), while DUs or remote radio head (RRH) are focused on low-level functions more related to signal acquisition. These functions can be decomposed in many different ways, the so-called functional splits. Figure 2.7 presents the different considered options at TS38.801 [7]. Each option is characterized by throughput, latency and jitter requirements for the underlying transport network and may range from a typical backhaul interface (Option 1) to an interface where digital samples of the air interface are exchanged (Option 8). These functional splits are motivated by the possibility to obtain centralization gains derived from multi-cell joint resource management, to shift functions to different locations based on use case requirements or to adapt RAN processing to different infrastructure deployments.

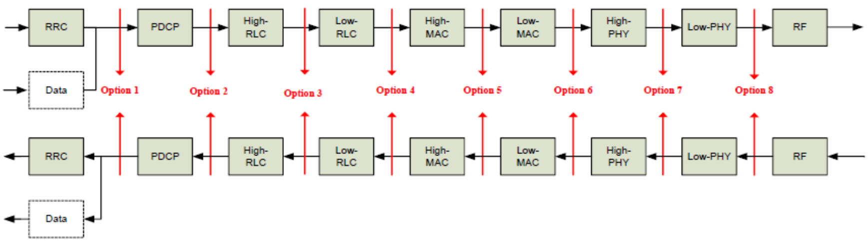


Figure 2.7: Functional Split between Central and Distributed Unit (extracted from [7])

In this context, 5G mobile networks face new challenges at the time to dimension and design the transport infrastructure, where fronthaul and backhaul traffic will be needed to be carried in a common infrastructure to control OPEX and CAPEX, hence, defining the so-called Crosshaul/Xhaul transport network [1]. In order to achieve so, this infrastructure should enable automatic reconfiguration of all network resources through a unified UP and CP interconnecting distributed 5G radio access and CN functions hosted on in-network distributed cloud infrastructures, as depicted in Figure 2.8. Such distributed cloud infrastructures will also be used to host constituents NFs of network services requested by vertical industries. The type of traffic (FH or BH), and its associated requirements, such as the kind of functional split for FH traffic, can shape the physical infrastructure deployment or constraint the possible kind of traffic within the transport network. Only the use of optical fibre (of constrained length segment) can satisfy the requirements in terms of throughput, latency and

[1]Next-Generation Fronthaul Interface (NGFI) defines newer variable bit rate (e.g., due to compression) fronthaul streams, which drives the interest in crosshauling due to the possibility of using cost-efficient packet switching schemes.

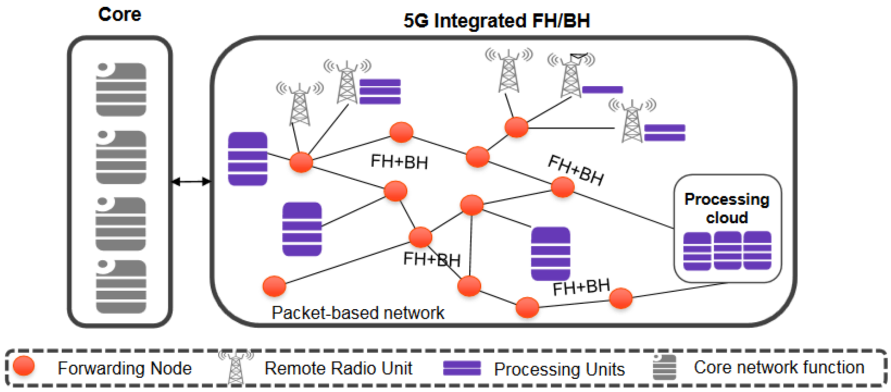


Figure 2.8: 5G Integrated FH/BH Crosshaul transport network (extracted from [8])

jitter of some functional splits. However, the use of fibre is not possible everywhere, being mostly available in urban environments, and its deployment is expensive and not fast enough to match the upcoming demand in mobile data. Such increase in the mobile data demands is envisioned to be supported, from a network perspective, by an increase of the network capillarity (densification), that is, larger number of base stations covering smaller areas. In this context, wireless solutions considering different spectrum bands, such as mmWave (E-band, W-band, D-band), microwave and sub-6GHz are key enabling technologies to transport the traffic from the RAN to the CN at the low aggregation level for highly densified deployments. Indeed, the trend is to move towards higher frequencies bands, such as E-band, with wider channels and more spectrum availability to cope with the transport needs of next-generation mobile networks [31]. Such wireless solutions should be deployed using multi-hop mesh topologies, as they show a higher level of availability and redundancy compared to other topologies such as tree or ring, being more effective in front of possible link failures and allowing load balancing strategies to mitigate congestion. Nevertheless, optical technology is still required at the high aggregation level, that is, aggregating several wireless *low aggregation sites*, as explained in [32]. This technology heterogeneity envisioned in 5G transport networks makes critical the interfacing between such wireless and wired technological domains from the user and data plane perspective because these domains can present very different protocol implementations for control and management purposes and provide very diverse level of capacity. In conclusion, next-generation mobile networks require that transport network shift from closed, static and inelastic networking infrastructures to open, scalable and elastic ecosystems integrating heterogeneous transmission technologies and distributed computing sites (close to the edge, close to the core) that will be used for different purposes according to the requirements of the orchestrated network services.

As a result of this thesis work, an end-to-end (E2E) network service orchestration architecture will be designed, implemented and evaluated by making use and evolving the concepts explained above as a contribution towards the transformation of the mobile transport network to realise this vision.

Chapter 3

State of the Art Review

As seen in the previous chapter, the next generation mobile network implies a redesign of its network architectural principles, requiring to be based on the presented technological paradigms to realise its vision. These paradigms are recent and still under development, but the research community is dedicating a lot of effort to design and develop (sometimes, overlapping) architectural frameworks to integrate and exploit its benefits in the mobile network landscape. This chapter presents a review of the existing literature that inspired this research and allowed us to formulate the research question presented in Chapter 4.

First, we review the use of SDN to configure and manage networking resources in mobile network scenarios. This section has a special emphasis on the transport segment because of the deep transformation that is expected to occur in next generation mobile networks, as previously explained in Chapter 2, passing from a static commodity role to an active and elastic ecosystem able to host and interconnect distributed computing resources to serve network services with different requirements.
Then, we introduce the network service orchestration concept, which will be further developed in Chapter 8, and review available initiatives and representative research projects proposing architectures performing the lifecycle management of network services implementing network slice instances in a flexible and dynamic manner, both from a computing and a networking perspective. To conclude this chapter, we review current research efforts considering the provision of network services spanning across multiple administrative domains managed by different organizations and counting each one with independent MANO stacks. This capability, greatly enabled by the introduction of SDN and NFV paradigms into the mobile network arena, will open the doors to the development of new business models.

3.1 SDN for mobile transport networks

If we had to fix the beginning of the "*SDN era*", we fix it with the famous Keynote given by Nick Mckeown at IEEE INFOCOM conference [13] back at the end of the previous decade. The initial research work in the SDN area focused on use cases mostly related to typical IT environments like university campuses, data centres and cloud networking. One of the first references presenting the use of SDN in mobile networks is the one by Li et al. [33] back in 2012. After that, several publications like MobileFlow [34] (2013), SoftCell [35] (2013) or Software Defined Wireless Networking (SDWN) [36](2014), proposed schemes to adopt SDN in mobile networks. In [37], we can find an exhaustive survey of all the surveys related to the use of SDN in mobile networks that have appeared during the last years. Among these references, the authors of [37] highlight the survey by Nguyen [38], which provides an interesting classification of proposed SDN-based mobile network architectures into revolutionary and evolutionary.

However, most of the references we can find in these surveys focus on the use of SDN in the RAN and the CN segments. Up to now, the transport segment was seen as a simple resource just connecting access networks with the core network mostly by means of optical communications. Nevertheless, the needs of next-generation mobile networks, such as the extensive integration of the use of wireless technologies as a cost efficient transport solution and the mix of backhaul and fronthaul traffic to keep OPEX and CAPEX under control is making more complex the management of the transport network. Additionally, we do not need to forget that distributed cloud infrastructures will be deployed throughout the transport network to host constituent network functions of network services that will require interconnection. In this sense, the SDN paradigm is called to play a relevant role in the transport segment to provide the required active and dynamic management of heterogeneous connectivities and networking resources.

With respect to the first item, the use of SDN to manage a wireless segment at the mobile transport network, the work in [39] presented and demonstrated an architecture to apply SDN for the wireless backhaul of small cells. Other works presenting the use of SDN for wireless small cell backhaul deployments are [40] and our previous work called WiseHAUL [41], which served as a basis for some of the work developed in the framework of this thesis work. With respect to the second item, the mix of backhaul and fronthaul traffic under the same transport infrastructure (and also integrating wireless as a transport technology) is an issue that has been tackled in the literature by recent European Horizon 2020 programme projects 5G-Crosshaul [8] and 5G-XHaul [42]. Up to then, literature solutions, like SoftAir [43], addressed the fronthaul issue as a separate problem and more focused on the access segment level. In particular, SoftAir, recommends to keep modulation and demodulation at the RRH and keep BBU close to the edge in order to reduce the data rate of possible CPRI traffic at the transport network. However, the authors of [44] present the 5G-Crosshaul solution, an architecture for integrating existing backhaul and new fronthaul networks into a flexible unified solution for 5G heterogeneous transport networks considering the capabilities of the SDN paradigm.

More generally, the benefits that the SDN paradigm can bring have motivated intense research in the last years to design and develop new control and management architectures and protocols to deal with the observed inflexibility and limited automation at the transport level. Current telco provider infrastructures are separated into multiple control domains, each using different network technologies at the transmission level, control interfaces and implementing forwarding policies with diverse goals. Management, configuration and troubleshooting processes rely extensively on human intervention requiring specialized knowledge to configure network elements due to the use of diverse vendor-proprietary solutions based on different versions of protocols like Simple Network Management Protocol (SNMP) protocol or cumbersome command line interface (CLI), hence limiting automation. Furthermore, service deployment is designed on paper by network managers. As a result, service deployment times are high (in the order of days). Per-vendor/per-technology transport domains, each with its isolated control employing proprietary extensions, hinder the dynamic end-to-end (E2E) global management of networking resources in transport networks.

With respect to control architectures, the preferred option in the literature is based on the use of a hierarchical approach within the same administrative domain, where a parent-child relationship is established between contiguous layers of SDN controllers dealing with complex deployments featuring multiple technologies at the control and at the data plane level. In particular, Casellas et al. presented in [45] a hierarchical solution based on stateful Path Computation Element (PCE) and OpenFlow to orchestrate network resources in a Generalized Multi-Protocol Label Switching (GMPLS) flexi-grid core transport optical network. This solution tackled the problem of the end-to-end provision of connectivity service between geographically remote data centres spanning multiple optical network domains. The hierarchical approach not only enables orchestration in the heterogeneous multi-technology, -domain, -vendor environment typical of envisioned 5G network deployments, but also improves scalability, modularity, and security as explained in the ONF architectural document [1].

The work in [46] evolved the work in [45], by introducing in the hierarchical architecture an implementation of the Applications-Based Network Operations (ABNO) framework [47]. This framework has been standardized by the Internet Engineering Task Force (IETF) and it is based on standard protocols and components to efficiently provide a solution to the orchestration of network resources of different control plane technologies. References [48], [49] or [50] are other examples of the use of hierarchical architectures for the orchestration of networking resources in multi-domain/technology transport environments. The 5G-Crosshaul project [8] has also considered the hierarchical approach to tackle network heterogeneity, as shown in our previous work ([51], [52]). Unlike the previous references on hierarchical approaches, our previous work presents preliminary design/architectural ideas including the joint orchestration of wireless (simple and mesh topologies) and optical transport domains and the initial characterization of a real experimentation setup. This work is evolved in Chapter 6 of this dissertation taking into account a more service management-oriented perspective (i.e., deployment and recovery of a set of coordinated connections), showing that the orchestration of network resources in a complex heterogeneous transport network can be done in the order of seconds, which is a big leap with respect to current telco providers' control mechanisms, based on more static procedures with a high degree of human intervention.

A relevant aspect in the hierarchical approach to allow the inter-controller coordination is the use of generic interfaces at the NBI interface, like the Control Orchestration Protocol (COP) [53], which has been developed and evolved within the context of European Research Projects ([54], [8]). This interface has been used in several research works, like the previously mentioned [51], [52] or a more recent one [55]. COP is the basis for similar efforts later carried out, such as ONF T-API (Transport API) [56], which is related to packet, optical and microwave equipment. Thanks to its open nature, the T-API interface and protocol has been increasingly considered to be used in the context of transport networks, as reflected in white papers covering industry interoperability events [57] and early proofs of concept (PoC) in more specific NFV scenarios [58].

Up to now, the previous paragraphs mainly focused on control operation aspects, which ultimately relied on the OpenFlow (OF) protocol to configure the forwarding/routing behaviour of the switches and routers in the network. As mentioned in Chapter 2, there are two differentiated types of SBI operations depending on their purpose: control and management. In the following paragraphs, we are going to tackle the other type of SBI operations, the management operations, which are related to the configuration and administration of NEs. The automation of both types of operations, control and management, introduced by the SDN approach enables a dynamic optimization of the network performance. Indeed, for a centralized network management, common abstractions and interfaces are essential in SDN deployments to enable open and vendor-agnostic management of networking equipment.

The work in [59] presents an overview of SBI interfaces for SDN in optical core transport networks. In this work, the authors cite recent efforts (*OpenConfig* [60] and *OpenDevice* [61]) based on Network Configuration/Representational State Transfer Configuration (NETCONF [62]/RESTCONF [63]) protocols and YANG [64] modeling to create vendor neutral data models addressing the management of different optical technologies and which are derived from operational needs, use cases and requirements from operators. However, it is worth mentioning that the first efforts to manage optical networking equipment in an SDN deployment were based on extensions of the OF protocol for optical networks [65].

For the case of wireless technologies, a similar evolution has happened. The ONF created the Wireless & Mobile Transport Group to apply SDN principles to microwave wireless transport networks. In their first PoC, back in 2015, they also considered extensions of the OF protocol to handle not only the control but also the management of wireless transport equipment. This approach has also been followed by other research work, such as [39], [40] or [66]. In these works, the proposed OF extensions allowed the SDN controller to retrieve statistics from Wi-Fi interfaces in the transport nodes so they can perform actions to reduce the effects of interferences or switch on/off devices for energy efficiency purposes. However, the Wireless Transport SDN group of the ONF changed the strategy and started to work on common information models of microwave network devices to enable a more automated and unified management of wireless network devices, hence, decoupling the management from the control operations. Then, the following PoCs of the Wireless Transport SDN group of ONF, demonstrated the capabilities and benefits from utilizing such common information models (ONF TR-532 [67]) for multi-vendor management of microwave transport network devices. This information model is exchanged by means of the NETCONF protocol [62], a general-purpose management protocol, which uses YANG [64] modelling language to implement the data model. At the network device, an adapter, what ONF calls mediator, is used for translating the NETCONF/YANG information model to/from the existing proprietary management procedures of each vendor's device.

3.2 Network Service Orchestration for Mobile networks

NFV and SDN technologies, mainly, through their software-centric approach, provide the network operators with new ways to flexibly and dynamically create, deploy, manage and provide the required resources (compute, network, storage) to network services serving end-to-end network slices in next-generation mobile networks. This process, which involves complex architectures and workflows, can be defined as network service orchestration (NSO). This is the key idea connecting the work performed in this thesis and will be further revisited in Chapter 8.

This simple definition expects to be the union of all the aspects covered in the definition of the NSO concept proposed at different research projects and contributed to the different related standardization bodies (e.g., ONF, ETSI, NGMN, ...), as covered in [68]. Next, this section presents relevant NSO frameworks to this work. These frameworks, most of them supported by operators and big organizations through different open source collaborative initiatives, are in continuous development and experiencing great architectural, operational and functional evolution during the last years thanks to the results produced in the context of different related research projects. Then, we review the efforts done to extend the NSO concept to coordinate deployments in multiple administrative domains, and more specifically from the network service perspective. Both topics are of relevance for the scope of this thesis work, as explained in the following chapters. We refer the reader to [68] for further analysis and a complete taxonomy study of the NSO concept.

3.2.1 Reference Network Service Orchestration frameworks

Open source and standardization around NSO are essential to provide fast innovation and increase the interoperability in 5G networks. Currently, there is still a continuous and intensive research effort to continue identifying relevant architectural options and challenging use cases to provide recommendations, upon which specifications will be defined to come up to a complete and unified NSO concept.

Next, we provide a brief description of some of the most representative frameworks providing NSO capabilities based on the available online information and research papers. Some of these frameworks are open source initiatives under the patronage of big operators, are supported by big networking organizations like the Linux Foundation, and base on the architectural guidelines provided by main standardization bodies, like the ETSI-NFV [69]. Other examples included in this section are the result of European research projects.In the last years, these research projects, conscious of the traction of the mentioned open source initiatives, are contributing actively to these open source initiatives. These contributions aim at developing and proposing architectural extensions enabling richer NSO procedures realising the next-generation mobile network vision. Indeed, in the provided NSO framework descriptions, some missing architectural aspects tackled in this thesis are highlighted.

Recently, comparative performance analysis among some of the orchestration frameworks included in this section have appeared in the literature ([70], [71], [72]). In these analysis, we can observe that the different alternatives are under different architectural development stages, including different set of features and they are in continuous evolution, releasing new versions periodically (e.g., every six months). Throughout these releases, these frameworks gain in stability and present a more solid architectural design thanks, in part, to the assimilated research contributions. An additional interesting remark mentioned in the analysis performed in [71], and also observed during the development of this thesis, is the fact that there are still gaps between what is being claimed and what features and functionalities are actually supported (and under which conditions) by the architecture of these NSO frameworks. The amount of available information, documentation, and experience in terms of the functional and operational capabilities supported by the architecture of these platforms and its technology readiness level is limited due to its continuous development process. Thus, there is still room for research to propose enhanced architectural options enriching the NSO concept to achieve higher degrees of automation and programmability.

1) **Open Source MANO (OSM) [73]**: It is an ETSI-hosted open source MANO community project released under the Apache Public License 2.0. OSM is an operator-led (Telefonica) project whose objective is to jointly innovate, create and deliver a MANO stack that is step by step closely aligned with ETSI-NFV information models to promote the integration between standardization and open source initiatives. OSM set out in May 2016, and in May 2018, they released its fifth major release OSM Release FOUR (OSM R4) which constituted a huge leap forward in terms of functionality, user experience and maturity [72]. Since its initial release, the development roadmap of OSM project considers the release of a new version of the project every six months and the last release, OSM R9 was delivered in December 2020. Throughout the successive releases, increased stability and better user experience have been achieved so the OSM platform is maturing into a production-ready solution, whose focus is to improve the orchestration experience for not only VNFs but also considering physical network functions (PNFs) and, more recently, introducing, container network functions (CNFs). Relevant features claimed through the successive release since OSM R5 are: the support to the deployment of network slices, continuous monitoring and policy framework enhancement to enable (auto)scaling capabilities, increased number of connectors to different VIM types (like FOG05 Edge Cloud or VMWare's VCloud Director) or TAPI-based transport networks, progressive adoption of ETSI-NFV standards and the capability of leveraging Kubernetes [74] to orchestrate CNF based network services (since OSM R7), hence bringing cloud-native applications to NFV deployments.

Based on the different architectural options presented in ETSI-NFV IFA009 [75], OSM implements a generic VNFM as an integral component of OSM and split the different functionalities into submodules with specific tasks. Thus, OSM architecture has a Resource Orchestration (RO) component to

handle cloud and physical resources, a VNF Configuration Agent (VCA) component to handle the interaction with VNFs/applications, and a monitoring component (MON) module in charge of monitoring procedures, on top of which sits a layer of Service Orchestration (SO), in charge of providing end-to-end coherency. It is worth noting that the generic VNFM implemented by OSM does not require the VNF to offer a specific type of interface. OSM can consume whatever particular interface is offered by a VNF and/or its element management system (EMS) by using VNF-specific code plug-ins, realised as proxy charms with Juju software [76] —ready to run in a container in the VCA— and which come as part of the corresponding VNF package, hence, supporting any of the interfaces which are currently presented in commercial VNFs.
OSM presents SDN assist capabilities. Through this capability, the RO module of OSM can manage the data plane underlay connectivity through an external SDN controller. However, this capability is limited to a deployment in an environment with only a single switch and requires the mapping between the switch ports and the NFVI element interfaces (VIMs). Hence, OSM cannot automatically handle the management of required network connectivity in complex scenarios like multi-hop heterogeneous transport networks at the WAN level. This kind of scenarios allow connecting different NFVI-PoPs managed by different VIM entities where the VNFs of the same network service (NFV-NS) have been distributed[1], hence limiting its VIM multi-site capabilities. Indeed, in OSM, such distribution of VNFs of the same NFV-NS in different NFVI-PoPs can only be achieved manually because of some architectural lacks. In order to automate such decision and make a contextual NFV-NS deployment based on NFV-NS requirements and the status of the NFVI infrastructure, the orchestration stack would need to receive information (or at least an abstraction) characterising the topology and the resource availability both at the VIM level and the WIM level. Furthermore, OSM framework counts with the additional lack that current information models used for the description of virtual links interconnecting VNFs only include bandwidth attributes and not other quality of service attributes like latency. Thus, its information model presents a limited view to express and map requirements for a NFV-NS. It is worth noting that ongoing efforts in this direction are included in latest releases of OSM (from OSM R8), where a placement optimization module based on user-provided models of compute and networking costs and latency and jitter metrics for inter-VIM connectivity. However, this approach lacks of enabling the required resource abstraction methods enabling the required degree of automation. Finally, OSM architecture covers only single administrative domain scenario, currently not supporting multi-administrative domain orchestration processes involving the interaction with other OSM instances for a coordinated deployment.

2) **Open Network Automation Platform (ONAP) [77]**: It is an open source networking project hosted by the Linux Foundation licensed under the Apache Version 2 License. Announced on the 23rd of February 2017 as the result of merging the AT&T OpenECOMP and Open-Orchestrator (Open-O) projects, ONAP released its unified first version (Amsterdam) in November 2017. Since then, new versions have been released every 6 months, being the Guilin version the last one delivered in December 2020. Throughout these releases, ONAP has developed multiple "blueprints" covering different use cases with the help of the added framework features being developed in alignment with ETSI and 3GPP specifications. One of the most relevant developed blueprints is the 5G blueprint, a multi-release effort with key initiatives around end-to-end service orchestration, network slicing and VNF/PNF lifecycle management. Built as a cloud-native application, ONAP is designed as a microservices-based system, with all components released as Docker containers to be a highly reliable, scalable, secure and easy to manage the platform. ONAP facilitates service agility by providing a common set of Northbound REST APIs that are open and interoperable, and by supporting YANG [64] and TOSCA [78] data models. However, as mentioned in [71], making it operational in a laboratory environment is not straight forward because it requires a lot of expertise in current cloud computing networking platforms

[1]During the development of this thesis work, since OSM R5, among its new features claimed the extension of information models to support inter-DC connections using T-API information models. However, there is not yet available public documentation illustrating this use case

including OpenStack [22], Kubernetes [74] and Docker [79] containers, among others, and presents a much bigger installation resource footprint when compared to other NSO frameworks, like OSM.
From a general point of view, ONAP architecture can be split into two basic groups: the ONAP Design Time Framework and the ONAP Runtime Framework. The main role of the design time framework is to provide the tools, techniques, and repositories for defining/describing resources, services and products. This component also defines recipes for instantiating, monitoring and managing VNFs, PNFs and service policies. It is also responsible for the distribution of these specific design rules into the runtime framework. The Runtime framework contains meta-data driven modules enabling VNF configuration and instantiation, and delivers real-time view of available resources and services, hence executing the rules, policies and other models distributed by the design time framework.
As with OSM, ONAP architecture does not consider the contextual NFV-NS deployment based on requirements and NFVI resource availability. With respect to the allocation of network resources in the transport domain, one of the latest releases (ONAP Frankfurt) claims ongoing architectural enhancements to support a blueprint use case tackling multi-domain optical transport network connectivity between different carrier providers involving multiple instances of ONAP. In the found online documentation of this blueprint, the connectivity of NFVI-PoPs within a single domain is not considered. Additionally, although being multi-domain, documentation only focuses on network resource connectivity between domains, but not dynamically handled in conjunction with the management of needed computing resources to deploy the different constituent parts of the NFV-NS.

3) **Cloudify [80]**: It is a quite mature open source solution for NFV orchestration licenced under the Apache License Version 2.0 that is already widely implemented in production environments. Despite of the open source nature of this product, Cloudify is a vendor solution and is developed by a single company named Gigaspaces. It works with YAML (Yet Another Markup Language) DSL (Domain Specific Language) TOSCA configuration files called *blueprints*. These *blueprint* files describe the execution plans for the full end-to-end NFV lifecycle orchestration of the service for installing, starting, terminating and orchestrating. Cloudify, by serving as the NFVO & VNFM of the ETSI-NFV MANO architecture, has a powerful workflow and flexible plugin mechanism to interact with external elements such as VIMs. Cloudify contributes to the development of NFV-MANO activities being member of various collaborative projects and open source communities like OASIS TOSCA, OpenStack or ONAP. Cloudify architecture also presents the limitations of the previous solutions to perform complex NSO operations mostly due to fact that does not handle topological and resource availability abstractions representing the underlying NFVI infrastructure. Among these limitations, its architecture prevents the consideration of NFV-NS requirements to determine a contextual deployment of VNFs attaining improved resource usage, lacks the management of transport network resources for distributed deployments of VNFs or misses multi-administrative domain orchestration capabilities, between others.

4) **Tacker [81]**: It is an official OpenStack project released under Apache License Version 2.0. It is based on the ETSI MANO architectural framework and builds a generic VNFM and NFVO to deploy and operate network services and VNFs on an NFV infrastructure managed by OpenStack. It uses TOSCA templates for VNF and NS descriptors, which are provided to the NFVO and VNFM. Among the NFVO tasks, we find the resource management in the VIM and the management of network services. The VNFM is in charge of the basic life-cycle of VNFs (create/update/delete) and facilitate initial configuration of the VNFs. TOSCA templates are used for VNF and network service descriptors. Tacker components are directly integrated into OpenStack, thus limiting the interoperability with other VIMs. It only works in single-administrative domain environments with ongoing support, under very restrictive conditions, of multi-VIM deployment. It is worth noting that this multi-VIM deployment does not consider dynamic configuration of network connectivity between sites, and in consequence, Tacker architecture cannot support the contextual deployment of NFV-NSs.

5) **OpenBaton [82]**: it is a project developed by Fraunhofer FOKUS Institute and the Technical University of Berlin. Released under Apache 2.0 licence, OpenBaton is an open source platform

providing a comprehensive implementation of the ETSI-NFV MANO specification using TOSCA as description language. The main components of its architecture are: an NFVO, a generic VNFM, a Juju VNFM adapter and an Autoscaling and Fault Management System. It incorporates a messaging system for dispatching events and presents a marketplace integrated with its dashboard for downloading available VNFs developed for the Open Baton framework. Open Baton is included as a supporting project (Orchestra[2] to integrate its functionality within the OPNFV initiative [83], another initiative of the Linux Foundation to deliver a standard NFV/SDN platform for the industry. Like in the previous cases, the proposed architecture is neither oriented to support the dynamic management of networking resources at the WAN level nor for multi-administrative domain service deployments.

The following NSO frameworks are examples of some of the first NSO open source solutions that have been developed within the context of European research and innovation programmes. In the last half of the decade, a large set of NSO-related projects have been funded by European Union Horizon 2020 programmes under the umbrella of 5G-PPP phases 1,2 and 3 [3], as described in [68]. Actually, the work in this thesis has been developed within the framework of some of them, namely the previously mentioned 5G-Crosshaul [8] and the 5G-TRANSFORMER [84] projects. Throughout these projects, different aspects of the NSO concept have been researched, which have derived in more complex architectures considering more complex concepts like network slicing or dynamic NFVI-PoP interconnection. However, the survivability of such contributions could be limited since these solutions may be tied to specific complements (like OpenStack), may have a smaller community (both of users and developers), are partly oriented to specific use-cases and prone to be discontinued because of its limited funding time horizon. For these reasons, through the mentioned phases, these research projects have been aligning with some of the previously mentioned initiatives, such as OSM, to contribute actively to its development and evolution by proposing alternatives to the detected architectural and functional lacks [4].

6) **Tenor [85]**: this orchestrator has been developed within the T-NOVA European project (01/2014-12/2016), one of the first research projects dealing with the NSO concept. The T-NOVA architecture presents a modular structure based on microservices, taking its concept from the generic definitions of ETSI-NFV ISG models at that time and expands it with specific add-on features. The main add-ons are: (i) at the NFVI management layer, besides the VIM, there is the WAN Infrastructure Connection Manager (WICM) to control the networking infrastructure and (ii) they also include a marketplace to promote VNF service offerings and facilitating commercial activity and seamless interaction among the various business stakeholders, like in the Open Baton platform. Tenor also implements automated placement intelligence of VNFs at deployment time, resulting in more optimal resource allocations and increased VNF performance. However, it is not oriented for multi-domain scenarios. In addition to this, its VIM relies only on OpenStack and the WICM on the OpenDayLight [19] SDN controller, potentially limiting its scope of operation.

7) **SONATA [86]**: it is the outcome of the European SONATA [86] project (07/2015-12/2017), which has been continued by the 5G-TANGO [87] project (06/2017-01/2020). Currently at the Release 5.1, SONATA platform aims to provide flexible programmability and deployment optimization of software networks for complex services/applications. This is achieved by applying an integrated development and operational (DevOps) process. The major components of the SONATA architecture are: (i) a service development kit (SDK) and a validation and verification (V&V) platform to support developers with the creation of innovative VNF/NS, and (ii) the SONATA Service Platform acting as the MANO framework to interact with the underlying VIM and WAN Infrastructure Manager (WIM). The SONATA Service platform includes a Slice Manager, which was also contributed to the OSM project, to deploy network slices between different VIMs in a single administrative domain. However, SONATA

[2] https://wiki.opnfv.org/display/PROJ/Orchestra
[3] https://5g-ppp.eu/
[4] https://osm.etsi.org/wikipub/index.php/Research

platform does not support multi-administrative domain deployments.

3.2.2 Multi-administrative domain network service orchestration

As explained in the previous subsection, one of the common points of the architecture of the described NSO frameworks is the lack of capability to orchestrate network services spanning multiple administrative domains (ADs). In next generation mobile networks, such scenarios are very relevant because these networks are expected to operate in highly heterogeneous environments, not only at the *"resource layer"* by supporting different access and transport technologies, but also at the *"service layer"*, where some necessary functions to deploy network services (NFV-NSs) mapped to network slice subnet instances can be provided by different organizations, hence requiring service federation capabilities, as we further develop in Chapter 10. Herein, we understand federation as a group of service providers agreeing upon standards of operation so they can work in a collective fashion.

In the literature, the service federation problem has been treated from two perspectives. First, some work deals with the problem of distributing the different component parts of an NFV-NS among multiple ADs and second, other work deals with the procedures and interfaces to make effective this distribution and allowing the real instantiation of NFV-NSs among different ADs.

With respect to the problem of mapping NFV-NSs across multiple domains, there are two main types of approaches proposed in the literature, namely *centralized* and *distributed*. In the *centralized* approach, a central third party entity collects information about computing resources (including its position) and the network topology in different ADs and decides on the NFV-NS partitions to deploy the different *"sub-chains"* in different domains, as in [88] and [89]. In the *distributed* approach ([90], [91]), the privacy of each domain can be maintained because infrastructure information is not shared between different domains. However, this lack of topology information and resource availability increase the difficulty of solving the multi-domain orchestration.

Previous references presented exhaustive simulation results of their proposed algorithms to map NFV-NS requests to generic multi-AD deployments. However, these works made strong assumptions like assuming that constituent parts of an NFV-NS are available in all the domains or considered a flexible VNF ordering, which could not be the case in real scenarios. Additionally, they do not delve into the required workflows and interfaces defining the required interactions between domains to make effective this partition and effectively instantiate the requested NFV-NS under current NSO frameworks. This last aspect is more aligned with the work performed in this thesis. With respect to this issue, the following paragraphs present related work where standard development organizations (SDOs) and research projects make recommendations, provide guidelines or present architectures and more or less detailed workflows covering such kind of deployments.

Regarding SDOs, from the point of view of the SDN architecture, the ONF presented an initial approach based on a peer-to-peer relationship at [15], where SDN controllers invoke services from other controller through the A-CPI interface establishing a client/server relationship. Another SDO, the Metro Ethernet Forum (MEF) defines the Lifecycle Service Orchestration (LSO) reference architecture [92], which handles multi-domain aspects. MEF proposes two inter-provider reference points: *LSO Sonata* and *LSO Interlude*. The first one addresses business interactions between providers, such as ordering, billing, trouble ticketing. The second one deals with the creation and configuration of connectivity services, as permitted by service policies. Notifications and queries on the service state are covered also by this interface. The Interlude interface is the one used by the mentioned ONAP blueprint tackling the connectivity of multi-domain optical transport networks. Finally, in 2018, ETSI-NFV presented a report on architectural options for multi-domain [9] based on the ETSI-NFV MANO architecture. In this report, the use case entitled *"Network Services provided using multiple administrative domains"* is of particular interest for this thesis work. This report presented a new reference point called *"Or-Or"* (Orchestrator-Orchestrator), which handles the peer-to-peer relationship between orchestrators in a provider/customer fashion. This reference point has been specified in the ETSI-NFV IFA030 [93] document. This specification effectively considers to use a subset of the procedures defined in ETSI-NFV IFA013 [94] for the *Or-Or* interface to perform the management of NFV-NS descriptors and the NFV-NS lifecycle management operations. However, the scope of the work in ETSI-NFV IFA028 and ETSI-NFV IFA030 presents a conceptual lack related to inter-domain resource orchestration aspects enabling the integrated orchestration of nested network services running in different ADs and the dynamic management of its required inter-domain connections, which can entail great complexity. This gap has to be also covered to effectively perform service federation among multiple ADs.

Regarding research projects tackling the multi-AD deployment problem for 5G networks, we find, for instance, the initial efforts from 5G-Exchange [95] and Vital [96] projects. 5G-Exchange relies on a peer-to-peer interaction between orchestrators, where each one is administered by an operator, to deploy services end-to-end in case an NFV-NS request cannot be honoured by the sole use of the own resources controlled by an orchestrator [97]. This consideration needs the different constituents parts of an NFV-NS to be available in all the involved ADs, which, as mentioned before, may not be the case. The Vital project developed the X-MANO framework [98]. Its architecture, introduced the concept of federation agents coordinated by the so-called federation managers (FMs), which worked also in a peer-to-peer manner with other FMs to orchestrate the network service across multiple ADs. However, this approach does not focus on the inter-domain connectivity solution at the data plane level, which is a relevant aspect to consider, as previously mentioned. In addition to these projects, more recent ones (5G-EVE [99], 5G-VINNI [100], 5GENESIS [101]) also consider relevant the creation of an end-to-end network ecosystem supporting the deployment and management of services across distributed domains in different testbed sites. These projects are still under development and a detailed description of its final architecture and implementation details are yet to be published to see if and how they handle the deployment and interconnection of parts of the same service deployed at the different sites and domains. To conclude this section, we mention the contemporaneous approach presented in [102], called 5GUK Exchange. This work, one of the limited examples going beyond architectural considerations and also experimentally assessed, proposes a thin hierarchical multi-domain orchestration layer where a central entity builds on top of existing MANO systems and performs only service orchestration and interconnection of domains, whereas resources are managed and controlled by the individual ADs, so they preserve the privacy over specific details and control of their networking and computing resources. This centralised approach is also considered by the three previously mentioned projects. However, the peer-to-peer approach seems to be better aligned with communications schemes and business relations between service providers, hence providing a more general framework for service deployments involving multiple administrative domains, as further developed in Chapter 10.

Chapter 4

Research Question

This chapter describes the research problem that this thesis addresses. First, Section 4.1 formulates the question that constitutes the basis of the research work developed in this thesis. After explaining the different aspects involving this question, we discuss in Section 4.2 the relevance of this question for the research community based on the shortcomings of previous research work presented in Chapter 3. Then, in Section 4.3 we present the potential impact of providing a solution answering the specific research question not only from a technical point of view but also from an economical point of view.

4.1 Question Formulation

Next-generation mobile networks aim to offer very different and diverse kinds of innovative services grouped under three big categories, namely eMBB, URLLC and mMTC, with the purpose of creating new business models attracting new players to the telecom network ecosystem, mainly, the vertical industries. This challenges current architectures and management schemes, implying a deep redesign of the mobile network architecture to allow the concurrent and timely deployment of such services in a shared infrastructure constituted by a heterogeneous set of resources. With this in mind, the research problem that this thesis tackles is:

How mobile network architecture functionality needs to be designed and developed for reducing the end-to-end service provisioning time in a heterogeneous and multi-provider computing and networking shared platform for fulfilling its service lifetime operational needs?

Next, we dissect the previous question commenting the meaning of the different underlined aspects:

- ***mobile network architecture***: Up to now, the design of mobile networks focused mainly on the design of the access and the core network segments, considering the transport segment as a *commodity or a dumb pipe* where the traffic flows. However, in next-generation mobile networks, the transport segment becomes a relevant and an active player to realize the 5G vision. As covered in Section 2.4, the transport network will have to multiplex FH and BH traffic into the same substrate network combining multiple transmission technologies and control schemes. Furthermore, it will require the integration of distributed processing sites where RAN and CN network functions, as well as those requested by vertical industries, will be deployed. Under such context, this work contributes towards the transformation and management of mobile transport network architecture to become an elastic ecosystem supporting the requirements of continuously and dynamically deployed network services and its interconnections. The following points develop the ideas considered in this work to achieve this transformation.

- ***designed and developed***: As mentioned in Section 2.4, next-generation mobile networks will be redesigned upon modularization and decomposition of network functions (NFs) to provide the requested logical network realizations to satisfy the different requirements. Softwarization techniques, such as SDN and NFV, are seen as the main technological pillars to achieve this vision because they can provide the needed flexibility, dynamicity, and automation capabilities to deploy the required network slice instances. In spite of the current level of development of such techniques, mostly for wired and fixed networks, they are quite recent and their deployment into the mobile field is still considered a gigantic task [37], where research will impact its development and realization. This research work not only addresses SDN and NFV separately, but also as a whole under the design of management architectures developing the multiple aspects of the NSO concept introduced in Section 3.2. The development process will benefit from the adoption of open source tools based on the different standardised protocol and architecture specifications defined for SDN and NFV. Among such open source tools, we find software switches or SDN controllers for SDN and the Open Source MANO [73] (OSM) or Cloudify [80] orchestration platforms for NFV. Furthermore, SDOs can also benefit from this, since these tools and their associated developer community can help providing open and early feedback in front of ambiguities or detected functional deficiencies.

- ***end-to-end service provisioning***: The foreseen dynamicity and flexibility to deploy network services in 5G networks and beyond impose an automated, integrated, and coordinated network-wide orchestration approach instead of network re-architecting and node-by-node management procedures as done in legacy systems. Furthermore, forthcoming 5G mobile networks will count with additional resource management tasks than the networking one with respect to legacy systems: the availability of reconfigurable and distributed pools of processing resources where virtualised NFs (VNFs) can be deployed. Hence, an appropriate architectural design of a management and orchestration (MANO) plane is essential to ensure an efficient, joint and coordinated utilization of the set of available resources placed in the mobile infrastructure to deploy network services while satisfying their requirements and trying to obtain the maximum revenue of the available deployed resources. In order to do so, the handling of proper resource abstraction models, the interaction with VNF placement algorithms, the application of network slicing concepts to provide isolated resources to the different network service instances, and the availability of open and standardized interfaces will be relevant.

- **heterogeneous and multi-provider**: In this work, we cover two different aspects when considering multi-domain transport networks. By heterogeneous, we mostly refer to networks which can feature different transmission and control technologies at different network segments/aggregation stages at the transport level, such as wireless (mmWave, sub-6GHz), multi-layer optical, and packet. The multi-provider aspect refers to the possibility of interacting and coordinating with other administrative domains (ADs) managing and controlling their own infrastructure with an independent MANO framework instance to deploy network services or some parts of it in a coordinated manner. This kind of interaction between ADs can be required due to business (e.g., cost) or technical reasons (e.g., service availability and coverage, lack or resources).

- **fulfilling its service lifetime operational needs**: The mentioned flexibility and dynamicity is not only important during provisioning time but also during operation time. Network services are defined to operate with a set of resources under certain conditions. Additionally, their definition may consider multiple structures and actions, like scaling the amount of VNF instances, to adapt to different changing service demands and keep their proper working. The orchestration tasks do not finish at service provisioning time. They also continue during the lifetime of the network service to detect the situations that require of corrective orchestration actions (scaling operations) to adapt the structure of the network service according to the network conditions and make use of the needed resources at each time if and where available.

4.2 Validity of the Question

As presented in Section 2.4, next-generation mobile networks entail a big revolution blurring the borders between network segments. It is unquestionable that SDN, NFV and network slicing will be the key pillars of the required transformation to realize this vision, but still it is needed more research to adapt these concepts to the mobile environment. In the state-of-the-art Section 3.1, it has been discussed how SDN provides solutions to the detected problems in current management schemes of telco provider heterogeneous transport networks. Thanks to a hierarchical SDN approach using common interfaces and resource abstraction models, the end-to-end global management of networking resources can be done reducing the provisioning time derived from current manual and isolated (per-vendor/per-technologies) procedures.

However, the presented hierarchical solutions mostly considered heterogeneity on technologies found in data centre environments, the original context of SDN deployments. Instead, this work delves into the inclusion of wireless technologies and also considering the different possible kinds of traffic available in the transport network (FH/BH) from a network service perspective, i.e., the integrated management of a set of coordinated connections. In next-generation mobile networks, wireless technologies will play a great role at the edge of the network to provide the needed densification to cope with the expected capacity increase demands. In addition to this, densification will rely on new emerging wireless transmission technologies, such as mmWave, that will need to be integrated into the softwarised paradigm to automate its control and management.

But the deployment of network services goes beyond the automation of network resource provisioning. Here is where the interaction between NFV and SDN is clearly identified. As further developed in Chapter 8, the basis of the NSO concept is to understand the semantics of the service to easily automate the interaction with control entities allowing the configuration of not only the required transport network resources at the WAN level, but also of computing resources at the distributed processing sites hosting VNFs to create end-to-end services implementing network slices while satisfying its requirements. A logically centralized NSO element could have a complete network infrastructure view independently of the underlying technologies employed at each technological domain supported by appropriate resource abstractions and the use of open interfaces. This NSO element is considered as a solution to optimally determine the distribution of VNFs composing a network service across different processing sites (NFVI-PoPs) managed by different VIM instances to make an efficient use of both cloud and network resources while honouring its requirements. Additionally, further orchestration actions may be needed to be carried out during service lifetime to adapt the structure of the service to provide the appropriate resources for a correct service working when needed and for the required time, hence coping with the network conditions and service demands at each time. However, such orchestration element is not found in current state-of-the-art solutions of service orchestrators or are recently starting to include such considerations into their architecture.

Finally, another common missing aspect among current analysed service orchestration systems is that they do not tackle the multi-provider network service deployment problem requiring service federation capabilities between involved administrative domains. Reviewed state-of-the-art contributions in Section 3.2.2 presented, on one hand, simulation studies focusing on different network service partition strategies across generic multi-domain networks and considering that VNFs can be placed anywhere. From the defined strategies, a peer-to-peer relationship between ADs seems to be more suitable than a centralized approach for the mobile network case. The peer-to-peer scheme may avoid issues like global trust among providers when compared with a scheme using a single logical exchange point. On the other hand, standardization bodies have generated reports with guidelines and research projects have designed some high-level architectures, but most of them without a clear definition of complete interfaces and mechanisms addressing all the aspects related to multi-provider network service deployments, like resource-oriented actions to coordinate and enable the inter-domain connectivity between elements of the same network service deployed in different ADs. In this sense, there is still room for the design of such interfaces and mechanisms to verify their feasibility under real conditions implying business/operational constraints, which is a lack in the literature. Moreover, as previously mentioned, the orchestration capabilities need to be maintained during service lifetime to provide the required dynamicity and flexibility coping with changes in the network conditions. Such consideration for this kind of service deployments implying multiple ADs is scarce in the literature, where only a reference work done by an SDO, the ETSI-NFV IFA028 [9] report, treats this problematic for a reduced set of scaling situations from the network service perspective, again disregarding the resource orchestration perspective related to the inter-AD service connectivity. This thesis also addresses this aspect for multi-AD scenarios, by extending previous mechanisms and interfaces devised for the instantiation phase to network service operation time while considering a broader set of scaling situations.

4.3 Is it a worthwhile question?

The value of the research question formulated in Section 4.1 must be considered from two perspectives. One is the impact that the proposed solutions may have in the research community and the other one is the economical implications of this research.

From the research perspective, on one hand, the proposed architectural design, associated procedures and interfaces aim at providing new insights to the NSO concept to increase the automated control and management of next-generation complex and heterogeneous mobile network infrastructures. In this sense, it is worth mentioning the importance of the experimental phase of the cyclic methodology followed in this work. Starting from a conceptual and theoretical approach, the concepts developed in this work are integrated in a real experimentation platform to be refined, assessed and to detect possible gaps not discovered during the concept phase, which may limit its realisation. Literature shows a lot of work about next generation mobile networks. This work usually focuses on formal concepts, theoretical architectures, and analytical or simulation-based performance evaluations. However, proof of concepts (PoCs) in real experimentation testbeds following realistic use cases in terms of network service descriptions or assumptions plausible to real scenarios under the framework defined by standardization bodies, like ETSI-NFV, are limited. Such PoCs become essential to demonstrate the correct operation and performance of related architectural concepts and associated technologies. Hence, it is clear that bridging this gap, providing measurements that could be used as reference for benchmarking purposes, and understanding deployment issues is of great interest for the research community. Moreover, the lessons learnt from the experimental approach can be very useful to propose evolutions on current standards and specifications and help/improve the design of network services understanding the needs of new mobile stakeholders, the vertical industries, hence fostering their integration in the ecosystem. This integration may not be as straightforward as expected due to the different knowledge backgrounds. On the other hand, the outcome of this work could be taken as the required basis to enable the next research challenge in next-generation mobile networks: the effective provision of autonomous networking [103]. The application of data-driven Artificial Intelligence (AI) and Machine Learning (ML) (AI/ML) techniques into the networking field is one complex task that researchers are currently facing and which is receiving a lot of interest from diverse working groups of different SDOs, like ETSI-Zero Touch Network & Service Management (ZSM) [104], ETSI-Experiential Networked Intelligence (ENI) [105], or O-RAN Alliance [106]. Effective architectures and interfaces supporting a high-degree of automated operations like the ones explored in this work overcome the limitations of current silo-based (per-domain manual operation) and vendor-dependent infrastructures. They are the base to ease the deployment of autonomous networking agents performing close-loop operations to provide efficient service provision, operation, fault management and resource orchestration capabilities.

From the economical perspective, the research question studied in this dissertation is the problem mobile network operators (MNOs) are currently facing with 5G networks. In a context of ever increasing mobile data traffic required per user at a decreasing Average Revenue per User (ARPU), the operators see the need of satisfying new types of services requiring URLCC and mMTC capabilities demanded by new telecom ecosystem players, the vertical industries, as a big driver for their business development. 5G will set the ground for the all-connected world of humans and objects, serving as a catalyst for disruptions in other technologies and business fields beyond information and communications technologies (ICT). Nevertheless, this will be viable if OPEX and CAPEX are controlled in ever increasing heterogeneous network deployments. The solutions proposed in this Thesis, based on softwarization techniques, will allow reducing such expenditures derived from the use of common resource abstractions, open interfaces, automated NSO procedures, and the use of commodity hardware implementing diverse network functions. The coordinated use of these diverse softwarization techniques will allow the dynamic and automated deployment of network slices serving network services when needed and as required. Such automation feature tackles the defined Key Performance Indicator (KPI) defined by the 5G Infrastructure Association PPP [107] of *"Reducing the average service creation time cycle from 90 hours to 90 minutes"*. In current telco networks, the time required to introduce a new service is long because of the manual procedures to reconfigure and re-architect the network with the inclusion of dedicated specific hardware appliances at a fixed location as a consequence of its monolithic system design, as previously mentioned. In addition to this, the introduced flexibility and dynamicity will open an unprecedented change in the value chain of the mobile communications industry. Further new players, besides vertical industries, will be introduced in the business ecosystem. These new players will specialize on activities currently performed by MNOs, such as connection or asset provisioning or will focus on new emerging ones, like service design. Finally, new dynamic business models based on the exchange/sharing of network service catalogues (e.g., network service brokering) will be feasible due to the multi-provider network service deployment and run-time automation capabilities.

Therefore, the provision of answers and solutions to the research question we are proposing is fundamental for contributing to the benefits next-generation mobile network will introduce in society.

Chapter 5

End-to-End Network Service Orchestration Architecture and Experimental Framework

This chapter presents a high-level overview of the network service orchestration architecture proposed in this thesis work, which has been developed mainly in the context of the H2020 European projects 5G-Crosshaul [8] and 5G-TRANSFORMER [84]. This description is a starting point, giving the reader the general context to understand the different problems tackled to incrementally build the answer to the proposed research question. The demonstration of this architecture and its associated concepts have been developed and assessed mainly over two interconnected heterogeneous experimental infrastructures at CTTC premises. The second part of this chapter provides a description of these two experimental infrastructures.

5.1 End-to-End Network Service Orchestration Architecture

Figure 5.1 presents an overview of the architecture of the end-to-end (E2E) Network Service Orchestration platform proposed and validated within this dissertation as an answer to the presented research question. In particular, the focus of this thesis work concentrates on the part of the architecture inside the red box. These building blocks perform the core orchestration functionalities and the management of the underlying infrastructure needed to perform the dynamic and flexible deployment of network services over heterogeneous transport networks and over multiple administrative domains. The concepts and architectural design behind such building blocks are developed in the rest of this dissertation. The remaining building blocks placed outside of the red box are part of the collaboration with other research groups within the context of the H2020 European project 5G-TRANSFORMER [84]. These building blocks complement the blocks inside the red box to perform advanced functionalities, such as the provision of monitored data and alerts to the core orchestration blocks to decide on scaling operations to maintain service level agreements (SLAs) during network service run-time operation.

The foundations of this architecture are based on the ETSI-NFV MANO architecture defined by the ETSI Industry Specification Group (ISG) and presented in Section 2.2. However, the analysis of this architecture and its associated interfaces revealed some lacks to accomplish the integration between SDN and NFV paradigms, which is one of the main drivers answering the proposed research question. This integration is essential to realise the vision of next generation mobile networks. Next, the functionality of each of the modules depicted in Figure 5.1 is briefly presented.

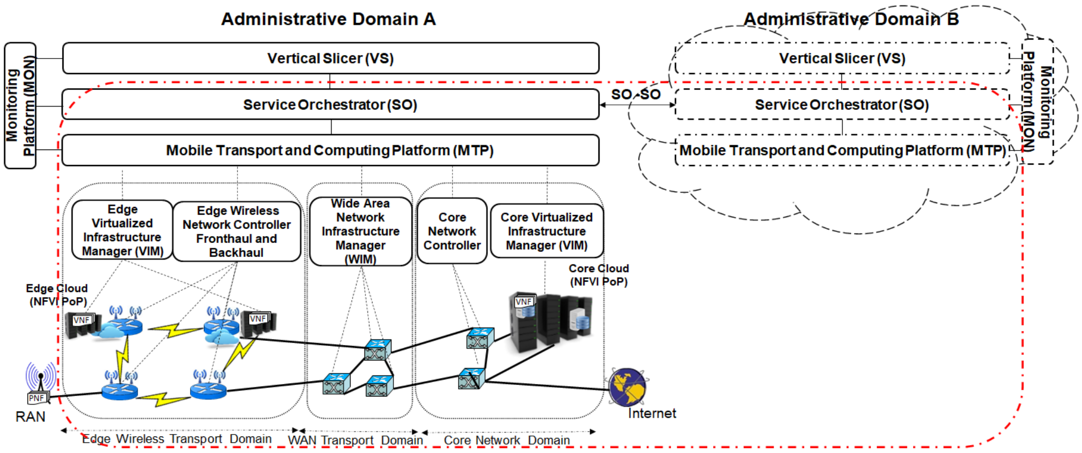


Figure 5.1: E2E Network Service Orchestration Architecture

- ***Vertical Slicer (VS)***: it is part of the Operations Support Systems and Business Support Systems (OSS/BSS), and provides a frontend for vertical industries to the E2E network service orchestration platform. It offers a high-level interface allowing users to easily request vertical services. The VS only focuses on the service and business demands without taking over how services are eventually deployed at the resource level. The VS offers a catalogue of vertical services, which are particularised by the vertical users with their requirements. The internal logic of the VS translates business-oriented service requirements into slice-related requirements to manage the lifecycle of network slices, which are mapped to NFV-Network Services (NFV-NSs).

- ***Service Orchestrator (SO)***: it is in charge of *deploying* the network slice instances by overseeing the E2E orchestration and the lifecycle management of the NFV-NSs based on the available resources advertised by the underlying Mobile Transport and Computing Platform (MTP). Note that NFV-NSs may embrace one or multiple administrative domains (ADs) owned by different service providers. It means that the SO, besides interacting with its local MTP (within a single AD), may interwork with other SOs governing remote domains through the depicted SO-SO interface, thus enabling the E2E deployment of multi-domain network slices in a transparent way to the vertical user through the network service federation (NSF) concept. This problem is tackled in the third part of this dissertation and it is one of its main contributions.

- ***Mobile Transport and Computing Platform (MTP)***: it is the unified controller responsible for executing the resource allocation process: VNF instantiation at the different NFVI-Points of Presence (PoPs) and, connectivity management at the underlying physical transport network interconnecting them and with other possible non-virtualised resources, like the Physical Network Functions (PNF). The MTP has full control of the underlying resources by interacting with different managers: the Virtual Infrastructure Manager (VIM), the Wide Area Network (WAN) Infrastructure Manager (WIM) or other network controllers present in the available infrastructure. For the sake of scalability and to simplify SO operations, the MTP applies abstraction mechanisms when exposing the resource view towards the SO.

- ***Monitoring Platform (MON)***: it collects metrics for the E2E network service orchestration platform so it can perform reactive actions to continuously ensure targeted SLAs. More specifically, the monitoring service at MTP collects data about the local physical and virtual resources; the SO monitoring service collects data about the managed VNFs of deployed NFV-NSs; and the VS monitoring service collects data about network slices and vertical services.
- ***Edge/Core Virtualized Infrastructure Manager (VIM)***: the VIMs are in charge of controlling the compute, storage and networking resources of their underlying NFVI-PoPs. Formerly, cloud deployments were centralized in big data centres (DCs) inside the network. Currently, the requirements of some envisaged networks services, such as low latency, push for the distributed deployment of these processing centres throughout the network, also embracing the edge part of the transport network closer to the end-users.
- ***Edge/Core Network Controller / Wide Area Network Infrastructure Manager (WIM)***: the network controllers and WIMs are in charge of controlling and managing the network resources of the transport infrastructure to interconnect different NFVI-PoPs where VNFs of the same NFV-NS have been deployed.

5.2 Experimental Framework

The demonstration of the architecture and concepts involved in the presented E2E network service orchestration framework has been developed and tested over the cloud computing and transport network capabilities of two interconnected experimental infrastructures deployed at CTTC premises, namely EXTREME Testbed ® [108] and ADRENALINE Testbed ® [109]. The following subsections present a description of the components and platforms provided by these two testbeds to build and assess the proposed orchestration architecture.

5.2.1 EXTREME Testbed

The Experimental Testbed for Research on Mobile nEtworks (EXTREME Testbed ®) [108] is an experimentation platform designed and developed by the CTTC Mobile Networks Department for testing wireless access and backhaul architectures featuring generic purpose server pools, cellular and other wireless equipment, ns-3 emulation/simulation and tools for fast prototyping and evaluation. The core of the EXTREME Testbed ® lies on two central servers, which are the interfaces between the experimenters and the SDN/NFV experimentation services. A series of reconfigurable multi-purpose network elements (9 Intel-based server with 40vCPUS and 6Gigabit Ethernet cards and more than 40 Intel-based servers with 4 Gigabit Ethernet cards) can be customized and used as network elements for experimentation purposes.

EXTREME also offers an edge wireless backhaul plus DC infrastructure for SDN/NFV experimentation. This is the part of the EXTREME testbed ® that will be used to develop and host the presented E2E network service orchestration architecture. It consists of eighteen nodes deployed within a skylight over three floors at CTTC premises. Each node is based on an Intel Core i7 processor (12-Cores at 3.3GHz x86 CPU) with 32GB and 1TB of hard disk, running Ubuntu 16.04 LTS distribution.

Every node is equipped with three Compex WLE900VX IEEE 802.11ac [110] Wi-Fi cards running the *ath10k* driver and the firmware and kernel provided by Candela Technologies [111], which is necessary to establish ad-hoc connections with adjacent nodes, hence forming the wireless data forwarding plane. The testbed facility also counts with up to eight mmWave links (TensorCom TC60G-USB3.0 EVK devices) to provide the wireless domain with IEEE 802.11ad [112] mmWave transport capabilities. All the nodes count with at least two Gigabit Ethernet ports and two of them are interconnected through 1 Gbps Ethernet links to two packet switches (OpenFlow-based) of the ADRENALINE Testbed ®. These links between EXTREME and ADRENALINE testbeds allow the communication of distributed cloud infrastructures through a transport network featuring multiple transmission technologies (wireless/optical).

The characteristics of the deployment site and the capabilities of the equipment allow the versatile use of the available nodes. Some of them are devoted to build the wireless transport network. They can be arranged forming multiple topologies. These nodes forming part of the transport network are provided with an instance of an OpenFlow software switch (Open vSwitch (OVS [21])) and they are controlled by our developed wireless SDN controller, called WiseHAUL [41]. Additionally, arbitrary transport network topologies, which can be easily connected to external equipment, can be emulated with the help of the GNS3 [113] emulator. Other nodes are devoted to host the different building blocks of the E2E network service orchestration framework. These nodes combine our software developments with open source MANO platforms, like Open Source MANO (OSM) [73] or Cloudify [80], or with open source cloud infrastructure software to deploy virtual machines (VMs) implementing the desired VNFs. For this last purpose, this framework relies on Openstack [22], a leading open source cloud orchestration platform, which, additionally presents a great degree of compatibility with the mentioned open source MANO platforms. More specifically, the DevStack [114] project is considered, which is a set of scripts and utilities to quickly deploy an OpenStack cloud from git source trees allowing easier prototyping.

Figure 5.2: EXTREME Testbed ®: (left) detail of the edge wireless NFV/SDN facility, (right) core facility

5.2.2 ADRENALINE Testbed

The ADRENALINE Testbed ® [109] is an SDN/NFV packet/optical transport network and edge/core cloud platform for E2E 5G and IoT services designed and developed by the CTTC Optical Networks and Systems Department.

The optical network is an all-optical Dense Wavelength Division Multiplexing (DWDM) mesh network with two colourless Reconfigurable Optical Add/Drop Multiplexer (ROADMs) and two Optical Cross-Connect (OXC) nodes interconnected through five bidirectional links using up to 610 km of G.652 and G.655 optical fiber. The optical network is controlled by a GMPLS/Active Stateful (AS)-Path Computation Element (PCE). ADRENALINE counts with a mesh network of packet switches deployed on commercial of the shelf (COTS) equipment based on OF protocol and using OVS. Some of them are connected to the optical transport network by means of a 10Gb/s XFP tunable transponder and others are part of a mesh network connected to the edge/core cloud platform. Both segments, optical and packet-switched networks are integrated in a wider SDN control architecture to provide E2E connectivity services, as covered in Chapter 6.

The core cloud infrastructure is composed of a core-DC with high-performance computing (HPC) servers and an intra-DC packet network connected to the optical network. The edge cloud infrastructure is composed of four micro-DCs in the edge nodes and two small-DCs in the central offices. These cloud infrastructures leverage VMs based on Openstack/Devstack open source software and container-based technologies oriented to offer the appropriate compute resources depending on the network locations. In particular, this work makes use of the resources available in the core cloud infrastructure, combining with those offered by the EXTREME Testbed ® to arrange the required setups and configurations to assess the proposed architecture and procedures.

Figure 5.3: ADRENALINE Testbed ®: detail of the optical transport network

Part II

End-to-End Orchestration of Network Services in Single Administrative Domains

Chapter 6

Control and management of heterogeneous transport networks

The contributions of this chapter have been published in [115] and [116]. This chapter starts the bottom-up building process of the E2E network service orchestration architecture presented in the previous chapter. This process starts with the analysis of the application of the SDN paradigm for the automated control and management of forwarding devices in heterogeneous transport networks featuring multiple transmission and different control plane technologies. An interesting point of the presented analysis lies in the integration of wireless forwarding elements as part of the edge transport network. Wireless transmission presents an interesting cost/performance trade-off to achieve the envisaged network densification and it will need to be smoothly integrated into the transport network, mostly dominated by optical transmission technology.

In Section 6.1, we present the need of the multi-domain network orchestration, considering the use of a hierarchical SDN approach as a way to handle the heterogeneity of transmission and control technologies within the transport network to manage the forwarding behaviour of the different elements in a single administrative domain. Section 6.2 presents the experimental findings, from a network service perspective, when applying a hierarchical SDN control scheme for the configuration of a transport network featuring multiple transmission technologies (wireless/optical) in the case of deploying and recovering a mobile service.

Next, this chapter deals with the other aspect tackled by the SDN paradigm, the management of network devices. We design, develop and deploy a wireless management agent, which is then used through the SDN controller to characterize the performance of an experimental mmWave network.

6.1 Multi-Domain SDN Orchestration

The appearance of the SDN paradigm has boosted the evolution of network management and control systems in recent years. As mentioned in chapter 2, SDN proposes the decoupling of the control plane logic from the data plane forwarding infrastructure by removing the "intelligence" of the forwarding elements and placing it in a logically centralized controller.

However, it is commonly accepted within the transport SDN community that a single, integrated controller for a large or complex network may present several technical issues, or may not be a practical solution. Among these technical issues, the following three reasons can be highlighted. First, we have the network size aspect in terms of controllable elements. This has a direct impact on the controller requirements (e.g., active and persistent TCP connections to control forwarding elements, memory requirements to store data structures representing network abstracted graphs or CPU load to implement control logic or processing message exchange). The second reason is the network complexity, in terms of having a network combining multiple data-plane technologies, such as a packet-switched layer for Layer2/Layer3 transport over a circuit-switched optical layer. The intrinsic parameters and attributes impact on functionalities and protocols to be implement at the controller level. Third, we have security and robustness aspects in terms of exposure of all network equipment towards the same control instance, which could easily be compromised.

Currently, the design of an SDN-based control plane considers the deployment of multiple controllers arranged under different schemes, also implying the definition of inter-controller protocols, to address such shortcomings. Normally, each of these controllers are in charge of a different set of network elements sharing some attributes (e.g., vendor, transmission technology), which defines a control domain. Thus, the deployment of a realistic SDN-based control plane will require the definition of multi-domain orchestration procedures to coordinate and automate the establishment and release of independent network connections through the different control domains. Note that nothing precludes the deployment of multiple SDN controllers for robustness/redundancy purposes.

Peer and hierarchical models are two basic approaches to organize the interconnection of multiple controllers managing different control domains to tackle the multi-domain orchestration problem. These approaches are depicted in Figure 6.1. In the peer model, the set of controllers are interconnected in an arbitrary mesh topology. The controllers use East/West interfaces to synchronize state, provide network topology abstraction, path computation and segment provisioning. In the hierarchical approach, controllers are organised in, normally, a tree topology with a given root being the top-most controller. This top controller, usually known as parent controller, acts as *"controller of controllers"*, handling the automation and high-level functions. Low-level controllers, usually known as child controllers, cover low-level, detailed functions and operations. Parent and child controllers interact through appropriate protocols via North/South interfaces. The application of generic interfaces enables a recursive use of this model, hence enabling multiple hierarchy levels.

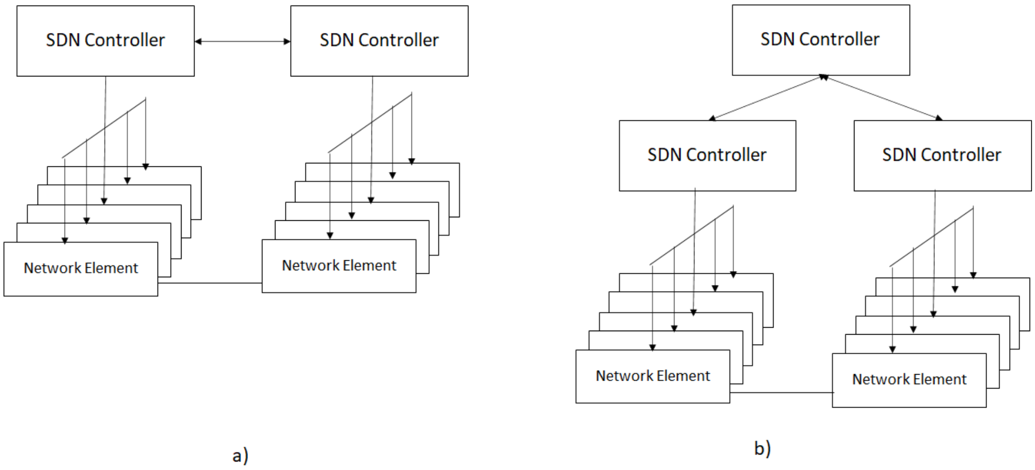


Figure 6.1: SDN interconnection models a) Peer, b) Hierarchical

6.2 SDN Orchestration of network resources in heterogeneous transport networks

The envisaged constraints and conditions involved in the deployment of next-generation mobile networks will require transport networks to mix equipment from multiple vendors featuring multiple technologies at the data-plane level (e.g., packet-switched over wireless links, circuit-switched optical networks) and at the control plane (e.g., OpenFlow, GMPLS).

In this case, within a single administrative domain, the hierarchical model may be a straightforward solution to handle this heterogeneity rather than the peering interconnection model, which would require a more complex framework to interconnect peering control domains, as explained in [117]. Under such deployment, the top-level orchestrator (parent controller) coordinates the topology management and the E2E service provisioning to ensure the inter-working between the different underlying domains. The hierarchy can be done by means of *"plugins"* covering the specificities of the technologies offered by the vendor NBIs.

The following sections analyse the deployment of the hierarchical interconnection model to perform the automated multi-domain SDN orchestration of network resources in a multi-technology transport network within a single administrative domain [115]. This analysis is presented from a network service perspective in terms of transport network resources, where the proposed hierarchical SDN-based control plane is assessed to automatically set up and perform the flexible recovery of the network connections required by a deployed virtual mobile network service effectively handling fronthaul and backhaul traffic simultaneously.

6.2.1 Experimental Hierarchical SDN Orchestration Architecture

Figure 6.2 presents the architecture under analysis. This architecture has been experimentally deployed in collaboration with other research groups within the context of the EU-H2020 5G-Crosshaul [8]. The system is composed of building blocks coming from three different geographically distributed sites throughout Europe: Castelldefels (Spain), Heidelberg (Germany) and London (UK), which are connected by dedicated virtual private network (VPN) tunnels. Next, we detail the elements and characteristics of the different planes present in the deployed SDN architecture following a bottom-up approach.

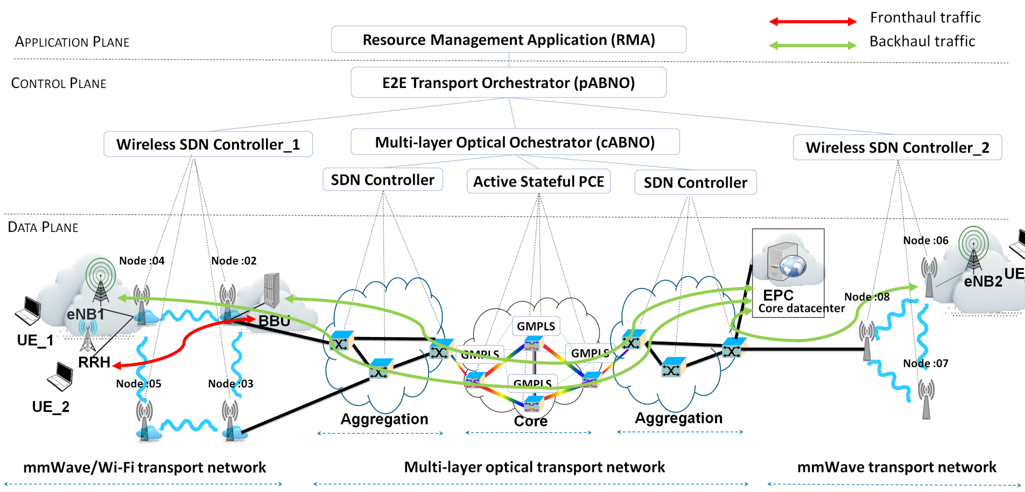


Figure 6.2: Hierarchical SDN orchestration of a multi-technology transport network

6.2.1.1 Data Plane

The data plane combines several domains using different transport technologies. The mmWave/Wi-Fi transport network (part of the EXTREME experimental framework [108]) and the mmWave transport network (in London) represent the wireless edge packet-switched domains of the transport network. All the forwarding elements (FEs) in these domains are equipped with wireless Gigabit interfaces based on mmWave (IEEE 802.11ad) links. Moreover, the equipment of the mmWave/Wi-Fi transport network also counts with Wi-Fi (IEEE 802.11ac) interfaces, hence supporting multiple wireless technologies. The multi-layer optical transport network, built with the ADRENALINE [109] testbed, represents the core transport domain, which features two layers separated into different sub-domains. First, there are two aggregation packet-switched Ethernet networks with tunable optical interfaces at edge nodes. Second, there is an all-optical dense wavelength division multiplex (DWDM) mesh network counting with two colourless Reconfigurable Optical Add/Drop Multiplexer (ROADM) and two Optical Cross-Connect (OXC) nodes, interconnected through five bidirectional optical links with a total of 610Km of optical fibre. Each optical node has multiple DWDM transceivers up to 2.5Gbps and one at 12.5Gbps with fully tunable laser sources.

6.2.1.2 Control Plane

The depicted hierarchical SDN-based orchestration entity has been deployed in three layers for which a parent-child relationship is established between contiguous layers. The hierarchical approach not only enables orchestration in a multi-technology, multi-vendor environment typical of envisioned 5G network deployments, but also improves scalability, modularity, and security, as explained in [15].

At the top of the hierarchy, the parent controller orchestrates the different child controllers, which handle the specificities of the different equipment and link technologies at the underlying forwarding elements. In this setup, all the control plane entities depicted in Figure 6.2 reside in the Castelldefels site (our experimental facilities) except the Wireless SDN Controller_2 of the mmWave transport network, which is placed in London.

In the deployed system, the parent controller is the E2E Transport Orchestrator, a multi-domain orchestrator that is based on the IETF Application-Based Network operations (ABNO) architecture [47]. It is referred to as parent ABNO (pABNO). The pABNO is in charge of composing the multi-domain topology and generating the corresponding connectivity requests from/towards the underlying child controllers. A complete and detailed description of this multi-domain orchestrator architecture can be found in [118].

The multi-domain orchestrator architecture supports hierarchical deployments with arbitrary depth allowing recursive and scalable deployments thanks to the use of a unified Southbound (SBI) and Northbound interface (NBI)[1]. In this case, the interface used in this system is the Control Orchestration Protocol (COP) [2] [53], which offers topology, connectivity and path computation services. The main reason of defining such protocol is to handle the heterogeneity of NBIs of SDN controllers (e.g., Ryu, OpenDayLight, ONOS). Thanks to COP, a single protocol providing a common NBI API can be used to orchestrate different SDN controllers.

COP defines an information model using the YANG modelling language [64], and it employs RESTCONF [63] as underlying protocol to exchange JSON objects between the different control entities. In addition to this, as a REST API does not allow notifications, websockets have been introduced in the system (as described in RESTCONF) in order to manage notifications about nodes, link status and updates on the established paths. The COP protocol, initially developed in the context of the European and Japan STRAUSS [54] project, has been further extended to account for wireless technologies. As mentioned in Chapter 3, COP precedes similar efforts later carried out at SDOs, such as ONF T-API (Transport API) [56]. While COP is a research-oriented multi-layer approach using YANG/RESTCONF, ONF T-API is focused on standardization efforts for REST NBI-based orchestration of SDN controllers. However, they are aligned in the objective of bringing the full benefits of programmable SDN transport networks to high-level application.

Figure 6.2 presents an example of recursion in the hierarchical orchestration architecture, where other multi-domain ABNO-based controller acts as Multi-layer optical orchestrator, referred to as child ABNO (cABNO). A recursive hierarchical deployment can help accommodate multiple underlying transport technologies, SDN controller types or heterogeneous network segments. In the deployed hierarchical SDN architecture, the introduction of a new hierarchy level serves to add multiple transport technology domains, e.g., the addition of the wireless transport domains. In turn, the cABNO also has child controllers. In fact, the cABNO provides E2E multi-layer (packet and optical) and multi-domain topology and network resource provisioning capabilities across its different child domains, as the pABNO does for the whole infrastructure. Two child SDN controllers serve the packet-switched domains represented at both sides of the all-optical network and a GMPLS/Active Stateful Path Computation Element (AS-PCE) manages the optical circuits of the core segment. Within the ABNO architecture [47], the cABNO also includes the Abstraction Manager (AM). The AM is able to provide several types of abstraction levels, such as node or link abstraction of the underlying network resources and correlates the requests based on the abstracted topology to the information of the underlying topology. Such abstractions serve to hide the unnecessary particularities of this domain to the pABNO, hence easing its job towards more modular and scalable deployments, as depicted in Figure 6.3. Figure 6.3 presents the pABNO Graphical User Interface (GUI), showing the topological view of the pABNO based on the information provided by the different child controllers. In this case, the cABNO abstracts all the components of the multi-layer network under its control as four nodes. The Wireless SDN controllers do not perform any abstraction, and the wireless domains are represented as depicted in Figure 6.2.

The Wireless SDN controllers are the child controllers for their respective transport domains. As such, they control the edge wireless transport segments to handle the topology and connectivity requests coming from the pABNO. In addition, in this setup, the Wireless SDN Controller_1 also counts with path computation capabilities for path recovery purposes. The Wireless SDN Controller_1 is based on

[1]Note that hierarchical deployments imply no direct communication between neighbouring controllers (that is, those controllers that are responsible for adjacent network segments or those which have lower or equal hierarchy level)

[2]https://github.com/5G-Crosshaul/COP

the WiseHAUL controller [41], which is part of our previous work. At its SBI towards the networking switching elements, they use OpenFlow (OF) protocol for the configuration of the forwarding behaviour.

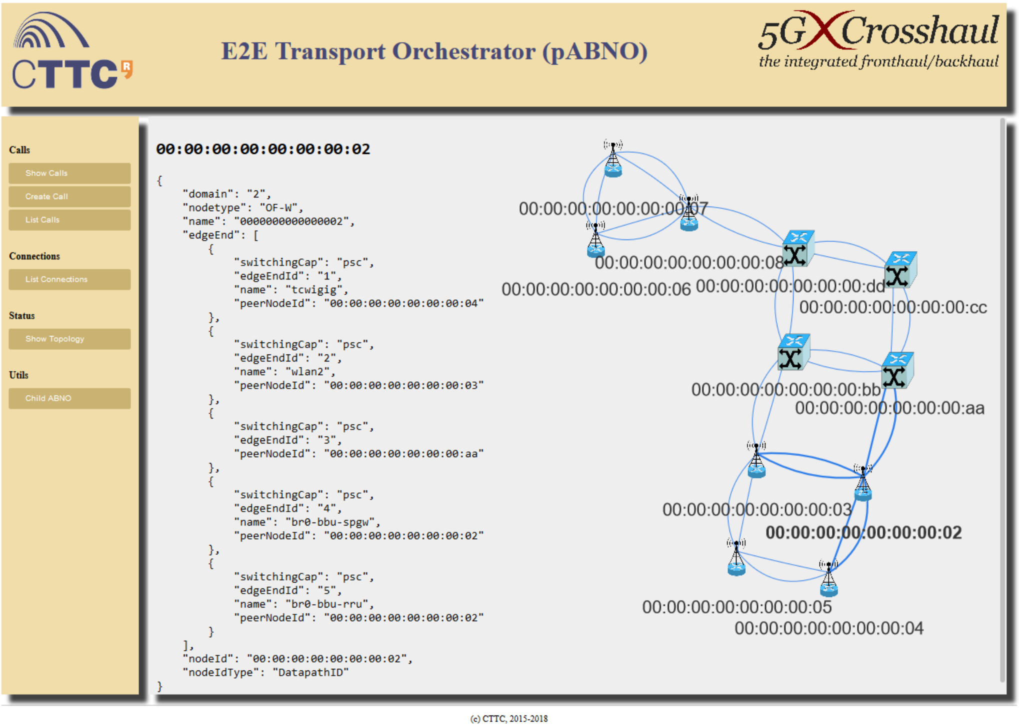


Figure 6.3: E2E Transport Orchestrator (pABNO) GUI presenting the topological vision of the parent controller based on the information provided by the different child controllers. Nodes in the figure correspond to those labelled in Figure 6.2, except for the optical domain, abstracted by the cABNO

6.2.1.3 Application plane

The Resource Management Application (RMA) is the application on top of the depicted system in charge of managing E2E transport network resources. Physically placed in Heidelberg (Germany), it makes use of the services offered by the hierarchical SDN control infrastructure by means of the previously described COP protocol via a VPN tunnel. When connecting with the pABNO, the RMA retrieves the current topology from the pABNO, which is updated dynamically thanks to the websocket notification system. Upon the reception of a service request, the RMA computes optimal paths for such request, where source, destination and traffic profiles (rate, delay) are defined. Then, the RMA generates the appropriate JSON objects communicating to the pABNO the computed paths satisfying the requirements of the different traffic profiles received in the request. When the service is no longer required, the RMA commands the pABNO to delete all the forwarding rules at the FEs associated to this service. Figure 6.4 summarises this and previous sections by presenting the workflow followed to provision E2E connectivity services in the hierarchical SDN architecture under analysis depicted in Figure 6.2.

6.2.2 Network resource provisioning performance evaluation

This section provides a characterization of the service setup time needed by the described architecture to provision the network resources required to interconnect the different elements of the deployed

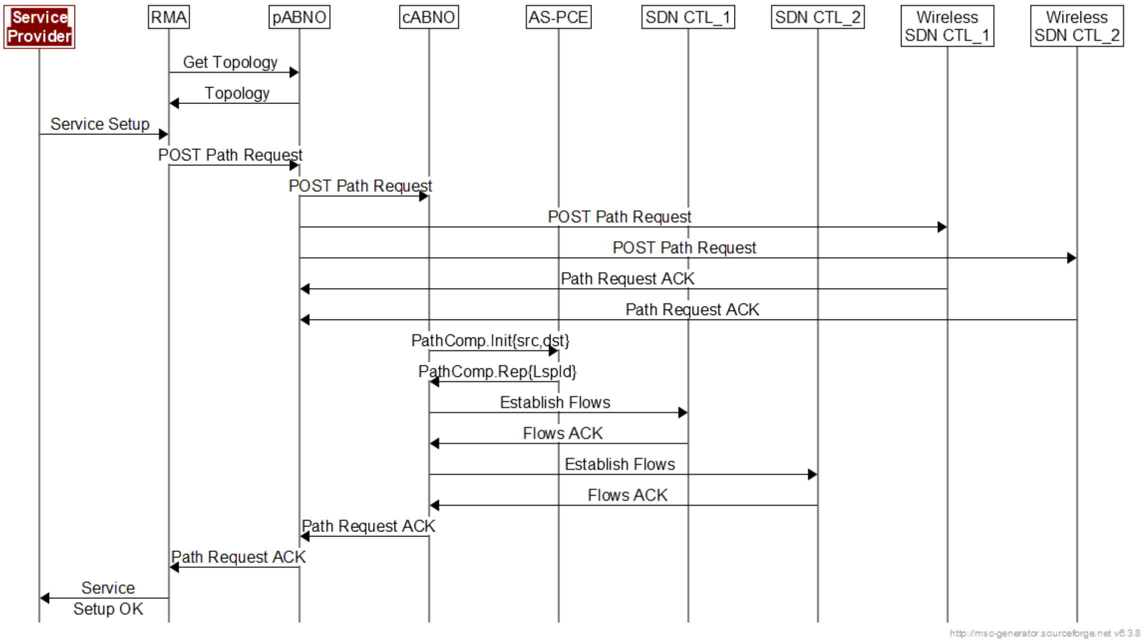


Figure 6.4: Message exchange workflow for provisioning of E2E connectivity services

mobile network layer. This characterization extends our previous work in [52] by providing an analysis from a service perspective level to align the provisioning process with the requirements of a vertical service and including a second wireless transport domain (both control and data plane) deployed in another physical site, showing the scalability and flexibility of the hierarchical orchestration approach. We define the service setup time as the interval between the reception of the service request from the RMA and the subsequent confirmation from the pABNO that all the requested paths have been installed at the different transport domains.

The deployed LTE mobile network service under study is based on the OpenEPC platform [119], and it is composed of several virtual machines (VMs) statically deployed as endpoints of the transport network. Two enodeBs and a remote radio head (RRH)-baseband unit (BBU) pair featuring a fronthaul packet data convergence protocol/radio link control (PDCP/RLC) split are placed at the edge wireless transport domains, as depicted in Figure 6.2. All of them connect to the Evolved Packet Core (EPC) entities placed at the Multi-layer optical transport network, namely mobility management entity (MME) and serving/PDN gateway (SPGW). Hence, the RMA will determine the appropriate paths among the different mobile network entities for the fronthaul and backhaul traffic profiles, represented by the different arrows in Figure 6.2. In order to establish the required network resources for the LTE mobile network service, eight paths (a total of sixteen flows to provide bidirectional paths) are required: from RRU to BBU, from BBU to MME, from BBU to SPGW, from eNodeB1 to MME, from eNodeB1 to SPGW, from eNodeB2 to MME, from eNodeB2 to SPGW, and from SPGW to MME.

Figure 6.5 presents the histogram and the cumulative distribution function (CDF) (seen from the RMA) of one hundred repetitions of the time to setup the described LTE mobile network service. As shown in Figure 6.5, the time it takes the RMA to calculate and validate the installation of the whole set of path requests to interconnect the different deployed mobile network entities is lower than 11 seconds in 90% of the samples, with an average value of 10.467 seconds and a minimum value of 9.343 seconds. Hence, the achieved results verify that the presented hierarchical control contributes to the 5G-PPP KPI target of enabling the introduction/provisioning of new network services on the order of minutes/hours. The application of an SDN-based orchestration provides a complete automation of the provisioning process, even in the presence of heterogeneous technologies, improving current

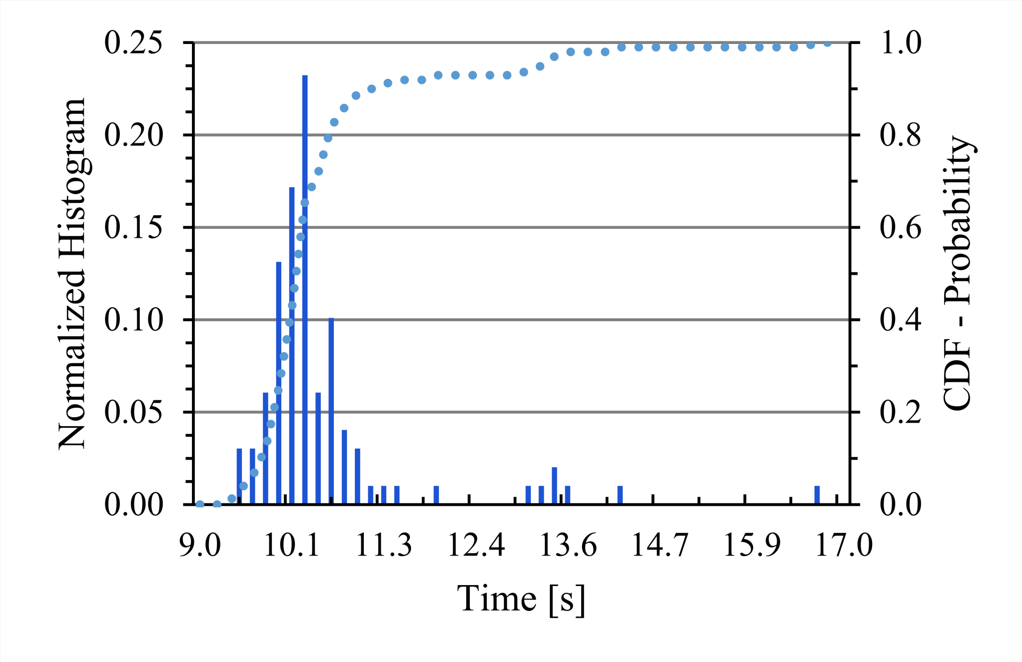


Figure 6.5: Normalized histogram and CDF of the service setup time seen by the RMA

management schemes employed in production networks based on the manual and sequential execution of operations applied during scheduled network operation windows.

Table 6.1 expands the previous figure by showing the experienced setup time at each entity of the hierarchical control plane. It shows how these values increase at the different stages of the hierarchical control plane due to the processing operation at each stage, the message passing time between control entities in the hierarchy (RMA-pABNO and pABNO-child controllers) and the delay introduced by the VPN connections (RMA-pABNO and pABNO-Wireless SDN Controller 2).

Control Entity	Average	Min	25-perc	75-perc	Max
RMA	10.467	9.343	9.962	10.544	16.658
pABNO	10.112	8.839	9.506	10.150	16.443
cABNO	9.764	8.507	9.147	9.808	16.160
Wireless SDN Controller 1	7.855	6.721	7.335	7.879	15.794
Wireless SDN Controller 2	7.659	5.649	7.332	7.814	10.889

Table 6.1: Service setup time seen from the different control plane elements [seconds]

From a deeper analysis of the measured experimental time at the pABNO level to set up each call, the most time-consuming operations in terms of setup time are those requests involving setting up a bidirectional multi-domain path traversing the multi-layer optical network. This is due to the amount of actions that have to be done in the set of network elements hidden under the abstraction provided by the cABNO controller. In particular, the first of such path requests is always the one which requires more time, around 2.9 seconds. As described in [52], this obeys to the need to tune laser interfaces at the nodes labelled as GMPLS in Figure 6.2 to set up the initial light path and the Ethernet service on top. After this first path is established, all the remaining path requests traversing the multi-layer optical network reuse the same light path and are established much faster, around 1 second.

Finally, a comment on the presented value for the Wireless SDN Controller_1. When the pABNO receives the different calls from the RMA, it decomposes them in different sub-calls for each network domain, which are then sent to the underlying child controllers. Then, all the operations for this call at the different domains are carried out and a validation is sent to the pABNO, which then sends and ACK to the RMA. As the most time-consuming call setup operations are done at the multi-layer optical domain and almost all call requests for the mmWave/WiFi transport domain traverse the multi-layer optical network, the child controller of this domain has to wait until the pABNO receives the acknowledge from the optical domain to process the following call request. Hence, the overall service setup time experienced at the mmWave/WiFi domain is mostly due to the pace at which calls are sent to be processed rather than the time itself to setup the different FEs in the mmWave/WiFi domain.

In relation to this, it can be added the fact that the calls are not ordered in time with respect to the involved domains. For instance, not all the calls directed to the same set of transport domains, (i.e., mmWave/WiFi domain and multi-layer optical domain with its calls from enodeB1 and BBU towards the EPC entities MME and SPGW) are processed consecutively. They are mixed with other calls involving other domains (i.e., mmWave domain and multi-layer optical domain with its calls from enodeB2 to MME and SPGW), increasing the time to consider that the service is setup. This fact modifies the experienced setup time distribution at the different stages/domains, as we can see if we analyse the times experienced by the mmWave transport domain. In this domain, only two calls have to be established (from enodeB2 to MME and from enodeB2 to SPGW). However, in the time between those calls are received at the Wireless SDN Controller_2 child controller, the RMA has sent other calls to the Wireless SDN Controller_1 to establish the path between service entities in the mmWave/WiFi domain and the optical transport domain.

6.2.3 Network resource service recovery performance evaluation

This section provides a characterization of the time needed by the described architecture to perform the service recovery of the deployed network service in the event of a link failure under two strategies: a) local recovery procedure performed by a child SDN controller and b) centralized recovery procedure performed at the RMA application.

In both cases, the link failure happens in the direct link connecting nodes labelled as *Node:04-Node:02* in Figure 6.2. In order to re-establish the mobile network service, three paths (six unidirectional flows) have to be re-established: from enodeB1 to MME, from enodeB1 to SPGW and from RRU to BBU.

Different control plane designs – and especially if centralized – need to account for heterogeneous recovery requirements, and it is well known that a pure centralized solution may not be suitable in all cases (e.g., in terms of recovery delay). Hybrid approaches (that is, that favour local decisions where possible and escalating to a higher level based on predefined policies and rules or when recovery is not possible) are commonly conceived to address the shortcomings. In any case, there are important trade-offs to be considered and an operator policy should decide when to apply local or centralized recovery. In addition to this, the type of failure can impose a specific type of recovery: an inter-domain failure implies corrective actions in multiple domains, which cannot be handled locally, or multi-layer / multi-technology connection constraints may limit the applicability of local recovery (e.g., coloured optical interfaces, asymmetrical ROADM nodes). Macroscopically, a local recovery may be more effective in terms of minimizing traffic disruption time, but the lack of topology visibility associated to multi-domain networks and the fact that local recovery is commonly constrained to maintain the original (local) endpoints preclude end-to-end optimality in terms of resource usage or the fulfilment of traffic requirements, such as delay. On the other hand, centralized recoveries may be potentially a good option, but at the expenses of higher restoration times, control bottlenecks and scalability issues (local recovery is to some extent distributed) and, finally, topology abstraction of hierarchical deployments may also limit its optimality. Herein, the aim of assessing different recovery strategies is to illustrate the architectural flexibility of the presented hierarchical system to allow both recovery approaches.

6.2.3.1 Local service recovery

Wireless SDN Controller_1 starts the recovery process upon the detection of an OF PORT_STATUS message. In parallel, it sends an update topology notification up the control hierarchy (which will eventually arrive to the application plane, RMA) to inform about the topology update and that the recovery process will be done locally. Since the child SDN controller maintains a database with all the installed flows, and based on the current *"domain"* network state, it checks which flows have been affected and proceeds to remove affected paths following a "break-before-make" strategy and to calculate an alternative path, if possible, to install it in the underlying FEs. In this case, the alternative path is Node:04-Node:05-Node:03-Node:02 (see Figure 6.2) to maintain the original endpoints and circumscribe the recovery within a single domain. If the path is not possible, the child SDN controller sends a notification up the hierarchy to inform that the paths are definitively removed from the system. Once Wireless SDN Controller_1 has received confirmation that the alternative flows have been installed in the corresponding XFEs, it notifies the RMA through the pABNO of the alternative paths set up in the network. Then, the RMA can update its own database to have the current network vision to be able to provide the appropriate paths for subsequent service requests.

In Figure 6.6, we can see the histogram and the CDF of performing one hundred operations of local recovery of the deployed LTE mobile network service. The measured recovery time values, with an average of 299ms, are similar to those reported in [66] when using OF Fast Failover (FF) group tables and using Bidirectional Forwarding Detection (BFD) protocol for a similar environment and even when repairing several flows. By using the BFD plus OF-FF approach, a link down event can be detected and re-routing decisions are triggered locally without the need to involve any SDN controller (if backup paths have been configured in advance). However, in this case, the BFD plus OF-FF would not be a good solution since not any control plane entity is involved in the recovery process, hence, RMA or other control-plane entity would not have a consistent or updated view of the network in terms of topology and installed paths. In addition to this, the OF-FF failover solution can present, potentially, additional drawbacks. A short BFD monitoring interval could lead to increased traffic and processing overhead or triggering false link transitions, which could affect the link performance for data transmission [120]. In order to provide backup paths, a denser transport network deployment implies the installation of a bigger number of forwarding rules in the nodes. Finally, the requirement of installing additional rules to perform crankback forwarding to send packets back towards a transport node with alternative active path towards the destination impacts on the experienced latency [121].

6.2.3.2 Centralized service recovery

In this case, after the link failure event, Wireless SDN Controller_1 notifies up the hierarchy the change in the topology and that no local recovery is going to be performed. The RMA receives this notification and proceeds to detect the affected flows to send back to the pABNO the corrective actions. First, it deletes those previous requests with affected paths. Then, it computes the new paths for the affected ones and re-installs them if they continue fulfilling the specified requirements at the service request, otherwise the path is not valid and the service is not recovered. Figure 6.6 shows the histogram and the CDF of performing one hundred operations of centralized recovery at the RMA. The measured times present more variance, increasing to the order of seconds (average value of 6.652 seconds) compared to the local recovery case evaluated previously. This is due to several factors. First, climbing up and

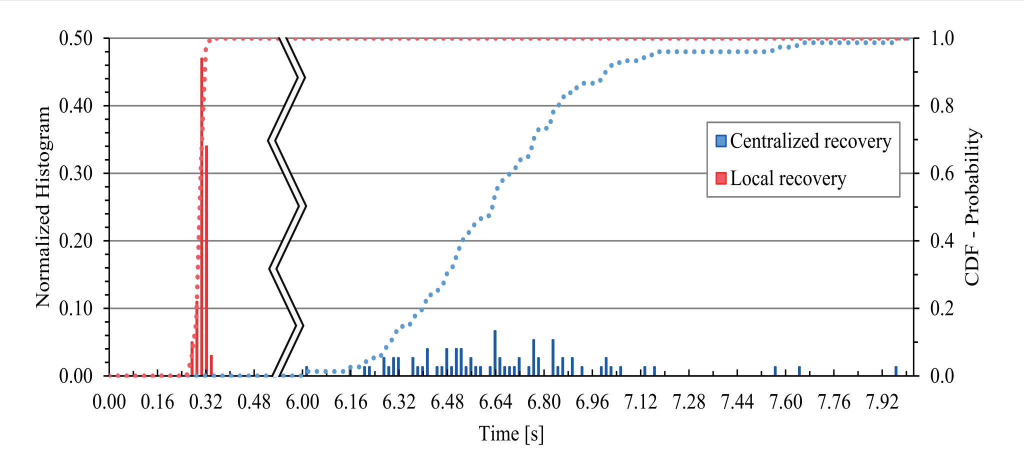


Figure 6.6: Normalized Histogram and CDF of local and centralized service recovery procedures

down in the hierarchy introduces delay, which is increased also due to the VPN connections. Second, the processing time at the RMA to update the topology and check the affected rules will be higher than the one in lower control level entities. Third, and most relevant, the corrective actions the RMA sends. Due to its global view, these corrective actions involve deleting and creating actions in multiple domains; namely, the mmWave/Wi-Fi and the multi-layer optical transport domain. In the case of local recovery, corrective actions only involve the mmWave/Wi-Fi transport network. As pointed out previously, actions performed in the multi-layer optical transport domain (both for creating and deleting) may potentially require more time due to the bigger amount of involved network elements both at control and data plane. In brief, with the above measurements, we validate the flexibility of our control plane infrastructure to have different and configurable levels of reaction in front of network events.

6.3 Management of wireless transport network devices using SDN

Besides controlling the forwarding behaviour of network elements, the SDN paradigm also considers the management of such elements, that is, the configuration of network interface transmission parameters. The definition and standardization of open interfaces at the SBI and open information models allow the seamless connection of multi-vendor network (wireless) devices to a single platform (open source SDN controllers), so the myriad of tools for network management can come together. This increases automation and reduces the duration and cost of integration processes, network-wide configuration errors derived from current manual human-based distributed management operations and, in general, operational expenditures.

Enabling the possibility of modifying the configuration of wireless network interfaces (e.g., power state, modulation and coding scheme (MCS), carrier frequency, channel bandwidth) of transport elements may allow advanced applications running at the application layer, such as energy management, to improve the performance of the overall network in terms of power consumption and throughput among others.

Following these principles, the next subsection presents the design of a wireless interface agent (WIA) running in the wireless devices under the control of the Wireless Controller_1 depicted in Figure 6.2. This WIA is used by the Wireless Controller_1 (the WiseHAUL SDN controller [41]) to modify the configuration of the mmWave wireless, which will allow us to characterize the performance of the mmWave part of the available experimental mmWave/Wi-Fi transport network.

6.3.1 Wireless Interface Agent Design

For the design of the WIA, we consider the inclusion of a new management agent in the wireless transport node decoupled from the available forwarding agent, based on OF protocol, provided by the instance of the software switch running at the wireless transport node. Following the idea of the previously mentioned COP protocol for modeling forwarding behaviour, the WIA is a management agent based on a RESTCONF server communicating with the client instantiated at the SDN controller over HTTP requests, where a uniform resource identifier (URI) identifies every exchanged resource. The WIA is able to integrate interfaces of available heterogeneous wireless technologies, namely, mmWave (IEEE 802.11ad) and Wi-Fi (IEEE 802.11ac), by means of an appropriate information model defined using the YANG modelling language. Then, the WIA is in charge of mapping the request of the Wireless SDN Controller to the specific configuration command of each wireless interface of the transport node. In addition to configuration parameters, the WIA considers asynchronous notifications of network events following a publish/subscribe pattern, which could be used to trigger actions in the SDN controller based on monitored events at network nodes. With this notification system, network events derived from monitored events happening at the wireless transport node, such as channel interference, can be reported to the SDN controller, which in turn can inform an upper network management application to trigger network configuration actions, such as a frequency switch in a network interface.

The decoupling between forwarding and management operations followed by the WIA approach (YANG-RESTCONF) provides flexibility and scalability capabilities to the scheme if we compare to the integration of both types of operations by custom extensions of OpenFlow (OF) protocol, since the addition of new features is handled easily by extending the information model and defining its associated URIs. An extension of the OF protocol, initially proposed in the 1st PoC of the Wireless Transport Group of ONF and other research works [66], requires the definition of new structures and parameters within the OF protocol structures/messages and the corresponding inclusion of code into the internals of the used open source software switch and the SDN controller to appropriately parse the bytes of new defined OF messages, hence limiting the scope and scalability of the solution. As mentioned in Chapter 3, the Wireless Transport Group of ONF, after the first PoC, decided to decouple forwarding and management operations, adopting a solution based on a YANG-NETCONF approach, hence having the same principles of the proposed WIA. However, our experimental approach decided to use RESTCONF instead of NETCONF to simplify the interface, easing the prototyping, since the set of NETCONF operations can be implemented with the reduced set of HTTP methods.

Table 6.2: Current parameters defined in the YANG model for wireless interfaces

Parameter	Config	Monitor	Inventory
Supported Channels			X
Channel central frequency	X	X	
Supported Channel Bandwidth			X
Channel Bandwidth	X	X	
Supported MCS			X
Current MCS	X	X	
Transmission Power	X	X	
Max Tx power per channel			X
Link Status	X	X	

Table 6.2 summarizes the set of currently supported parameters by the WIA, based on capabilities of available hardware using Wi-Fi (Compex WLE900VX IEEE 802.11ac) and mmWave (TensorCom TC60G-USB3.0 EVK IEEE 802.11ad) technologies. The available mmWave devices count with reduced configuration capabilities with respect to the considered Wi-Fi device because it is a pre-commercial

device. In particular, current capabilities of the mmWave device drivers are reduced to read/configure its MCS and link status, and read its central frequency.

6.3.2 Experimental characterization of a mmWave network through the Wireless Interface Agent

This subsection presents the characterization of the mmWave transport network under the control of the Wireless SDN Controller_1 depicted in Figure 6.2. This transport network is a ring of four nodes, as depicted in Figure 6.7. This characterization considers two case studies, where the presented WIA, triggered from the GUI of the WiseHAUL SDN controller, is used to manage the configuration of the available wireless interfaces. The first case study provides results showing the experienced downtime after a change in the status of the mmWave link, while the second one studies the experienced TCP throughput in single and multi-hop mmWave setups when varying the MCS and the maximum transmission unit (MTU). The reported values represented in the following figures show the statistical distribution of the corresponding experiment from the 25th to the 75th percentiles, and the whiskers from the 5th to the 95th percentiles. The marker represents the average value.

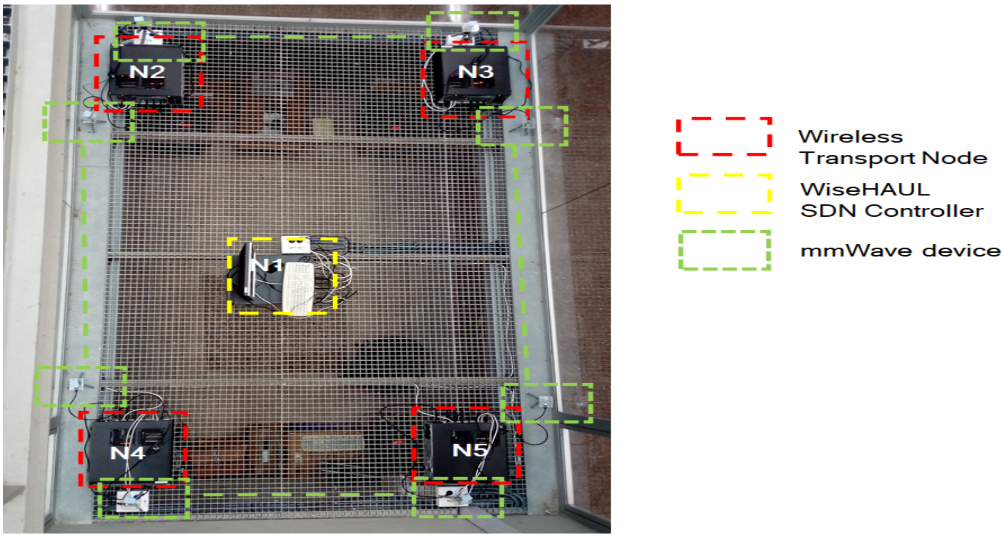


Figure 6.7: Experimental mmWave/Wi-Fi transport network setup under assessment

6.3.2.1 Link Status modification

The following results compare the experienced network downtime when there is a change in the status of the mmWave/Wi-Fi interfaces present in the wireless transport nodes. Herein, this change is introduced with the WIA client running at the SDN controller. This change emulates a management decision, where an upper layer application may decide, for instance, to switch on/off transport interfaces to make an efficient energy consumption based on the throughput required by the services deployed at the network at different periods of the day.

We use the Iperf [122] tool to generate a data flow using UDP protocol between the endpoints present at nodes labelled as N5 and N3 in Figure 6.7. After an instant, we switch off the mmWave interface of N5 connecting with N3 using the WIA. This action triggers the local recovery procedure at the WiseHAUL SDN controller, which re-configures the affected flows through the alternative path composed of nodes N5-N4-N2-N3. After some instant, we use the WIA to switch on again the mmWave interface of node N5. This message triggers the local recovery procedure, which re-establishes the former path, i.e., N5-N3. We define the downtime as the time interval during which there has been an interruption in the reception of data packets at the endpoint of node N3 as a consequence of the switch off/on action. We repeat the same experiment 20 times with a mmWave link (IEEE 802.11ad) and a Wi-Fi link (IEEE 802.11ac) in the 5GHz band. Experiments were performed at night to reduce possible interference effects in the Wi-Fi link.

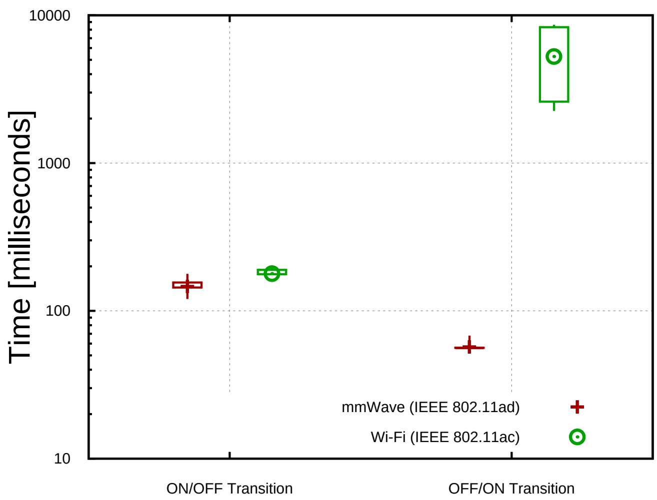


Figure 6.8: Experienced downtime in different kinds of wireless interfaces

Figure 6.8 presents the experienced downtime when considering the different transitions and the different available wireless technologies. It shows that the mean measured downtime after a switch OFF event is of 150ms for the mwWave interface, and slightly higher for the Wi-Fi interface, around 180ms. The packet reception resumes as soon as the forwarding agents install the associated flow rules at the corresponding wireless transport nodes, as all the remaining interfaces in the alternative path are in ON state. However, the case for the OFF/ON transition is different. As we can see in Figure 6.8, the *"mmWave downtime"* is less than half the value (around 56 ms on average) with respect to the ON/OFF transition, and it is much lower than the Wi-Fi case, which is on the order of seconds. From the point of view of control operations requested by the WiseHAUL SDN controller to reconfigure the forwarding behaviour of the wireless transport nodes, a lower downtime for the OFF/ON is expected because the recovery procedure implies more flow deletion operations in the different nodes (N5-N4-N2-N3) and less creation operations (N5-N3), which are more time-consuming. In this case, the difference in experienced downtime comes due to the time it takes to the wireless device to make effective the request sent by the WIA. The mmWave link is established automatically after the interface is ON again. However, in the Wi-Fi case, it is required to establish again the ad-hoc network forming the link. Since the information of previous link (e.g., frequency, basic service set identifier) is lost after the OFF operation, the creation of the new Wi-Fi link incurs higher delays (around 5.2s on average). In the best cases, this delay is of around two seconds; however, in the worst cases, it could be as high as eight seconds.

6.3.2.2 Modification of Modulation and Coding Scheme

The following results show a TCP throughput characterization of the deployed mmWave network when setting the different available MCSs with the WIA under single and multi-hop mmWave communications and considering different MTUs. Lower MCS, requiring less transmission power and power consumption, can match the throughput requirements of deployed network services. Differently to [123], we consider TCP connections between endpoints. TCP is selected given its significant share on Internet traffic. Additionally, this evaluation gains relevance for our setup, since the performance of TCP connections can be substantially affected due to the nature of mmWave communications, i.e., high data rates transmitted over the different links of a multi-hop communication with potential channel variations.

The evaluated flows last for ten seconds in each repetition, and are generated using the Iperf [122] tool. Every conducted experiment is repeated 10 times. For the case of one hop, TCP traffic has been generated between the endpoints of the nodes labelled as N2 and N3 in Figure 6.7. For the case of two hops, traffic has been generated between the endpoints of the nodes labelled as N2 and N5, using N3 as forwarding node. This forwarding behaviour has been proactively configured thanks to the use of the COP protocol at the WiseHaul SDN controller. Additionally, this characterization also explores the effect of the MTU on the throughput network performance. The maximum value of supported MTU of TC60G-USB3.0 EVK devices is 7912 bytes. However, we also consider 1500 bytes, which is popularly used in Wi-Fi networks and in Internet. As commented before, WiseHAUL platform combine different wireless technologies. This heterogeneity will be the norm in 5G network deployments and it is interesting to see how this can impact the performance of the network, since the full capabilities of mmWave hardware would not always be possible to be exploited.

In our setup, we have considered that all mmWave links operate in Channel 2 (60.48GHz), differently to [123], which considered different channels for the different mmWave links. As a member of the IEEE 802.11 family, the multi-hop communication in the presented setup is likely to be affected by co-channel interference (CCI), because both links share the same channel without being aware that they are in each other's carrier sensing range. In addition to this, the CCI is also likely to occur due to the nature of the mmWave equipment under evaluation. As explained in [124], pencil-shape beams in mmWave networks are a myth. The reason for this is that real-world phased array antennas suffer from significant side lobes, and its effect is particularly noticeable in consumer-grade hardware due to its cost-efficient design. Finally, it is worth mentioning that the IEEE 802.11ad standard only defines four channels, reducing the possibilities to design the frequency planning and making CCI more likely for close devices.

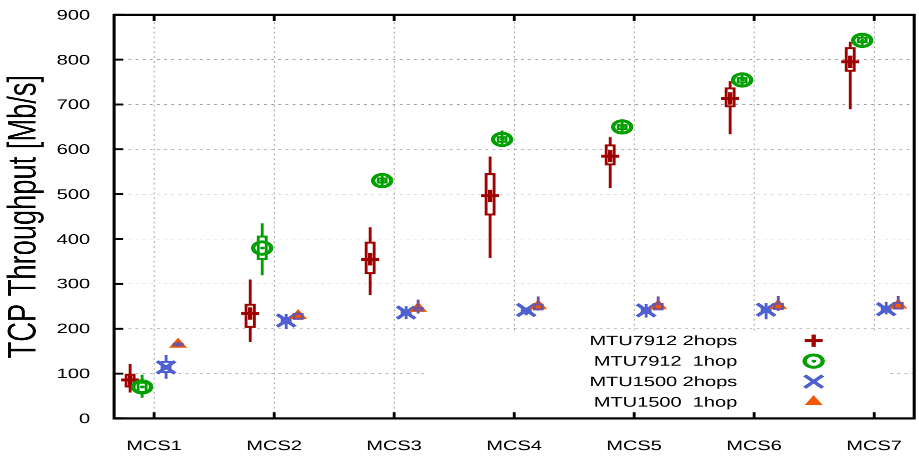


Figure 6.9: mmWave data plane performance characterization

Figure 6.9 shows the performance comparison of E2E application throughput when considering the variables previously mentioned: available MCS, number of hops and different MTU values. As expected, the maximum throughput has been achieved using MTU 7912 bytes with one hop and using the highest available MCS (MCS7). This value is of 843 Mbits/s on average. Notice that this is

TCP throughput and not link layer rate. When comparing the same MCS but different number of hops, we can see that the value is very similar, around 800Mbits/s. However, the boxplots are wider (and in general for all the considered MCSs in the 2-hop communications), showing more performance variability due to possible fluctuations of link quality at each hop of the multi-hop communication. Furthermore, we can see that the setup does not suffer from CCI introduced by radiated side lobes. As indicated in [124], the cause of a good performance at the receiver of the second hop is not only explained due to the selection of a good beam pattern capturing as much as possible the main lobe of the desired transmitter but also due to very residual reception of side lobe transmissions from the other interfering transmitter.

If we refer to the values obtained with MTU 1500 bytes, we can see that using higher MCS schemes does not provide further gains in terms of E2E throughput and there is a saturation as depicted in Figure 6.9, and also experienced with TCP connections in [125], but for the case of one hop. In such cases, having network elements with a value of MTU equal to 1500 bytes, it could be convenient to use a lower MCS in mmWave devices (giving a similar performance than higher MCS) because they provide more reliable communications requiring less sensitivity at the receiver side, and at the same time, providing energy consumption gains due to reduced processing costs at the transmitter/receiver units.

6.4 Conclusions

This chapter presents the capabilities of the SDN paradigm to automate the control and management of heterogeneous transport networks featuring different transport technologies and control planes.

Such heterogeneous transport networks will be the norm of next generation mobile networks, where different multi-domain orchestration schemes can be used to control network resources. This is the focus of the first part of the chapter, where the deployment of a hierarchical scheme within a single administrative domain has been analysed due to its scalable and modular capabilities. Extending our previous work [52], we experimentally characterise this hierarchical control scheme from a network service level perspective. In particular, we assess the deployment and recovery of the connectivity services required by a virtualised mobile service. This assessment includes the orchestration of multiple wireless transport networks with a multi-layer optical network while featuring simultaneously backhaul and fronthaul traffic. Experimental results show that the provisioning of transport network resources requires, on average, 10.467 seconds. This value accounts for the time to compute and allocate the suitable resources according to the network conditions and contributes to attain the target set for 5G networks of reducing service setup times down to minutes. With respect to the service recovery time, two approaches are assessed in the event of a link failure, namely local (at the lower hierarchical level) or centralized (at the top hierarchical level) depending on the desired path optimality versus recovery time trade-off. On average, recovery takes 0.299 seconds and 6.652 seconds, respectively. This confirms the flexibility of the proposed architecture, showing also recovery times down to the order of seconds, hence contributing to achieve the expected performance targets of next generation mobile networks.

The second part of the chapter focuses on the management of wireless interfaces through the SDN architecture. For doing so, a wireless interface agent (WIA) decoupled from the forwarding agent is presented and implemented. The WIA integrates interfaces of available heterogeneous wireless technologies, namely mmWave (IEEE 802.11ad) and Wi-Fi (IEEE 802.11ac), by means of a common information model. This agent is used to characterise the behaviour of a mmWave transport network when changing the status of the interfaces, hence modifying network topologies, and when varying the MCS of the used interfaces. These capabilities are of interest for applications pursuing different network-wide optimization objectives, like efficient energy consumption based on the throughput required by deployed services.

In conclusion, SDN constitutes one of the main pillars in the network service orchestration architecture that we are building in this thesis to enable dynamic and flexible E2E network service deployments in distributed points of presence. To continue the work presented in this chapter, there is the need to integrate the SDN control of network resources present in the transport network with the control of computing, storage and networking resources present in data centres implementing the VNF component of the NFV architecture. This is the focus of the next chapter, where the architecture of one of the main modules of the pursued network service orchestration framework, the Mobile Transport Platform (MTP), is proposed.

Chapter 7

Integrated management of networking and computing resources

The contributions of this chapter have been published in [126] and [127]. This chapter provides the description of the first core element in the proposed E2E network service orchestration presented in Chapter 5, the mobile transport and computing platform (MTP). It acts as the unifier controller presenting the status of available NFVI resources and executing the resource allocation process in collaboration with the different infrastructure managers, i.e., VIMs and WIMs. For instance, the hierarchical SDN architecture presented in the previous chapter is an example of a WIM instance.

Section 7.1 presents the rationale behind the definition of the MTP block. Making a mapping with respect to the ETSI-NFV architecture, the MTP functionality is performed at the NFVO block. A very relevant aspect of the MTP is its abstraction capabilities, which provides scalability and allows simplifying the network service orchestration process performed at higher layers of the proposed architecture by hiding the heterogeneity of infrastructure resources. This clear decoupling between resource allocation and network service orchestration operations is a different aspect with respect to the architectures presented in Section 3.2 and allows a more modular architecture where the different stakeholders can develop a higher degree of specialization. In Section 7.2, we propose an architecture for the MTP block and then, we present different abstraction alternatives of transport network resources in Section 7.3. Section 7.4 describes the procedures allowing the MTP to dynamically interconnect VNFs of the same network service placed in different NFVI-PoPs connected by a WAN transport network. Note that the work in this chapter mostly focuses on transport network aspects, which was an architectural lack we detected in the ETSI-NFV work during the development of this Thesis, as commented in the Chapter 3. Thus, the presented MTP acts as the *"connection bridge"* between the SDN and NFV paradigms, that is, connecting the work presented in the previous chapter with the one in the next chapter. Due to its bridge role in the proposed E2E architecture, the validation of the MTP block is included within the performance evaluation presented in the following chapter, which studies the overall E2E orchestration of network services in a real experimental setup.

7.1 Orchestration of computing and transport networking resources

As presented in Section 2.4, the envisaged architecture for next generation mobile networks is based on softwarization and modularity principles and relies on the distributed deployment of interconnected NFVI-Points of Presence (PoPs) to deploy network functions satisfying the heterogeneous requirements of innovative network services demanded by new stakeholders in the telecom network ecosystem, the vertical industries.

NFVI-PoPs can offer non-virtualised resources, also known as physical network functions (PNF), and virtualised compute, storage and networking resources based on commodity hardware to implement the virtual network functions (VNFs). These virtualised resources are offered by cloud orchestrator software stacks (e.g., Openstack), also known as VIM. These NFVI-PoPs are interconnected by means of transport networks controlled by the WIM entity, which will need to rely on SDN-based network control solutions to offer the required dynamic connectivity services. Thus, the orchestration of the heterogeneous resources required by the different network services consists of the integrated management of the capabilities offered by several entities, i.e., multiple VIMs and WIMs.

The scale of network deployments, in terms of amount of NFVI-PoPs, transport network density, and the heterogeneity of resources implying different technological domains (wireless, optical transport connectivity types), increases the complexity of the E2E orchestration of network services. For this reason, resource abstraction strategies are applied to represent the NFVI resources so as to simplify the E2E orchestration procedure.

Beside this, and what is more interesting in this work, is the application of abstraction strategies together with the use of *standardized* interfaces, which have been extended. This allows the decoupling of the main responsibilities assigned to the NFVO block [69] proposed in the ETSI-NFV architecture in terms of resource orchestration operations between multiple building blocks of the orchestration architecture placed at different levels, as in our proposed architecture with the MTP and the Service orchestrator (SO) modules. Furthermore, this decoupling enables the specialization of some of the different stakeholders involved in the deployment and management of next generation mobile networks, i.e., infrastructure providers and service providers, and enables more advanced and evolved scenarios, in terms of involved stakeholders, where a single infrastructure provider may serve multiple service providers or a service provider may interact with multiple infrastructure providers.

Based on this reasoning, the proposed E2E network service orchestration defines the Mobile Transport and Computing Platform (MTP) building block, hence decoupling the E2E network service orchestration process performed by upper layers over an abstracted view, from the execution of detailed resource orchestration operations in the different involved segments. This entity performs the unified control of NFVI resources in collaboration with the different infrastructure managers, i.e., VIMs and WIMs. Some of the most relevant operations performed by this unified resource controller are:

- The collection and storage of the status of the underlying NFVI resources reported by VIMs and WIMs in a repository.
- The application of abstraction mechanisms to provide resource information to upper layers, easing the E2E orchestration procedure.
- The registry of the stitching information between NFVI-PoPs managed by VIMs and the transport networks managed by a WIM, as well as connections between different interconnected transport networks controlled by multiple WIMs to dynamically determine the interconnections between deployed network endpoints.

- The allocation/release of isolated virtualised resources satisfying the VNF requirements as well as isolated transport resources satisfying the specified quality of service constraints for VNF interconnections (e.g., rate and latency) and the provision of PNF connectivity to the virtualised resources.

7.2 Mobile Transport and Computing Platform Architecture

Figure 7.1 presents the architecture of the proposed Mobile Transport and Computing Platform module. Briefly, the MTP provides abstracted NFVI resource information to the SO based on the information retrieved from the different infrastructure managers and executes its allocation/deallocation requests for the different NFV-NSs instances. To do so, the MTP selects the appropriate computing resources for the VNFs and the appropriate networking resources for the Virtual Links interconnecting the VNFs of a network service (NFV-NS) to satisfy its requirements and ensure the proper isolation between NFV-NSs instances. Next, a detailed description of the components of the MTP architecture is presented:

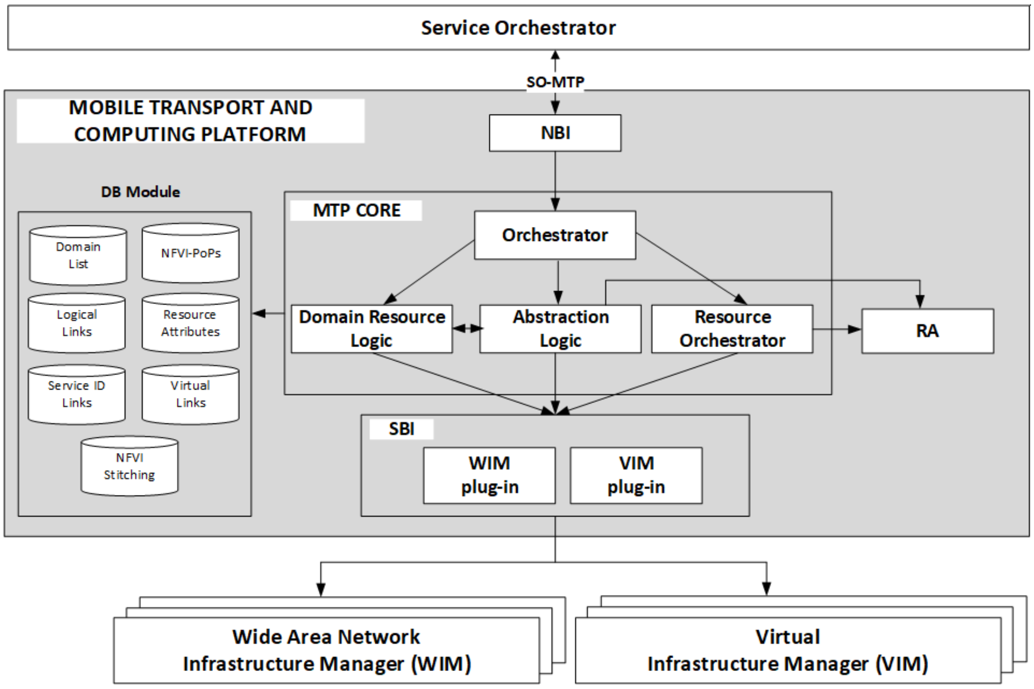


Figure 7.1: Mobile Transport and Computing Platform Architecture

- **MTP Core**: the main component of the MTP architecture, in charge of executing the management and configuration of computing, storage and networking resources at the underlying VIM and WIM entities. It contains four sub-modules:
  - *Orchestrator*: this module receives the requests from the NBI and it is in charge of coordinating the rest of modules at the MTP Core to execute the received requests.
  - *Domain Resource Logic*: it retrieves the networking, computing and storage resources announced from the available WIM/s and VIM/s entities. Then, it combines this information with the stitching information available in the database (DB) module and transforms it into graph elements to be processed by other sub-modules.
  - *Abstraction Logic*: this sub-module implements the logic required to pass from a detailed representation of resources as exposed by the infrastructure managers (i.e., WIM/s and

VIM/s) to a *simplified* view, which allows hiding the specificities of the underlying infrastructure. An example of this abstraction operation is the transformation of sets of chained transport network links sharing some aggregated characteristics (e.g., rate, latency) into the logical links (LLs) view representing point-to-point links interconnecting NFVI-PoPs. More on this in Section 7.3.

 – *Resource Orchestrator (RO)*: this sub-module implements the methods to create/modify/remove the actual resources in VIM/s and WIM/s. These resources support the components of the NFV-NS (i.e., VNFs and virtual links) through the different lifecycle phases (e.g., instantiation, scaling, termination). As part of this process, it also interacts with the Resource Allocation (RA) module.

- **NorthBound Interface (NBI)**: this module receives and validates the requests from the SO to retrieve the status of NFVI resources and perform their allocation. The NBI provides a unified reply presenting the status of all NFVI resources, i.e. those managed by the different VIMs and the available interconnections among NFVI-PoPs. In the case of allocation, the NBI handles separately different requests directed to the multiple kinds of NFVI resources because they are allocated during different steps of the NFV-NS instantiation process. On one hand, there are those requests related to VIM operations, like the creation and allocation of VNFs, and virtual network resources to attach the port of VNFs. On the other hand, there are those requests handling WIM-related operations, like the configuration of the interconnections among NFVI-PoPs through the underlying transport network to communicate VNFs.

- **SouthBound Interface (SBI) module**: it implements the logic to communicate with the underlying VIM and WIM entities through a plugin system. Thanks to this plugin system, different types of entities implementing VIM/WIM functionality can be easily supported by the MTP.

- **Database (DB) module**: this module manages the different DBs containing the system information. Currently, the available databases are:

  – *Domain List*: it includes the registries of the WIM, VIM entities in the underlying infrastructure and their corresponding IP addresses and access credentials.

  – *NFVI-PoPs*: it contains association information between NFVI-PoPs and the registered VIMs they belong to.

  – *Resource Attributes*: it includes collected resource data in terms of CPU/RAM/Storage provided by each registered NFVI-PoP.

  – *Logical Links (LLs)*: this DB includes the description (e.g., offered bandwidth, delay) and status of the link abstractions representing the available interconnections between registered NFVI-PoPs.

  – *ServiceId*: it contains the identifiers and the information related to all current established interNFVI-PoP connections created in the MTP module across the successive network service instantiation processes ordered by the SO module. These interNFVI-PoP connections are characterised by the used LL, its requested bandwidth and latency requirements, and the NFV-NS instance associate to the creation of such connections.

  – *NFVI Stitching*: it registers the connection points between the transport network managed by a WIM entity and the available NFVI-PoPs of a VIM entity and the connections points between the transport networks managed by different WIMs.

  – *Virtual Links (VLs)*: it includes information related to the virtualised network resources created in the different VIMs to support the required communication between VNFs of the same NFV-NS, as expressed in the network service descriptor (NSD). Then, the VNFs attached to these virtualised networks may require to be interconnected by allocating resources in the available LLs. Among this information, this DB keeps track of the used virtual network name, the VLAN identifier, the Classless InterDomain Routing (CIDR) value, and IP address Pools determined by the RO module of the MTP to avoid collisions and to provide the required isolation between the traffic of VNFs attached to the different virtual network instances. In the MTP, the address Pool is represented as an integer value representing a set of IP addresses within an assigned CIDR. This is needed to coordinate the assignment of IP addresses among the different VNFs of an NFV-NS connected to the same VL and deployed in different VIMs.

- **Resource Allocation (RA) module**: this module is in charge of requesting the calculation of detailed paths in the underlying transport network satisfying the requirements expressed in the request done by the RO or the Abstration Logic sub-modules. It communicates with the *RA Engine*, an external process doing the actual path calculation, through a well-defined REST API. In this way, multiple external algorithmic modules can be easily integrated in the MTP.

After presenting the different modules composing the architecture of the MTP, the following lines provide additional details about the design choices for the interfaces used by the MTP at the NBI and the SBI. The MTP NBI offers a REST-API considering different ETSI-NFV specifications and reports, but they have been extended and adapted to provide all the required functionality. In the case of presenting the status of NFVI resources, the unified answer provided by the NBI is based on one hand, on the structure of information elements (IEs) available in ETSI-NFV IFA 005 [128] specification for describing NFVI-PoP resources. On the other hand, a new IE is required to model the link abstractions representing the available interconnections between distributed NFVI-PoPs. This integrated representation of NFVI resources is a lack detected in the ETSI-NFV work. This new IE considers the work presented in the ETSI-NFV IFA022 report [129], which only provides high-level guidelines on the management and connectivity of multi-site services. In the case of allocation/deallocation-related operations, the NBI considers the next. For VIM-related operations, it bases on ETSI-NFV IFA005 operations, but for WIM-related operations, new operations have been defined to allocate/deallocate resources in the underlying transport network according to the new IEs modelling the available link abstractions. Actually, this topic is further developed in Section 7.4 and it is one of the main achievements of this work showing the integration of SDN and NFV paradigms.
With respect to the SBI, the VIM plugin considers OpenStack software [22] as the reference VIM entity and the WIM plugin considers the Control Orchestration Protocol (COP) protocol [53] to interact with the WIM entity, hence allowing the direct integration of the work presented in the previous chapter. Other types of VIM/WIM entities could be supported through the definition of its corresponding plugins. A prototype of the MTP is released as open source code under Apache 2.0 licence [130].

7.3 Management alternatives of transport network resources at the MTP based on abstractions

One of the most relevant aspect of the MTP architecture is its capabilities to apply abstractions on the resources under its control. These abstractions serve to hide the specificities of the multiple underlying technological domains, hence simplifying upper layer operations related to the network service orchestration, as explained in next chapter.

Since the nature of the NFVI resources are intrinsically different, the abstraction mechanisms applied by the Abstraction Logic sub-module differ. In the case of VIM-related resources, the abstraction presented by the MTP to upper orchestration layers represents an amount of CPU units, RAM memory and storage capabilities. More specifically, it presents the total, used and available capacity for each of the resources of interest. This abstraction may be further refined to represent these values for each of the hosts under the control of a VIM instance. In the case of WIM-related resources, the abstraction of the transport network provided by the MTP is based on the concept of Logical Link (LL), as previously mentioned in Section 7.2. A LL characterises a unidirectional point-to-point interconnection between two NFVI-PoPs present in the NFVI with certain QoS characteristics, e.g., rate, latency, round trip time, or cost. Figure 7.2 shows the transformation of the detailed view observed by the MTP to the abstracted view generated by the MTP. Note that, as seen in the previous chapter for the hierarchical SDN control, an infrastructure manager could offer as well a partial abstracted view of its own resources.

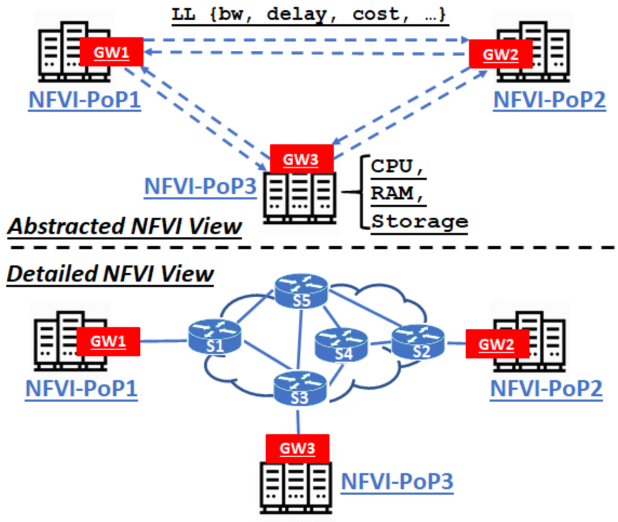


Figure 7.2: Detailed versus abstracted NFVI view

Based on the LL concept, we devise two operational modes for the presented MTP architecture, which determines how these LLs are derived, exposed and ultimately allocated. These operational modes, which have been integrated in the different MTP workflows, are the Infrastructure Abstraction (InA) and the Connectivity Service Abstraction (CSA) [127].

In the InA operational mode, every LL is always associated to a physical WAN path computed with the help of the RA module. As an example, an LL connecting NFVI-PoP1 and NFVI-PoP2 in Figure 7.2 is derived from the computed path formed by NFVI-PoP1, S1, S5, S4, S2 and NFVI-PoP2. The LL attributes inherit the features of the underlying WAN path supporting such an LL. That is, the available LL bandwidth is inferred from the unused bandwidth on the most congested path link. Moreover, the LL delay is computed as the total accumulated delay over each WAN path link. Notice, that once the upper SO module selects a specific LL for the NFV-NS, it implicitly determines the WAN nodes and links advertised by the corresponding WIM/s to accommodate the traffic flow between the respective NFVI-PoPs. In the InA operational mode, the MTP computes and presents to the SO up to K paths between pairs of NFVI-PoPs when requesting the status of the NFVI resources (including also the status of computing resources) at each NFV-NS instantiation operation. The rationale behind targeting K WAN paths deriving K LLs for a NFVI-PoP pair is to enable the SO to use different LLs if multiple connections need to be deployed between a given NFVI-PoP pair. In the InA operational mode, the *Abstraction Logic* sub-module triggers the path computation at the *RA module* every time the NFVI resource status is requested. We insist on the fact that this path computation is then not needed with the actual network resource allocation request arriving from the SO to interconnect the VNFs of the same NFV-NS deployed in different NFVI-PoPs because the path has been already defined.

In the CSA operational mode, the infrastructure operator controlling the MTP module characterises its transport network with a set of connectivity service types/classes defining supported characteristics for the connectivities between NFVI-PoP pairs. Each offered connectivity type/class provides its own parameters, e.g., guaranteed bandwidth and maximum delay. Therefore, no explicit abstraction computation is triggered at the *Abstraction Logic* sub-module to derive the exposed LLs to the SO when it requests the status of the NFVI resources during the NFV-NS instantiation operation. Conversely to the InA operational mode, when the SO requests the allocation of resources on such LL during the NFV-NS instantiation process, the MTP is required to conduct an explicit WAN path computation to find the nodes and links supporting the selected LL attributes, i.e., the associated service type/class parameters. This path computation is what we denote as the *expansion* mechanism. Hence, the LLs are not bound to a pre-computed WAN path and the same service type/class could follow multiple physical paths. This may lead to a better use of the WAN resources because there is not a direct mapping between LL and physical path. In this case, the *Resource Orchestrator* sub-module triggers the computation at the *RA module* for each requested LL during the actual resource allocation operations.

In this work, the algorithm running in the *RAEngine* to compute the LLs between NFVI-PoP pairs is based on the well-known Yen algorithm [131]. In the case of InA operational mode, this algorithm is run without constraints in terms of bandwidth or latency between each possible pair of NFVI-PoPs. The K best WAN paths are determined sorting the possible solutions by selecting those with the largest end-to-end available bandwdith, lowest end-to-end delay and path with lowest cost in terms of number of hops. In the case of the CSA operational mode, the path-computation algorithm runs with constraints between the specified NFVI-PoPs to find the path fulfilling the bandwidth and latency requirements associated to the service class.

The InA operational mode provides a bigger degree of detail of the underlying NFVI with respect to the CSA operational mode. Hence, it seems plausible to use the InA operational mode in scenarios where the NFVI is controlled by the same actor performing the network service orchestration on top of that (in this case the SO module of the proposed architecture). Accordingly, the CSA operational mode may be an alternative in a scenario with multiple stakeholders, where for e.g., confidentiality reasons, a lower level of detail of the NFVI can not be provided. With respect to performance, aspects to take into account at each operational mode are the size of the underlying WAN topology and the complexity plus degree of distribution of the VNFs of the NFV-NSs requested to be deployed. Larger WAN infrastructures increase the abstraction computation time in the InA mode, which in turn enlarges the required time to present the NFVI resource view to the SO. However, in the case of a complex service (i.e., one with a considerable amount of VNFs and multiple connections between them) which has been distributed among multiple NFVI-PoPs, the CSA mode needs to trigger the expansion mechanism for each of the possible requested LL, hence requiring additional time to complete the deployment of the LLs due to the additional interactions with the RA module to determine the available transport paths fulfilling the LL requirements. Needless to say, that in this case, the size of the WAN infrastructure also impacts the time required to find the requested path.

In the rest of this work, when presenting performance evaluation results, the use of the CSA operational mode is considered in the MTP because of its possibility to determine multiple physical paths for the same LL, which may help to attain a better use of the networking resources. It is left for future work an exhaustive evaluation to quantify the advantages and drawbacks of each option. However, it is worth highlighting the flexibility of the proposed MTP architecture to allow the use of these different operational modes.

7.4 Interconnection of NFVI-Points of Presence through the MTP

This section describes the different operations performed at the MTP level to interconnect VNFs of the same NFV-NS deployed in different NFVI-PoPs. It illustrates the required integration of the SDN and the NFV paradigms to enable the dynamic E2E orchestration of NFV-NSs across geographically distributed NFVI-PoPs interconnected by a transport network. This is one relevant aspect covered by the proposed E2E Network service orchestration architecture with respect to other orchestration approaches presented in Chapter 3.

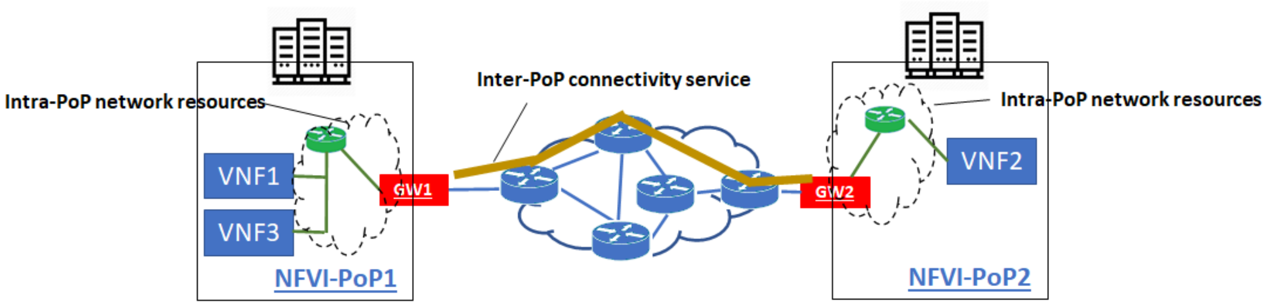


Figure 7.3: Multi-site connectivity scenario

This process is divided in two operations that are executed in different moments of the NFV-NS instantiation process controlled by the SO module, as detailed in the next chapter. These operations are: (i) the creation of intra-PoP network resources at the VIM level to attach the ports of allocated VNFs, and (ii) the allocation of inter-PoP network resources between NFVI-PoPs performed at the WIM level, as depicted in Figure 7.3.

We denote here as intra-PoP network resources those managed by a VIM entity within its NFVI-PoPs under control. They are created to map the VLs interconnecting the VNFs of a given NFV-NS, as expressed in the NSD. They consist of a virtual isolated Layer-2 broadcast domain inside the cloud infrastructure and an associated Layer-3 IPv4 or IPv6 subnet. The virtual network interface card (vNIC) of the attached VM implementing the VNF is assigned an IP address from this subnet. We refer to the inter-PoP network resources as the connectivity services established in the underlying transport network resources offered by the WIM. They allow the interconnection of the VNFs of an NFV-NS deployed in different NFVI-PoPs while satisfying the QoS characteristics required for the VLs.

Figure 7.4 presents the workflow followed by the MTP to create the intra-PoP network resources. The SO sends an *Allocate Virtualised Network Resource* ETSI-NFV IFA005 [128] request to the MTP previous to the VNF allocation operation (step 1). Among other parameters, this request specifies the network name and the VIM where the required intra-PoP network needs to be created. The SO generates this network name so it is unique for each NFV-NS instance and virtual link interconnecting VNFs, hence allowing the concurrent deployment of multiple vertical services and multiple instances of the same vertical service at the logical level. This request arrives to the Resource Orchestrator sub-module of the MTP, where it checks whether this network name entry is present in the Virtual Links DB (step 2). If it is not present, the Resource Orchestrator determines a new VLAN identifier, CIDR range and the address Pool parameters to use in this VIM to create the intra-PoP network (step 3), hence allowing the physical isolation of resources between concurrent virtual links of the same network service and of other network service instances. If the network name is present in the DB, then with the parameters retrieved from the DB, the MTP proceeds to the creation of the new intra-PoP network at the specified VIM. The network name can be present in the DB due to the fact that the SO has decided to split the components of an NFV-NS in multiple PoPs and a previous intra-PoP network with the

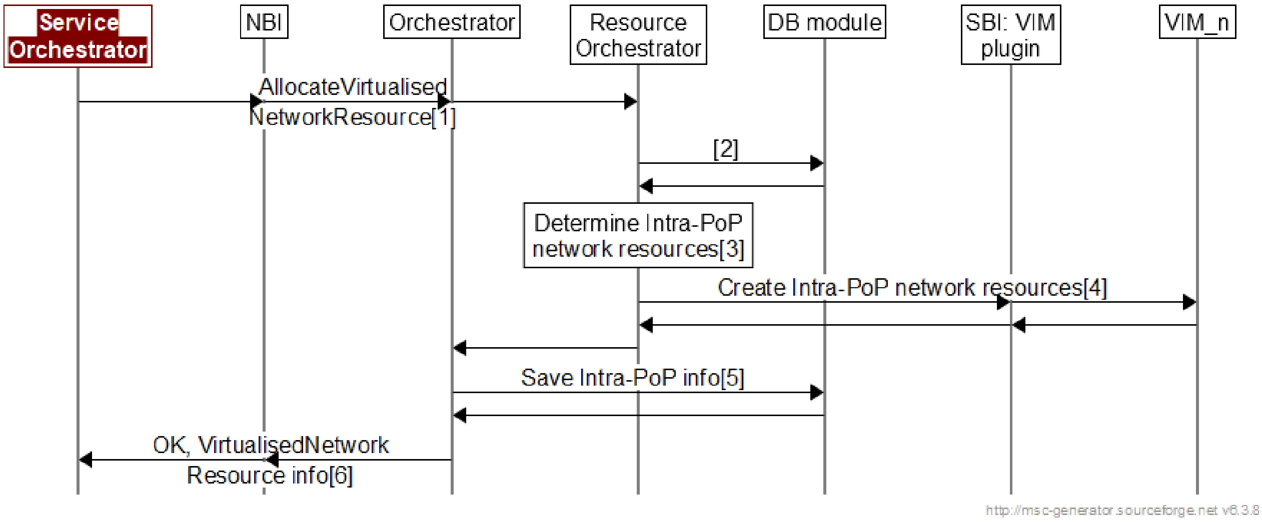


Figure 7.4: MTP workflow for the creation of intra-PoP network resources

same network name in another VIM has been requested. Then, the Resource Orchestrator requests the creation of the intra-PoP network to the SBI VIM-plugin, which contacts the corresponding VIM (step 4). After the intra-PoP network is created, the Resource Orchestrator reports the Orchestrator, which saves the information of the created virtual network resource at the Virtual Link DB (step 5). Finally, the Orchestrator validates the creation of the requested network resource and sends the associated details to the SO (step 6).

This proposal differs from other contemporary work [132] in the fact that the MTP uses the same VLAN identifier in all VIMs requested to create the intra-PoP network with the same network name. On one hand, this allows a better usage of the VLAN identifiers, limited to 4096, because the same identifiers are used on a VL basis and not on a VIM basis. On the other hand, this avoids having to swap VLAN identifiers in the transport network when configuring inter-PoP connectivity services, hence simplifying the configuration of forwarding operations and the complexity of forwarding operations. For NFVI-PoPs sharing the same network name, the MTP assigns a different address Pool, which serves to avoid IP address collision because the intra-PoP network DHCP clients are configured in disjoint sets of IP addresses at the different involved VIMs.

After the VNFs have been allocated and attached to the created intra-PoP networks, the SO will send a single request to the MTP with a list of required inter-PoP connectivities between VNFs of the same NFV-NS placed in different NFVI-PoPs and attached to an intra-PoP network with the same name. The workflow followed by the MTP to allocate resources on the requested LLs depends on the MTP operational mode (i.e., InA and CSA), as described previously. Figure 7.5 presents the workflow to create the required inter-PoP connectivity services for the CSA operational mode. After performing de-abstraction operations (steps 3 and 4), both operational modes differ in step 7. In InA operational mode, the *Resource Orchestrator* does not need to interact with the *RA* module to handle the computation of transport paths satisfying the QoS characteristics specified in the request of step 1 because they have been calculated previously when presenting the status of the NFVI resources. Instead, the *Resource Orchestrator* interacts with the DB module to retrieve the information of the requested LL, prior to allocating network resources on it.

Listing 1 presents the format of the request received by the MTP in step 1 of the workflow described in Figure 7.5. This is a new request answering to the detected lack in the ETSI NFV architecture, and partially handled in ETSI-NFV IFA022 [129] report, as commented previously.

Listing 2 presents the response to this request, which is a list containing the identifiers and the

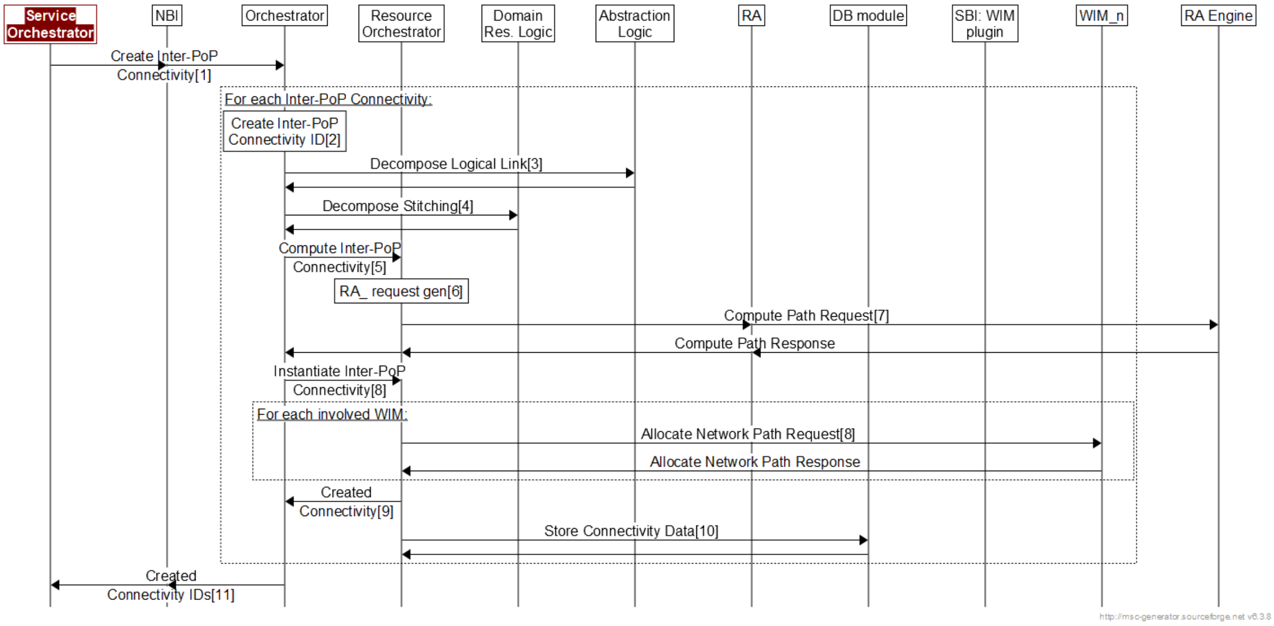


Figure 7.5: MTP workflow for the creation of inter-PoP network resources (CSA operational mode)

Listing 1 SO request to MTP to establish inter-PoP connectivity services

```
Request: Inter-PoP connectivity creation
URL =  "http://mtp_ip:port/5gt/mtp/v1/abstract-network-resources"
body: { "logicalLinkPathList": [
           "logicalLinkAttributes": {
              "srcGwIpAddress": string,
              "dstGwIpAddress": string,
              "localLinkId": string,
              "remoteLinkId": string },
           "reqBandwidth": string,
           "reqLatency": string,
           "metaData": {
              {"key": "srcVnfIpAddress",
               "value": string},
              {"key": "dstVnfIpAddress",
               "value": string},
              {"key": "networkName",
               "value": string} }
        ],
        "interNfviPopNetworkType": string,
        "networkLayer": string,
        "metadata":{
           "key": "ServiceId",
           "value": string }
       }
```

type of established inter-PoP connectivity services. These identifiers are then saved in the *ServiceId* DB presented previously.

Listing 2 MTP inter-PoP connectivity service response

```
Request: Response to inter-PoP Connectivity
body: { [ { "key": "interNfviPopConnectivityId",
            "value": string },
          { "key": "interNfviPopNetworkSegmentType",
            "value": string } ] }
```

Next, we comment three aspects with respect to the creation of inter-PoP connectivity services with the proposed MTP architecture. These considerations enable the support to more generic deployments of VNFs of the same NFV-NS among an arbitrary number of NFVI-PoPs while validating the content of the request for the intended NFV-NS deployment. First, the content of metadata field in the inter-PoP connectivity request contains the source and destination IP addresses of the VNF ports it connects and the network name they share. Thanks to the network name, the MTP is able to recover from its DB the associated parameters of the intra-PoP network (VLAN identifier, CIDR and address Pool) and verifies that the specified IP addresses are within the registered CIDR and address Pools, hence validating the content of the request. Second, once the request is validated, the MTP incorporates in the subsequent request to the involved WIMs not only the VLAN identifiers, but also the specified IP addresses of the VNFs so the WIM can configure the forwarding elements (FEs) available in the transport network. With the use of these two parameters (VLAN identifier and IP addresses), the proposed E2E network service orchestration architecture supports the location of VNFs of the same NFV-NS in more than two NFVI-PoPs interconnected by partially overlapped physical paths. Third, the MTP generates four requests to the involved WIMs for each inter-PoP connectivity service in the *logicalLinkPathList* allowing bidirectional ARP and IP traffic.

7.5 Conclusions

This chapter deals with the description of the MTP building block of the proposed E2E network service orchestration architecture. The architectural design of this block addresses the creation of a unified NFVI controller, which handles two main basic set of processes. First, it provides *a vision* of the NFVI resources under its control to upper orchestration layers, and then, it executes the resource allocation/deallocation requests needed to deploy the network services over this NFVI composed of distributed and heterogeneous computing, storage and networking resources.

The use of abstraction/de-abstraction mechanisms, appropriate information DBs and interfaces are required to present, handle the relation, and manage the status of the different underlying NFVI resources. Thanks to this, the whole set of resource orchestration functions defined by the ETSI-NFV architecture can be decoupled into the multiple building blocks of our proposed E2E architecture. This helps simplifying the service orchestration process performed at upper layers of the orchestration stack and allows more diverse network management scenarios in terms of involved stakeholders.

This chapter focused mainly on the management of networking aspects related to the interconnection of the distributed NFVI-PoPs where the multiple VNFs of an NFV-NS can be deployed. This is a lack detected in the ETSI-NFV work. The proposed MTP module considers the abstraction of the underlying transport network presented by the WIM/s entities under its control as a set of links between NFVI-PoPs with certain QoS characteristics, in what we refer to as Logical Links (LLs). In this work, we propose two different operational modes for the MTP which differ mainly in the definition of the LLs. This difference impacts in the design of the workflows associated to the mentioned set of MTP basic processes: the presentation of the NFVI resource status and the allocation/release operation. The use of one or other definition may be suitable depending on the relation between involved stakeholders. However, the proposed architecture for the MTP is flexible enough to allow such operational modes.

The allocation of networking resources in the transport network performed at the WIM level to map the LLs together with the allocation of networking resources at the VIM level to attach the different VNFs show the tight integration between the SDN and the NFV paradigms. The MTP coordinates its different submodules to perform this allocation process while avoiding collision between assigned resources, and enabling isolation and support to deployments of VNFs of the same NFV-NS in an arbitrary number of NFVI-PoPs.

We conclude that all these mentioned aspects are essential to enable the dynamic orchestration of multiple network services instances over the same shared infrastructure. The next chapter delves into this process by presenting the architecture of the brain module of the proposed E2E network service orchestration architecture, the Service Orchestrator building block.

Chapter 8

End-to-End Orchestration of Network Services

The contributions of this chapter have been published in [133], [134], [135], [10], [11], and [136]. The resulting architecture has also been included in the European Commission's Innovation Radar programme, which highlights excellent innovations resulting from the different funded research projects. Following the bottom-up building process initiated in Chapter 6, here, the focus is on the Service Orchestrator (SO) module, which performs the end-to-end (E2E) orchestration and lifecycle management of NFV-Network Services (NFV-NSs) mapping the requested network slice instances based on the resource availability provided by the beneath MTP block.

This chapter begins with Section 8.1, where we explain the concepts and the related considerations required to perform the integrated E2E orchestration of network services. Section 8.2 presents the proposed architecture for the Service Orchestrator module to achieve the described end. Sections 8.3 and 8.4 focus on how the proposed architecture performs some of the most relevant lifecycle management operations of NFV-NSs: instantiation and service level agreement (SLA) fulfilment. Both operations are evaluated in an experimental setup considering a real automotive vertical use case. This evaluation shows service creation times and the execution of SLA management actions in the order of few minutes, in line with the 5G-PPP KPI objective. Appendix A has been included at the end of the document, where we can find a description of other relevant lifecycle management operations, namely NFV-NS onboarding and termination. This chapter closes the first part of this dissertation, devoted to the automated E2E orchestration of NFV-NSs in single-administrative domain scenarios.

8.1 Orchestration of NFV-Network Services

The synergies between SDN and NFV paradigms, together with cloud computing capabilities, as described in previous chapter, provide the network operators with new technological enablers to flexibly and dynamically create, deploy, and manage networks services in next-generation mobile networks. This is known as network service orchestration (NSO). Briefly, the NSO concept can be defined as the process of understanding the semantics of the requested service, selecting and controlling automatically multiple available kind of heterogeneous resources to meet the requirements expressed by the network service request and perform its E2E lifecycle management ensuring the mechanisms to fulfil defined service level agreements (SLA) not only at instantiation time but also during run-time [68].

In order to do so, first, it is required a model of the network service expressing its structure, characteristics, and requirements to have the ability to select and automate the control of available physical and virtual resources delivering the service. In current MANO schemes, an NFV-NS is modelled by means of a descriptor as a set of constituent parts, normally a list network functions (NFs). They are arranged following an unspecified connectivity between them or according to one or multiple forwarding graphs (FG), the so-called VNF-FG, expressing the topology of the NFV-NS (or a portion of it). The constituent parts may be virtual network functions (VNFs), physical network functions (PNFs) or another NFV-NSs, called nested NFV-NS (more on this in the second part of this dissertation). VNFs are also described by means of a model including, among others, its requirements regarding virtualised resources (CPU, RAM and storage), the image needed to implement its functionality and the offered connection points (interfaces) to interact with other NFs. It is worth mentioning, that for the case of a PNF, the descriptor expresses the connectivity details to enable the interaction of this already deployed element (PNF) with the virtual elements (VNFs, nested NFV-NS). The constituent parts of an NFV-NS are interconnected by means of virtual links (VLs), expressing the kind of connectivity and the Quality of Service (QoS) performance requirements (e.g., dedicated rate, delay, jitter, packet loss, etc.) in the inter-NF communication as well as the communication with other nested NFV-NSs. The set of performance requirements of a VL constitutes its deployment flavour (DF). At the NFV-NS level, the DF concept is also present, where multiple DFs could also be defined, each one containing a different set and number of instances of NFs to form multiple versions of the same NFV-NS with different structures, which represent the multiple instantiation levels (IL) of this NFV-NS. The NFV-NS model shall include monitoring parameters to be tracked during the lifetime of an NFV-NS instance and associated actions to enable mechanisms to inform and/or react in front of possible SLA violations.

The NSO process can be split into two main types of operations [69]. First, there is the service orchestration logic, which deals with the reception and understanding of the different and concurrent network service requests and with the management of its lifecycle by triggering the execution of its different associated workflows. These workflows define a set of tasks to be executed in a determined order to transition between the different states of the lifecycle operation (e.g., from *instantiating* state to *instantiated* state). Second, there is the resource orchestration logic, which is in charge of deciding the optimum placement of NFs and the request of the required resources to be allocated to implement the VNFs and its interconnections, based on the NFV-NS requirements and the provided (possibly abstracted) resource view of the underlying NFVI. Note that here, we refer only to the functions selecting the resources and not the execution of resource allocation operations, which in the proposed E2E network service architecture is performed by the MTP block. As mentioned in the previous chapter, this approach proposing a decoupling of resource orchestration responsibilities is a characteristic of the proposed E2E architecture. This allows flexible scenarios with multiple stakeholders, which can achieve a higher degree of specialization, i.e., infrastructure providers, service providers.

The following list highlights the most relevant lifecycle management operations performed during the continuous NSO process. These are the operations addressed by the proposed Service Orchestrator module developed in the following sections.

- Onboarding and maintenance of NFV-NS and associated NF (VNFs, PNFs) descriptor catalogues.
- Instantiate an NFV-NS by pointing to the specific artefacts present in the previously stored descriptors (DFs, ILs), satisfying the expressed requirements, and configuring the mechanisms to verify the continuous fulfilment of such requirements.
- Scale Network Service, e.g., change the structure of the deployed NFV-NS by modifying the amount of VNF instances to react in front of a detected SLA violation.
- Retrieve NFV-NS information, to check and validate the operational state of the NFV-NS instance.
- Terminate NFV-NS, i.e., request the termination of its constituent NF instances to release and return the associated NFVI resources.

To offer all its potential, the NSO process, or better said, the entity executing it, needs to integrate and interwork with other management entities like the OSS/BSS entity (e.g., like the Vertical Slicer (VS) in the proposed E2E orchestration architecture depicted in Chapter 5)). The automation of these operations is achieved through a joint evolution of both type of entities implying the use of open and standardized interfaces with common information models. Additionally, the use of open standardized interfaces brings increased interoperability capabilities to the system, promoting scenarios with multiple stakeholders, as previously mentioned.

8.2 Service Orchestrator (SO) Architecture

Figure 8.1 presents the architecture of the proposed Service Orchestrator module, showing also the relation with the VS, the MTP and the MON building blocks of the E2E Network Service Orchestration architecture presented in Chapter 5. This architecture has been designed to drive the NSO process according to the considerations and lifecycle management operations mentioned in previous section. Its design has flexibility and modularity in mind to exploit already available functionalities from well-known MANO frameworks (e.g., OSM or Cloudify) and focus on the addition of innovative functionalities allowing the desired E2E orchestration under the considered scenarios. Briefly, the SO receives from the VS the requirements of the network slice instances to deploy in the form of NFV-NSs and requests the available infrastructure status (e.g., network topology, available computation capabilities) from the MTP. The placement algorithm (PA) maps these requirements to the available infrastructure. Then, the SO requests the allocation of the service components requested by an NFV-NS, i.e., VNFs and virtual links (VLs), over the underlying infrastructure in the determined locations and configures the monitoring mechanisms verifying the fulfilment of defined SLAs to achieve a successful service delivery.

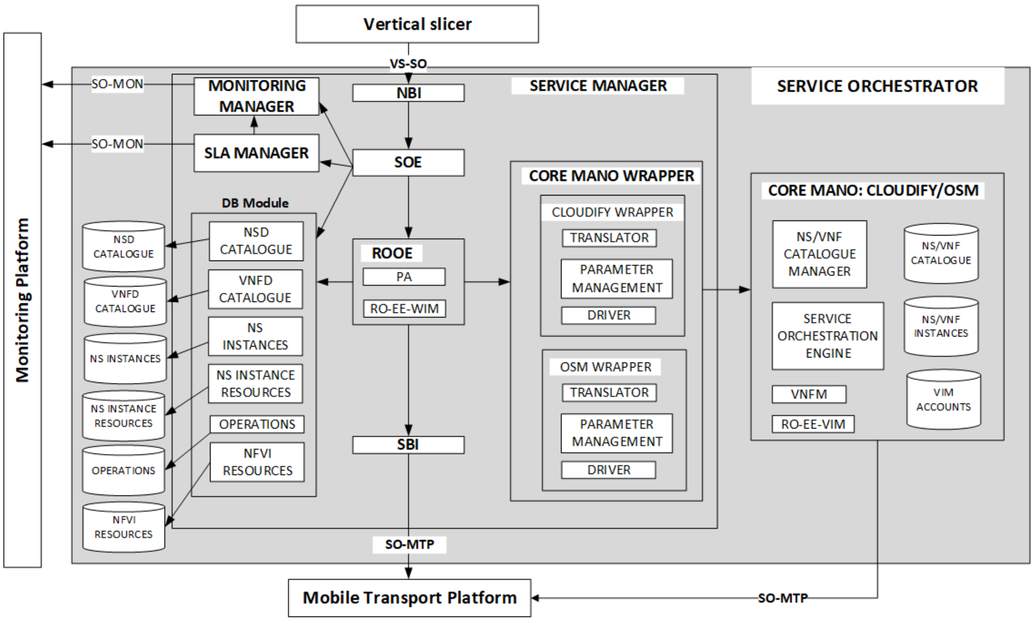


Figure 8.1: Service Orchestrator Architecture

At a high-level, the architecture of the SO follows a modular design organized in four main building blocks. First, the Service Manager (SM), which is the brain element of the SO. This block is in charge of performing the service and resource orchestration logic, dispatching relevant tasks to the other building blocks according to the operational workflow it handles. Second, the Core MANO platform (e.g., OSM or Cloudify) handles the management of computing resources by interacting with the MTP. It also provides the VNF Manager (VNFM) entity to interact with the deployed VNFs. The SM uses the application programming interface (API) offered by each MANO platform through its wrapper system including translation and adaptation functionality. Third, even if the PA is depicted as part of the resource orchestrator of the SO, it is highlighted here as one main building block, since it is an external component accessible through a well-defined REST API. In this way, multiple external algorithmic modules can be easily integrated to the SO. The PA determines the distribution of the multiple VNFs in an NFV-NS among the available NFVI-PoPs and the required allocation of associated networking resources (i.e., logical link) to interconnect those VNFs placed in different NFVI-PoPs during the NFV-NS instantiation process. Some examples of such algorithms adapted to work with the proposed SO architecture can be found in [137], where cluster-based or genetic algorithms are considered. Finally, the databases are in charge of maintaining the state of the SO in terms of service offering (NFV-NS and VNF catalogues), instantiated network services, or network instance resources used among others.

8.2.1 Service Manager Architecture

This subsection presents the internals of the SM, the core module of the SO architecture. This represents a first complete design, which will be further extended in the following chapters. Among the novelties introduced by the SM, there is the capability to interact with a PA and deal with WAN network connections in an automated way when multiple NFVI-PoPs are involved the deployment of an NFV-NS, which is a functionality hitherto with limited support in most of available open source MANO platforms, as highlighted in Chapter 3. Another novelty introduced by the SM is that the available wrapper system allows handling the specificities of the supported open source MANO platform in a way that does not alter the SO internal operational workflows. Hence, the architecture of the SO is flexible enough to integrate and exploit the already available functionalities from well-known

MANO frameworks (e.g., OSM or Cloudify), avoiding the reinvention of the wheel, and focus on the addition of advanced functionalities like the lifecycle management of network services involving multiple administrative domains, as developed in the following chapters. Next, a detailed description of the components of the SM:

- ***NorthBound Interface (NBI)***: this module offers an API towards the VS to support requests for VNF and NFV-NS onboarding, NFV-NS creation, instantiation, status query, scaling and termination operations.
- ***Service Orchestration Engine (SOE)***: this module receives the requests from the NBI and it is in charge of coordinating the different lifecycle management operations. For the NFV-NS query-related operations it interacts with the appropriate databases to fulfil the request and retrieve NFV-NS instance information status. If the request involves resource management (e.g., for NFV-NS instantiation, scaling or termination requests), the SOE delegates the operation to the ROOE module. It also coordinates the operations with the Monitoring Manager and the SLA manager modules to perform NFV-NS monitoring actions allowing SLA fulfilment operations.
- ***Resource Orchestration Engine (ROOE)***: it handles the requests from the SOE to determine required resources and manage the resource allocation process in collaboration with the MTP during the different lifecycle management operations. It interacts with the Placement Algorithm (PA) submodule, the *Core MANO wrapper* module and the Resource Orchestration Execution Engine WIM (RO-EE-WIM) submodule to accomplish the requested requirements. The RO-EE-WIM is in charge to communicate with the SBI to handle network resource management operations at the MTP to intercommunicate VNFs of the same NFV-NS placed in multiple NFVI-PoPs. Note that the counterpart RO-EE-VIM, in charge of handling compute resources, is provided by the used Core MANO platform.
- ***Core MANO wrapper***: it allows the interaction between the SM and the available open source orchestration platform (e.g., Cloudify or OSM). The different wrappers also manage the requests coming from the ROOE to handle compute, storage and network (intra-PoP) resource operations at the MTP.
- ***SouthBound Interface (SBI)***: this submodule is a client of the API offered by the MTP to support the retrieval of NFVI resource availability and to request to the underlying WIM/s the allocation (and de-allocation) of network resources in the logical link abstractions representing the interconnections among available NFVI-PoPs.
- ***Monitoring Manager***: this submodule interacts with the MON platform to configure the performance monitoring jobs measuring the specified VNF metrics in the NFV-NS descriptor (NSD) and to configure the corresponding visualization dashboards. Examples of such metrics are the CPU or RAM consumption or the amount of incoming bytes in an interface of a VNF.
- ***SLA Manager***: this submodule interacts with the MON platform to configure and receive alerts from it in case there is a time kept violation of a measured metric (threshold-based) compromising the fulfilment of SLA conditions as expressed in the NSD. After the reception of an alert, the SLA manager evaluates it and may react by requesting a scaling operation to the NBI as expressed in the NSD to solve the SLA violation.
- ***Database (DB) module***: this module contains several sub-modules to interact with each of the external databases containing the system status information. These external databases are:
  - *NSD catalogue*: stores the information of NFV-NS descriptors.
  - *VNFD catalogue*: stores the information of VNF descriptors.

- *NS Instance Database*: contains the information associated to the instantation of an NFV-NS, like the kind of used NSD, the DF, the IL or the information regarding the different NFV-NS service access points.
- *NS Instance Resource Database*: stores information of resources used by an NFV-NS Instance like VNF placement, IP/MAC addresses of the different deployed VNFs, intra-PoP network resource characteristics (e.g., used VLAN, CIDR, addressPool) and inter-PoP network resource identifiers, between others.
- *Resources Database*: stores information of the status of the different resources (networking, computing and storage) of the underlying NFVI as provided by the MTP.
- *Operations database*: stores information about the status of the different NFV-NS operations carried out at the SO, e.g., instantiate, terminate an NFV-NS.

Finally, after presenting the architecture of the SM, next lines are going to provide some relevant considerations about implementation aspects. The implementation of the SM is based on different ETSI-NFV specifications and recommendations to promote its interoperability thanks to the adoption of open and standardized interfaces. At its NBI, the SM implements a REST server based on the HTTP protocol and JSON language for message encoding. The information models used at the NBI for NSDs, VNFDs, and Network Service lifecycle management actions (e.g., instantiation or query of Network Service instance information) follow the ETSI-NFV IFA specifications 014 [138], 011 [139] and 013 [94], respectively. It is worth noting that extensions in the NSD models have been designed and integrated to describe additional characteristics of the NFV-NSs. An example of this is the information element modelling service constraints or policies to be applied during run-time operation of a deployed NFV-NS to define and react in front of an SLA violation (*autoScalingRule* parameter), which was not defined in the ETSI-NFV IFA014 specification. To interact with the MTP, as explained in previous chapter, the ETSI-NFV IFA specification 005 [128] is used to retrieve resource availability and to allocate/de-allocate resources at the VIM-level. The ETSI-NFV IFA022 [129] report inspired the exchanged messages with the MTP to allocate networking resources interconnecting NFVI-PoPs at the transport network level. The Core MANO wrapper, besides interacting with the offered APIs by the corresponding open source MANO platforms, is also in charge of translating NSDs and VNFDs ETSI-NFV IFA-based descriptors exchanged at the SM NBI during onboarding operation time into the open source MANO specific format. For example, TOSCA-based descriptors are used for Cloudify, while YAML-based descriptors are used for OSM, each of them with their own proprietary information models. Finally, one last remark is that the SM has been designed as a stateless software application, where the status of all the managed entities is maintained and updated through the available external databases. Concurrent requests about active lifecycle actions on new or existing NFV-NSs are managed in an asynchronous manner, with per-lifecycle operation dedicated threads. The feasibility of each requested action is verified based on the current status of the related service, as reported in the database, and refused in case of conflicts. A prototype of the SO is released as open source code under Apache 2.0 licence [140].

8.3 Instantiation of NFV-Network Services

After explaining the architecture of the SO module, next subsection presents the defined workflow to perform the instantiation of an NFV-NS. Then, a performance evaluation considering different kinds of NFV-NSs and its different ILs analyses the time contribution of the different operations involved during the instantiation process.

8.3.1 NFV-NS Instantiation workflow

Figures 8.2 and 8.3 present the operational workflow followed by the proposed SO architecture to perform the instantiation of NFV-NSs. Prior to the instantiation of the network service, the VNFDs and the NSD must be onboarded at the SO platform. The onboarding of an NSD is not allowed if the descriptors of the associated VNFs are not onboarded previously. The description of this onboarding operation has been included at the end of the document in Appendix A.

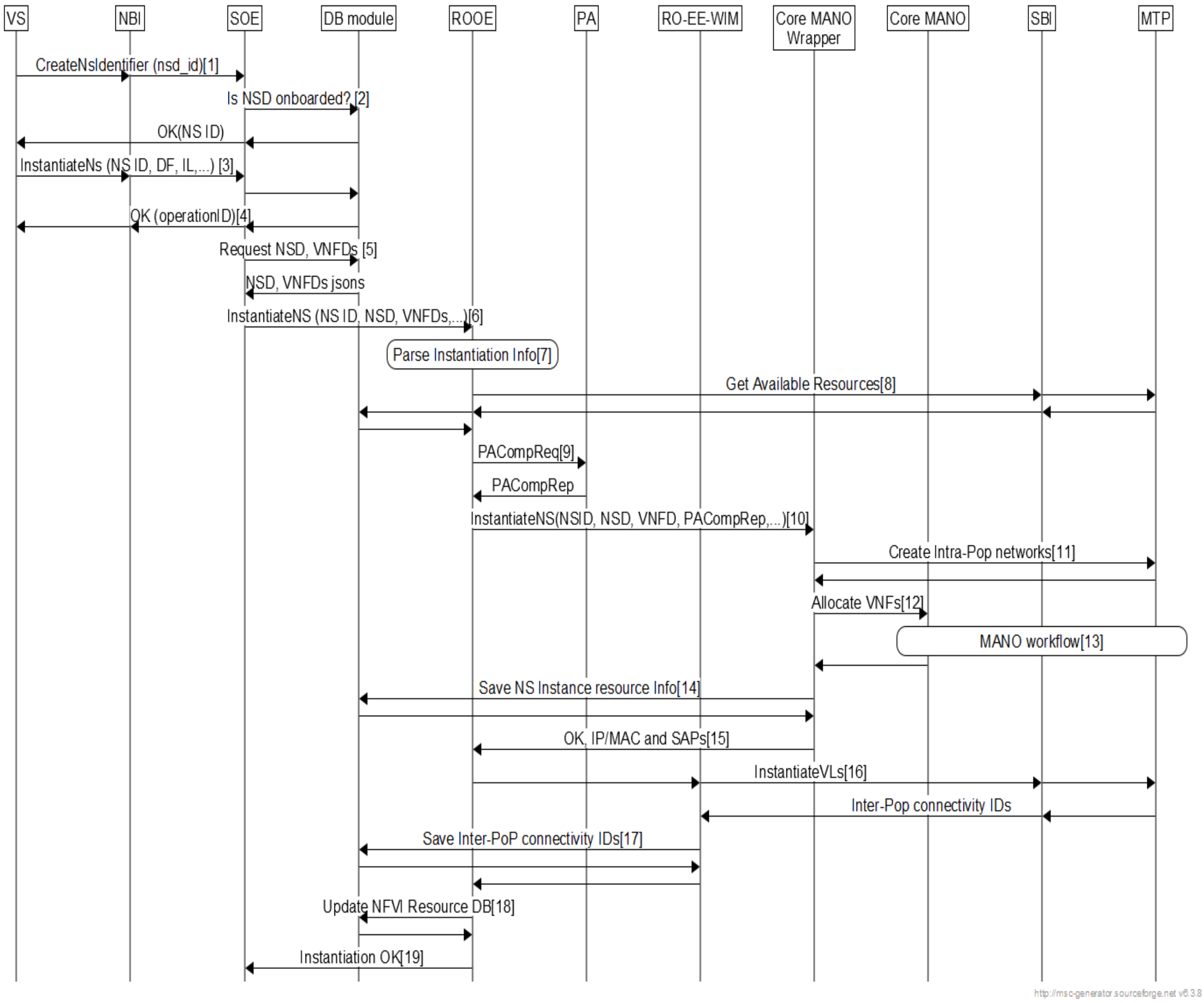


Figure 8.2: Instantiation workflow for NFV-Network Services (Part 1)

First, the Vertical Slicer (VS) module, as part of the process to deploy the network slice instance, requests the instantiation of the NFV-NS according to the procedure specified in ETSI-NFV IFA013 [94]. In this process, first an NFV-NS instance identifier (ID) is created specifying the NSD ID. The SOE module confirms with the NSD Catalogue DB the availability of the requested NSD, and once validated, it returns an NFV-NS instance ID. After receiving the NFV-NS instance ID, the VS sends the *InstantiateNS* request specifying, at least, the received NFV-NS instance ID, the deployment flavour (DF) and the instantiation level (IL). The SOE registers the operation at the Operation DB and while it proceeds with the instantiation process, it returns the operation identifier to the VS (steps 1-4). This operation identifier will be used by the VS to periodically poll the status of the instantiation operation.

The SOE retrieves the NSD and the associated VNFDs from the NSD, VNFD Catalogue DBs, respectively (step 5). Then, the instantiation process continues with the SOE issuing a request to the ROOE including the NS instance ID, the NSD, the VNFDs and the parameters embedded in the *InstantiateNS* request (step 6). The ROOE parses the information in the descriptors based on the parameters of the instantiation request (step 7) and asks the MTP for available aggregated resources through the SBI interface (step 8), which are then saved in the NFVI Resources DB. With both informations, the ROOE composes a request to the PA (step 9), which will determine the distribution of the VNFs across the available NFVI-PoPs and the required logical links interconnecting them while satisfying the requirements expressed in the instantiation request. Based on the PA output, the ROOE sends a request to the Core MANO wrapper, which will manage the allocation and instantiation of computing resources (VMs) and establishing the intra-PoP connectivity (step 10).

The Core MANO wrapper translates the incoming information from the ROOE request to (i) contact the MTP to create the needed intra-PoP networks implementing the VLs at the requested VIMs (step 11), and (ii) to invoke the API procedures of the associated Core MANO platform triggering the instantiation of computing resources implementing the VNFs in the form of virtual machines (VMs) (step 12 and 13). With respect to point (i), it is worth mentioning that the Core MANO wrapper generates a unique ID to name the intra-PoP virtual network. This ID is based on the ID of the VL in the NSD and the generated NFV-NS instance ID. This unique ID allows the MTP to apply the resource allocation policy explained in the previous chapter, which enables and avoids collisions among concurrent deployments of multiple types of NFV-NSs and multiple instances of the same NFV-NS. Regading aspect (ii), the Core MANO wrapper specifies the generated IDs of the recently created intra-Pop virtual networks when using the API procedures of the Core MANO platform, so the VMs to be created are attached to them.

Once finished the operations in the Core MANO platform, the Core MANO wrapper stores the resource instance information in the Network Service Instance Resource DB module (step 14) and sends back to the ROOE the instantiation confirmation along with the IP/MAC addresses of the different ports of the created VNFs and its associated service access points (SAPs)[1] (step 15). Based upon this information and the PA output, the ROOE sends a request to the RO-EE-WIM with the list of connectivity services that need to be established over the selected logical links to interconnect the VNFs allocated at the different NFVI-PoPs (inter-PoP connectivity). This request is processed and redirected from the RO-EE-WIM to the MTP via the SBI (step 16). The MTP, after following the process described in the previous Chapter (see Section 7.4), sends a confirmation identifier associated per each requested connectivity service back to the ROOE, via the SBI and the RO-EE-WIM, which stores the information at the Network Service Instance Resource DB (step 17). The ROOE updates the NFVI Resource DB (step 18) and sends an NFV-NS instantiation confirmation to the SOE (step 19).

During the final steps of the instantiation process (see Figure 8.3), the SOE triggers the configuration of possible monitoring jobs and alerts. The SOE contacts the Monitoring Manager (step 20), who parses the NSD looking for the presence of performance monitoring job information elements (IEs) (step 21). In case there are defined monitoring job IEs, the Monitoring Manager contacts the MON platform to configure them. The configuration of the monitoring jobs at the MON platform consists of two operations, first the creation of *exporters* elements to periodically collect the monitoring information offered by an exporter program running at the VNFs[2] (step 22), and second, the creation of a *dashboard* element to visualize the requested monitored info described in the NSD (step 23). The MON platform returns to the Monitoring Manager the identifiers of the created exporters and dashboards

[1] The service access points are connections points from where an NFV-NS can be accessed

[2] It is supposed that if monitoring information is requested for a VNF, the image used by the VM implementing this VNF in the VIM will count with the appropriate software to expose this information to the MON platform.

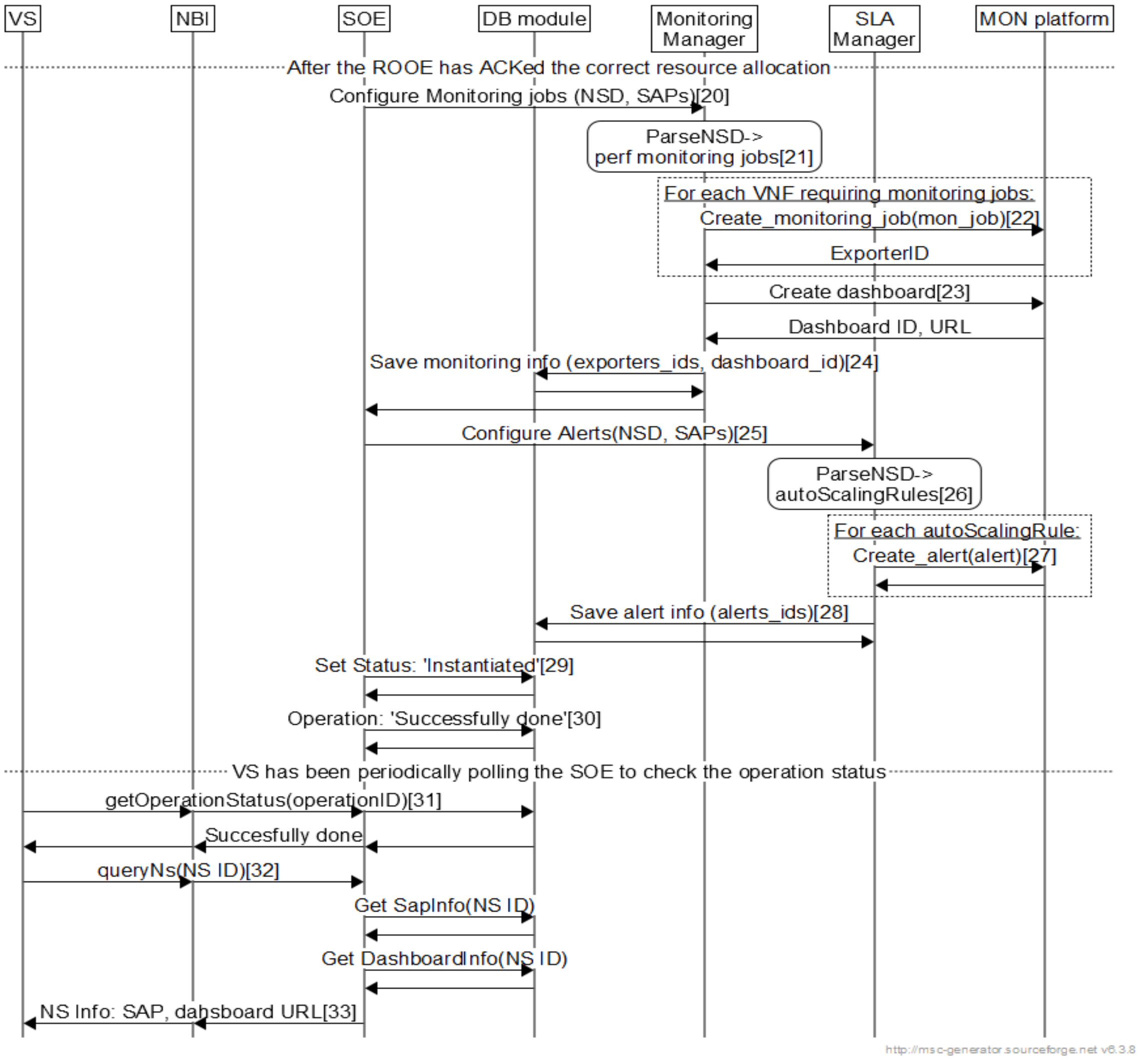


Figure 8.3: Instantiation workflow for NFV-Network Services (Part 2)

and the universal resource locator (URL) of the dashboard as a result of the previous operations. The Monitoring Manager saves the monitoring associated information (exporters and dashboard information) at the NS Instance DB (step 24). After that, the SOE triggers the SLA Manager (step 25), who parses the NSD looking for the presence of *autoScalingRule* IEs (step 26). In case such rules are defined, the SLA Manager contacts the MON platform to configure and send alerts in case of a violation of the criteria expressed in the rule (step 27). The SLA Manager saves the information associated to the configured alert (alerts IDs) in the NS Instance DB (step 28). After that, the SOE updates the DB module setting the status of the NFV-NS as *"Instantiated"* (step 29) and the status of the operation as *"Successfully done"* (step 30). When the VS polls the SO for the operation status and receives a positive confirmation (step 31), the VS requests a *QueryNs* operation (step 32) to get the NFV-NS SAP information and the URL of the dashboard to visualize the defined monitoring information. The SOE gets this information from the NS Instance DB and sends it back to the VS (step 33).

8.3.2 Performance Evaluation of the NFV-NS Instantiation process

This subsection presents a performance evaluation of the E2E orchestration architecture proposed in Chapter 5 in an experimental setup configured mainly in the EXTREME Testbed ®. This experimentation focuses on the experienced service creation time (SCT) and the profiling of the different operations involved in the instantiation process to quantify their impact. In this evaluation, we define the service creation time (SCT) as the elapsed time between the moment the Vertical Slicer (VS) receives the instantiation request until the VS receives the NFV-NS SAP information after a correct instantiation. Different NFV-NSs with different amount of VNFs included in their defined instantiation levels are considered to study the impact of the allocation of different amount of resources in the total experienced SCT.

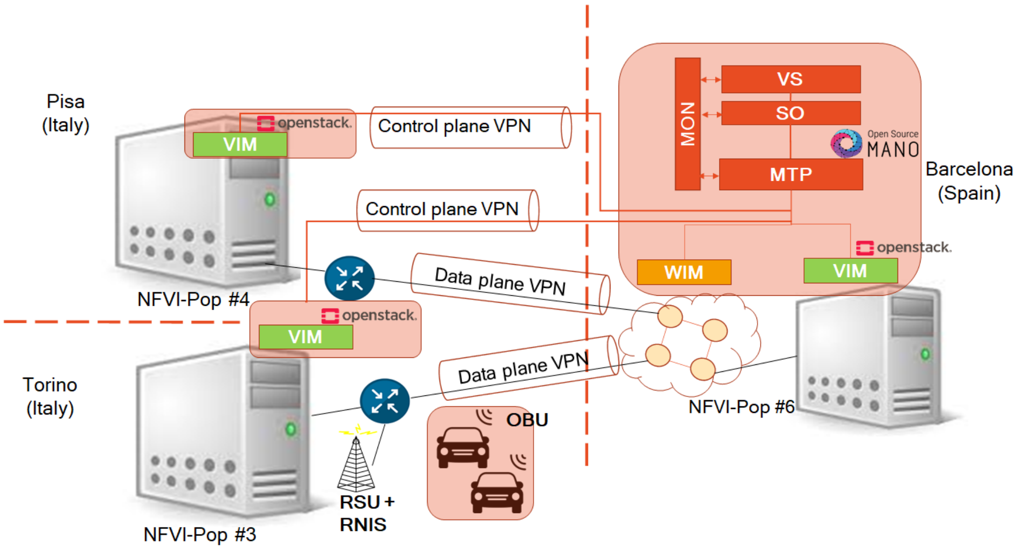


Figure 8.4: Experimental Testbed setup to evaluate the NFV-NS Instantiation process

Figure 8.4 shows the experimental evaluation setup built mainly with the edge facility of the EXTREME Testbed ® in collaboration with other partner institutions of the 5G-TRANSFORMER project [84]. The EXTREME Testbed ® hosts a complete instance of the E2E network service orchestration platform described in Chapter 5, an instance of an Openstack-based VIM and an instance of a WIM controlling a wireless transport network consisting of a ring of four forwarding elements (FEs). The platform uses OSM Release 6 as Core MANO platform. The WIM consists of a simplified version of the IETF ABNO architecture mentioned in Chapter 6 using the COP protocol [53] to communicate with the MTP and interacting with the FEs of the underlying transport network through the WiseHAUL SDN controller [41]. The other two instances of Openstack-based VIMs are placed in Pisa and Torino, Italy. Torino site also hosts an IEEE 802.11p Roadside Unit (RSU) physical network function to provide radio access to vehicles equipped with On-Board Units (OBU) and a Radio Network Information Service (RNIS) providing channel state information to the applications requiring it. The three geographical sites are connected through VPN tunnels encapsulating both control and data traffic.

In this evaluation, we consider two different services that have been defined within the context of the automotive use case of the 5G-TRANSFORMER project [84]. These services are (i) vehicle collision avoidance at intersections and (ii) video streaming, which are examples of, respectively, safety and entertainment services for vehicular users.

The vehicle collision avoidance service exploits the information in the Cooperative Awareness Messages (CAMs) defined by ETSI [141], which are transmitted periodically by vehicles to detect dangerous situations and generate warnings accordingly. Such warnings are encoded following the ETSI Decentralized Environmental Notification Messages (DENMs) [142] and delivered to human drivers, or to an emergency braking system aboard vehicles. Three different types of VNFs compose the vehicle collision avoidance service. First, the CIM (Cooperative Infrastructure Manager), which receives, decodes, and stores CAMs messages sent by the vehicles within the area covered by the vehicle collision avoidance service. Second, the Collision Detector (CD), which queries the CIMs for new CAMs and runs a trajectory-based algorithm (e.g., the one presented in [143]), to detect pairs of vehicles on collision course over the considered area. Third, the DENM Decider, which timely encodes the warning messages and sends them to the vehicles deemed to be on collision course. In this service, the processing latency experienced by the collision detection algorithm is very relevant. For this reason, the vehicle collision avoidance service defines two different ILs, which basically varies the amount of instances of the CD VNF. Based upon the vehicle traffic density, the area assigned to each CD VNF instance will be reduced to keep its processing latency under recommended limits. In the following, we will refer to this service as Extended Virtual Sensing (EVS), since it leverages vehicular communication as a virtual sensor, through which data related to vehicle mobility can be collected.

The video streaming (VStr) service provides multimedia entertainment to vehicle passengers. Two different types of VNFs compose this service. First, the Video Streaming Controller (VStrC), which is able to exploit radio channel link information to select the most suitable bit rate for video streaming to the user. Second, the Video Streaming Server (VStrS), which is a front-end application that contains a video catalogue and edits the media presentation description files according to the suitable bit rate. The video streaming service presents two different ILs. The reduced video streaming service, only including the VStrS, which applies a default constant bit rate for the video streaming and the full-fledged version, counting with the two VNFs, which allows adapting the video encoding to the user channel conditions. Figure 8.5 and Table 8.1 present the structure of the presented network services, its different ILs and the footprint of the associated VNFs.

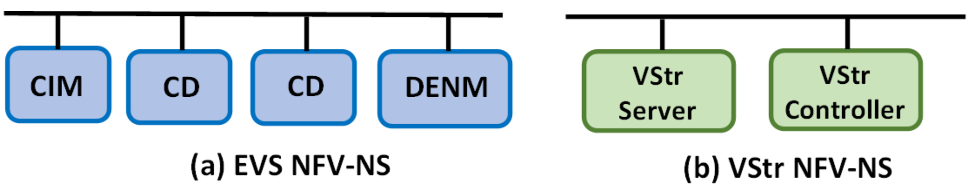


Figure 8.5: Structure of the NFV-NSs under evaluation: (a) EVS NFV-NS, (b) VStr NFV-NS

Next paragraphs present the profiling of the different operations of the SCT and the overall experienced SCT for the presented EVS and video streaming services considering the different ILs available for each NFV-NS, as described in Table 8.1. Due to the big amount of involved operations in the

Table 8.1: Description of NFV-Network Services under evaluation

NFV-NS IL	**Constituent VNFs**	**Resource FootPrint (CPU/RAM(GB)/Storage(GB))**
Reduced VStr	1VStrS VNF	VStrC VNF (2/4/50)
Full-fledged VStr	1VStrS VNF, 1VStrC VNF	VStrS VNF (2/2/50)
EVS (with 1CD)	1 CIM VNF, 1 DENM VNF, 1 CD VNF	CIM VNF (2/4/25) DENM VNF (1/2/25) CD (1/2/25)
EVS (with 2CDs)	1 CIM VNF, 1 DENM VNF, 2 CD VNFs	

instantiation process, the following graphs present them grouped by the following criteria, the module in charge of triggering the operation and the relative impact in the overall experienced SCT. These graphs present the statical distribution of such operations by means of boxplots, which cover the experienced maximum, minimum, average, median, 20^{th} and 80^{th} percentile values of the ten repetitions performed for each experiment. We remark that the same boxplot representation is used in the rest of boxplots graphs presented in the rest of this chapter.

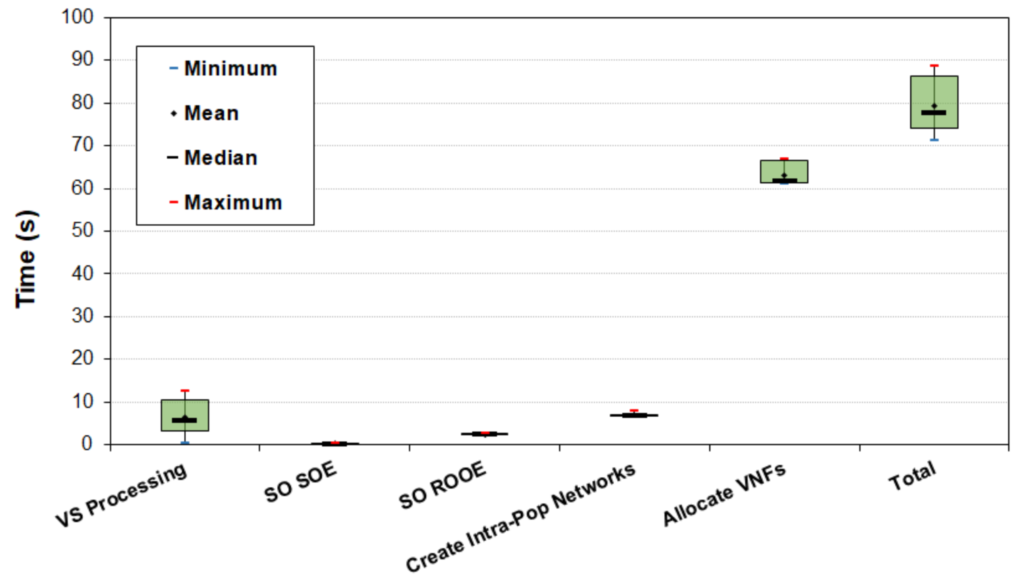


Figure 8.6: Service Creation Time profiling of EVS NFV-NS (with 2CDs)

Figure 8.6 presents the statistical distribution of the various components of the SCT when deploying the EVS service consisting of 2 CD VNFs. The most time consuming operation is the *Allocate VNFs* one, which is roughly 6 times higher than the biggest of the remaining components. This phase accounts for the time it takes to the Core MANO wrapper to interact with OSM platform, which, in turn, deploys the VMs implementing the EVS service by interacting with the MTP and finally the VIM (Openstack). In this case, there are four VMs to deploy, i.e., 1 CIM VM, 1 DENM decider VM, and 2 CD VMs. The following components in order of importance are the creation of the intra-PoP networks (6.986 s on average) and the Vertical Slicer (VS) processing (6.385 s on average). The former accounts for the time it takes to the Core MANO wrapper to interact with the MTP, which, in turn, interacts with the Openstack-based VIM to create the required intra-PoP network of the EVS service. In this case, the EVS service only presents a single VL to communicate all the VNFs. In addition to this, due to the latency constraints expressed in the NSD, the PA decides to allocate all VNFs in the same NFVI-PoP, so this intra-PoP network only needs to be created in a single location. The latter one has a much larger dispersion due to the polling process that the VS performs to determine if the instantiation process at the SO has finished (in addition to the internal processing, which is in the range of ms). The configured polling period is of 20 s. Next, the ROOE processing accounts for the interaction with the MTP to retrieve the topology and available resources (the biggest component with 1.964 s on average), the interaction and placement calculation in the Placement Algorithm (PA) server (522.3 ms), and other components like the creation of the requests and interaction with the MTP to allocate the networking resources in the logical links (LLs). In this case, the impact of this component is almost null since all the VNFs are placed in the same NFVI-PoP. Finally, the processing in the SOE lasts 292.6 ms on average. This operation accounts for the time it takes to the SOE to interact and coordinate the operations at the different entities in the SO module. Among these operations, besides coordinating with the ROOE module or updating DB modules, the SOE, in this case, interacts with the Monitoring Manager and the SLA manager submodules to configure the monitoring jobs checking the CPU load of the CD VNFs and to establish the alerts reporting SLA violations. Actually, this set of interactions with the monitoring platform accounts for a total of the 85% of the SOE time.

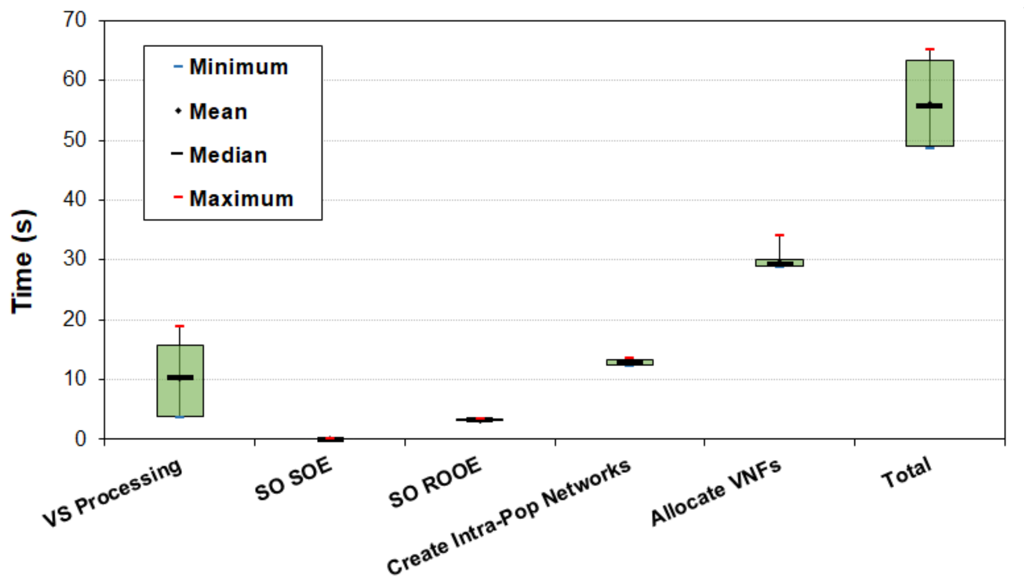


Figure 8.7: Service Creation Time profiling of Full-fledged VStr NFV-NS

The same pattern in terms of relative importance of the components is observed when deploying the other NFV-NSs involved in the scenario under evaluation. For instance, Figure 8.7 presents the SCT of the full-fledged video streaming service. The most remarkable differences compared to the above EVS service are the larger time for intra-PoP network creation (12.862 s on average) and the smaller time for VNF allocation (29.732 s on average). As for the former, this happens even if the service to deploy is simpler (i.e., two VNFs instead of four) because in the previous case, all four VNFs were deployed within the same NFVI-PoP and the same intra-PoP network was used by all of them. Conversely, in this case each VNF is deployed in a different NFVI-PoP. Therefore, two intra-PoP networks must be created, one at each NFVI-PoP, and the SO must interact twice with the MTP, which, in turn, interacts with different VIM instances to create them. Furthermore, since these two intra-PoP networks must be stitched to allow both VNFs to interact as part of the vertical service logic, as explained in Section 7.4 of previous chapter, the allocation time of inter-PoP LL resources is not zero (293.2 ms on average), which also makes the processing time at the ROOE larger (3.271 s vs. 2.501 s on average), despite being a simpler service. This time also includes the interaction with the MTP, which, in turn, interacts with the WIM to configure the required transport network connection.

Note that the time to allocate resources in this LL is relatively low compared to the ones reported in Chapter 6. This is mainly explained due to the less complex structure of the WIM component, the less amount of FEs to configure, and the less number of connections to establish.

As for the latter, the smaller VNF allocation time in this case is precisely due to the fact that only two VNFs, instead of four, have to be deployed by the respective Openstack instances at each NFVI-PoP. Finally, a remark on the SOE component, which in this case presents an average value of 31 ms, almost ten times smaller than the average value for the EVS service (292.6 ms). Differently to the EVS service, the video streaming service does not require the creation of performance monitoring jobs or alerts, so there is no need of interaction with the MON platform. Equivalent observations can be made for the other services under evaluation (i.e., 1-CD EVS and reduced VStr). The most time consuming operations are the ones related to the actual resource allocation performed in interaction with the MTP, namely the VNF allocation process and the time for creating the intra-PoP networks.

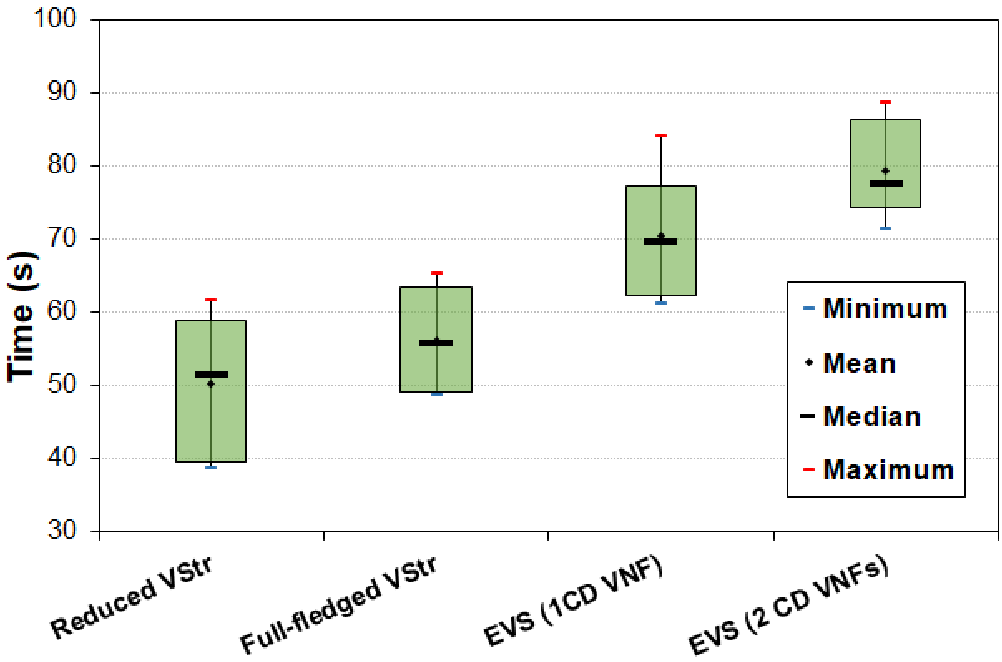


Figure 8.8: Service Creation Time for the different considered NFV-NSs and ILs

To finish the analysis, Figure 8.8 presents the experienced SCT for the different NFV-NSs and ILs presented in Table 8.1, ordered by the total amount of VNFs in the NFV-NS. As it can be observed, the experienced SCT presents a linear trend with the number of VNFs because, as seen in previous Figures, this is the most relevant time component in the SCT. The dispersion is mostly introduced by the polling operation of the Vertical Slicer (VS), as commented previously. This linear trend is also observed in [70], where a comparison between different MANO solutions for 5G Networks in presented. In particular, this reference compares OSM, Cloudify (the two MANO platform currently supported in the E2E network Service Orchestration architecture proposed in this work) and the SONATA MANO platforms. The values reported in the mentioned work for the instantiation time of similar NFV-NSs (at least in terms of amount of VNFs) are aligned with the ones presented in this evaluation, except for the case of using OSM directly. However, a direct comparison between works is difficult to be established because of several reasons: (i) the differences in the deployment of the experimental setup (i.e., geographical distribution of components in our work versus a local deployment [70]), (ii) the amount of operations involved in the instantiation process (e.g., VS processing, resource availability check with MTP to run PA, possible interconnection of VNFs distributed in multiple NFVI-PoPs), which are not done when using directly OSM and Cloudify MANO platforms, and finally, (iii) the hardware capabilities of the tested NFVI.

8.4 SLA Fulfilment of NFV-Network Services

As mentioned in Section 8.1, one important aspect related to the NSO process is the fulfilment of SLAs not only during instantiation time, with a proper decision on the resource allocation process, but also during run-time. Generally, SLAs specify the set of service quality parameters (e.g., maximum CPU load allowed in a given kind of constituent VNF) that service providers have to guarantee to verticals for a running service instance. This implies the support of mechanisms to detect the specified SLA violations and the capability to perform lifecycle management operations aimed at solving such SLA violations, such as a scaling operation changing the structure of the deployed NFV-NS.

In the proposed SO architecture, the mechanisms to enable the automated detection and reaction in front of an SLA violation are handled by the SOE, the Monitoring Manager and the SLA Manager submodules. As explained previously, during the last steps of the NFV-NS instantiation process, the SOE triggers the Monitoring Manager and the SLA Manager to interact with the MON platform to configure the performance monitoring jobs and the alerts described in the NSD. When there is an SLA violation in any of the configured monitoring jobs of interest, the SLA Manager receives an alert from the MON platform, and if the conditions expressed in the *autoScalingRule* field of the NSD are met (e.g., the value of a metric exceeds a threshold value for more than a given time), it issues a *ScaleNS* request to the NBI of the SO with the solution proposed in the NSD. In the proposed SO architecture, this solution considers a change in the IL, which may imply the aggregation or the deletion of multiple VNF instances, which are known as scale out/in operations, respectively. This kind of scaling operations can be performed without service interruption. It is worth mentioning that ETSI specifications also consider the case of scale up/down operation, where computing and storage resources (CPU/RAM/Storage) associated to a VNF may change. However, this kind of operations are not yet supported by current open source MANO platforms and VIMs without a service interruption.

Finally, it is worth highlighting the availability of the mentioned NBI endpoint enabling the request of scaling operations. It provides flexibility to the proposed SO architecture to allow other kind of proactive approaches triggered by other entities, like the VS, aiming at the fulfilment of other more system-oriented SLAs obeying to resource quota agreements established between service providers and verticals.

Next subsection explains the designed workflow followed by the SO module to perform a scale operation on a running NFV-NS instance to react in front of a detected service SLA violation. Then, this workflow is evaluated when considering both scaling out/in operations for the EVS service deployed in the infrastructure presented previously. The section concludes presenting and evaluating a complementary SLA fulfilment action driven by the VS and executed by the SO based on the prioritization of vertical services owned by the same vertical user.

8.4.1 NFV-NS Scaling Workflow

Figures 8.9 and 8.10 present the operational workflow followed by the SO to perform a scaling operation in the instantiated NFV-NS requesting a change in the current IL. In this case, the presented workflow is automatically triggered by the SLA manager upon the detection of an SLA violation as expressed in the NSD. As mentioned previously, it is worth noting that the source of this operation could also be the Vertical Slicer module as a result of applying further SLA operations, as explained next in Section 8.4.3.

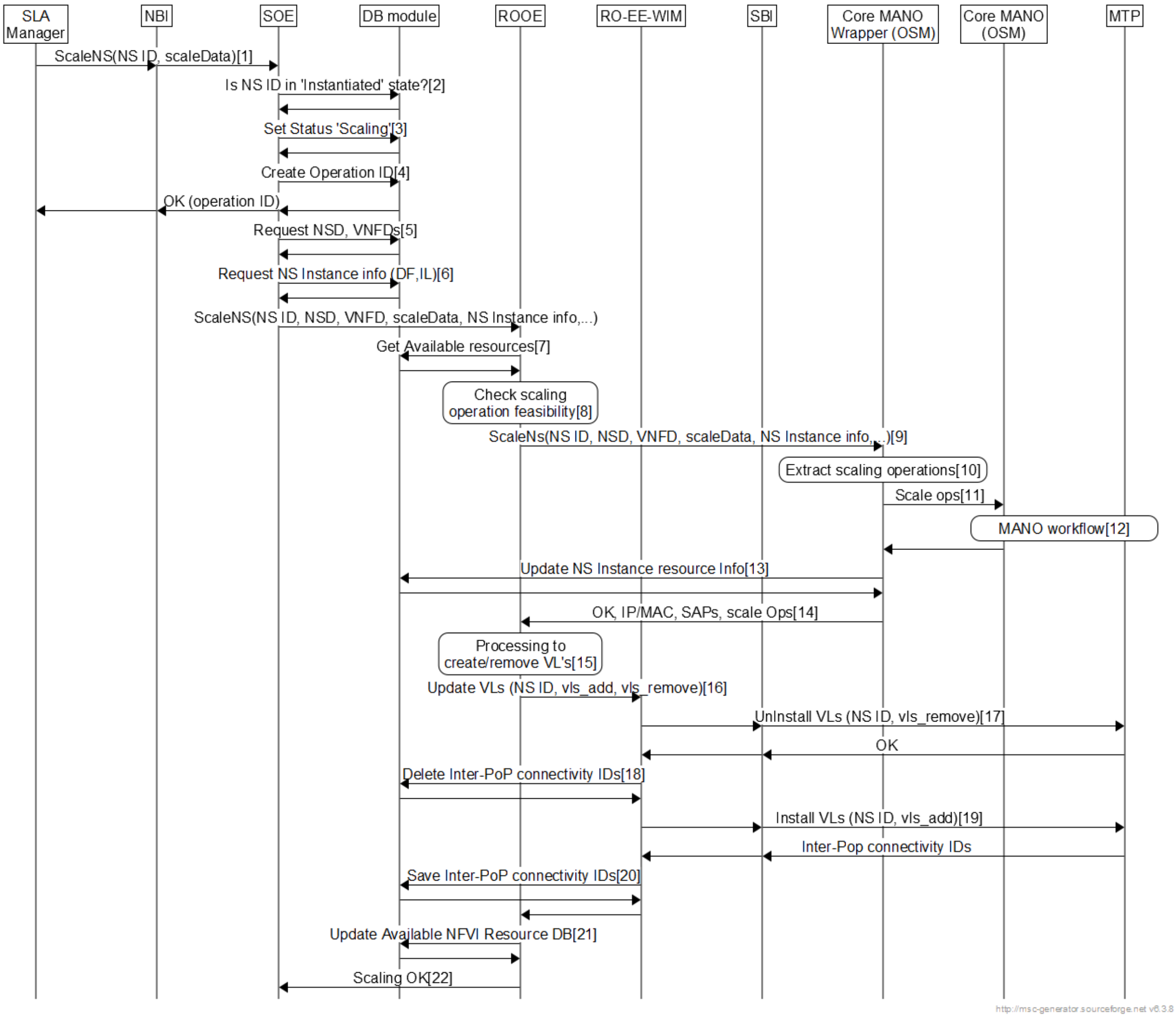


Figure 8.9: Scaling workflow for NFV-Network Services (Part 1)

The SLA manager upon the received alert from the MON platform and after verifying the conditions expressed in the NSD, decides to issue a *ScaleNS* request to the SO reporting the target IL needed by the running NFV-NS instance (step 1). This request arrives to the SOE through the NBI, which checks with the NS Instance DB if the NFV-NS is in "Instantiated" state (step 2), otherwise the request cannot be processed. After a positive check, the SOE sets the NFV-NS status as *"Scaling"*, and progresses with the scaling request while creating, registering and returning an operation identifier to the entity requesting the scale operation, in this case, the SLA manager (steps 3 and 4). This operation identifier will be used by the SLA Manager to check the status of the scaling operation periodically. To progress with the scaling request, the SOE retrieves the NSD and the associated VNFDs from the corresponding catalogue DBs, and the network service instance info (DF, IL) from the NS Instance DB (step 5 and 6).

The scaling operation process continues with the SOE issuing a request to the ROOE including the NS instance ID, the NSD, the VNFDs, the target IL (included in the *ScaleNS* request) and the NS Instance info retrieved in previous step (step 7). The ROOE gets the available resources from the NFVI Resource DB and combining this information with the one received by the SOE checks if the scaling operation is feasible from a resource perspective point of view (step 8). Upon a positive check, the ROOE sends a request to the Core MANO wrapper (step 9), which extracts the scaling operations derived from the change of IL (step 10). Note that a change in the IL may imply simultaneous and combined scale out/in operations on multiple VNF instances. Then, the Core MANO wrapper translates these scaling operations to invoke the API procedures of the associated MANO platform (step 11), which performs the termination (scale in) or the instantiation (scale out) of needed computing resources (step 12). Note that scale in operations are considered to be performed first to release resources at the underlying infrastructure. When finished, the Core MANO wrapper updates the resource information consumed by this NFV-NS instance at the corresponding DB and returns it back to the ROOE (steps 13 and 14). Now, the ROOE determines the VLs to be updated (creation/deletion) based on the distribution of new/removed VNF instances and contacts the RO-EE-WIM with the list of VLs that need to be deleted and the list of VLs that need to be created (steps 15 and 16). Then the RO-EE-WIM, through the SBI and MTP, performs the VL-related operations (steps 17 and 19). As for the case of computing resources, the release of networking resources is performed prior to the allocation of additional networking resources. After a positive confirmation at each step, the RO-EE-WIM updates the information at the Network Service Instance Resource DB (steps 18 and 20). The ROOE updates the NFVI Resource DB and sends confirmation to the SOE (steps 21 and 22).

Once all the scaling operations related to infrastructure resources are finished, the SOE triggers the update of monitoring jobs and alerts (see Figure 8.10). The SOE contacts the Monitoring Manager (step 23), which based on the new VNF distribution and the required monitoring jobs, determines which ones need to be updated (step 24). To perform the update operation (create/delete), the Monitoring Manager contacts the MON platform (steps 25 and 26). This operation considers also the update of the associated dashboard element (step 27). The Monitoring Manager saves the monitoring information at the NS Instance DB (step 28). The same logic is followed to update the alerts (steps 29 to 33)[3]. After that, the SOE updates the DB module setting the status of the NFV-NS as *"instantiated"* (step 34) and the status of the operation as *"successfully done"* (step 35). In this case, the scaling operation finishes when the SLA Manager polls for the operation status and receives a positive confirmation (step 36).

[3]Due to current implementation of the MON platform, it expects the dashboard and alerts to be configured to represent the average value of a measurement taken among the different VNF instances of the same type. For these VNFs, in case the new IL requests to change the number of such instances, there is no need to update the dashboard and the alert info because it will be automatically updated

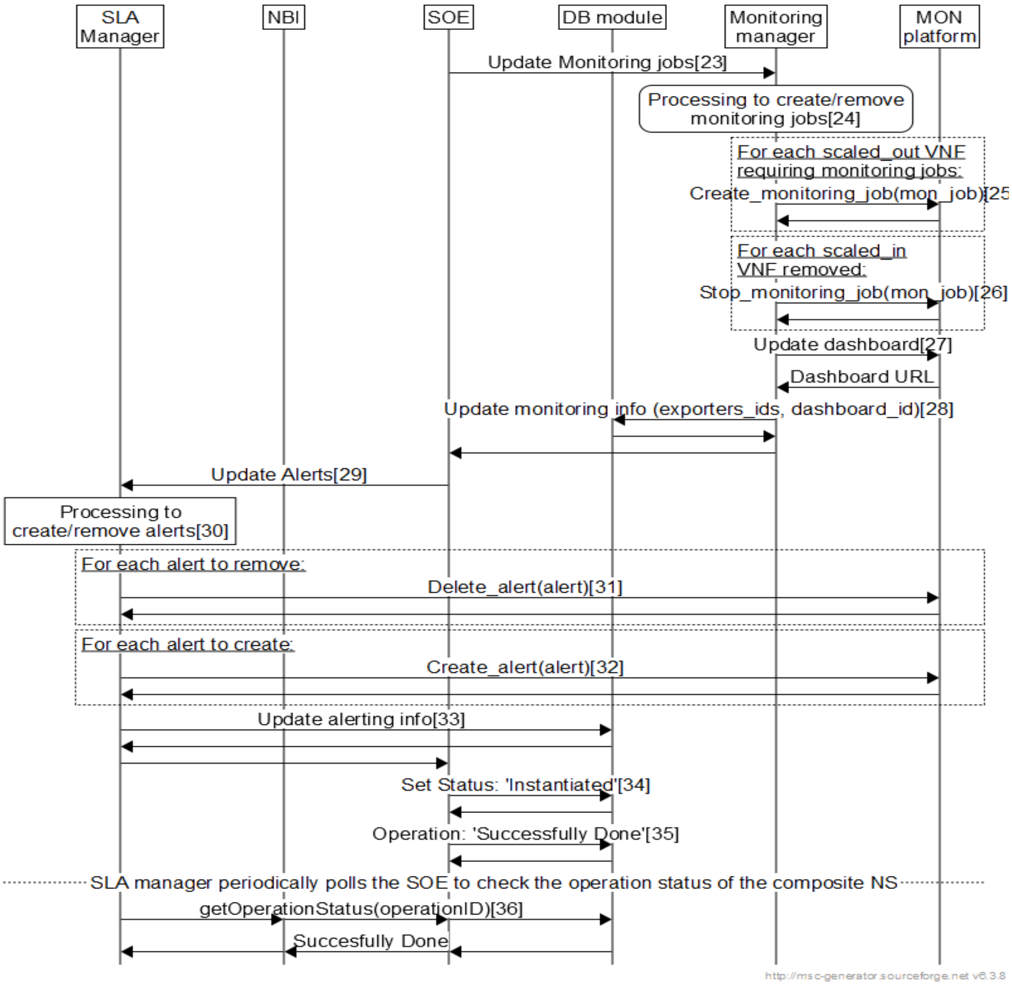
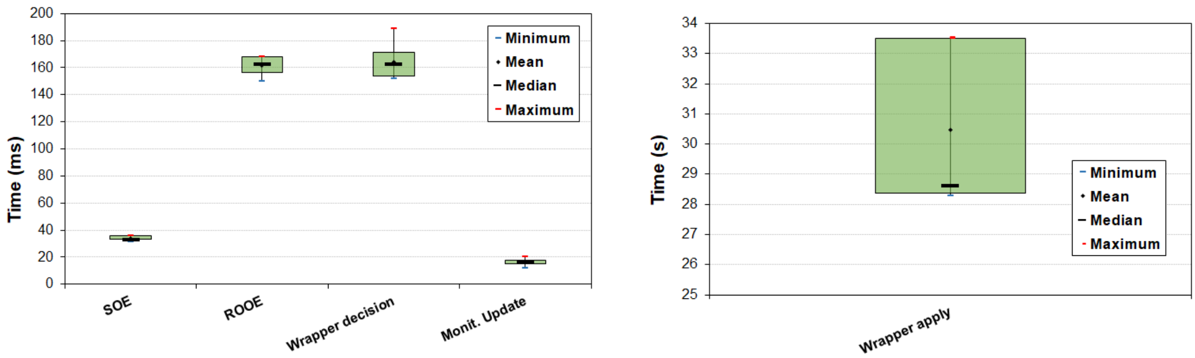


Figure 8.10: Scaling workflow for NFV-Network Services (Part 2)

8.4.2 Performance Evaluation of the NFV-NS Scaling process

This subsection focuses on the performance evaluation of the scaling operation (out/in) of the EVS service presented in Section 8.3.2. As explained previously, the processing latency experienced by the collision detection algorithm is relevant to deliver the collision warning messages on time. This processing latency depends on the amount of vehicle traffic density in the area (i.e., amount of CAM messages registered by the CIM DB VNF) covered by the CD VNF, and presents a correlation with the CPU consumption of the CD VNF. This is the rational behind the defined two ILs for the EVS service, where the number of CD VNF instances changes from one to two (scale out) to handle high traffic densities, as experimentally demonstrated in [10]. Conversely, the EVS service can return to its original IL (with only one CD VNF) to avoid using extra resources in periods of light traffic conditions, where the service demand is lower. The CPU load of the CD VNF is continuously monitored so the SO, coordinated by the SLA Manager, can adapt the IL (and the associated resources) of the EVS service to the most suitable one depending on the traffic conditions. Next paragraphs profile the operations performed at the SO during the scale out/in process of the EVS service.



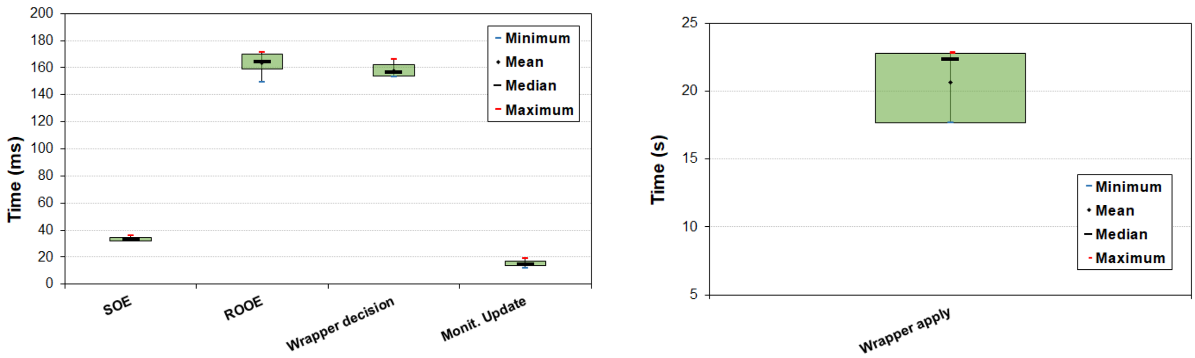
(a) Scale out time (small components)

(b) Scale out time (big component)

Figure 8.11: Scale out breakdown time analysis for the EVS service

Out of the total scale out time (30.862 s on average), the biggest time component corresponds to the deployment of the new instance of the CD VNF VM (see Figure 8.11(b)). In this case, this time is included in the Wrapper apply step (30.481 s on average), which accounts for the time from requesting scaling to the Core MANO wrapper (sent by other building blocks of the SO), the interaction with OSM platform and the MTP, and the interaction with Openstack to deploy the new VM and to attach it to the corresponding intra-PoP network. The remaining components are much smaller (tens or hundreds of ms) (see Figure 8.11(a)). Notice the difference of scales in both figures (seconds vs. milliseconds). First, the SOE processing (33.8 ms on average) measures the time it takes to the SOE to process the scaling request coming from the SLA manager as a consequence of an alert triggered by the MON platform. Database update operations are also included in the SOE processing time. Second, the ROOE processing (161.8 ms on average) accounts for the time it takes to check and prepare all the information to be sent to the wrapper to switch from the initial EVS (the running service) to the scaled-out EVS where the additional CD VNF instance allows to balance the vertical service load, thus, reducing the CPU load and allowing the CD processing latency time to be in the required limits. In this case, there is not a need to allocate network resources in logical links to connect the new CD VNF instance and the rest of VNFs of the service because they all are placed in the same NFVI-PoP. Third, Wrapper decision (163.9 ms on average) accounts for the time to prepare/translate the received scaling request received by the Core MANO wrapper to trigger the associated interaction with the OSM platform, i.e., launch the different scaling out/in operations to pass from the current IL to the targeted IL[4]. Finally, update monitoring (16.3 ms on average) measures the time to update the monitoring jobs, i.e., create the new exporter in case of a scaling out operation to allow the monitoring of the metric/s of interest for the new created CD VNF instance and to make this data accessible through the corresponding interface. Note that this value of interaction with the MON platform is much lower than during the instantiation phase because only the new monitoring job needed to be created, while the dashboard and the alert/s were already configured and did not required an update, as commented in the workflow of Section 8.4.1.

In the same way, the scale in operation was also evaluated, as depicted in Figure 8.12. This is the process through which the SO downscales the resources assigned to the service by changing the scaled-out EVS to the initial IL counting only with one CD VNF instance. This operation is possible when the vehicle traffic density reduces to a level where the average CPU consumption of the CD VNFs is below a threshold capable of fulfilling the requirements of the network service. The scale-in process is equivalent to the above one but for releasing resources (e.g., logical links, VMs, monitoring jobs) instead of creating them. Terminating services and releasing resources (scale in) takes less time (21.051 s on average) than allocating resources (30.862 s on average for scale out), as can be observed in Figures. 8.12(a) and 8.12(b). More specifically, all components described above other than those related to the Core MANO platform (i.e., OSM) are very similar: (i) SOE processing (33.5 ms), (ii) ROE processing (163.7 ms), (iii) Wrap decision (157.9 ms), and (iv) Monitoring update (15.0 ms). However, the main component (Wrap apply), i.e., that devoted to directly releasing the resources by interacting with the MTP (and Openstack) is 10 s smaller (20.676 s vs. 30.481 s on average). Notice also the difference of scale in Figures 8.12(a) and 8.12(b). As a final remark, when comparing our results with the work in [70], the same trends are observed for the difference in time performance between scale out and scale in operations with respect to their analysis. However, their analysis is not as exhaustive as the one presented herein, hence difficulting a direct comparison.



(a) Scale in time (small components)

(b) Scale in time (big component)

Figure 8.12: Scale in breakdown time analysis for the EVS service

8.4.3 Network Service Arbitration

During the network service instantiation process, the resource allocation performed by the SO aims at satisfying the SLAs expressed in the NSD. Additionally, further SLAs can be established between the service provider and the vertical. These SLAs can obey to the definition of service priorities and the resource quota in terms of computing storage, and network resources that the service provider must make available to match the fee paid by the vertical. This is the concept of network service arbitration, as explained in [144]. This process is coordinated at the Vertical Slicer (VS) level of the

[4]Note that passing from one IL to another may imply multiple scaling operations, either out and in

E2E Network Service Orchestration system as part of its network slice management function. More specifically, the *Arbitrator* module of the VS is in charge of making use of service priorities to solve resource allocation conflicts during resource shortage situations derived from the consumption of the assigned quota. Then, the VS, upon the decisions of the Arbitrator module, will instruct the SO the required lifecycle management operations, like termination or scaling of certain NFV-NS instances marked with lower priorities.

In this section, the Network Service Arbitration concept is evaluated in the scenario and with the services described in Section 8.3.2. Initially, two instances of the video streaming (VStr) service are instantiated. These instances present different IL, one instance is the reduced version (1 VStr Server VNF) and the other instance is the full-fledged (2 VNFs: 1 VStr Server and 1 VStr Controller). Different priorities have been assigned to the different ILs of this service. The reduced version of the video streaming service has higher priority than the full-fledged because it consumes less resources and the primary goal of the vertical is to offer a video streaming service, no matter which video quality can be provided. We configure the E2E Network Service Orchestration system so these two video streaming service instances consume most of the available resource budget paid by the vertical using them. At some point in time, the vertical requests the instantiation of the EVS service, which due to its safety nature possesses the highest priority. However, when the Arbitrator module at VS processes the requests, it realises that the vertical has not paid for enough resources to accommodate the three instances of the network services simultaneously. For this reason, the Arbitrator decides to terminate the enough quantity of network service instances with lower priority, in this case the full-fledged VStr service, to make room to the new network service request with higher priority.

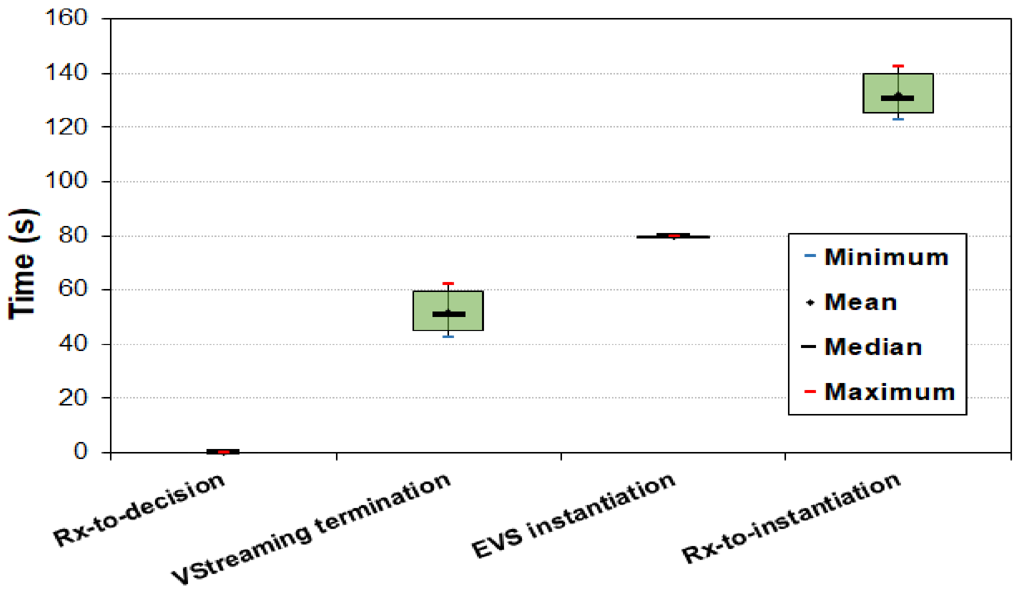


Figure 8.13: Network Service Arbitration breakdown

Figure 8.13 presents the time profiling of the three different steps involved in the Network Service arbitration process. First, the Rx-to-decision time accounts for the time it takes to the Arbitrator to realize that the new service request does not fit in the vertical's resource budget and to decide to terminate the full-fledged video streaming service. This is the smallest component, since it only involves internal processing inside the VS (165 ms on average). Out of this time, 45% (on average) is spent inside the Arbitrator module per se.

Second, video streaming termination accounts for all the interactions inside the E2E network service orchestration stack to deallocate the associated resources, i.e., inter-PoP logical links, VMs, and the intra-PoP networks created in the NFVI-PoPs between others. The description of the workflow associated to the termination operation has been included at the end of the document in Appendix A. These interactions are triggered by a message from the VS to the SO NBI once the decision has been made at the Arbitrator, and it involves the SOE, the core MANO wrapper, the core MANO platform, the ROE inside the SO and its interaction with the MTP. The latter, in turn, interacts with the underlying network infrastructure.Finally, the databases at all layers are also updated. The whole process takes on average 52.020 s.

Third, once enough resources are released for the new service, the EVS instantiation phase accounts for the deployment of the EVS service (80.009 s on average). In this case, the deployed EVS NFV-NS counts with a single instance of the CD VNF. As shown in Figure 8.13, the addition of these three components (Rx-to-instantiation) results in a total average of 132.194 s, from the reception of the high priority service request to its instantiation after solving the resource shortage situation by terminating other low priority service.

8.5 Conclusions

This chapter concludes the first technical part of this work, dealing with the E2E orchestration of network services in single administrative domain scenarios. It presents the key building block of the proposed E2E Network Service Orchestration architecture, the Service Orchestrator (SO), in charge of coordinating the whole NSO process to implement the NFV-NSs realising the network slice instance requests.

The SO architecture considers the interaction and coordination with the other architectural building blocks (VS, MTP, MON) to achieve reduced network service provisioning time and SLA fulfilment capabilities in heterogeneous computing and networking NFVI deployments. The core of the SO is the Service Manager (SM) block, which provides flexibility and modularity to the SO architecture to perform the service orchestration and resource orchestration operations of the NSO process while integrating and exploiting the available functionalities of production-level MANO frameworks without altering SO internal workflows. Additionally, the SM concept enables to put the focus on the addition of required functionalities missing in the used MANO frameworks to perform the desired E2E orchestration, like the automated distribution and interconnection of components of the same NFV-NS distributed in multiple NFVI-PoPs and multiple SLA management schemes. Two of the most relevant NFV-NS lifecycle management operations managed by the SO have been analysed, namely instantiation and scaling. Moreover, they have also been evaluated in a collaborative experimental environment using part of the EXTREME Testbed ® infrastructure when applied to an automotive vertical use case. The description of other lifecycle management operations, namely onboarding and termination, has been included in an Appendix A at the end of the document.

The instantiation case was analysed for different NFV-NSs consisting of different amount of VNFs to check the impact of the amount of VNFs in the service creation time. Deployment times were proportional to the amount of VNFs and in the order of few minutes (1-2 minutes), in line with the 5G-PPP KPI of achieving a service creation time reduction from 90 hours down to 90 minutes. These obtained results show the benefits of the automation capabilities provided by the introduction of NFV and SDN techniques during the NSO process. The profiling of the instantiation process revealed that the most time consuming operations are the ones related to the actual resource allocation process at the NFVI and that the impact in time of the overhead introduced by the SM during the lifecycle management operations is limited, thus confirming the validity of the proposed architectural design.

Regarding the scaling operation, obeying to SLA fulfilment actions, the presented workflow corresponds to the case when the scaling operation is automatically triggered by an alert issued due to the violation of a monitored parameter included in the network service descriptor. The profiling of the scaling operation considered the scale out/in cases, where a new instance of a VNF in an NFV-NS is added/deleted according to the targeted IL, respectively. The time to release resources (scale in) is lower than the one to allocate resources (scale out), being both operations in the range of several tens of seconds (20s-30s). As part of SLA fulfilment operations, this chapter analyses also the network service arbitration case driven by the VS and executed by the SO to solve resource allocation problems by using service prioritization.

These results led us to conclude the validity of the proposed architectural design for the SO and MTP building blocks, their workflows, interfaces and interactions with the other blocks (VS, MON). They will be the basis to extend the E2E network service orchestration architecture towards scenarios considering multiple administrative domains.

Part III

End-to-End Orchestration of Network Services in Multiple Administrative Domains

Chapter 9

Network Service Composition: Deploying and Sharing Composite Network Services in a Single Administrative Domain

The contributions of this chapter have been published in [145]. Previous chapters presented a complete NFV/SDN-based network service orchestration architecture to automate the deployment of NFV-NSs mapping the requested network slice instances (NSIs) in a single administrative domain (AD). This chapter evolves this architecture to support the deployment of composite network services, as a foundation for the deployment of NFV-NSs in multiple ADs. In brief, composite network services are those network services constituted by multiple NFV-NSs.

In particular, Section 9.1 presents the concept of network service composition and why it is useful for next generation mobile networks to cover the needs of verticals. Section 9.2 describes the new architectural components added to the Service Orchestrator (SO) module, its associated functionalities and the proposed workflow to support this concept through the deployment and sharing of composite NFV-NSs implementing the NSI lifecycle management requests issued by the Vertical Slicer (VS) module. The validation and performance evaluation of the designed procedure is included in the next chapter, to provide a benchmark of the deployment of composite network services in a single AD versus a multiple AD scenario, which is one of the relevant aspects covered by this work.

9.1 Network Service Composition Concept

As previously mentioned, next generation mobile networks aim to provide flexibility, dynamicity, and programmability capabilities to satisfy the requirements of vertical industries while making and efficient use of the available computing, storage and networking resources. In the context of network slicing, as presented in Section 2.3, where verticals will be provided a dedicated set of heterogeneous resources fulfilling the characteristics required by the service instances, this can be achieved by dynamically building NSIs with multiple network slice subnet instances (NSSIs). Hence, more complex and tailored offerings can be provided when needed and for the required time to make an efficient resource usage. For instance, a sports vertical can create a fan engagement service with multiple sub-services, whose availability depends while a sport event lasts; or an e-health vertical can activate/deactivate additional sub-services working in collaboration of baseline services upon the detection/solution of an emergency situation.

To be able to realize the aforesaid vision, the service provider, by means of its MANO framework, needs to offer the possibility to chain multiple NFV-NSs and be able to establish the required relations between them. We refer to the possibility of chaining different NFV-NSs to work in a coordinated manner as network service composition (NSC). This is a fundamental mechanism to enable the instantiation of concurrent NSIs in the form of a dynamic puzzle of NSSIs over a shared infrastructure.

9.2 Orchestrating Composite Network Services

In previous chapters, the considered NFV-NSs were an integral unit constituted, normally, by a set of VNFs that work on their own. From now on, we refer to this kind of NFV-NSs as *regular NFV-NS*. However, NFV-NSs can also be defined by a set of NFV-NSs. Those NFV-NSs that are constituted by a set of NFV-NSs are called composite network services, and each of the constituent NFV-NSs are called nested NFV-NSs, as shown in Figure 9.1. This figure also shows that multiple composite NFV-NSs definitions can consider the use of some of the same constituent nested NFV-NSs, enabling the possibility of sharing scenarios between regular and composite NFV-NS instances. Note that the composite network service requires its own descriptor, which follows a similar structure to that of a regular NFV-NS. This descriptor specifies the relation among constituent nested NFV-NSs and it needs to be maintained and onboarded at the orchestration platform to be able to perform the different lifecycle management operations.

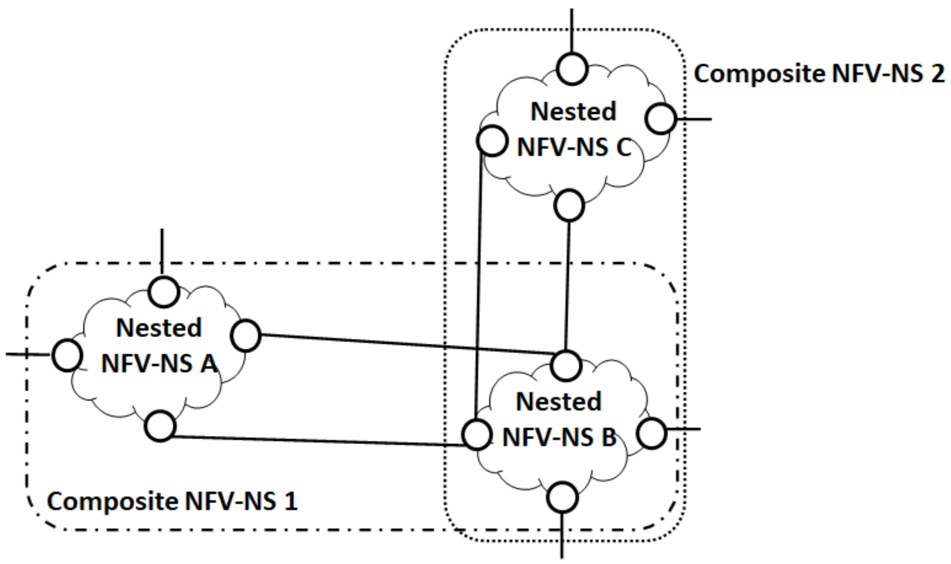


Figure 9.1: Composite network service and nested NFV-NSs examples (adapted from [9])

The deployment of composite network services adds additional complexity to the orchestration process beyond the instantiation of the different nested NFV-NSs. This complexity relates to:

- *The management of the interconnections among the different adjacent nested NFV-NSs within the definition of a composite network service.* During the composite NFV-NS instantiation process, the orchestration system needs to understand how the different nested NFV-NSs are related to apply a proper network resource allocation policy ensuring that the required communication between the VNFs of the different nested NFV-NS can be established. This relation is defined with virtual links (VLs) at the composite NFV-NS descriptor level connecting with the VLs defined at the nested NFV-NS descriptor level so the orchestration system can establish a mapping between the VLs at the different levels (composite, nested) to apply the mentioned appropriate network resource allocation policy enabling the interconnection. This policy consists of a coordinated creation and configuration of virtual networks in the involved virtual infrastructure manager/s to implement such VLs.

- *The use of an already deployed regular NFV-NS to instantiate the composite network service*: The instantiation request may provide a reference to an already deployed regular NFV-NS. In that case, the orchestration process needs to verify if this is a valid reference for the composite NFV-NS (i.e., if the type of the referenced regular NFV-NS is one of the constituent nested NFV-NSs and if it presents a compatible structure (deployment flavour and instantiation level) with the requested composite NFV-NS). Then, it determines and establishes the relation with the remaining parts of the composite NFV-NS to be deployed, so the same resource allocation policy is applied to enable the communication between the required VNFs of the different constituent nested NFV-NSs.

- *The fact that a given regular NFV-NS may be nested (and possibly shared) by several composite NFV-NSs*: The regular NFV-NS needs to register which composite NFV-NS/s is/are using it because it may affect the policy of the orchestration system to handle its lifecycle management. For instance, a regular NFV-NS cannot be terminated without terminating first the instances of associated composite NFV-NSs, otherwise they will be disrupted. Another example is when terminating a composite NFV-NS instance using a reference to a regular NFV-NS. The process needs to be transparent to the regular NFV-NS and the other possible concurrent composite NFV-NS using this regular NFV-NS to avoid possible malfunctioning in the remaining composite and regular NFV-NS instances.

Following subsections describe the enhancements introduced in the architecture of the SO module and the operational workflow allowing the orchestration of composite network services as the ones depicted in Figure 9.1 by taking into account the previous aspects.

9.2.1 Service Orchestrator Architecture Evolution

Figure 9.2 highlights the two new modules added to the baseline architecture of the Service Manager (SM) block of the Service Orchestrator presented in Chapter 8 to handle the lifecycle management of composite network services. The modular architecture of the SM allows adding the advanced capability of orchestrating composite network services while maintaining backwards compatibility with the existing architectural design and the associated instantiation, termination, scaling, and onboarding lifecycle management procedures.

Following the same approach as for the orchestration of regular NFV-NSs, where there are the SOE and the ROOE modules, the Service Orchestration Engine parent (SOEp) and the Composite Resource Orchestration Engine (CROOE) modules have been introduced to handle the logic of the corresponding service and resource orchestration operations for composite network services, respectively.

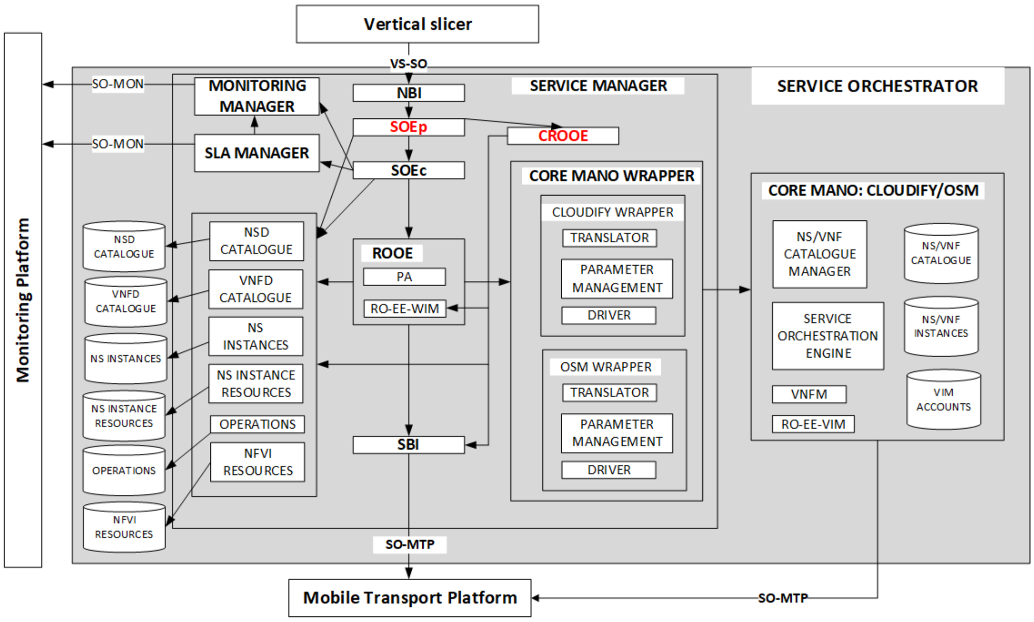


Figure 9.2: Service Orchestrator Architecture enabling NSC

This new architecture introduced a hierarchical approach at the SOE module (parent-child). The SOEp processes all the requests received at the NBI, coordinating the underlying modules to provide the E2E orchestration of composite network services, including the management of references to instantiated regular NFV-NSs in sharing scenarios and the interconnection among the constituent nested NFV-NSs. After validating the format of the request (e.g., checking the compatibility between the requested composite NFV-NS and the possibly provided reference), the SOEp delegates the instantiation of the different nested NFV-NSs of a composite network service to the SOE child (SOEc), the former SOE module presented in Chapter 8. The SOEc follows the same procedure as that of a regular NFV-NS to instantiate a constituent nested NFV-NS. In case of receiving a request for a regular NFV-NS, the SOEp delegates the instantiation process directly to the SOEc since there is not any action associated to composite network services.

The CROOE, triggered by the SOEp, is in charge of analysing the composite NSD to perform the mapping between VLs defined at the composite level and those defined at the nested level to determine how the different constituent nested NFV-NSs are interconnected. This information is ultimately provided to the SOEc module (via SOEp) to apply the coordinated network resource allocation policy at the MTP level when instantiating the different nested NFV-NSs. This information will also be used to determine the inter-nested connections among the VNFs of the different adjacent nested NFV-NSs after instantiated. For that, the CROOE counts with a Link Selection Algorithm (LSA) procedure, which selects the suitable logical links satisfying the requirements expressed in the composite NSD for such interconnections. The CROOE also manages the establishment of these required interconnections at the underlying transport infrastructure, done through the MTP block. As happens with the establishment of connections between VNFs of a regular NFV-NS deployed in multiple NFVI-PoPs, the establishment of such interconnections between nested NFV-NSs deployed in multiple NFVI-PoPs is a distinctive operation of this NSO architecture with respect to state-of-the-art open source MANO approaches, like OSM. To the best of our knowledge, the contemporaneous work in [146] using the SONATA platform is the sole one considering such interconnection operation for the deployment of NFV-NSs using the NSC concept. However, this approach is constrained by the fact that the whole nested NFV-NS has to be deployed in the same NFVI-PoP, which is not a limitation in our proposed architecture.

9.2.2 Composite Network Service Instantiation Workflow

Figure 9.3 presents the operational workflow followed by the proposed architecture to perform the instantiation of composite network services. This workflow focuses on the actions performed by the SO and its different modules, which manages the E2E orchestration of composite network services. Prior to the instantiation of the composite network service, the descriptors of the constituent nested NFV-NSs and the composite network service itself must be onboarded at the SO platform. The onboarding workflow of a nested NFV-NS is the same as of a regular NFV-NS, and it is described in Annex A.1. It is worth mentioning that in the case of composite network services, the composite descriptor is kept in the DBs of the SO, without needing further interaction with the Core MANO platform. The SO is the only entity managing the relation among nested NFV-NSs, being this process transparent to the Core MANO platform. This is the reason why the Core MANO platform does not require to handle composite NFV-NS descriptors. The onboarding of a composite NSD is not allowed if the NSDs of the constituent nested NFV-NSs are not onboarded previously.

First, the Vertical Slicer (VS) module requests the instantiation of the composite network service to the SO according to the procedure specified in ETSI-NFV IFA013 [94] (steps 1-3). After receiving the identifier (ID) of the network service instantiation operation, the VS issues the *InstantiateNS* request, where at least the deployment flavour (DF) and the instantiation level (IL) for the composite network service is specified. The SOEp retrieves the NSD from the NSD catalogue DB and checks that the requested service is a composite network service (steps 4-5). This is indicated by the presence of the parameter *"nestedNsdId"* in the NSD. This parameter, included in the ETSI-NFV IFA014 [138], is the list of constituent nested NFV-NSs used by this composite network service.

Then, the SOEp requests the CROOE to map the connection points among constituent nested NFV-NSs (step 6). The CROOE maps the virtual links (VLs) profiles defined at the composite NSD with the connections points defined at the nested NFV-NS level. In turn, the connections points defined at the nested NFV-NS level are mapped to VLs defined in the nested NSD. In the end, the CROOE maps the VLs defined at the composite level with those defined at the nested level. The CROOE stores this information in the Network Service Instance Resource DB (step 7). More specifically, the rational behind this VL mapping is to base on the ID given to the VL included in the composite NSD to create the virtual networks implementing the *inter-connected* VLs defined in the nested NSD. This *VL mapping* information will be provided to the SOEc during the instantiation of the nested NFV-NS to manage the creation of virtual networks at the NFVI-PoP level implementing the requested VLs. The MTP block can apply the same logic explained in Section 7.4 to determine and assign the appropriate parameters (e.g., VLAN Id, Classless InterDomain Routing (CIDR), addressPool) to create these intra-PoP networks at the involved NFVI-PoPs. This will enable the establishment of connectivity among nested NFV-NSs while avoiding resource collision (e.g., IP addresses) among the elements of this composite NFV-NS deployment and other concurrent NFV-NS instances.

After that, the SOEp starts instantiating each of the nested NFV-NSs sequentially (steps 8-10). In order to do so, it first parses the composite NSD to extract the (DF, IL) pair associated to the nested NFV-NS according to the requested (DF, IL) pair for the composite network service. Then, the SOEp provides this information and the previously extracted VL mapping information to the SOEc, which instantiates the nested NFV-NS following the procedure explained in Chapter 8, as if it were a regular NFV-NS. During this procedure, the ROOE stores resource information related to the nested NFV-NS as a separated entry for each nested NFV-NS in the Network Service Instance Resource DB. This information, among other values, includes the distribution of VNFs among available NFVI-PoPs determined by the PA, the IP and MAC addresses associated to the ports of the VNFs or the assigned CIDR, addressPool values the MTP block determined to create the intra-PoP virtual networks at the NFVI-PoP level, as previously mentioned. After the SOEc finishes the instantiation of a nested NFV-NS, the SOEp updates the entry of the composite network service at the NS Instance DB, with the

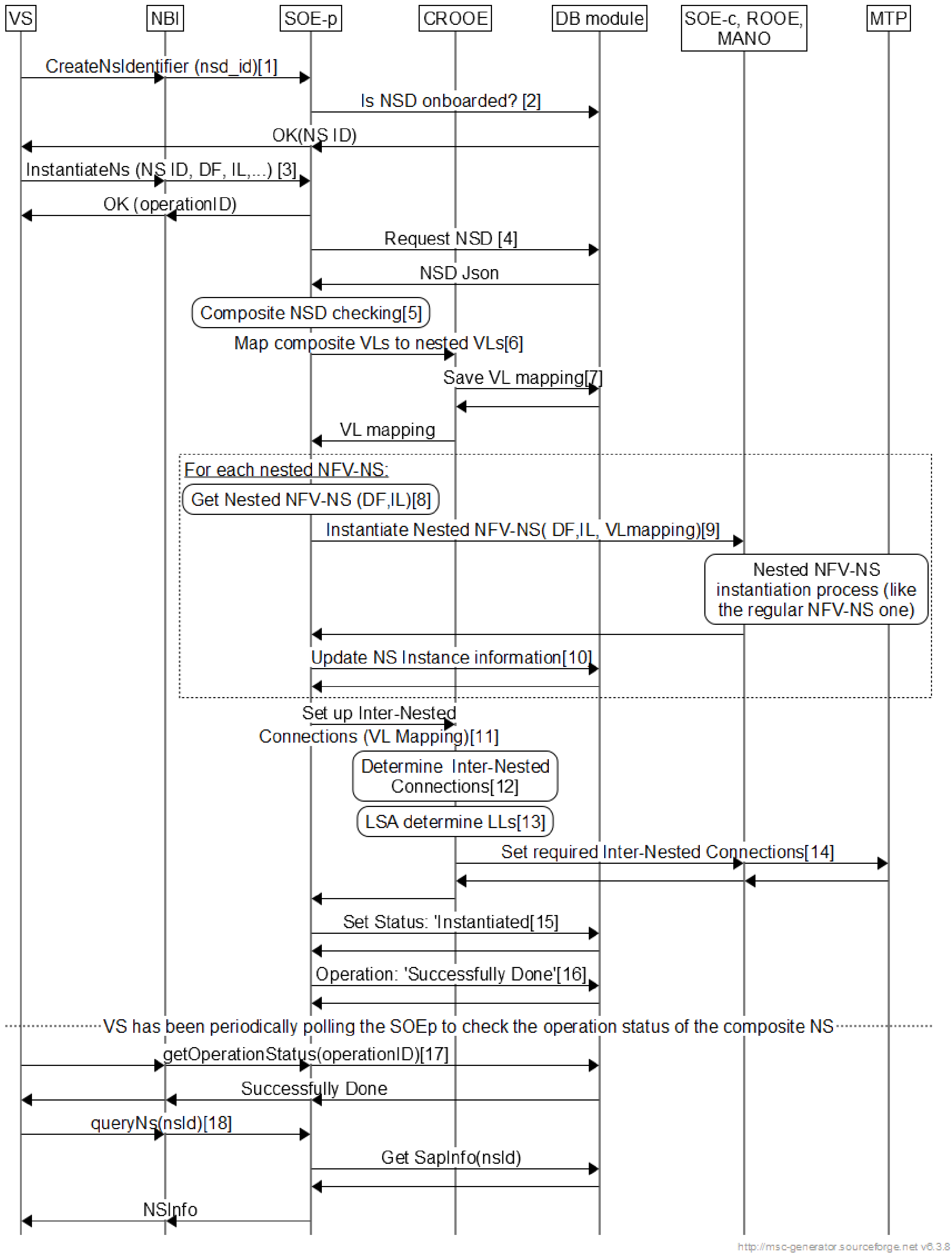


Figure 9.3: Instantiation workflow for composite Network Services

(DF,IL) pair, the used nested NFV-NS descriptor identifier, the instantiation order and the information regarding the service access points (SAP). In this DB entry for the composite network service, there is a list of elements storing the mentioned information for each of the constituent nested NFV-NSs. The instantiation order is a relevant parameter to perform the termination operation and handle properly resource deallocation aspects related to the created intra-PoP virtual networks, as explained in Appendix A.2.2.

Once all the nested NFV-NSs have been instantiated, the SOEp requests the CROOE to set up the inter-nested connections (step 11). Using the VL mapping information determined in step 7 and the information stored in the NS instance resource DB for each of the nested NFV-NSs, the CROOE determines the pairs of VNFs belonging to different nested NFV-NSs, mapped to the same composite VL, and placed in different NFVI-PoPs which need to be connected according to the information included in the NSD (step 12)[1]. The CROOE retrieves the resource information from the MTP to get the status of the Logical Links (LLs). All this information is passed to the Link Selection Algorithm (LSA), which selects the LLs satisfying the requirements of the VLs expressed in the NSD (step 13). From all the LLs advertised by the MTP interconnecting the desired NFVI-PoPs, the LSA selects the one that, satisfying the requirements of latency and bandwidth, presents lower bandwidth availability. After that, the CROOE contacts with the RO-EE-WIM module to establish these connections through the MTP (step 14). The identifiers of the connections are stored by the RO-EE-WIM at the Network Service Instance resource DB.

Finally, the CROOE acknowledges the SOEp, which updates the NS Instance DB declaring the composite network service as instantiated (step 15) and the Operation DB declaring the operation as successfully done (step 16). The VS notices the successful instantiation when polling the operation status and retrieves the information related to the composite network service SAPs (step 17-18).

Previous workflow presented the case when the composite network service is instantiated without referencing an already instantiated regular NFV-NS. But this instantiation process can be done using such reference. The specification of such reference allows NSSI sharing scenarios, where multiple composite NFV-NSs can use the same deployed regular NFV-NS instance as one of its constituent nested NFV-NSs. To allow this possibility in the proposed orchestration architecture, the *shareable* attribute has been introduced in the NSD management procedure. By design, all NSDs corresponding to regular NFV-NSs are stored in the NSD catalogue DB with the *shareable* attribute activated during the onboarding process, so any composite network service can use an instance of an already deployed regular NFV-NS to implement any of its nested NFV-NS if there is compatibility between the requested (DF,IL) pairs[2]. Next paragraphs present further considerations done at the SOEp and CROOE modules during the instantiation process to effectively achieve network service sharing during instantiation time.

A reference to an already instantiated regular NFV-NS is provided by the VS in the *InstantiateNS* request towards the SO (step 3). After validating that the provided reference belongs to an instantiated NFV-NS, the SOEp checks if the provided reference is compatible with the received *InstantiateNS* request in step 3 (step 5). This check validates that the type of referenced regular NFV-NS is within the list of nested NFV-NSs defined for this composite network service, and if the (DF,IL) pair values of the referenced NFV-NS is compatible with the requested (DF, IL) values for the composite network service. Next, the SOEp updates the NS Instance entry of the regular NFV-NS to register the composite network service sharing it. It is worth mentioning that in the proposed workflow nothing precludes

[1]Note that: (i) those VNFs belonging to the same nested NFV-NS and allocated in different NFVI-PoPs were interconnected during the nested NFV-NS instantiation process, and (ii) those VNFs belonging to different nested NFV-NSs and allocated in the same NFVI-PoP do not require further operations to connect them because they are attached to the same virtual network created at the NFVI-PoP to implement the VL as determined during the *VL mapping* process

[2]By current design, NSDs corresponding to composite network services cannot be shared

a regular NFV-NS to be shared among multiple composite network services of the same or different type.

In the *VL mapping* operation between composite VLs and nested VLs (step 6), the CROOE needs to take into account the already created VL instances of the referenced regular NFV-NS. In this case, the allocated nested VLs of the referenced regular NFV-NS, which are part of the inter-nested VLs at the composite level, will be used as a reference to map the rest of shared VLs among the remaining nested NFV-NS. During the instantiation of the *remaining* nested NFV-NSs (step 9), this mapping will allow the MTP creating the required virtual intra-PoP networks with the proper configuration or relying on the configuration of the ones previously created during the instantiation of the referenced regular NFV-NS. As previously mentioned, the proper resource allocation policy of virtual networks avoids resource collisions and enables the establishment of connectivity among VNFs belonging to different nested NFV-NSs, which is later handled by the CROOE module (step 11-14).

Finally, the SOEp, when updating the NS instance DB entry of the composite network service (step 15), records the relation between the composite network service and the referenced regular NFV-NS. The record of this relation is needed to avoid problems when performing lifecycle management operations on the *shared* NFV-NS instances (i.e., regular, composite). For instance, in the proposed implemented design, during a termination operation, a regular NFV-NS cannot be terminated if it is being used/shared by one or more composite network service instances to avoid malfunction in these associated composite instances.

9.3 Conclusions

This chapter has presented the network service composition (NSC) concept, which allows the creation of more complex NFV-NS offerings chaining multiple regular NFV-NSs. This concept is essential to enable the dynamic deployment of NSIs and its associated NSSIs, hence providing better adaptation to the needs of the vertical user. The NSC concept entails multiple challenges that requires the enhancement of the architecture of the SO to be able to manage the deployment and sharing of composite network services in a single AD. The modular design of the SO, and more specifically of the SM, allows adding the orchestration of such type of network services on top of the already available regular lifecycle management operations with a negligible impact in previously defined workflows.

One of the key points in the described orchestration process is the understanding of how the different nested NFV-NSs in the composite network service are connected. It allows coordinating proper resource allocation policies with the MTP block enabling the establishment of the required inter-nested connections so the different parts of the composite network service work as if they were a single entity. This is what we refer to as the *VL mapping* operation in the presented workflow. This mapping process is also relevant in shared deployments to interconnect the instantiated regular NFV-NS with the remaining part of the composite NFV-NS to be deployed.

The fact that a composite network service is constituted by multiple nested NFV-NS is the foundation for the extension of this approach towards deployments involving multiple ADs, where each of these ADs counts with its own management and orchestration platform. This topic is developed in the following chapter.

Chapter 10

Network Service Federation: Deploying and Sharing Composite Network Services in Multiple Administrative Domains

The contributions of this chapter have been published in [147], [148], and [12]. Previous chapter focused on the orchestration of network slice instances and its associated subnet instances using composite NFV-NSs in a single administrative domain (AD). This kind of network service definition, made up of multiple components, allows an enrichment of the orchestration process, opening the door to the lifecycle management of network slices spanning multiple ADs, which will introduce new business models among service providers. In brief, this thesis considers an AD as other organization performing lifecycle management of network slices/services with its own MANO platform. This chapter presents the evolution of the proposed orchestration architecture to achieve the deployment of network slice instances through composite NFV-NSs in multiple ADs.

In particular, Sections 10.1 and 10.2 present the concept of network service federation (NSF) and the new required considerations to be taken into account during the NSO process. Section 10.3

describes the new architectural enhancements introduced at the SO architecture and the proposed workflow to achieve the end-to-end (E2E) deployment of network slice instances in the form of composite NFV-NSs spanning multiple ADs. Appendix A at the end of the document complements the mentioned workflow by describing the termination operation. Section 10.4 validates and provides an experimental performance evaluation of the proposed orchestration architecture considering the deployment of a composite NFV-NS implying NSF (multiple AD scenario) versus the deployment in a single AD presented in previous chapter. This section includes also the description of the application of the proposed NSF procedure in a Proof-of-Concept (PoC) for an eHealth vertical use case. This PoC has been included in the H2020 5G Infrastructure PPP - Trials & Pilots Brochure Number 2 [149], recognising its value and impact for next-generation mobile network empowerment.

10.1 Network Service Federation: Concept and Challenges

The highly heterogeneous environments in which next generation mobile networks are expected to operate entail the management of network scenarios involving different technologies within each of the different network segments (access, transport, core), hence requiring multi-domain orchestration. Moreover, the management and orchestration of network services on top of the available infrastructure is adding a new meaning to the *domain* term. While satisfying the needs of vertical users, some necessary functions to compose networks services and/or the infrastructure resources required to instantiate such functions may be provided by different service providers (SPs), thus requiring federation capabilities. Each of these providers is a different administrative domain (AD).

Examples of the criteria by which a SP (consumer domain) may require resources from another SP (provider domain) may be diverse, including business (e.g., cost) or technical reasons (e.g., network service availability, coverage, lack of resources). Depending on how this process is done between ADs, we distinguish between resource and network service federation (NSF). *Resource federation* can be defined as the process by which a consumer AD requires the control and management of infrastructure resources to a provider AD to deploy a part of a network service (e.g., VNFs) in a NFVI as a Service (NFVIaaS) fashion. On the other hand, *network service federation* (NSF) is the process where the consumer AD requires the deployment of a regular NFV-NS (known also as network service delegation) or a nested NFV-NS (in case of a composite network service) to a provider AD, while the provider domain keeps the full management of its infrastructure resources. Note that in the NSF process, the provider AD performs the lifecycle management of the deployed network service in its NFVI in coordination with the consumer AD, differently to resource federation process, where the provider AD only *"lends"* the NFVI resources, being the consumer AD exclusively in charge of managing these resources to perform the lifecycle procedures.

In this dissertation, the focus is put on the NSF case. Research into this direction may be of interest to SPs to enable the offering of a broader spectrum of services to vertical industries, hence, opening the door to new business models based on the sharing and exchange of network service catalogues that fully realises the vision of 5G and beyond mobile networks. For instance, a vertical industry may set up a business relationship with a single SP to satisfy its requests, and, in turn, this SP may establish agreements with other SPs to extend its portfolio of service offerings, thus increasing its profit acting as a service broker while avoiding the rejection of service deployment requests due to service unavailability.

Several challenges arises to make effective the NSF vision. One is the embodiment of service/business-level agreements or partnership between organizations. Providers prefer to preserve low-level details of their infrastructure (topologies, network configurations) that can expose their business. Hence, the relations among providers will likely to be established following a peer-to-peer fashion, where each relation can be negotiated bilaterally to control the level of exposure. Additionally, information is not available to third party operators as it would be if relations are established following a central exchange point. Another important challenge is the need for a standardized interface to manage the information exchange process handling the multi administrative lifecycle management operations.

10.2 Orchestration of Network Services in Multiple Administrative Domains

The orchestration of network services in multiple ADs is based on the idea of distributing the constituent nested NFV-NSs of a composite network service among those ADs able to implement them while fulfilling the associated requirements. In order to perform such distribution, the orchestration architecture needs to consider, at least, the following aspects to enable such kind of deployments:

1. Physical/logical interconnection schemes among ADs.
2. Procedures to perform lifecycle management operations (i.e., instantiation, termination) among ADs.
3. Procedures to perform NSD/VNFD management operations among ADs.
4. Procedures to exchange NFVI resource availability.
5. Procedures to manage the required inter-domain connectivity among elements of nested NFV-NSs deployed at different ADs.
6. Domain selection to deploy the nested NFV-NSs of a composite network service among the suitable ADs.

The following sections cover the enhancements of the proposed orchestration architecture, mostly in the SO module, and the definition of an operational workflow and an interface to deal with the mentioned aspects.

10.3 Service Orchestrator Architecture Evolution

Figure 10.1 presents the architecture of the SO module highlighting the modules and interfaces of the Service Manager block involved in the NSF procedure.

At a first glimpse, the main difference with respect to the architectural figure presented in Chapter 9 (Figure 9.2), is the presence of a new reference point, the *SO-SO* interface. This interface serves for the purposes of items 2, 3, 4 and 5, which are triggered by the SOEp and CROOE modules, in charge of handling operations related to the deployment of composite NFV-NSs. SOEp extended role deals with item 6. Item 1 refers to the interconnection of the different ADs at the management plane and at the data plane level. The management plane interconnection corresponds to the communication channel between MANO frameworks at different ADs to perform lifecycle management operations, i.e., the channel where the messages exchanged through the *SO-SO* interface flow. The data plane interconnection corresponds to the communication channel among NFVIs of the different ADs through where, the traffic among the different deployed VNFs belonging to different nested NFV-NSs flow.

As mentioned in Chapter 3, the ETSI-NFV IFA028 [9] identified a new reference point in the ETSI NFV architecture called *"Or-Or"* to handle the relationship between orchestrators. This interface has

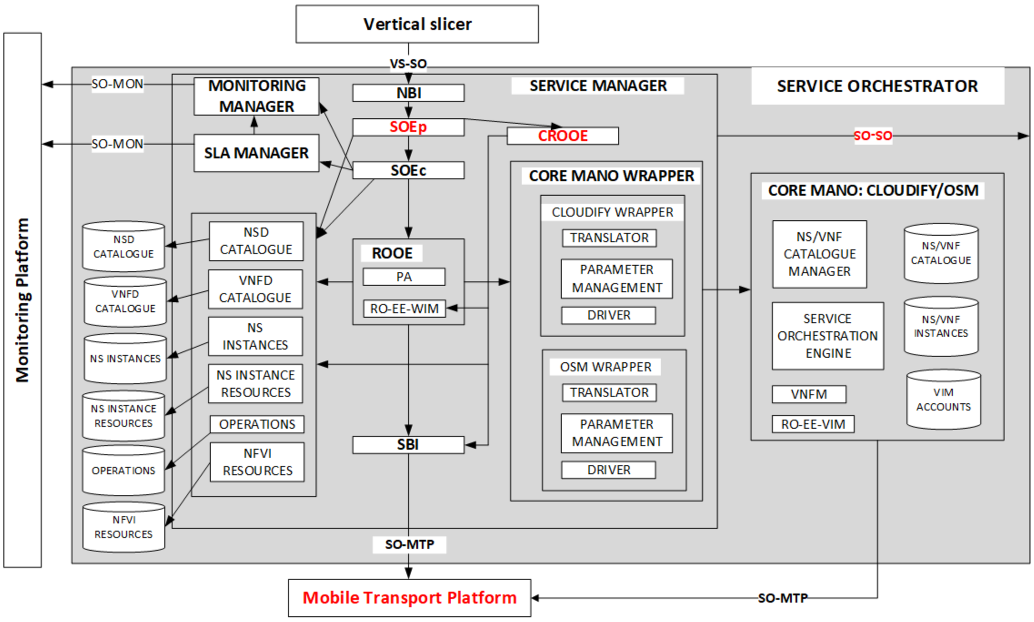


Figure 10.1: Enhanced Service Orchestrator Architecture

been specified in the ETSI-NFV IFA030 [93] document, by considering to use a subset of the procedures defined in ETSI-NFV IFA013 [94] specification to perform the NFV-NSD Management and the NFV-NS lifecycle management operations. However, this approach only considers the problem from the service orchestration perspective, that is covering item 2 and 3. However, resource orchestration logic is fundamental to make the E2E multi-domain orchestration and network slicing a reality in practice. Actually, item 5 is the corner contribution of this chapter ([147], [148]), thus concentrating the focus of following subsections. The proposed instantiation procedure and interface covers the detected gap in most of related work presented in Chapter 3 covering all aspects to achieve an effective multi-AD orchestration of network services.

Before that, additional practical considerations on items 1, 3, 4, and 6 regarding the addition of NSF capabilities to the SO are provided. The proposed orchestration architecture assumes that there has been a previous *"offline"* bilateral agreement between peering ADs where they establish the terms and conditions of its relationship for the provision of services. As commented previously, this is the most likely relation model between AD due to business considerations. In this offline process, service providers, each one having compatible orchestration platforms:

- Establish physically the management plane and the data plane links between ADs (item 1) in a peer-to-peer fashion and update their MTP modules to incorporate information about the peering ADs. The MTP abstracts the peering ADs as a NFVI-PoP presenting a set of characterised logical links (LLs) interconnecting the local NFVI-PoPs with the point at the underlying transport network where the data plane between ADs have been established. Note that only the networking resources at the transport level are relevant in the abstraction of a provider AD and there is no need of knowledge about computing, storage resources in order to preserve privacy between peering domains.

- Establish a pay-per-use agreement so it is considered that peering ADs have NFVI resource availability (storage, computing and networking) to satisfy the instantiation request operation (item 2). Hence there is no need to exchange NFVI availability between peering ADs (item 4)[1].

[1]Another simple approach would be that peering ADs agree on a fix amount of available NFVI resources. Then, each AD takes into account this resource availability at the time to request instantiation operations to other ADs. Note that this approach does not imply the disclosure of topological information, which is a sensitive information for operators.

- Onboard the composite and nested NFV-NSDs in the orchestration platform of the different peering ADs (item 3). In this process, each AD onboards the shared nested and the composite NSDs in the peering ADs. With this procedure, the SO can easily register in which peering AD a nested NFV-NS can be deployed (item 6). Such onboarding operations at the SO are only granted if there is an agreement between ADs. Note that for nested NFV-NS descriptors onboard requests of a provider AD, the consumer AD will not need to verify if the corresponding VNFs are onboarded, as done with a regular NFV-NS, as explained in Appendix A.1.

10.3.1 NSF Instantiation workflow

Figure 10.2 and Figure 10.3 present the operational workflow followed by the proposed E2E network service orchestration architecture to perform the instantiation of a composite network service in multiple ADs, thus requiring NSF. Before starting with the full detailed description of such workflow, it is worth refreshing/mentioning the following aspects.

First, regarding terminology, the consumer domain (CD) is the one receiving the composite network service instantiation request from the Vertical Slicer (VS) and the provider domain (PD) is the domain/s satisfying the nested NFV-NS instantiation requests derived from a CD during the NSF process. Note that the consumer/provider role could be exchanged depending on which SO receives the request from its VS, differently to other approaches like [102], where there is only a central point receiving network service requests.

Second, this workflow is inspired on the top-down approach presented in ETSI-NFV IFA028 [9] report. However, the proposed workflow extends the ETSI-NFV IFA028 workflow by specifying the interface and the required steps to perform the interconnection between the VNFs of different nested NFV-NSs deployed in different ADs.

Third, the proposed workflow has been built on top of the designed workflow for the network service composition (NSC) case developed in Chapter 9, thus ensuring backwards compatibility with all previous lifecycle management operational procedures defined for the SO. For the sake of completeness, all steps have been included in the following description, providing more details on the additional steps with respect to the NSC case.

First, the Vertical Slicer (VS) module requests the instantiation of the composite network service to the SO according to the procedure specified in ETSI-NFV IFA013 [94] (step 1). This is going to be the domain acting as CD, hence coordinating the whole instantiation process. The SOEp retrieves the NSD from the NSD catalogue DB and checks that the requested service is a composite network service (step 2). At this point, the SOEp decomposes the composite network service between nested NFV-NSs to be instantiated locally (consumer domain) and nested NSs to be instantiated in a federated domain (provider domain) (step 3). Then, the SOEp requests the CROOE to map the connection points among constituent nested NFV-NSs (step 4).

After that, the SOEp starts the sequential instantiation of those nested NFV-NSs to be instantiated in the CD (step 5). Based on the (DF,IL) values requested for the composite network service, the SOEp derives the appropriate (DF,IL) values for the nested NFV-NS and delegates this operation to the SOEc, which follows the procedure defined for a regular NFV-NS (step 6), using also the *VL mapping* information derived in step 4. Once, the nested NFV-NS has been instantiated, the SOEp stores its associated instantiation information (DF, IL, instantiation domain, nested NFV-NS descriptor identifier, instantiation order, service access point information) in the NS Instance DB (step 7).

When finishing the instantiation of nested NFV-NSs at the CD, the SOEp launches the instantiation of federated nested NFV-NSs (step 8) contacting the corresponding PD through the *SO-SO*

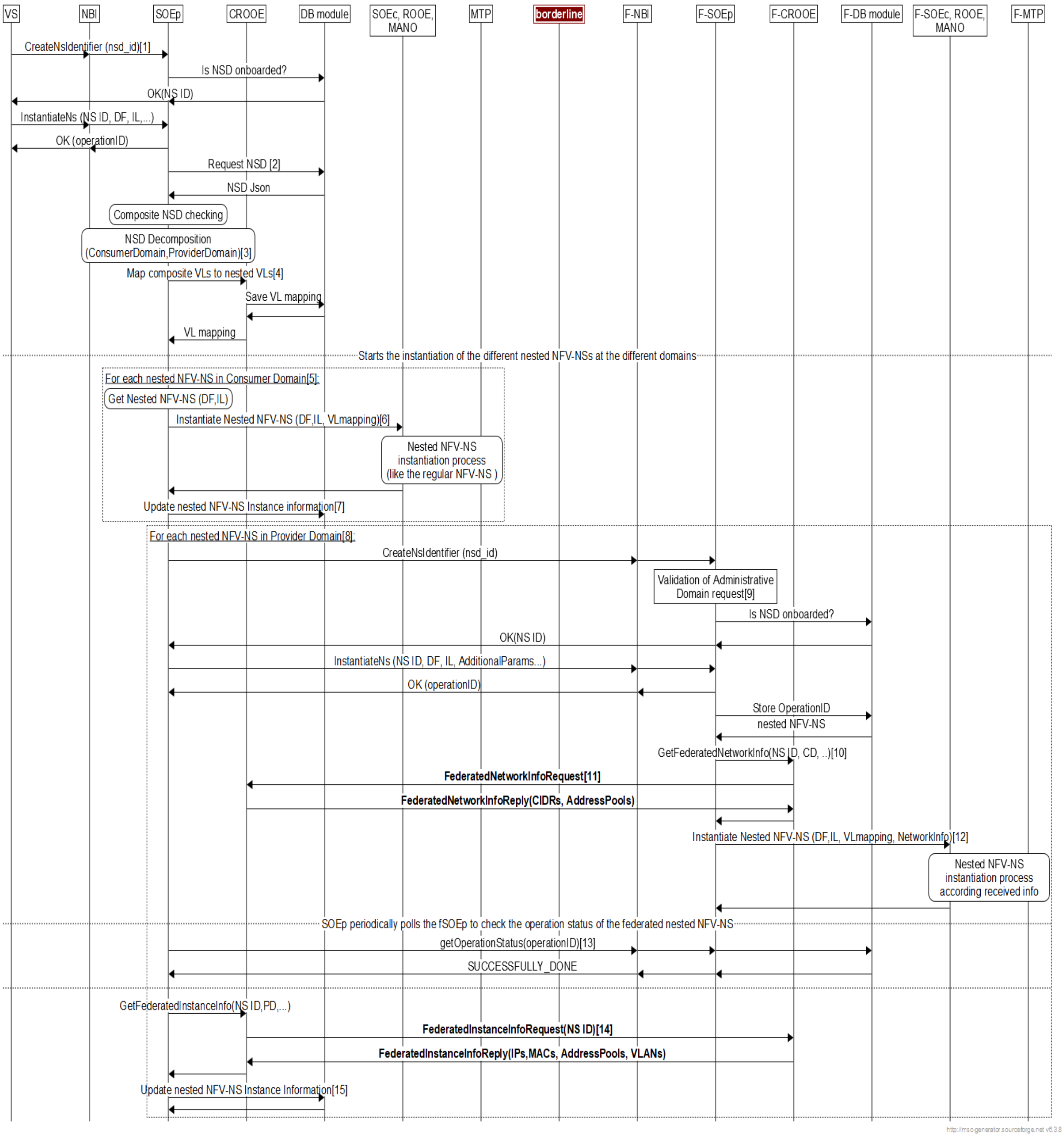


Figure 10.2: Network Service Federation Instantiation workflow (Part 1)

interface following the ETSI-NFV IFA030/013 procedure, as in step 1. That is, the CD SO replicates the procedure of the VS with the required PDs.

At the PD, the NBI first validates that the request comes from an AD allowed to solicit the NSF procedure (step 9). The presence of additional parameters in the instantiation request indicates the provider SOEp that this is a request from a CD and asks the provider CROOE to contact the consumer CROOE to obtain the required information needed to instantiate this nested NFV-NS at the provider domain (step 10). This information exchange between respective CROOE modules is done through the *SO-SO* interface (step 11). Then, the provider SOEp relies on its SOEc to instantiate the requested nested NFV-NS according to the received information from the CD (step 12). While the operation progresses, the CD SO polls periodically the PD SO through the *SO-SO* interface to check the instantiation status of the nested NFV-NS at the PD (step 13). For that, it uses the operation identifier received when issuing the instantiation request to the PD.

Once the CD has confirmed that the nested NFV-NS at the PD has been correctly instantiated, its CROOE module asks its counterpart at the PD about the instantiation details of the nested NFV-NS (step 14). This information is stored at the NS instance DB entry of the composite network service (step 15) and it will be used to setup the inter-nested NFV-NS connections between the different VNFs and coordinate the instantiation of other possible subsequent nested NFV-NSs (at other ADs) while avoiding address collisions between ADs. Note that in this composite DB entry, like for the local deployed nested NFV-NS, the associated instantiation information of the federated nested NFV-NS (DF, IL, instantiation domain, nested NFV-NS descriptor identifier, instantiation order) is also stored.

After the nested NFV-NSs have been instantiated at the different PDs, the SOEp at the CD starts coordinating the process to interconnect the VNFs of the different nested NFV-NSs as expressed in the composite NSD (step 16).

First, it asks its CROOE to determine the interconnections among VNFs of different nested NFV-NSs deployed locally at the multiple NFVI-PoPs under its control according to the information included in the NSD (step 17). Then, it runs the Link Selection Algorithm (LSA) to determine the LLs to serve the required connections (step 18). After that, the CROOE block interacts with the local MTP through the ROOE module to set up these connections (step 19).

Second, the SOEp asks the CROOE to determine the interconnections of the VNFs of different nested NFV-NSs deployed in NFVI-PoPs of PDs with the ones instantiated at the local NFVI-PoPs of the CD according to the information included in the NSD. This is done for each combination of instantiated *"consumer/provider"* nested NFV-NSs (step 20)[2]. For each of these pairs, peering CROOEs (CD-PD) exchange the interconnectivity information (e.g., the addresses of pairs of VNFs at each AD requiring interconnection) through the *SO-SO* interface (step 21). Then, the CROOE at the PD runs the LSA to select the appropriate LLs to allocate these interconnections from PD to CD (step 22) and contacts its MTP through the ROOE to set their segment of the inter-domain inter-nested path, that is, from the different NFVI-PoPs to the point in its transport network where the selected inter administrative domain link is established. The resulting connectivity identifiers are stored in the Network Service Instance Resource DB. After the CROOE at PD validates the establishment of these connections to the CROOE of the CD, the CROOE at the CD completes the operation by running the LSA (step 23) and contacting its MTP (step 24) to set their segment of the inter-domain inter-nested path (from consumer to provider).

Finally, the CROOE at CD acknowledges the SOEp (step 25), which updates the NS Instance DB declaring the composite network service as instantiated (step 26) and the operation DB declaring

[2]Note that the following logic could be applied for each *"provider/provider"* pair of adjacent nested NFV-NSs. Communications between nested NFV-NSs in different PDs is done through the transport network of the CD, as a way to preserve the privacy on the relations set by the CD

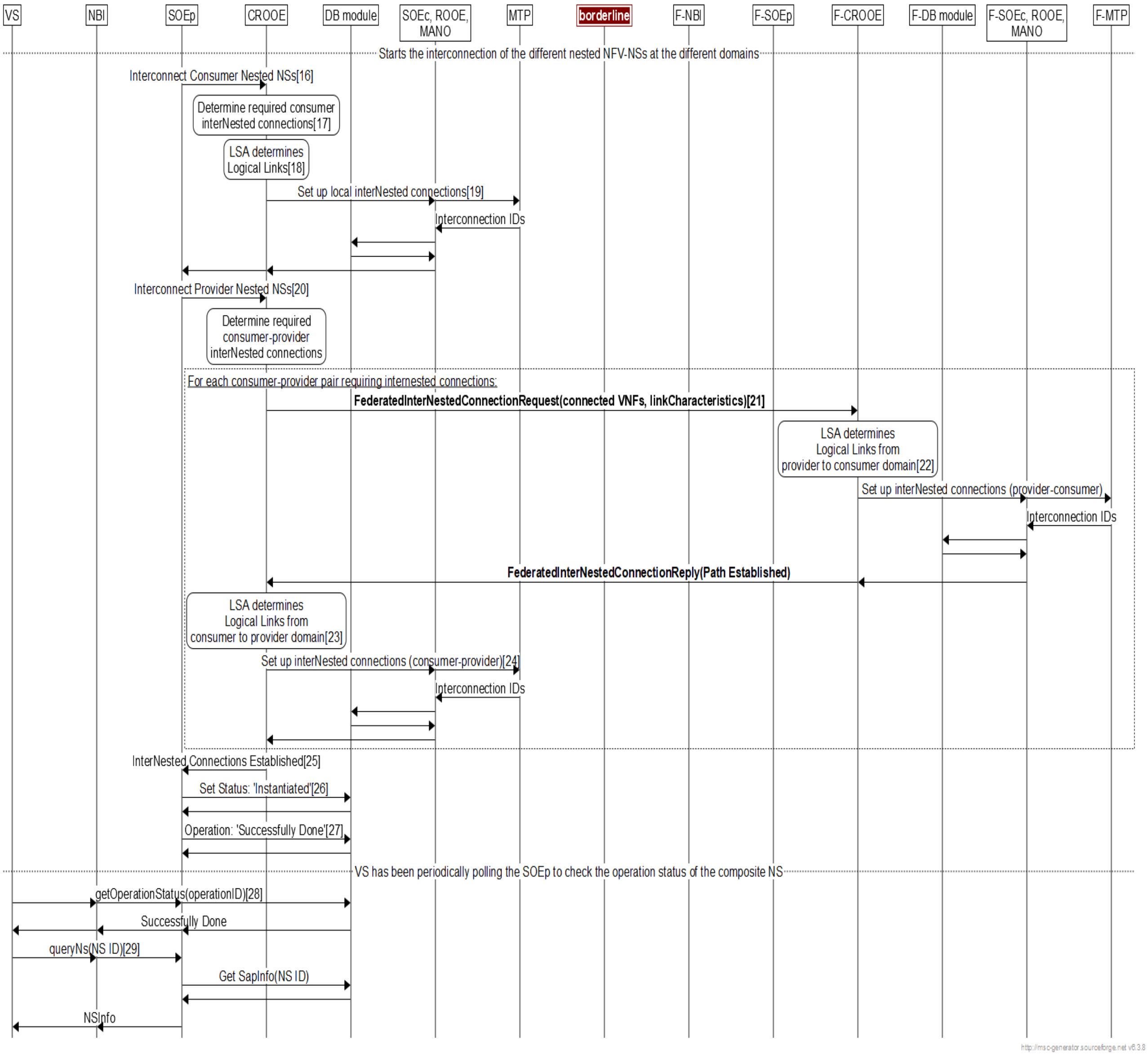


Figure 10.3: Network Service Federation Instantiation workflow (Part 2)

the operation as successfully done (step 27). When polling the operation status, the VS notices the successful instantiation and retrieves the information related to the composite network service access points (steps 28-29).

This instantiation workflow also supports the orchestration of composite network services in multiple ADs sharing an already instantiated regular NFV-NS at the CD, which is the domain driving the orchestration process. In this case, the SOEp starts the multi-AD orchestration process after validating the provided reference by the VS, i.e., checking that the referenced regular NFV-NS exists as an instantiated network service in the CD, and its (DF,IL) values are compatible with the (DF,IL) values received at the *InstantiateNS* request for the composite network service. If all these checks are valid, the SOEp updates the NS Instance DB entry of the regular NFV-NS to register the composite network service sharing it. This operation is done in step 2 of the presented workflow, previous to the decomposition of the NSD between those nested NFV-NSs to be deployed at the CD and those to be deployed at the PDs.

As in the case of network service composition, the key part of the process lies in the VL mapping operation performed in step 4 of the presented workflow. This will later allow the MTP the creation of virtual intra-PoP network resources at the NFVI-PoP level with a proper configuration (i.e., aiming to avoid address collision and allowing a smooth interconnection between VNFs) when the CROOE module at a PD asks for required information to instantiate the nested NFV-NS (step 9). Finally, the SOEp, when declaring the network service as instantiated in the NS Instance DB, records the relation between the composite network service and the reference regular NFV-NS.

In Section A.2.2 of Appendix A included at the end of the document, we can find the workflow describing the NSF termination operation. One relevant aspect of the termination operation is that nested NFV-NSs will be terminated in inverse order to which they were created to avoid potential conflicts in resource deallocation operations.

10.3.2 NSF Interface

The presented workflow in previous subsection shows that multiple messages are exchanged through the *SO-SO* interface depicted in Fig. 10.1 during the multi-AD orchestration process. In particular, the SOEp triggers the operations in steps 8 and 13. These operations are also included in the workflow proposed in ETSI-NFV IFA028 [9] report and, as commented previously, the used interface is defined in the ETSI-NFV IFA030 specification [93] based on the ETSI-NFV IFA013 [94] specification. Specifically, the SOEp at the CD triggers the *Create NS Identifier*, *Instantiate NS* and *Query NS* operations. The proposed approach considers the use of the *additionalParamForNs* parameter of the *InstantiateNS* operation defined in ETSI-NFV IFA013/030 to indicate the PD that the instantiation request comes from a CD. In this parameter of the instantiation request, the CD includes the reference to the composite network service and the VL mapping determined by the CROOE module of the CD in step 4. This information is fundamental to allow organising the information that CROOE modules exchange in later steps of the instantiation process.

Next paragraphs present the messages exchanged between CROOE modules (steps 11, 14, 21) related to the proposed resource orchestration logic aspects allowing the interconnection of the different nested NFV-NSs in different ADs, thus covering the detected gap in the mentioned reference work in Chapter 3. These operations have been highlighted in the proposed instantiation workflow (see Figures 10.2 and 10.3).

10.3.2.1 Federated Network Information

This is the request from the PD to the CD done in step 11, where the PD asks about information to instantiate the requested nested NFV-NS. The Universal Resource Locator (URL) of the request contains the NS identifier (*nsId*) in the CD, so the CD knows to which composite network service instance the PD refers. As mentioned previously, the PD knows this identifier because it is included in the instantiation request triggered by CD in step 8 as part of the *"additionalParamForNs"* parameter included in ETSI-NFV IFA013 specification.

The CD reply provides the information of the CIDR/addressPools[3] that the provider domain has to use/not to use, respectively, for each shared VL between nested NFV-NS in different ADs. The information about shared VLs among nested NFV-NSs is part of the information extracted during the VL mapping process of step 4, and the PD receives this information when receiving the *InstantiateNS* request from the CD, as previously mentioned. Listing 3 presents the detailed format of the request and the reply of this interaction.

Listing 3 Federated Network Information

```
Request: provider domain
URL = "http://consumer_domain:port/5gt/v1/ns/{nsId}/federated-network-info"
body:{
    "nsdId": "NS descriptor identifier in the provider domain"
     }
Reply: consumer domain
body:{
    "networkInfo": {
      "cidr": {
       "internested_VL1": "CIDR_1",
       "internested_VL2": "CIDR_2"},
      "addressPool": {
       "internested_VL1": [Pool_A],
       "internested_VL2": [Pool_B,Pool_C]}
    }
```

10.3.2.2 Federated Instance Information

This is the request from the CD to the PD done in step 14, where the CD asks the PD about the nested NFV-NS resource instantiation parameters. The URL of the query contains the NS identifier (*nsId*) in the PD. The CD knows this identifier as part of the nested NFV-NS instantiation operation launched by the SOEp in step 8.

The PD reply includes the following information for each shared inter-nested VL[4]:

- the IP and MAC addresses of the ports of the VNFs connected to these shared inter-nested VLs to be able to determine and establish the interconnections between VNFs of different nested NFV-NSs.
- the used addressPools in the PD, which will be recorded in the network service Instance DB of the CD (step 15) for subsequent requests of nested NFV-NSs. To avoid potential address collisions, the network resource allocation policy applied at the MTP to create the intra-PoP network implementing the VLs will use different addressPools to those received in the previous *Federated Network Information* request.

[3]Remember that, in the proposed orchestration architecture, the addressPool is an integer value determined by the MTP building block representing a set of IP addresses within a CIDR

[4]Note that all this kind of resource information out of common VLs between nested NFV-NSs is not shared among ADs

- the VLAN identifiers associated to the VLs: in order to preserve the independency of each AD by avoiding the exchange of further messages to coordinate the use of VLAN identifiers among ADs, each AD chooses its own VLAN identifiers. The CD will be in charge of configuring the forwarding elements of its transport network to perform VLAN Id exchange when traffic from/to the provider domain goes to/from the consumer domain.

Listing 4 presents the detailed format of the request and the reply of this interaction.

Listing 4 Federated Instance Information

```
Request: consumer domain
URL = "http://provider_domain:port/5gt/v1/ns/{nsId}/federated-instance-info"
body:{
    "nsdId": "NS descriptor identifier in the provider domain"
}
Reply: provider domain
body:{
    "InstanceInfo": {
        "IP":
          {"internested_VL1":[{"IP_VNF1_VL1","MAC_VNF1_VL1"},
                              {"IP_VNF2_VL1","MAC_VNF2_VL1"}],
           "internested_VL2":[{"IP_VNF1_VL2","MAC_VNF1_VL2"},
                               {"IP_VNF3_VL2","MAC_VNF3_VL2}"]},
        "addressPool":
              {"internested_VL1":[PoolX],
               "internested_VL2":[PoolY, PoolZ]},
        "vlanId":
              {"internested_VL1": vlanId_A,
               "internested_VL2": vlanId_B},
     }
 }
```

10.3.2.3 Federated Internested Connections

Request from the CD to the PD done in step 21 listing the pairs of connected VNFs between respective nested NFV-NSs and the characteristics of each of the shared inter-nested VLs in terms of bandwidth and latency as included in the composite NSD. The URL of the query contains the NS identifier (*nsId*) in the PD and the body request contains the NSD identifier in the PD, so it can verify that the reference in the URL corresponds to an instance using the requested NSD.

The provider domain replies "OK" or "KO" depending on the result of the path establishment operation. Listing 5 presents the detailed format of the request and the reply of this interaction.

Listing 5 Federated Internested Connections

```
Request: consumer domain
URL = "http://provider_domain:port/5gt/v1/ns/{nsId}/fed-internested-connections"
body:{
 "nsdId": "NS descriptor identifier in the provider domain"
 "connectedVNFs": {
     "internested_VL1": [[(IP_VNF1p, MAC_VNF1p), (IP_VNF1c, MAC_VNF1c)],...],
     "internested_VL2": [[(IP_VNF2p, MAC_VNF2p), (IP_VNF2c,vMAC_VNF2c)],...]},
 "linkChar":{
      "internested_VL1": {
          "latency": latency_vl1,
          "bw": bandwidth_vl1},
      "internested_VL2": {
          "latency": latency_vl2,
          "bw": bandwidth_vl2}}
}
Reply: provider domain
body:{
 "pathEstablishment": {"OK"/"KO"}
}
```

10.4 Experimental Deployment of Composite Network Services using NSF

This section presents the validation and performance evaluation of the proposed orchestration architecture and the NSF instantiation process presented in previous subsections in an experimental multi-administrative/multi-technology setup configured between the EXTREME and ADRENALINE testbeds described in Chapter 5. This experimentation considers the service creation time (SCT) of a composite network service under different deployment schemes, using the NSC orchestration process in a single AD presented in previous Chapter 9 as a benchmark to evaluate the NSF process.

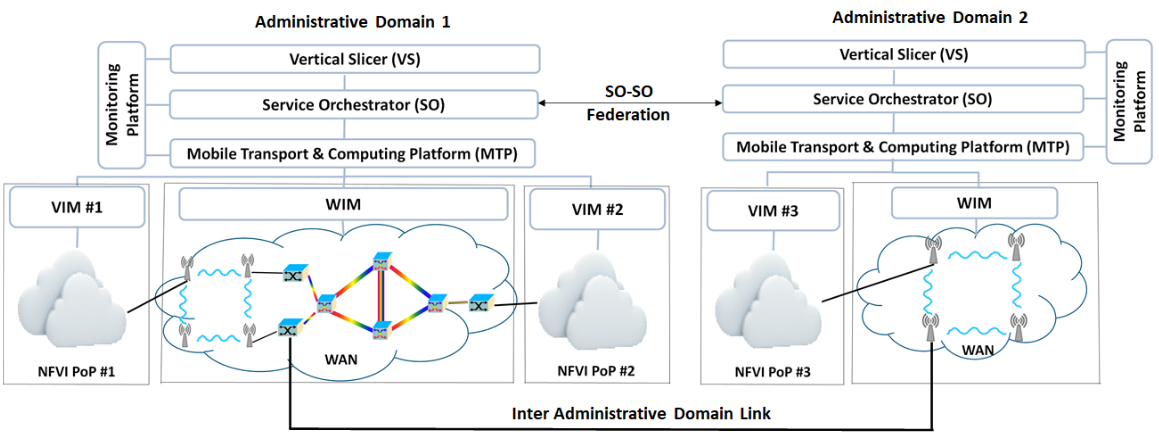


Figure 10.4: Experimental Multi-administrative/multi-technology domain orchestration platform setup

10.4.1 Experimental Setup Description

Figure 10.4 shows the experimental multi-administrative/multi-technology setup, in which, each AD presents an instance of the whole orchestration platform. At the control plane level, ADs communicate each other through the *SO-SO* interface, which traverses over a L3 Virtual Private Network (VPN) connection. At the data plane level, the inter administrative domain links interconnect the transport networks of peering ADs. In this setup, there is a single inter AD link, implemented via a L2 VPN connection.

The SO module of both orchestration platforms uses Open Source MANO (OSM) Release 6 as Core MANO platform. AD1 presents two NFVI-PoPs. Each one is managed by its own Virtual Infrastructure Manager (VIM) (Openstack all-in-one Devstack Queens release). These NFVI-PoPs are connected through the multi-technology transport network described in Chapter 6. As a quick recap, this transport network presents, at the control plane level, a hierarchy of SDN controllers following the ABNO controller architecture [47] to perform the E2E SDN transport orchestration of network connectivities. The element on top of this hierarchy acts as the Wide Area Infrastructure Manager (WIM), hence interacting with the MTP. At the data plane level, this transport network combines a wireless mmWave/WiFi (IEEE 802.11ad/802.11ac) network residing in the EXTREME Testbed ® with a multi-layer network combining packet and optical (based on wavelength division multiplexing) switching technologies placed in the ADRENALINE Testbed ®. AD2 has a single NFVI-PoP, which is managed by its own VIM (Openstack all-in-one Devstack Queens release). At AD2, the transport network features a single wireless WiFi (IEEE 802.11ac) network with four forwarding elements as presented in Figure 10.4. This transport network is controlled by an SDN controller using the same interface (COP protocol [53]) as the SDN controller of AD1 to handle connection requests coming from the MTP module.

10.4.2 Performance evaluation

In order to characterise the NSF procedure and compare the orchestration of composite services under a single AD versus a multi-AD scenario, the SCT has been chosen as the reference measurement to be evaluated in the experimental setup of Figure 10.4. Since the focus in this chapter is on the NSF procedure followed by the SO, the following evaluation considers the SCT as the elapsed time between the SO at the consumer domain receives the network service instantiation request until this SO declares the network service as successfully instantiated.

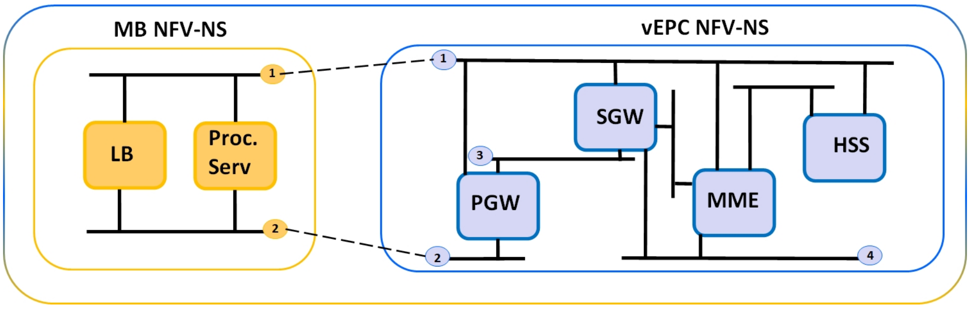


Figure 10.5: Composite network service under evaluation. Dashed lines show the inter-nested network service connections and numbers represent the service access points

The composite network service under evaluation is shown in Figure 10.5 and it has been defined within the scope of the eHealth use case of the 5G-TRANSFORMER project [84]. It consists of two nested NFV-NSs. The first nested NFV-NS emulates a Virtualised Evolved Packet Core (vEPC), and consists of four VNFs, namely MME, HSS, SGW and PGW. The second nested NFV-NS is a monitoring backend (MB) NS, and consists of two VNFs, namely a load balancer (LB) and a processing server.

Figure 10.6 shows the statistical behaviour of the SCT experienced when considering different deployment schemes given the presented composite network service and the experimental setup, as explained in Table 10.1. These experiments cover a wide range of deployment schemes to provide the reader with a broader context to better understand the following analysis. The covered deployment

Table 10.1: Network Service deployment schemes under evaluation

Component	Description	Label in Figure 10.6
Single Nested NFV-NS MB	MB NFV-NS in NFVI-PoP#2 of AD1	Nested-MB
Single Nested NFV-NS vEPC	vEPC NFV-NS in NFVI-PoP#2 of AD1	Nested-vEPC
Composite NS Single-Pop	Composite NFV-NS, both Nested NFV-NS in NFVI-PoP#1 of AD1	Compo S-Pop
Composite NS Multi-Pop	Composite NFV-NS, vEPC NFV-NS in NFVI-PoP#2 and MB NFV-NS in NFVI-PoP#1 of AD1	Compo M-Pop
Federated NS	Composite NFV-NS, vEPC NFV-NS in NFVI-PoP#2 of AD1 and MB NFV-NS in NFVI-PoP#3 of AD2	Federation

schemes include simple ones: instantiation of a regular NFV-NS; intermediate: composite NFV-NS (both single and multi-NFVI-PoP in a single AD); and complex: federation, involving multiple ADs. The box stretches from the 20^{th} and 80^{th} percentiles, including the mean and the median values. The whiskers represent the maximum and minimum values. Each experiment has been repeated ten times.

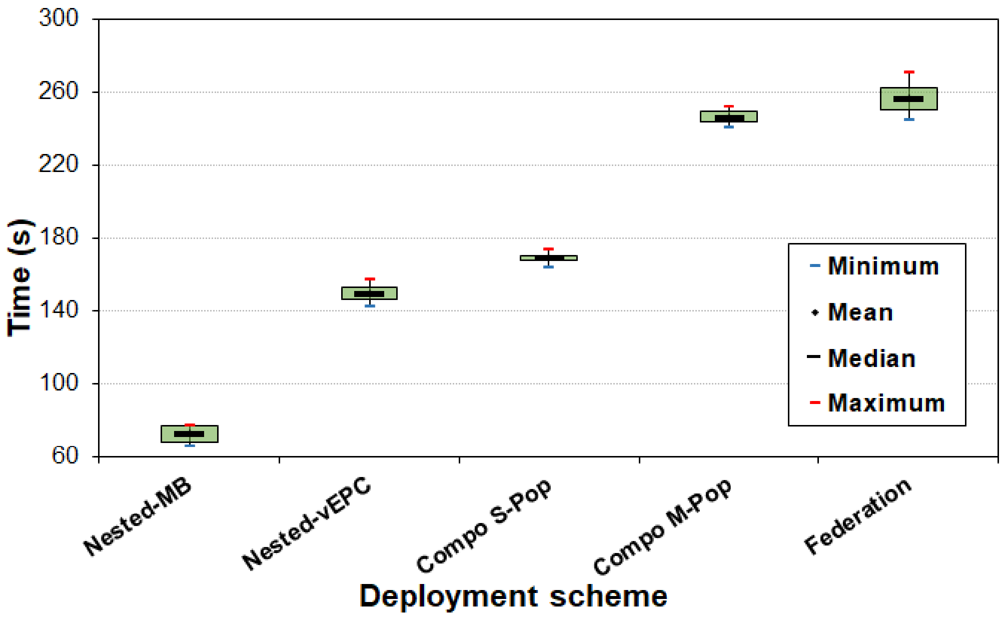


Figure 10.6: Service creation time boxplots for each of the considered deployment schemes

As it can be observed in Figure 10.6, the total experienced SCT increases as the orchestration process considers a more complex deployment scheme. Thus, as expected, the experienced SCT for the deployment of the composite network service in multiple ADs using the described NSF procedure (named as *Federation* in Figure 10.6) is the highest one among all considered cases, presenting a maximum value of around 270 seconds, which is in line with the 5G KPI target of deploying network services in the order of minutes. Next paragraphs dig into the characterization of the time sources contributing to the experienced SCT while focusing on the *Federation* case.

Figure 10.7 presents a profiling of the average value of the operations involved in the SCT of the *Federation* case when compared with the deployment of the same network service in multiple NFVI-PoPs of a single AD (labelled as *Compo M-PoP* in Figure 10.6). This deployment scheme presents a

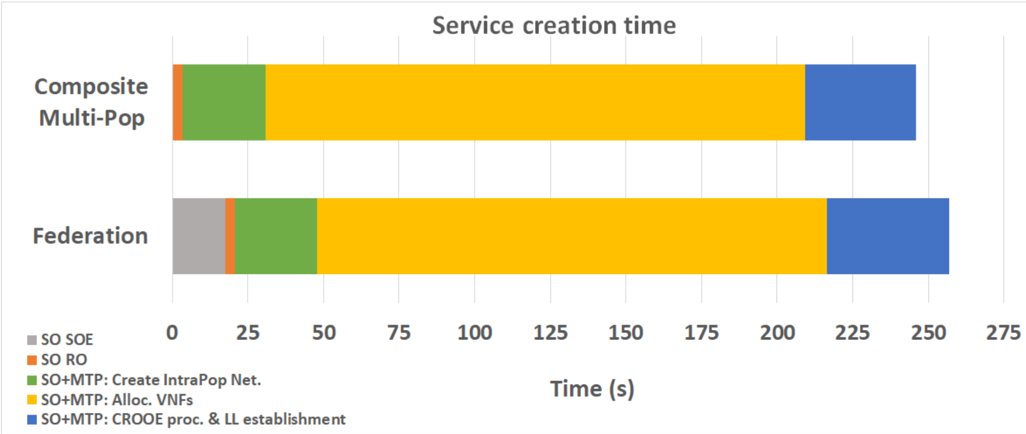


Figure 10.7: Service creation time profiling: federation versus composite multi-pop deployment

similar complexity to the *Federation* case excluding the operations related to the interaction among different ADs.

From Figure 10.7, it is observed that the average service creation time of the *Federation* case is of 257 seconds. This value is around 5% higher than the average one experienced in the *Composite Multi-Pop* case, which is of 245 seconds.

The allocation and activation of VNFs (in this case VMs) is the most relevant component in the SCT. For the *Composite Multi-Pop* case, it accounts for the 72.5% (178.6 seconds) of the total SCT. For the *Federation* case, it accounts for the 66% (168.8 seconds).

After the operation of allocation and activation of VNFs, the next most significant components in the SCT are *(i)* the creation of virtual networks at the NFVI-PoP level to support the different VLs of the network service and *(ii)* the determination and establishment of LLs through the transport network at the corresponding ADs to interconnect the different nested NFV-NSs done by the CROOE module[5]. Summing up the contribution of these three operations to the SCT, they account for the 98.5% (*Composite Multi-Pop*) and the 92% (*Federation*), respectively, of the total average value.

In these three operations, which are the actual resource allocation operations, different modules of the SO (Core MANO wrapper, CROOE, ROOE) interact with the MTP. As expected, its impact in the SCT presents a similar trend when compared to the case of a regular NFV-NS deployment in a single AD using any state-of-the-art MANO platforms (e.g., OSM, Cloudify), as mentioned in Chapter 8. Its value depends mainly on the specific virtualization technology used (e.g., VMs vs. containers), the amount of interconnections to be established, the management complexity of the transport network and the performance of the hardware equipment at the VIM/NFVI-PoP sites. With respect to the last point, the performance of the hardware equipment, a clear example can be derived from Figure 10.6 when comparing the *Nested-vEPC* and the *Compo S-PoP* cases. The small difference in SCT among cases is explained due to the lower performance of VIM labelled as VIM#2 in Figure 10.4 with respect to VIM labelled as VIM#1.

Figure 10.7 also shows another relevant component impacting in the SCT of the *Federation* case labelled as SO SOE. This component considers the elapsed time in the different operations at the Service Orchestration Engine (SOEp-SOEc) to coordinate the orchestration process. Its contribution

[5]The time represented in Figure 10.7 related to the allocation of network resources at the LLs corresponds exclusively to the interconnection of the different nested NFV-NS. Networking resources at the LLs are not allocated when deploying the nested NFV-NS because NSD descriptors have been defined so as the Placement Algorithm running in the SO places all the component VNFs of a single nested NFV-NS within the same NFVI-PoP

is around the 7% of the overall experienced average time. This comes mostly from the periodic polling operation (every 10 seconds) that the SOEp of the CD makes to the implied PD/s to verify the instantantiation status of the requested nested NFV-NSs in other ADs (step 13 of presented workflow in Section 10.3.1) prior to continue with the orchestration process. With the exception of the effect of this polling operation and considering that communication between ADs would be in the order of milliseconds, the processing overhead introduced by the SOE module to coordinate all the orchestration process is limited. Actually, the value of the SO SOE component in the *Composite Multi-PoP* case (around 61ms) could be considered as an approximate measure of the time it takes the SOE sub-modules (SOEp and SOEc) to decompose the NSDs and to create the appropriate requests to later coordinate the rest of modules of the SO architecture. Finally, the remaining time component is the SO RO, which accounts for 3 seconds on average for both cases. This component considers the time the ROOE module takes to parse the nested NFV-NSD and its associated VNFDs, retrieve NFVI resource status through the MTP and interact with the placement algorithm when instantiating the nested NFV-NS.

After this analysis and obeying to the values presented in Figure 10.7, an initial conclusion is that the difference of 5% in SCT between considered cases (*Federation* vs *Composite Multi-PoP*) is mostly explained by the periodical polling operation done from the CD to verify the instantiation status of the nested NFV-NS at the PD in the performed experiments.

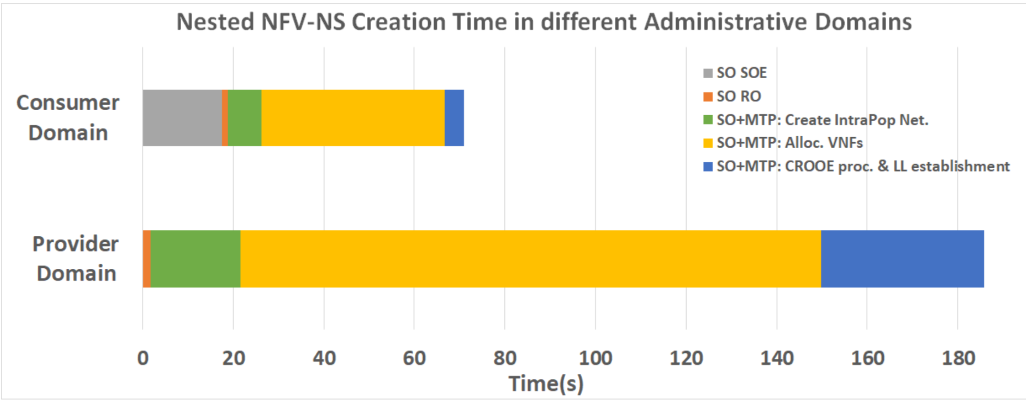


Figure 10.8: Service creation time profiling: Consumer domain versus provider domain

Figure 10.8 breaks down the *Federation* case presented in Figure 10.7 by presenting a comparison on the SCT time of the different nested NFV-NSs in the different ADs depicted in Figure 10.4. In this case, the CD is the AD2 and the PD is the AD1. This graph allows focusing on the complexity of the different nested NFV-NSs to understand its impact in the SCT. Figure 10.5 showed that the MB nested NFV-NS consists of two VNFs and two VLs and the vEPC nested NFV-NS consists of four VNFs and six VLs. Thus, this explains the difference in employed time at different ADs to allocate the corresponding VNFs and to create the virtual networks mapping the VLs to which the created VNFs are attached.

Comparing the time for the determination and creation of LLs to interconnect the different nested NFV-NSs at the different ADs, a clear asymmetry can be observed although the same amount of inter-nested NFV-Ns connections are required to be established from/to CD and PD. In the PD, this time presents and average value of 36.1 seconds, while in the CD, it presents an average value of 4.1 seconds. This difference in time is mostly explained due to the more complex WIM structure (hierarchy of SDN controllers) dealing with different transport technologies and the bigger amount of forwarding elements to be configured in AD1 with respect to AD2. The impact of the time required at the CD to determine the interconnections between VNFs of different nested NFV-NSs according to specified information in the NSD and to select the appropriate LLs with the LSA algorithm (both at the CD and the PD) is extremely low, in the order of milliseconds. Thus, this asymmetry in the management of the transport network to allocate resources for the LLs between ADs and the size of the transport network itself may also have a big impact in the total experienced SCT of the whole *Federation* case. The impact of the SOE component is only relevant in the CD (17s), accounting for the 25% of the time at CD operations (recall that the impact in the whole federation process is of 7%). As previously mentioned, this is mainly due to the polling operation the CD does to the involved PD/s while coordinating the NS deployment employing the NSF procedure. In the PD, this component contributes, on average, with only 44 ms.

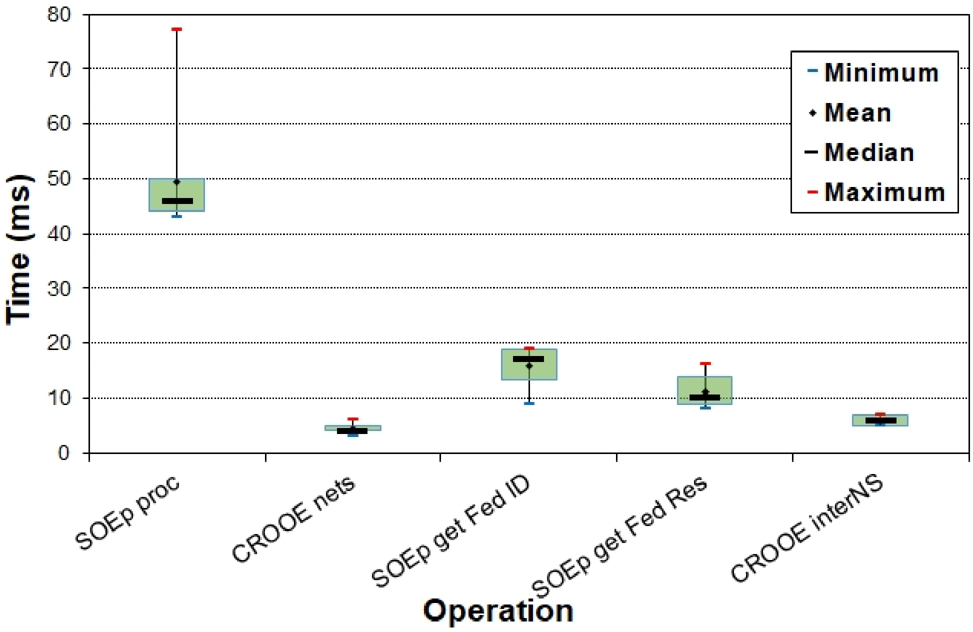


Figure 10.9: Statistical behaviour of processing operations involved in the NSF process

As a final consideration in this study, Figure 10.9 shows the statistical behaviour, from the CD perspective, of the time elapsed in some of the processing and message exchange operations triggered by the SOEp and the CROOE modules to handle the orchestration of composite network services and enable the NSF procedure. Table 10.2 gives a brief explanation of the considered operations, which have been detailed in previous subsections. As it can be seen in Figure 10.9, and has been mentioned during the analysis of the experimental results, the magnitude of elapsed time of these operations is in the order milliseconds, not influencing in the total experienced SCT, which, for the presented case, is in the order of hundreds of seconds.

10.4.3 Use Case: NSF enabling Multi-Provider eHealth Emergency Services

This section presents a Proof of Concept (PoC) of the proposed NSF procedure applied to the eHealth vertical use case developed in the framework of the 5G-TRANSFORMER project in a close to real scenario ([12], [149]). The aim of the PoC is to help the SAMUR-Civil Protection service of the City of Madrid (henceforth SAMUR) to use the kindness of next generation mobile network technology to reduce latency coordination mechanisms for emergency services and improve medical on-site care, hence increasing the patient's survival possibilities in the case of a heart attack alerted by a wearable device.

The description of the use case is as follows. Users wear a smart wearable device, which periodically reports its health status (heart beat rate) and position to a monitoring server. In the case the

Table 10.2: Description of processing and operations associated to the communications between ADs during the NSF procedure

Component	Description	Label in Figure 10.9
SOEp processing	Composite NFV-NS decomposition between local/federated, iteration over the different nested NFV-NSs during instantiation	SOEp proc
CROOE VL mapping	Mapping between composite VLs and nested NFV-NSs	CROOE nets
SOEp creating Federated Nested NS ID	SOEp at CD performing ETSI-NFV IFA030/013 procedure to instantiate a nested NFV-NS at the PD	SOEp get FedID
SOEp getting Federated instantiation result	After instantiation of nested NFV-NS, CD asks the result to provider domain to update DB registries	SOEp get Fed Res
CROOE determining inter-nested VNFs connections	After all nested NFV-NSs have been instantiated, CROOE at CD determines the interconnected VNF of different nested NFV-NSs	CROOE interNS

monitoring server detects a potential issue, it issues an alarm to the wearable device. If the user does not provide feedback within a certain interval, the alarm is confirmed. In that case, the emergency procedure is started and the central server automatically requests the presence of paramedics in the user's location and triggers the deployment of an edge emergency network service close to the patient's location to support the emergency team and improve the medical on-site care by reducing the latency to access patient's health records or by offering Augmented Reality/Virtual Reality services. Note that depending of the user location, the edge emergency network service may be requested to other service provider, hence requiring NSF to satisfy the needs of the emergency request. For this work, this is the interesting part of the use case.

This PoC is realised through the deployment of two NFV-NSs. First, the regular Monitoring NFV-NS and second, the Emergency composite network service. This composite network service consists of two nested NFV-NS, namely the Monitoring NFV-NS (that is already deployed) and the Edge NFV-NS.

These NFV-NSs are deployed in the experimental setup depicted in Figure 10.10. A detail of the structure of the different nested NFV-NS is provided in the bottom right part of the figure. This experimental setup is split between two physical sites placed in Castelldefels (CTTC premises) and Madrid (5TONIC Lab [150]) and divided into two administrative domains.

AD1 is a service provider (SP) that has its infrastructure in multiple physical locations (Castelldefels and Madrid). Actually, AD1 consists of a complete orchestration platform managing a total of three NFVI-PoPs interconnected by a multi-layer/multi-technology transport network. AD1 uses OSM Release 6 as Core MANO platform. Two of those NFVI-PoPs and the transport network are placed in Castelldefels premises and are part of the experimental setup described in previous section and depicted in Figure 10.4. The additional NFVI-PoP, placed in the Madrid site, is colocated with the Radio Access Network functionality. Both physical sites are interconnected by a L3 control plane VPN to interact with the VIM and a L2 data plane VPN to forward traffic among VNFs deployed at different sites. AD2 is a service provider, whose infrastructure is placed in a single physical location (Madrid). AD2 consists of a complete orchestration platform managing a total of two NFVI-PoPs. Differently to AD1, which uses OSM, AD2 deployment uses Cloudify software as Core MANO platform. This

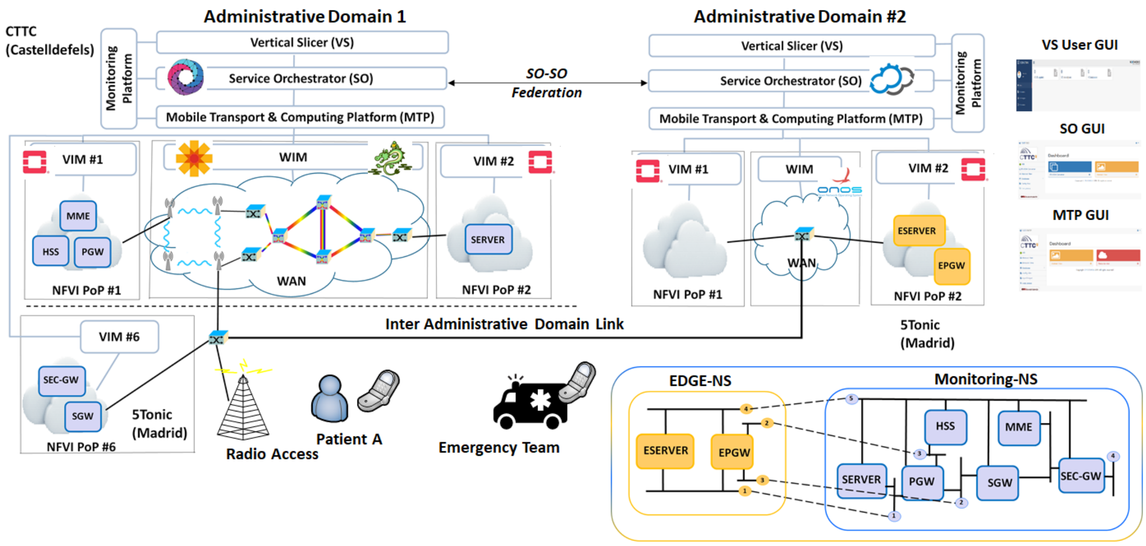


Figure 10.10: eHealth PoC experimental setup

deployment allowed the validation of the multi-MANO capabilities of the Service Manager concept described in Chapter 8, thus confirming the kindness of the proposed architectural design. Both ADs are connected by a L3 control plane VPN where the messages of the *SO-SO* interface flow and a L2 data plane VPN implementing the Inter Administrative Domain Link.

Initially, the regular Monitoring NFV-NS is deployed at AD1 among the available NFVI-PoPs as depicted in Figure 10.10. This distribution is possible due to the non-demanding latency requirements of the VLs defined in the Monitoring NFV-NS. Moreover, it is worth remembering that the distribution and interconnection of VNFs of the same NFV-NS in multiple NFVI-PoPs is another relevant feature of the orchestration platform. This regular NFV-NS counts with a total of six VNFs. The Server VNF is in charge of gathering the vital sign data of registered users, processing it and triggering alarms in case of anomaly. The rest of VNFs implement the software entities of a vEPC mobile core deployment, namely SEC-GW, SGW, PGW, MME, and HSS. They are represented with blue boxes in Figure 10.10.

On confirmed emergency alarm, the Server VNF triggers a request for the deployment of the Emergency composite network service at the VS of AD1[6]. The VS of AD1 requests the deployment of the composite network service combining the already instantiated monitoring NFV-NS and a new edge NFV-NS to the SO of AD1. The edge NFV-NS consists of two VNFs, where the edge PGW redirects the traffic from the emergency team to the processing edge Server. They are represented with yellow boxes in Figure 10.10. After analyzing the request, the SO (of AD1) considers the AD2 to deploy the edge NFV-NS due to NFV-NS availability and proximity to the patient location to satisfy low-latency requirements imposed by the emergency situation. Then, the NSF procedure is performed through the *SO-SO* interface between ADs. When the Edge nested NFV-NS is deployed, the SO in AD1 manages the configuration of the nested NFV-NS interconnections at both ADs through the available inter administrative domain link. By the time the emergency medical team arrives to the patient's location, the Emergency composite network service is deployed and the data from the user equipments of the emergency team flows from the RAN to the newly deployed EPGW at the edge through the inter-administrative domain link, while satisfying the low-latency requirements imposed by the emergency situation. A demonstration video showing this use case can be found online at *https://www.youtube.com/watch?v=meGSPmV5of0*.

[6]Additionally, it reports the user location to the emergency service central office to send the emergency medical team. This description omits some details on this part of the process since the focus is on the network service orchestration operations.

10.5 Conclusions

This chapter has presented the Network Service Federation (NSF) concept, its relevance for SPs in next generation mobile networks and a set of aspects that an orchestrator architecture must consider to effectively handle the E2E deployment of network slice instances in the form of composite network services spanning multiple administrative domains.

From this starting point, the modules of the proposed SO architecture have been evolved to tackle the identified challenges. As mentioned in previous Chapter 9, the modular and flexible architectural design of the SO allows adding the NSF/multi-domain functionality on top of the already available single-domain orchestration capabilities. The NSF concept required, between other aspects, the development and integration in the SO architecture of the *SO-SO* interface coordinating the orchestration operations between ADs at the management plane level. This interface handles not only service orchestration operations but also resource orchestration operations. Indeed, the required logic and the execution of these resource orchestration operations is a disregarded functionality in most of other approaches proposed in the literature. The main contribution of this chapter focuses on this and proposes the definition of an interface and a detailed workflow covering this detected gap in related work. Thanks to this, nested NFV-NSs can be deployed in multiple ADs and the inter-nested inter-domain connections are set up according to the requirements while avoiding potential resource conflicts in identifiers of different domains (e.g., IP addresses). These considerations are fundamental to make E2E multi-domain orchestration and network slicing a reality in practice.

The introduced changes in the architecture and the proposed interface and workflow have been validated and the service creation time (SCT) for a defined composite network service has been profiled in an multi-administrative/multi-technology experimental setup. Experimental results show that the SCT for a composite network service requiring NSF is in the order of minutes (around five minutes for the evaluated case). The main components contributing to the SCT are the intrinsic operations associated to the instantiation of the nested NFV-NSs, like booting up virtual machines or setting interconnections through the transport network between different NFVI-PoPs. The impact in SCT of operations related to the processing required by the federation process and the latency in the management plane link between ADs is limited.

The chapter concludes with the presentation of the applicability of the NSF procedure to a close to real eHealth scenario. Thanks to the dynamic and flexible capabilities introduced by the NSF concept in the orchestration process, the eHealth vertical user can automatically deploy composite network services when/where required to meet certain service constraints, like low-latency communications between paramedics and the hospital. This can contribute to improve the quality of care by reducing the time required to detect the emergency and coordinate the medical resources and by enabling the possibility of using new innovative low-latency services based on augmented reality.

Chapter 11

Scaling Composite and Federated Network Service deployments

The contributions of this chapter have been published in [151] and [152] and have been submitted to [153]. This is the last technical chapter of this work, were we present the enhancements of the SO architecture to handle the scaling of network slice instance deployments using composite NFV-NS, also those including NSF. These kinds of operations aim to contribute to the SLA maintenance during run-time operation despite changing service demands.

This chapter begins with Section 11.1, where we present the multiple scenarios and considerations that need to be taken into account to evolve the main modules and interfaces of the SO architecture to perform the scaling operation of composite NFV-NS deployments. The operational workflows associated to these scenarios are then, presented in Sections 11.2 and 11.3. Section 11.4 validates and provides an experimental performance evaluation of such proposed scaling workflows when applied to composite NFV-NS deployments in single AD and multi AD scenarios. This evaluation shows that scaling operations for composite deployments can be executed in the order of few minutes, shaping the NFV-NS structure according to dynamic service demands while making use of the underlying resources when needed.

11.1 Scaling Composite NFV-NS deployments: scenarios and considerations

As mentioned in Chapter 9, next generation mobile networks benefit from the flexibility and dynamicity provided by composite NFV-NS deployments to build E2E network slice instances providing more complex and tailored offerings to the vertical industries, hence better adapting to its needs.

As considered by the NSO concept presented in Chapter 8, SLAs need to be fulfilled not only at instantiation time but also during service run-time despite network traffic changing conditions. As seen also in Chapter 8, SLAs can be honoured during run-time operation by means of scaling operations aimed to change the structure of the NFV-NS to provide it with the adequate amount of resources. However, the multiple deployment options and components of a composite NFV-NS increase notably the complexity of the scaling process when compared to the procedure presented for regular NFV-NS deployments.

The kind of required operations involved during the composite NFV-NS scaling process depends on: *(i)* how it has been deployed, and *(ii)* which is the destination of the scaling request. With respect to the first consideration, a composite NFV-NS can be deployed in either a single AD or in multiple ADs (by using the NSF concept presented in Chapter 10). Moreover, such a deployment can be done all-at-once or using a reference to a deployed regular NFV-NS, which may be shared with other composite NFV-NSs. With respect to the second consideration, the scaling operation can be directed to the whole composite NFV-NS deployment or it may be an auto-scaling operation triggered by one of its constituent nested NFV-NSs. Additionally, in the sharing scenario, if a scaling operation is directed to a shared regular NFV-NS, this operation may imply changes to associated composite NFV-NSs. All these scenarios have been considered in this work, broadening the scope of the ETSI-NFV IFA028 [9], which, to the best of our knowledge, is the sole work in the literature considering the scaling operation of composite NFV-NSs deployments. The ETSI-NFV IFA028 report only considers the service orchestration perspective of the scaling operation acting at the composite level, disregarding some of the other mentioned scenarios and also the resource orchestration perspective of the problem, as mentioned in Chapter 10 with the instantiation procedure. In this case, the resource orchestration perspective of the problem deals with the update of inter-nested NFV-NSs connectivity associated to successive scaling operations and possibly involving multiple ADs, which is essential to achieve effective deployments, as mentioned in previous chapters.

In the SO architecture proposed in previous chapters, the modules in charge of handling the instantiation of composite NFV-NSs and the NSF process are the parent Service Orchestration Engine (SOEp) and the Composite Resource Orchestration Engine (CROOE). The same modules are the ones handling the scaling operation. The SOEp orchestrates all the operations in case of a composite NFV-NS, relying on the child SOE (SOEc) to perform the instantiation/scaling of the different nested NFV-NSs in a composite NFV-NS as if they were a regular NFV-NS ([134], [135]). Its logic has been extended to understand the destination of the scaling operation while verifying its compatibility with the defined composite NFV-NS deployment flavour (DF) and instantiation level (IL) to coordinate the rest of the modules of the SO and to contact other SOs at other involved ADs, if needed. The CROOE, triggered by the SOEp, is in charge of determining the inter-nested NFV-NS connections and of handling its establishment by interacting with the local MTP or the CROOE module of other involved ADs. When a scaling operation is performed, it also handles the required updates (creation/deletion) of such inter-nested connections based on the information stored in the Network Service Instance Resource DB and the new IL of the component nested NFV-NSs.

11.1.1 NSF Interface Extension

As with the instantiation operation presented in Chapter 10, all message exchange during the scaling operation between respective SOEp and CROOE modules is performed through the available *SO-SO* interface, when such operation requires the interaction of multiple ADs.

Messages exchanged between peering SOEp modules are related to lifecycle management operations and follow the ETSI-NFV IFA030/013 specifications [93]. This work includes the support to the *Scale NS* operation in the SOEp. In the ETSI-NFV IFA028 report, this operation is thought to be only used at the consumer domain (CD) side to launch the scaling operation at the provider domain (PD). However, a particular case handled in this work, which is further developed in Section 11.3, is the use of this operation at the PD to inform the CD that an auto-scaling operation is being held in a nested NFV-NS instantiated at the PD. When issuing this message, the PD besides informing the CD of the new nested NFV-NS IL in the *scaleNsData* information element of the *ScaleNS* request, it also includes as an *additionalParamForNs* attribute, the *operationID* of the scaling operation at the PD, so the CD can poll the status of the operation at the PD before continuing with the scaling process.

Messages exchanged between CROOE modules are related to resource orchestration operations aiming to the interconnection of nested NFV-NSs deployed in different ADs. As mentioned in Chapter 10, these messages cover a gap detected in related state-of-the-art work. This interface has been extended with the *Update Federated Internested Connections* message to handle the update of inter-nested connections (either create/remove) during the scaling operation. It is worth noting that the change from one IL to another IL of a composite NFV-NS or one of its nested NFV-NSs may imply the simultaneous creation/deletion of VNF instances (scale out/scale in) and, hence, of its associated interconnections. Listing 6 presents the detailed format of the request issued by the CD to the PD. In this request, the CD includes a list of the *"new"* pairs of VNFs to be connected (provider/consumer) as a consequence of possible scale out operations and a list of *"old"* pairs of VNFs whose connection needs to be deleted as a consequence of possible scale in operations. This message also includes the characteristics of each of the shared inter-nested virtual links (VLs) in terms of bandwidth and latency to determine the appropriate logical links (LLs) satisfying such requirements. The query URL contains the NFV-NS identifier (*nsId*) in the PD. In the body of the query, the NSD identifier in the PD can also be found, so the PD can verify that the *nsId* reference in the URL corresponds to an NFV-NS instance using the requested NSD. The PD replies *"OK"* or *"KO"* to the CD depending on the result of the path establishment operation.

Listing 6 Update Federated Internested Connections

```
Request: Consumer domain
URL = "http://provider_domain:port/
5gt/v1/ns/{nsId}/update-fed-internested-connections"
body:{
    "nsdId": "NS descriptor identifier in the PD"
    "connectedVNFs_add": {
      "internested_VL1": [VNF3p-VNF1c,...]
      "internested_VL2": [VNF3p-VNF2c,...]},
    "connectedVNFs_del": {
      "internested_VL1": [VNF4p-VNF2c,...]
      "internested_VL2": [VNF2p-VNF3c,...]},
    "linkChar":{
      "internested_VL1": {
        "latency": latency_vl1,
        "bw": bandwidth_vl1},
      "internested_VL2": {
        "latency": latency_vl2
        "bw": bandwidth_vl2}  }  }
Reply: Provider domain
body:{
   "pathEstablishment": {"OK"/"KO"}  }
```

11.2 Composite NFV-NS scaling workflow

Figure 11.1 and Figure 11.2 present the operational workflow followed by the proposed E2E network service orchestration architecture to perform the scaling of a composite NFV-NS. In this description, we follow the same terminology as in the workflow for the NSF case in Section 10.3.1. This description is considering a generic case, where the scaling operation requesting the change of IL of the composite NFV-NS can imply the scaling of nested NFV-NS either in the CD or in the PD, hence requiring NSF operations.

First, the Vertical Slicer (VS) requests the scaling of the composite NFV-NS to the SO of the CD according to the format of the *ScaleNS* operation included in the ETSI-NFV IFA 030/13 specification [94] (step 1). After checking that the composite NFV-NS is in "Instantiated", the SOEp sets its status to "Scaling" and returns an operation ID to the VS (steps 2,3 and 4), so it can request the operation status to the SO.

After that, the SOEp retrieves the composite NSD from the *NSD Catalogue DB* and the NFV-NS instantiation info from the *NS Instance DB* and performs a mapping between the target IL for the composite NFV-NS specified in the received scaling request and the corresponding ILs for the different constituent nested NFV-NS (steps 5 and 6). With this mapping operation, the SOEp knows which nested NFV-NSs need to be scaled, which are their target ILs and in which domains these nested NFV-NSs are deployed.

First, the SOEp starts the scaling of those nested NFV-NSs deployed at the CD, in case it is needed (step 7). For each of these "consumer" nested NFV-NSs, the SOEp gets its information from the *NS Instance DB*, set its status to "Scaling" and creates the corresponding *ScaleNS* request, which is delegated to the SOEc (steps 8, 9, 10 and 11). The SOEc proceeds with the scaling of the nested NFV-NS according to the procedure explained in Section 8.4.1 (step 12). When the SOEc has finished the scaling of the nested NFV-NS, the SOEp updates the information related to the nested NFV-NS deployment at the *NS Instance DB* (step 13). Among the updated information, the SOEp sets the nested NFV-NS to "Instantiated" status, its new IL and the updaded info of its associated SAPs.

Next, the SOEp starts the scaling of those nested NFV-NSs deployed at the PD, in case it is needed (step 14). For each of these "provider" nested NFV-NSs, the SOEp gets its information from the *NS*

Instance DB, set its status to "Scaling" and creates and sends the corresponding *ScaleNS* request to the PD through the *SO-SO* interface (steps 15, 16, 17 and 18). At the PD, the SOEp validates the scaling request and returns the operation ID to the CD (steps 19 and 20). This operation ID is used by the SOEp at the CD to poll the status of the scaling operation at the PD. As in the CD, the SOEp at the PD relies on its SOEc to perform the scaling operation (steps 21, 22). Once the CD has confirmed that the nested NFV-NS at the PD has been correctly scaled (step 23), the SOEp at the CD requests its CROOE to get the updated information of the federated nested NFV-NS instance (step 24). Once received the information, the CD SOEp updates the information related to the provider nested NFV-NS deployment at the *NS Instance DB* (step 25), like previously done with the consumer nested NFV-NSs.

After all the required nested NFV-NSs, either at the CD or at the PD, have been scaled, the SOEp at the CD starts coordinating the process to update the inter-nested connections according to the new ILs of the different nested NFV-NS.

First, the SOEp contacts the CROOE to update the inter-nested connections among nested NFV-NSs specifically deployed at the CD (step 26), if needed. The CROOE determines the inter-nested connections based on the corresponding NSDs and the instantiation information of the different consumer nested NFV-NSs available in the composite entry of the DBs (step 27). The CROOE compares the new set of inter-nested connections with those that were stored previously (after instantiation or after other scaling operation) in the *NS Instance Resource DB* to determine those connections that need to be established or those that need to be deleted (steps 28 and 29). Then, the CROOE updates the associated DB entry with the *new* set of inter-nested connections to take them into account for possible subsequent scaling operations (step 30). After that, the CROOE proceeds with the establishment/termination of the updated set of inter-nested connections. It first contacts the RO-EE-WIM, which contacts the MTP through the SBI to remove the required interconnections (step 31). The termination of connections is done first to free resources for the connections that need to be established. Then, the CROOE collects the resource information from the MTP, runs the LSA to select the appropriate LLs interconnecting the desired NFVI-PoPs at the CD and contacts its associated MTP to establish them (steps 32, 33 and 34).

Second, the SOEp contacts the CROOE to update the inter-nested connections between pairs of nested NFV-NSs deployed at the CD and PD/s (step 35), if needed. The process followed by the CROOE is similar to the previous one. Initially, the CROOE determines all the connections between each pair of *"consumer/provider"*[1] nested NFV-NSs, retrieves the former inter-nested connections from the *NS Instance Resource DB* and updates this entry with the new set of inter-nested connections (steps 36, 37 and 38). Then, with this information and for each pair of *"consumer/provider"* nested NFV-NS, the CROOE determines the set of connections that need to be established and those that need to be removed (step 39). Next, the CROOE at the CD issues an *Update Federated InterNested Connection* request to the CROOE at the PD (step 40) with the determined sets of inter-nested connections. The PD-CROOE first contacts its associated MTP to delete the specified connections, and then, it runs the LSA to select the appropriate LLs (from PD to CD) serving the inter-nested connections between ADs, contacting its associated MTP to establish them (steps 41, 42, 43 and 44). Once finished, the PD-CROOE acknowledges the CROOE about the correct path establishment operation. Then, the CROOE at the CD deletes the required inter-nested connections with the help of the MTP, runs the LSA to determine the appropriate LLs (from CD to PD) to map the new set of required inter-nested connections and contacts the MTP, through the RO-EE-WIM to establish them (steps 45, 46, 47 and 48).

After the SOEp receives the confirmation from the CROOE that all the *"consumer/provider"* pairs have been processed (step 49), the SOEp at the CD updates its DBs, setting the composite

[1] Note that the following logic could be applied for possible *"provider/provider"* pair of adjacent nested NFV-NSs.

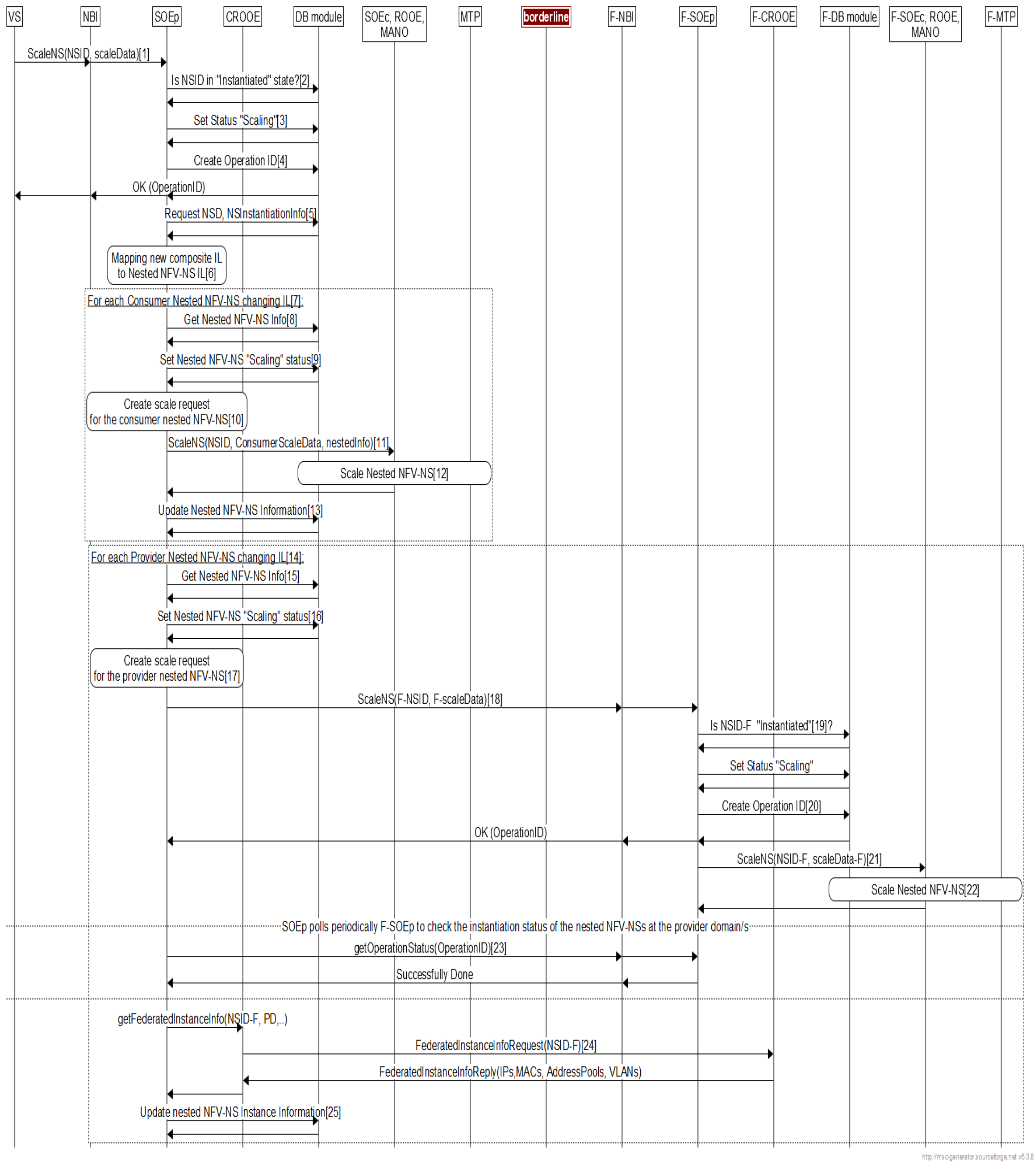


Figure 11.1: Composite Network Service Scaling workflow (Part 1)

NFV-NS status to "Instantiated" and declaring the scaling operation as ”Successfully done” (steps 50 and 51). When polling the operation status, the VS notices the successful operation and retrieves the information of the SAPs associated to the new IL of the composite network NFV-NS (steps 52-53).

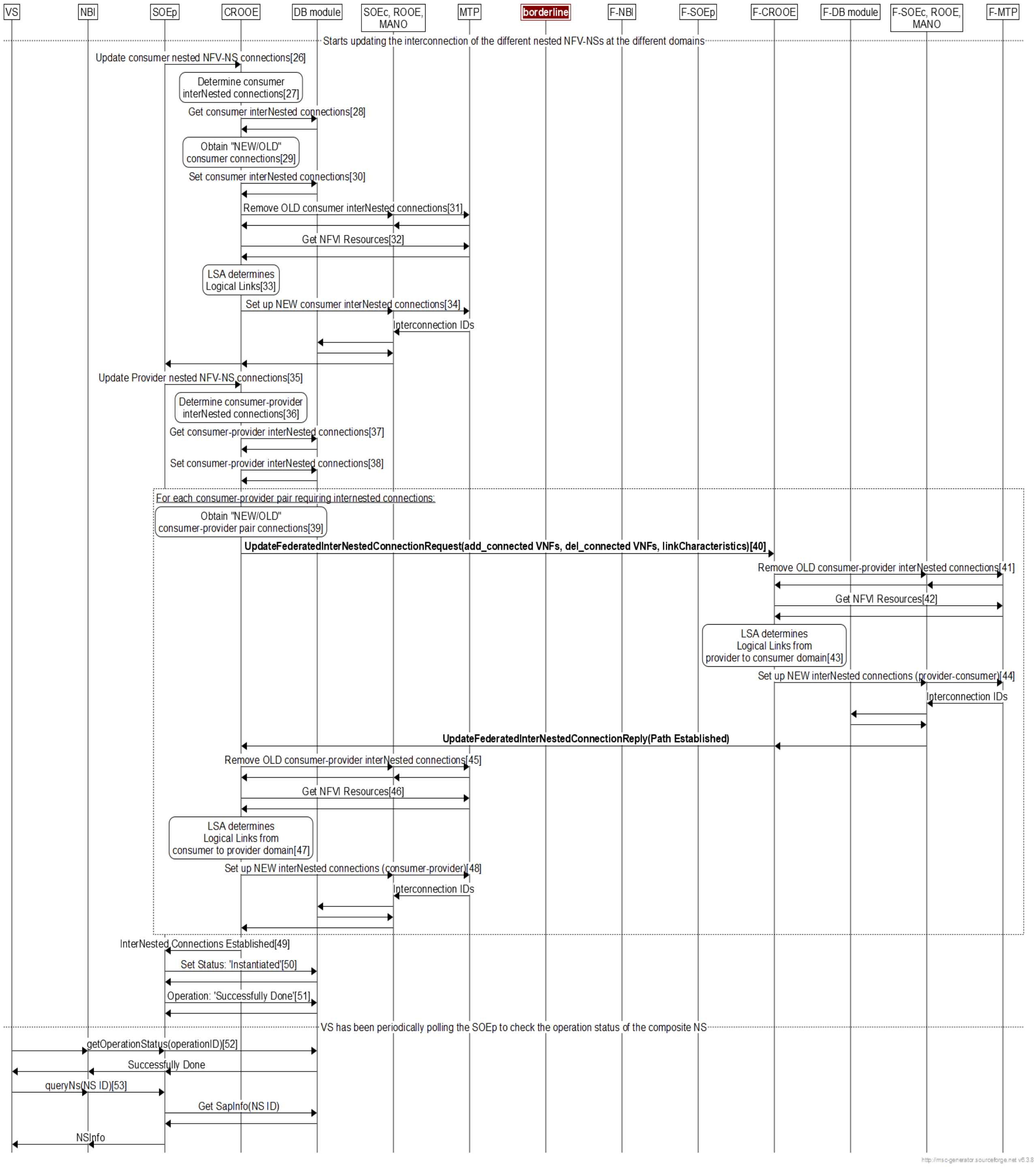


Figure 11.2: Composite Network Service Scaling workflow (Part 2)

11.2.1 Composite NFV-NS scaling workflow considering NFV-NS sharing

Figure 11.3 considers the case of scaling multiple composite NFV-NSs when there is a scaling operation in a regular NFV-NS that has been used as a reference to instantiate the composite NFV-NS/s. In this case, we have considered that the VS initiates the operation, but it could be also as a result of an auto-scaling operation initiated by the SLA Manager of the SO as a reaction in front of a detected SLA violation in the shared regular NFV-NS. The following workflow description focuses on the differential aspects with respect to the previously presented workflow.

The scaling operation starts as the previous one, with the SOEp receiving the *ScaleNS* request. In this case, after getting the NFV-NS info from the corresponding DBs (step 5), the SOEp realises that the regular NFV-NS has associated one or multiple instances of composite NFV-NSs. Then, the SOEp generates a separated process where it delegates the scaling operation of the regular NFV-NS to the SOEc. Like in the previous case, the scaling operation of the regular NFV-NS (step 7) follows the procedure explained in Section 8.4.1. While the scaling operation of the regular NFV-NS progresses, the SOEp in the main thread keeps polling the *NS Instance DB* to check that the regular NFV-NS has finished its scaling process, that is, it presents an "Instantiated" status (step 8). After a correct verification, the SOEp proceeds to update the interconnections between the regular NFV-NS and the different nested NFV-NSs of the composite NFV-NS/s using this regular NFV-NS as a reference.

For each associated composite NFV-NS, the SOEp gets the instantiation information from the *NS Instance DB*, set the status of the composite NFV-NS to "Scaling" and makes a mapping operation to derive the new composite NFV-NS IL after the change in the regular NFV-NS (steps 9, 10 and 11). After that, the SOEp proceeds to update the interconnections between the nested NFV-NSs of the composite NFV-NSs and the regular NFV-NS. This operation follows the same steps as in the workflow described in Figure 11.2, starting by updating the interconnections exclusively at the CD and then, updating interconnections between CD and involved PD/s. After the interconnections have been correctly established, the instantiation information of the composite NFV-NS is updated (new IL) and the status of the composite NFV-NS is set to "Instantiated" (steps 20, 21).

Once all the interconnections with all the associated composite NFV-NSs have been updated, the SOEp updates the DB entries for the regular NFV-NS, setting its status to "Instantiated" and declaring the scaling operation as "Successfully done" (steps 22, 23). At the next polling operation, the VS will be able to retrieve the updated information of the regular NFV-NS (steps 24,25).

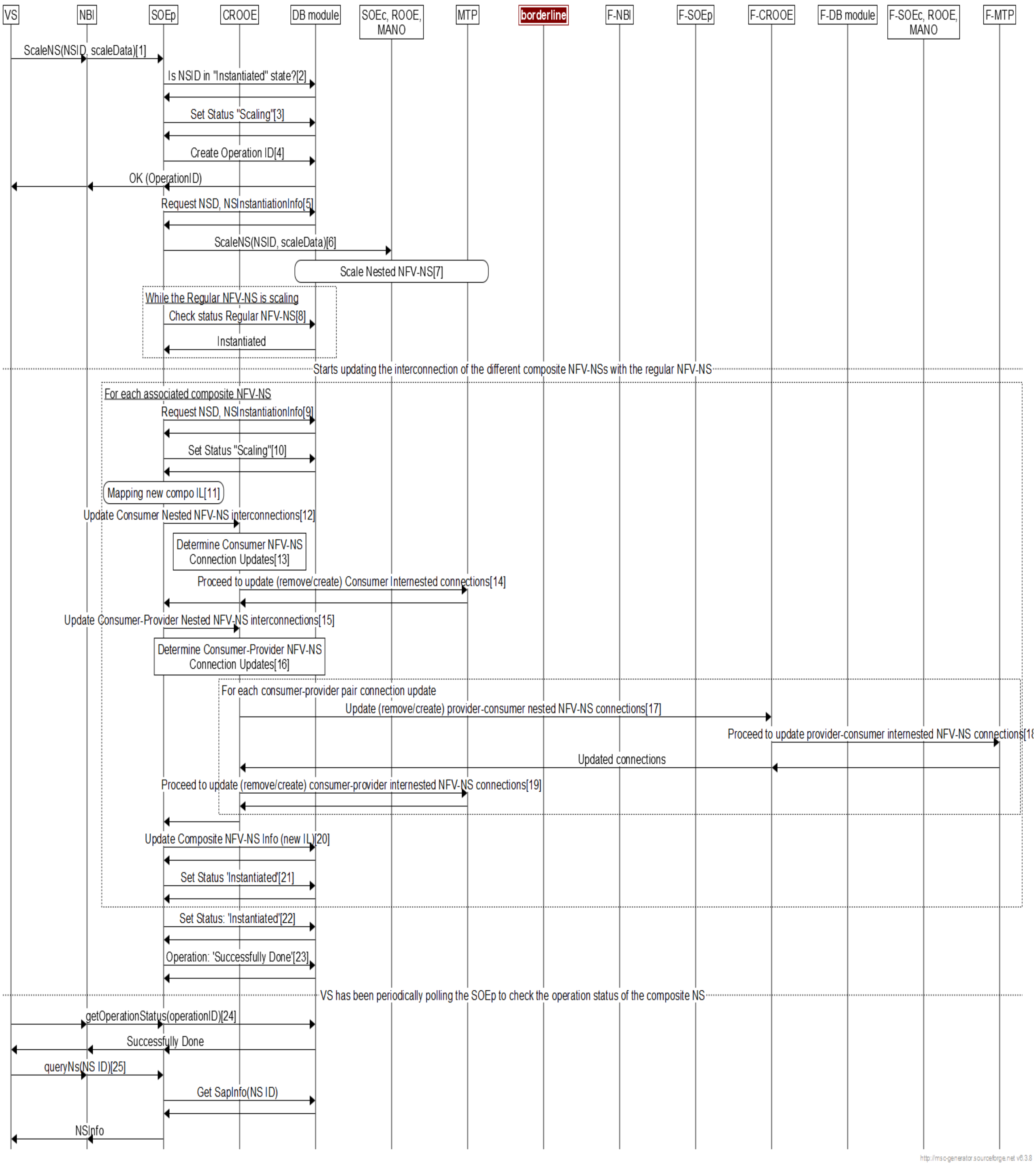


Figure 11.3: Composite Network Service Scaling workflow considering a regular NFV-NS as a reference

11.3 Nested NFV-NS scaling workflow

This section presents the operational worflow followed by the SO after receiving an auto-scaling operation triggered by its SLA manager module upon the detection of an SLA violation as expressed in the NSD of a nested NFV-NS part of a composite NFV-NS. As with the workflow of Section 11.2, it can be divided into two parts. The first one presents different initial steps considering where the nested NFV-NS has been instantiated, at the CD or at the PD. This part of the workflow is the focus of this section.The second one relates to the update of the inter-nested connections, which is common for both cases and driven by the CD, exactly as described in Figure 11.2. It is worth mentioning that the change of IL of a nested NFV-NS derived from the scaling operation needs to be also included in one of the ILs defined for the composite NFV-NS, otherwise the SO would not continue with the scaling operation due to a fail in the operation checking IL compatibilities.

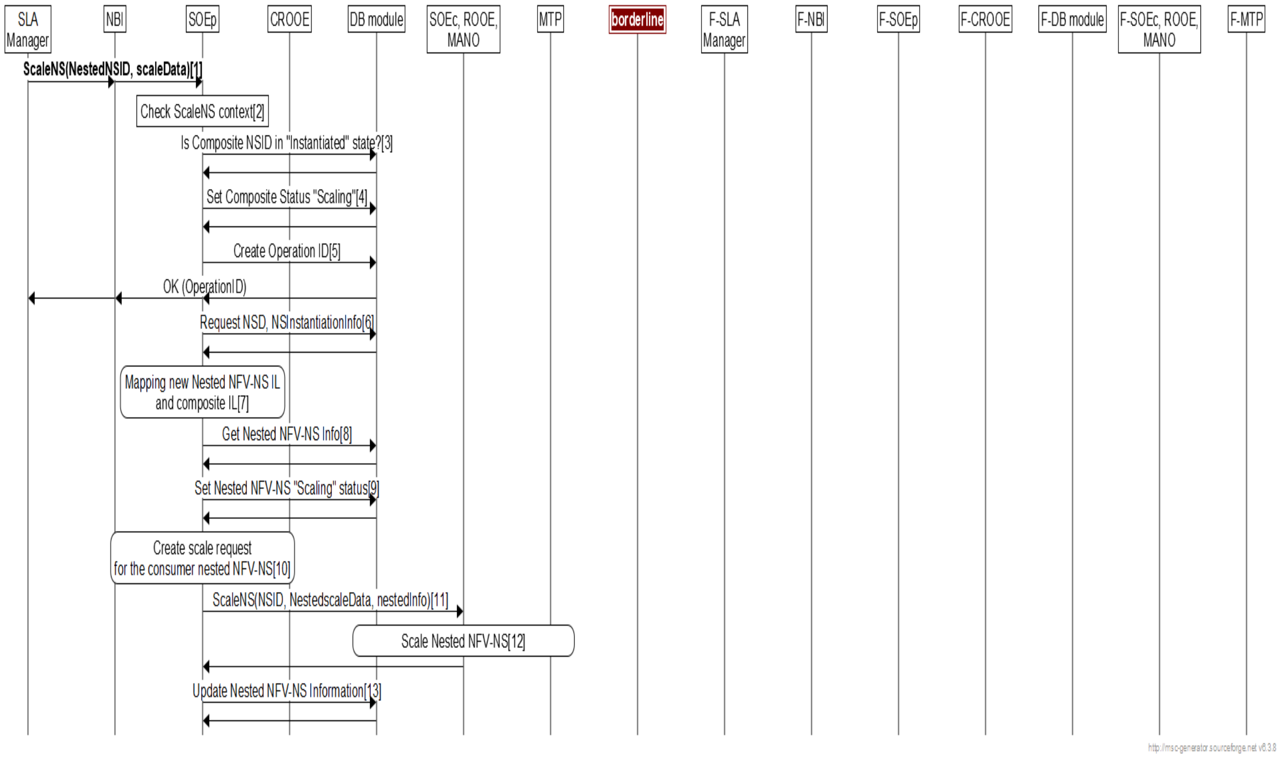


Figure 11.4: Consumer Nested NFV-NS Scaling workflow (Part1)

Figure 11.4 presents the initial steps of the workflow considering that the scaling operation has been issued by a nested NFV-NS placed in the CD. In front of a detected SLA violation in a nested NFV-NS deployed at the CD, the SLA Manager decides to issue a *ScaleNS* request specifying the nested NFV-NS ID and the target IL expressed in the corresponding *autoScalingRule* of the nested NSD as solution to the detected SLA violation (step 1). This request arrives to the SOEp through the NBI, which checks the context of the scaling request (requester and network service instance destination) that determines the scaling procedure and sets the composite NFV-NS to "Scaling" status (steps 2, 3 and 4). The SOEp, then, returns an operation identifier to the SLA Manager (step 5). After that, the SOEp gets the NSD of the associated composite NFV-NS and its instantiation information from the *NS Instance DB* (step 6). With this information and the scale request, the SOEp performs the mapping operation to check the compatibility between the target IL for the consumer nested NFV-Ns and the ILs described in the NSD of the composite NFV-NS (step 7). Once validated, the SOEp gets the instantiation information of the nested NFV-NS and sets its status to "Scaling". Then, the

SOEp delegates the scaling of the nested NFV-NS to the SOEc (steps 10, 11 and 12), which follows the workflow explained in Section 8.4.1 to perform the scaling of the nested NFV-NS. When the SOEc finishes with the scaling, the SOEp proceeds to update the information of the nested NFV-NS at the composite NFV-NS entry of the *NS Instance DB* (step 13). After that, the SOEp at the CD proceeds with the second part of the workflow to update the inter-nested connections from the scaled consumer nested NFV-NS to the rest of nested NFV-NS in the composite NFV-NS, applying the corresponding steps described in Figure 11.2.

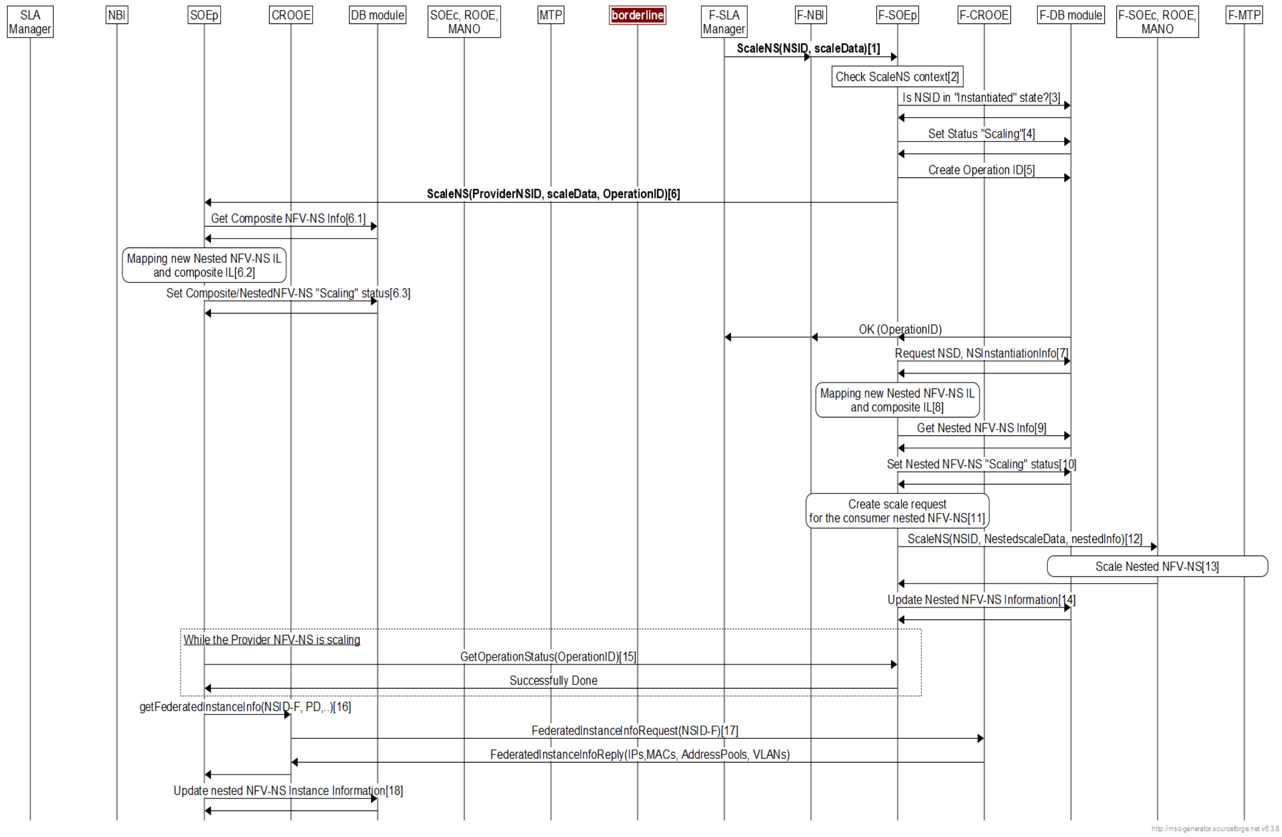


Figure 11.5: Provider Nested NFV-NS Scaling workflow (Part1)

Figure 11.5 presents the initial steps of the workflow considering that the scaling operation has been triggered by the SLA Manager of a PD as a reaction to an SLA violation in a nested NFV-NS deployed at the PD. This request arrives to the PD-SOEp, which checks the context of the scaling request (step 2). The PD-SOEp realises that the NFV-NS to be scaled is part of a composite NFV-NS and after setting the provider nested NFV-NS status to "Scaling" and creating the operation identifier (steps 3, 4 and 5), it spawns a parallel process, where the PD-SOEc handles the scaling of the requested NFV-NS (steps 7 to 14), as explained previously. In the main process, the PD-SOEp issues a *ScaleNS* request to the corresponding CD to inform of the launched scaling process being done at the PD (step 6). Among the information in this request, the PD includes the new IL of the nested NFV-NS and the identifier of the operation being done at the PD as part of the *additionalParamForNs* attribute. The SOEp at the CD verifies with the *NS Instance DB* that the request coming from the PD is valid. It checks if the nested NFV-NS instance entry exists as part of a composite NFV-NS instance and if the composite NSD includes an IL with the IL the nested NFV-NS is scaling to. Then, the SOEp at the CD updates the status of the entries of the composite/nested NFV-NS, setting them to "Scaling" state (steps 6.1 to 6.3). The CD SOEp starts a periodical polling to check the status of the operation at the PD (step 15). Upon validating that the scaling operation of the nested NFV-NS at the PD has been successfully done (step 15), the SOEp requests the CROOE to contact the PD-CROOE to retrieve the updated instantiation information of the nested NFV-NS and updates the corresponding nested NFV-NS information within the composite NFV-NS entry at the *NS Instance DB* of the CD, declaring the nested NFV-NS in "Instantiated" status(steps 16 to 18).

After that, the SOEp at the CD proceeds with the second part of the workflow to update the inter-nested connections from the scaled nested NFV-NS deployed at the PD to the rest of nested NFV-NSs in the composite NFV-NS, applying the corresponding steps described in Figure 11.2.

11.4 Experimental scaling of Composite NFV-NS deployments

This section presents the validation and performance of the proposed scaling workflows presented in previous sections in an experimental multi-administrative setup configured between the EXTREME ® and ADRENALINE ® Testbeds described in Chapter 5. This experimentation considers the service scaling time of a composite network service under different deployment schemes (single vs multi AD scenarios) and when issuing scaling requests at the different levels of the network service deployment, composite and nested NFV-NS level.

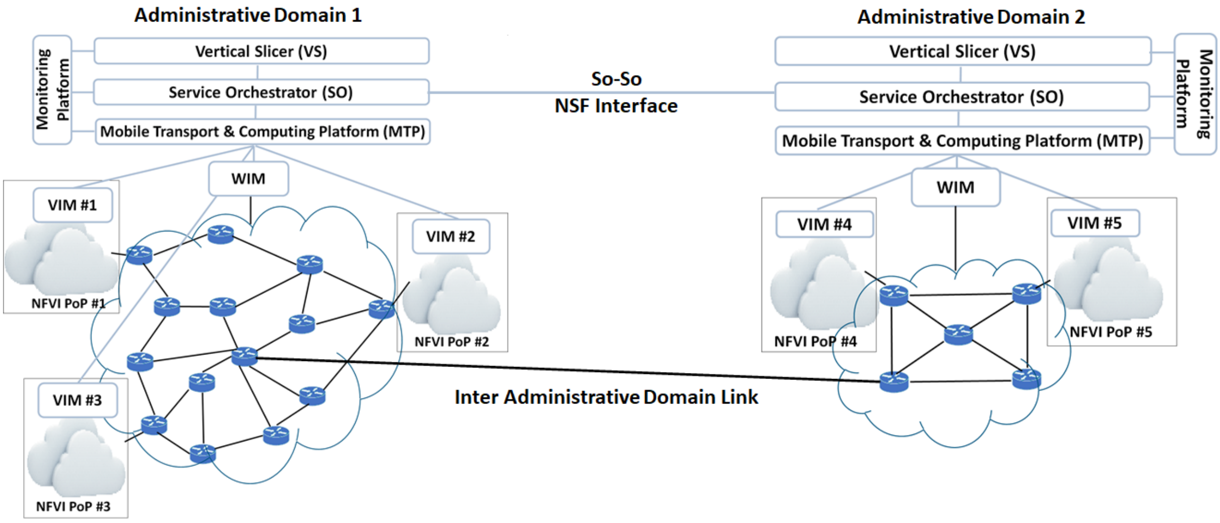


Figure 11.6: Experimental Multi-administrative domain orchestration platform setup

11.4.1 Experimental Setup Description

Figure 11.6 shows the experimental multi-administrative setup used in this validation, where each AD counts with a whole instance of the E2E orchestration platform. The SO module of both orchestration platforms uses Open Source MANO (OSM) as Core MANO platform – namely, AD1 uses OSM Release 6 and AD2 uses OSM Release 7. Different OSM releases are considered in the targeted validation to validate the backward compatibility of the proposed E2E orchestration platform with external components along with increasing the heterogeneity of the experimental setup. AD1 presents three NFVI-PoPs, which are managed by dedicated VIMs (Devstack Queens release). These NFVI-PoPs are interconnected by a GNS3 [113] emulated transport network topology of 14 packet-switch nodes and 22 bidirectional links managed by an instance of the ONOS SDN controller [18] acting as WIM. AD2 has two NFVI-PoPs, also managed by its own dedicated VIM (Devstack Queens release) instances, which are interconnected by an emulated GNS3 transport network featuring 5 packet-switch nodes and 8 bidirectional links. Like in AD1, this transport network is managed by an instance of the ONOS

SDN controller. At the data plane level, AD1 and AD2 are linked through a L2 VPN connection implementing the Inter-AD Link depicted in Figure 11.6. At the control plane level, ADs communicate each other through the *SO-SO* interface, which traverses over a L3 VPN connection. With respect to the setup employed in previous Chapter 10, we have introduced the use of the GNS3 network simulator. This tool gives us the flexibility to emulate arbitrary network topologies of SDN-based software switches (OpenVSwitch), which can be easily connected to the physical network interface cards available at the host running the emulation to interact with the NFVI managers (VIM/WIM) controlled by the orchestration platform. As commented in Chapter 7, the proposed architecture for the MTP module, using a plugin system to communicate with the NFVI managers, allowed a smooth integration of the ONOS SDN controller into the orchestration platform to manage the networking resources at the underlying emulated transport network.

11.4.2 Performance Evaluation

In this performance evaluation, we present the assessment of the presented workflows in the setup of Figure 11.6 when considering different scaling operations of the composite NFV-NS depicted in Figure 11.7.

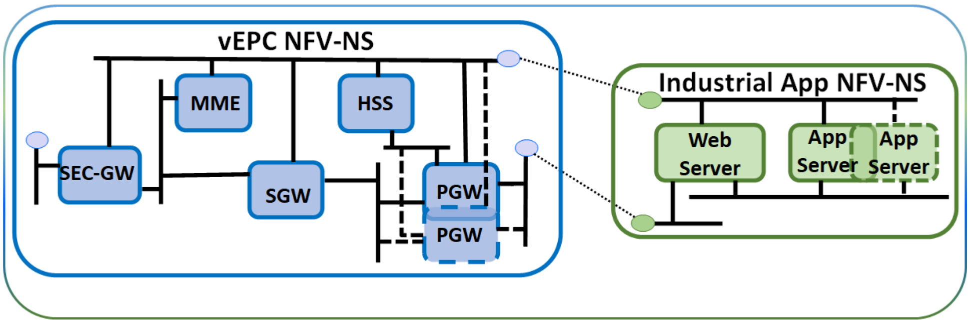


Figure 11.7: Composite network service under evaluation.

This composite NFV-NS consists of two nested NFV-NSs, which consider an Industry 4.0-like scenario. First, we have the nested NFV-NS implementing a standalone Non-Public Network (NPN) [154] by means of a Virtualised Evolved Packet Core (vEPC). This vEPC nested NFV-NS is made up of five VNFs, namely SEC-GW, MME, HSS, SGW and PGW. The second nested NFV-NS represents an *"Industrial App"* and is initially formed by two VNFs: the webserver (WS) VNF, which also includes load balancing capabilities, and an App server VNF. The dashed lines in Figure 11.7 represent the multiple ILs defined for the composite NFV-NS and its nested NFV-NS to dynamically adapt its structure to accommodate new necessities or address different conditions within the factory. For instance, the composite NFV-NS defines an IL where the vEPC NFV-NS presents an additional instance of the PGW to attain traffic load balancing in front of an increase of sessions, or the Industrial App NFV-NS presents an additional instance of the application server to cope with a surge in the number of operations upon an increase in the factory production. Additionally, the deployment of this composite NFV-NS could be considered in multiple steps. First, the regular vEPC NFV-NS can be deployed to provide the mobile infrastructure. Then, different composite NFV-NSs can be deployed using the regular vEPC NFV-NS as a reference. In this way, multiple *"Industrial Apps"* can be attached to the regular NFV-NS, as explained in [151].

Figure 11.8 and Figure 11.9 present the statistical behaviour of the experienced time to perform scaling out/in operations on the composite NFV-NS, respectively, when considering its multiple ILs as described in Table 11.1. The boxplots cover the experienced maximum, minimum, average, median values and the 20^{th} and 80^{th} percentiles of the ten repetitions performed for each experiment. The scaling time considers the elapsed time between the SO at the consumer domain receives the scaling request until the SO declares the operation as successfully done. The initial deployment state of the composite NFV-NS considers the nested vEPC NFV-NS with a single instance of the PGW VNF and the nested Industrial App NFV-NS with a single instance of the App Server VNF. Then, the scaling operation directed to the composite NFV-NS considers passing from this initial IL to another IL with additional VNF instances (scale out) and passing from the "scaled out" IL to the initial IL (scale in). In all the composite NFV-NS deployments, the nested vEPC NFV-NS is deployed in NFVI-PoP#1 of AD1 and the nested Industrial App NFV-NS is deployed in NFVI-PoP#4 of AD2 or NFVI-PoP#2 of AD1 to compare the impact of the multi AD deployment (implying NSF) on the scaling operation when compared with a single AD deployment.

Table 11.1: Composite NFV-NS ILs under evaluation

Instantiation Level	**Description**	**Label in Figure 11.10 and Figure 11.11**
Big_vEPC	vEPC NFV-NS with 2 instances of PGW VNF	Compo-A, Fed-A
Big_App	Industrial App NFV-NS with 2 instances of App Server VNF	Compo-B, Fed-B
Big_vEPC_App	vEPC NFV-NS and Industrial App NFV-NS with two instances of PGW VNF and App Server VNF, respectively	Compo-C, Fed-C

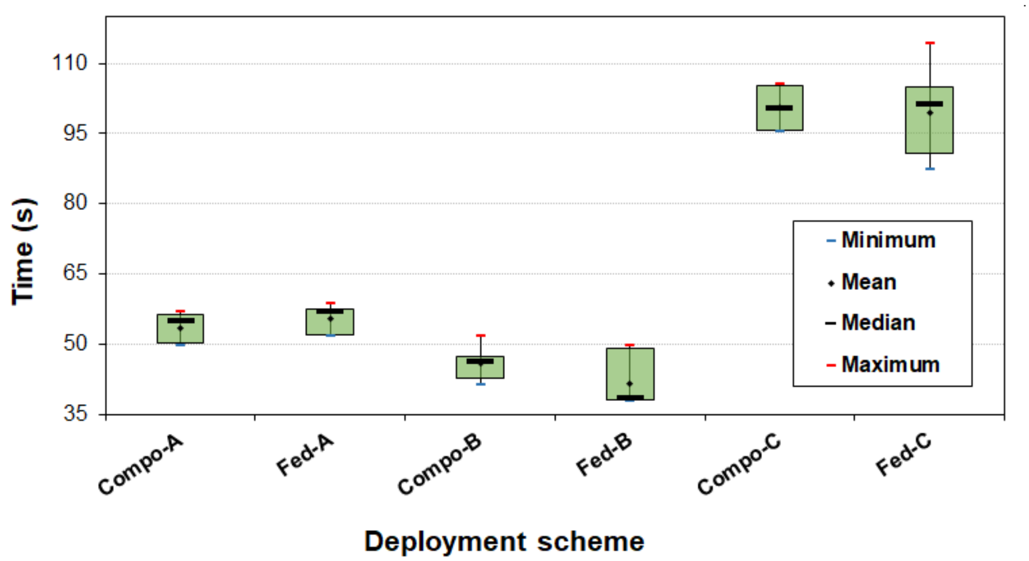


Figure 11.8: Composite Scale Out Operation Time vs Deployment scheme

Figure 11.8 shows that the time to pass from the default IL to the Big_vEPC IL (*Compo-A* case) is bigger than the one needed to pass to the Big_App IL (*Compo-B* case) (53.3 s on average vs 46 s), although in both cases, the considered ILs are adding an instance of a VNF with respect to the default IL and in the *Compo-A* case is needed to create three additional inter-nested connections and in the *Compo-B* case, five. After analysing the scaling logs provided by the SO module, and as observed in Chapter 8, most of the scaling time is due to the creation of the additional VM. In particular, although the added VNFs at the different nested NFV-NSs (PGW VNF in the vEPC NFV-NS and App Server VNF in the Industrial App NFV-NS) request the same amount of resources in terms of CPU, RAM and storage, the additional time to create the PGW VNF can be explained due to the load at the

NFVI-PoP#1 to support the rest of VNFs of the vEPC NFV-NS and due to the need of creating more networking ports (4 vs 2) and attach them to the different intra-PoP networks implementing the VLs. These experienced scaling times are in line with the ones presented in Chapter 8, and the different values experienced in this evaluation with respect to the one in Chapter 8, can be explained due to the different kind of used hardware and the amount of implied operations (e.g., update of inter-nested connections) derived from the structure of the associated NFV-NSs.

In the case of *Compo-C* case, the experienced time is around the sum of the previous times (around 100 s on average), since the scaling operations on the different nested NFV-NSs are performed successively as described in the workflow of Section 11.2. In this case, the added inter-nested connections are nine, more than the sum of the previous considered cases (eight). It is worth noting that OSM performs scaling operations sequentially on a per VNF basis rather than applying a certain degree of parallelization as occurs with the instantiation operation, so the OSM wrapper at the SO required to be developed accordingly. This explains why scaling operations are relatively more time-consuming than the overall instantiation time, which in this experimentation is around 230 s, in line with the values obtained in Chapter 10.

With respect to the cases were NSF is implied, specially in cases labelled as *Fed-B* and *Fed-C*, the experienced scaling times present more variability when compared to the *Compo-X* counterpart cases. This can be explained due to the polling operation the CD makes periodically (10s in this experimentation) to the PD to check the status of the scaling operation. In the case of *Fed-A*, the nested NFV-NS that it is scaled is in the CD domain, only requiring interaction with the PD to update the inter-nested connections, operation that it is not affected by the effect of the polling operation. Precisely, this interaction to establish inter-nested connection between ADs (from PD to CD and from CD to PD) explains the slightly bigger value of *Fed-A* with respect to *Compo-A* case.

In Figure 11.9, the scale in operation shows similar trends with the scale out operation attending to the variability of the experienced time between *Compo-X* and *Fed-X* cases and the cumulative time when applying multiple scaling operations. However, the overall experienced time is almost half the time with respect to the scale out time due to the time it takes to delete the previously added VNF instance, as observed in Chapter 8 and pointed out also in [70]. In addition to this, the time required to delete the inter-nested connections is less than the time required to create them.

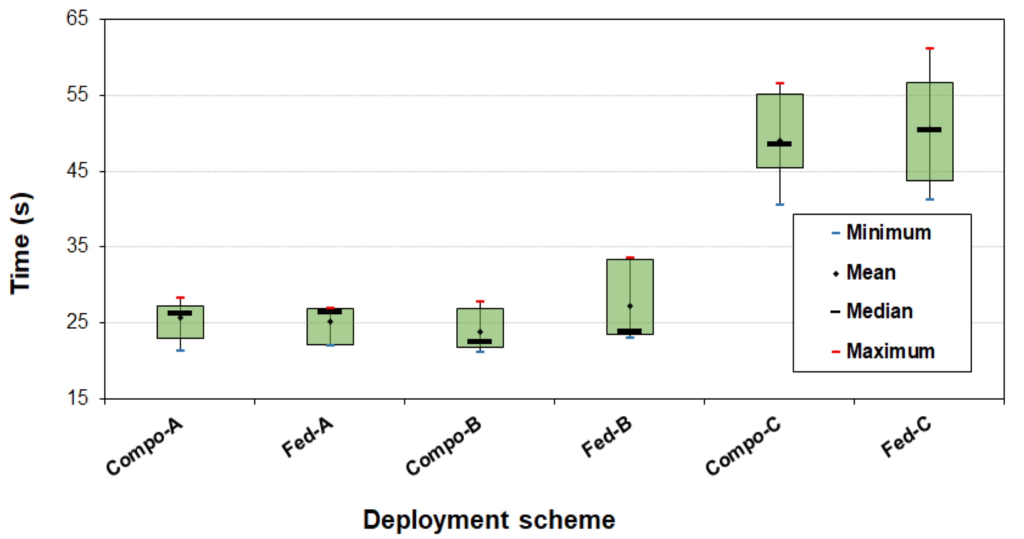


Figure 11.9: Composite Scale In Operation Time vs deployment scheme

Next, we present a more detailed profiling of the time sources contributing to the experienced scaling time when considering that a nested NFV-NS triggers an auto-scaling operation as a consequence of a SLA violation, as explained in Section 11.3. In particular, the following figures presents the average obtained results when considering the Big_App IL case of Table 11.1. The case when the composite NFV-NS is deployed in multiple NFVI-PoPs of a single AD is labelled as *Composite Multi-Pop*, and the case when the composite NFV-NS is deployed between two different NFVI-PoPs managed by different ADs is labelled as *Federation*. As in the former analysis, experiments have been repeated ten times.

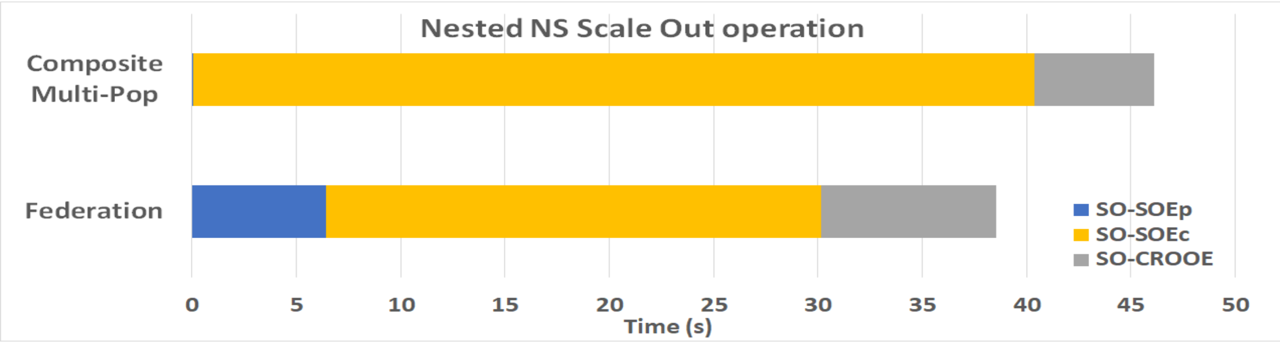


Figure 11.10: Nested NFV-NS Scale Out Operation Profiling

Figure 11.10 compares the average scaling time when performing the scaling out operation that adds a new App server VNF. It shows that the average scale out time of the *Federation* case is of 38.5 s. The different steps of the workflows presented in Section 11.3 have been grouped into three main groups. First, the SO-SOEp time accounts for the time spent at the SOEp module of the different ADs to process the corresponding scaling requests, retrieve and update information at the DBs and the time derived from the polling operation the CD makes to the PD to check that the nested NFV-NS has been successfully scaled. Actually, most of this experienced time is due to this polling operation, whose interval in this evaluation is set to 10 s. Second, the SO-SOEc time accounts for the time required by the SOEc to coordinate the scaling operation with the rest of sub-modules of the SO. Actually, the most time consuming operation (85% of the SO-SOEc time) is the interaction with the Core MANO platform to create the virtual machine (VM) for the new instance of the App server VNF. The SO-SOEc represents 62% out of the total time of the scale out process for the *Federation* case.

Finally, the SO-CROOE value, which represents the 22% of the total time, comprises the determination of the new inter-nested connections and its configuration at the different ADs after launching the LSA and interact with the corresponding MTP modules. Out of the total value of the SO-CROOE time of the *Federation* case, 72% corresponds to the time required to update the inter-nested connections at AD1 forwarding elements, hence reflecting the more complex structure of the underlying transport network. This component of the SO-CROOE time in the *Federation* case is similar to the overall SO-CROOE one experienced in the *Composite Multi-PoP*. This is due to the similar amount of forwarding elements to configure between NFVI-PoP#1 and the Inter-AD Link and between NFVI-PoP#1 and NFVI-PoP#2, respectively. In both cases, the time to determine the new inter-nested connections after creating the new VNF instance is less than 20 ms on average.

Surprisingly, the scale out operation at the *Federation* case requires less time than at the *Composite Multi-Pop* case although performing the polling operation and requiring the configuration of new inter-nested connections at both ADs. This is explained due to the time the different Core MANO platform instances, using different software releases, require to create the new instance of the App server VNF instance. After detecting this effect, additional tests revealed that OSM Release 7 instance at AD2 provided a faster VM creation procedure than OSM Release 6 at AD1. The average improvement in the SO-SOEc component is of around 40% between one version and the other of the Core MANO software, suggesting its maturing throughout its different releases towards a production-ready solution.

Finally, mention that the impact of the SO-SOEp time in the *Composite Multi-Pop* case is minimal (in the order of tens of ms) because it is mainly devoted to check the content of the scaling request and update the corresponding DB entries.

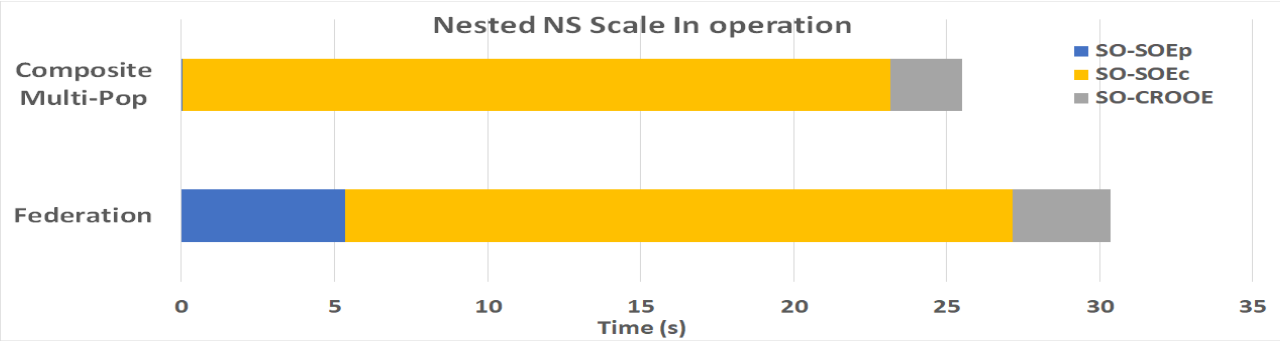


Figure 11.11: Nested NFV-NS Scale In Operation Profiling

Figure 11.11 compares the scale in case, when the nested Industrial App NS turns to its original IL with a single instance of the App server VNF once the processing load turns back to normal values as described in the NSD. The average value of this operation in the *Federation* case is of 30.7 s. Scale in operation is performed faster than scale out, as previously mentioned. In this situation, the *Federation* case presented a higher value than the *Composite Multi-Pop* case, as expected. Mostly, the difference in time comes from the polling operation time done in the CD and included in the SOEp value, which represents around 18% of the average experienced value. More noticeable differences in the overall time between considered deployments could be experienced as a contribution of the SO-CROOE component if AD2 presented a more complex transport topology. Finally, in this case, both releases of the employed Core MANO platform at the different ADs attain a similar performance when removing the VM associated to the previously added App server VNF.

11.5 Conclusions

This chapter concludes the second technical part of this work, considering the provision of SLA management of network slice instances deployed as composite NFV-NSs (also implying NSF) during run-time by means of scaling operations.

The complexity of this operation stems from the multiple possibilities at the time to deploy a network slice instance as a composite NFV-NS and the destination of the scaling request, which translate into the definition of multiple workflows. These workflows have been integrated on top of the SO architecture without disrupting any previous functionality thanks to its modular design. Additionally, the *SO-SO* interface coordinating the orchestration operations between ADs has been extended to handle the update of the inter-nested connections after successive scaling operations. Precisely, the handling of such resource orchestration logic and the consideration of the multiple scaling possibilities for composite NFV-NS deployments are the main contributions of this chapter, covering the gap found in related work.

The proposed workflows and the SO-SO interface extension have been validated in a multi-domain experimental setup, where the scaling time of a composite NFV-NS deployed in a single AD and in multiple AD has been evaluated and profiled. Experimental results showed that the time required to scale a nested NFV-NS part of a composite NFV-NS could be less than a minute (tens of seconds), being the operations related to the allocation/release of underlying resources (above all booting up new virtual machines for the new instances of VNFs), the most time consuming operations. These results are in line with the ones presented in Chapter 8, but a straightforward comparison is not fair due to the different kind of employed experimental setup and hardware equipment, and the implied operations derived from the structure of the implied NFV-NSs. Besides the used hardware, the kind of technique used for virtualization of computing resources, and the complexity of the underlying transport infrastructure (both at control and data plane level) connecting the involved NFVI-PoPs and ADs, we realised about the impact on the experienced scaling time of the performance of different releases of the same open source MANO platform at the time to allocate computing resources. All these aspects need to be taken into consideration in the SLA management of generic end-to-end network slice instance deployments.

Part IV

Conclusions and Future Work

Chapter 12

Conclusions and future work

12.1 Conclusions

This thesis has proposed and discussed an E2E management and orchestration architecture allowing the automated deployment and SLA maintenance of network slice instances (NSIs) in the form of network services (NFV-NSs) in the complex scenarios envisaged for next generation mobile networks. Mainly, the complexity of these scenarios consists of the heterogeneity of resources, its distributed presence across the network (within the same administrative domain (AD) or implying multiple administrative domains), and the need of continuously satisfying the diverse requirements demanded by the multiple network services sharing the same network substrate.

To achieve this end and provide answers to the research question posed in Chapter 4, the proposed E2E orchestration architecture bases on the integration of recent networking paradigms, namely SDN, NFV and Network slicing. Along the chapters 6 to 11 of this dissertation, we describe the design of this architecture following a bottom-up approach and we evaluate it and its associated workflows using an experimental infrastructure under different scenarios. It is worth mentioning that its design is aligned with the models proposed by related SDOs, hence promoting the applicability of the resulting architecture as well as its interoperability with other approaches sharing these guidelines.

First, we deal with the provision of networking resources in heterogeneous transport networks. Current transport networks suffer from a partitioned management driven by technology (optical, wireless) and equipment supplier, and from a tightly coupled control with the switching hardware, implying very complex and slow procedures to operate the network. This motivates the introduction of SDN as the technology enabler to achieve the required automation and dynamicity to *control* the forwarding behaviour of heterogeneous multi-layer networks. In particular, Chapter 6 analyses a hierarchical SDN approach suggesting that certain levels of hierarchy can cope with the network heterogeneity from a multi-technology perspective both at the data (wireless, optical) and the control plane (OpenFlow, GMPLS) in single administrative domains. Experimental results show that transport network resources implementing a mobile network service can be configured in the order of seconds (around 10.5 seconds on average). The proposed approach allowed also flexible recovery schemes in case of a link failure, whose recovery time depend on the degree of centralization, spanning this recovery time from hundreds of milliseconds to seconds. Moreover, this chapter also explores the *management* capabilities of the SDN paradigm by proposing an agent to configure heterogeneous wireless interfaces present at the forwarding elements thanks to the use of a common information model. This contributes to the design of applications pursuing network-wide optimization objectives.

The next step in the process of automating the deployment of network services, implies on one hand the allocation of further kind of resources, like computing resources to implement their constituent virtualised network functions (VNF) and on the other hand, the understanding of what these VNFs need and how they are related to orchestrate this allocation process. The canonical ETSI-NFV architecture concentrates all these functionalities in a single entity. However, our proposed E2E orchestration architecture splits them into the building blocks presented in Chapters 7 and 8. This is possible thanks to the use of resource abstraction mechanisms and an appropriate interface between these blocks. This split enhances the scalability of the proposed architecture, simplifies the operations at the upper orchestration level, and allows more flexible management scenarios where the multiple involved stakeholders (i.e., infrastructure providers and service providers) can present a higher degree of specialization. Chapter 7 focuses in the resource allocation process and presents the Mobile Transport and Computing platform (MTP) building block. A particular aspect tackled by the proposed MTP architecture is the abstraction of transport network resources. We propose two different alternatives to derive them and describe their implications in the allocation process, as a part of the whole orchestration process. Indeed, for this allocation task, we required the definition of an extended interface to solve the lack detected in the NFV architecture. This lack prevents the synergies of SDN and NFV paradigms to fully exploit the distributed computing resources available in single administrative domain scenarios while satisfying the requirements posed by network services. The network softwarization process introduced by the mentioned paradigms enables more modular architectures using an increasing number of interfaces, which may end up in an ecosystem of more specialised network stakeholders. Under such context, the proper definition of such interfaces is very relevant to promote interoperability and coordination among the different network actors.

Chapter 8 presents the Service Orchestrator (SO), which is the main element of the proposed orchestration architecture and manages the E2E control of services and resources required to deploy the requested network slice instances (NSIs) in the form of NFV-NSs. The architecture of the SO relies on the Service Manager (SM) concept, which acting as the real brain, performs the service and resource orchestration logic. Among the novelties introduced by the SM, we find the capability of dealing with transport network connections in distributed NFV-NS deployments, a wrapper system that allows the integration and the handling of the specificities of well-known MANO frameworks (e.g., OSM or cloudify) without altering the SO internal operational workflows and multiple SLA management schemes. This completes a *first version* of the E2E orchestration architecture focusing on single administrative domain scenarios. Furthermore, this chapter presents the evaluation of the resulting architecture over a real experimental infrastructure considering the instantiation and SLA maintenance of different network services involved in an automotive vertical use case. This evaluation, besides validating the operation of the proposed stack, shows that NSIs requiring different amount of resources can be deployed in the order of few minutes, in line with the 5G-PPP KPI of achieving a service creation time reduction from 90 hours down to 90 minutes. Additionally, it shows that SLA actions in the form of scaling operations changing the structure of the deployed network service can be performed in the same time order of magnitude. These results show similar trends with respect to comparable state-of-the-art approaches, where the most time consuming operations are those involving the allocation of resources (e.g., creation of VMs) at the underlying NFVI. It is worth noting that a direct benchmarking with other approaches is difficult due to different aspects, like availability of common network service descriptors and access to the same NFVI hardware and network topology/scenario.

The next part of the thesis investigates the required evolution of the *first version* of the E2E orchestration architecture to deal with multiple ADs. It is in this process when the most interesting concepts behind the network slicing paradigm are better reflected in the proposed architecture. Chapter 9 explains the network service composition (NSC) concept that allows representing an NFV-NS, the so-called composite NFV-NS, as a chain of multiple NFV-NSs, the so-called nested NFV-NSs. Thus, the required NSI is deployed as a composite NFV-NS, where its multiple nested NFV-NSs implement the different network slice subnet instances (NSSIs). The key concept to build such deployments is to understand how the different nested NFV-NSs are linked within a composite NFV-NS to perform appropriate orchestration and allocation operations aiming to the establishment of the required connectivities between nested NFV-NSs. Besides allowing the deployment of composite NFV-NSs from *scratch*, i.e., all at once, this understanding process needs to consider the case of using references to previously instantiated regular NFV-NSs to allow the sharing of an NSSI between multiple NSIs mapped to composite NFV-NSs deployments. The orchestration of composite NFV-NSs in a single AD scenario has been added at the SO block on top of the initial architecture, showing the modularity and flexibility of the SM concept to add new designed functionalities with a limited impact in previous available workflows.

Based on this, Chapter 10 finally tackles the problem of network service federation (NSF), where the different nested NFV-NSs (or NSSIs) part of a composite NFV-NS (or NSI) are deployed in multiple ADs, each one counting with its own MANO platform. To this end, we propose a detailed workflow and a complete definition of the *SO-SO* interface to communicate the SO blocks of the different involved ADs. The resulting procedure, conceptually initiated during the analysis of the NSC problem, and the associated interface handle not only the logic of service orchestration operations but also the logic of resource orchestration operations, which are a disregarded functionality in the literature. This consideration is fundamental to make E2E multi-domain orchestration and network slicing a reality in practice and allows opening the door to new business models based on the sharing and exchange of network service catalogues between service providers. This chapter also includes an evaluation of the resulting architecture in an experimental multi-administrative/multi-technology setup and how the NSF concept has been applied to an eHealth vertical use case in a close to real scenario. Evaluation results show that federated network services can be deployed in the order of several minutes (4-5 minutes in our experiments) and this time is mostly conditioned by the intrinsic operations associated to the deployment of the different nested NFV-NSs. The impact of federation-related operations and the latency between ADs is limited when compared with the overall creation time. The technical part of this thesis concludes with the chapter 11, where we mix the NSC/NSF concepts with the SLA assurance. Dealing with the scaling of composite NFV-NSs deployments adds another degree of complexity in the orchestration process due to the amount of deployment cases, the scope of the scaling operation (composite vs nested) and the interaction, implications among connected NSIs/NSSIs instances. Such considerations go beyond the work available in the literature. This reinforces the idea of flexibility, dynamicity, and programmability needed to realise 5G and next generation mobile network vision. The proposed procedures considering different deployment scenarios and scaling target have also been evaluated in a multi-administrative experimental setup considering an Industry 4.0 scenario, showing results in the same line that those obtained in Chapter 8 for SLA operations.

In general, we conclude that the time required to execute the orchestration processes, namely instantiation and scaling, evaluated throughout the different chapters depends on the actual resource allocation process. This process is impacted by the used hardware, the kind of technique used for virtualization of computing resources (e.g., virtual machines), and the complexity of the underlying transport infrastructure (both at control and data plane level) connecting the involved NFVI-PoPs and ADs. However, in this last evaluation considering scaling of composite NFV-NS, we also realised about the impact of different releases of the same open source MANO platform at the time to allocate computing resources.

Finally, we would like to stress that on the process to demonstrate the validity and practicality of our architecture and its concepts in real experimental infrastructures, different use cases in the context of multiple vertical industries, e.g., automotive and eHealth, have been considered. Besides the technical aspects, we realise that the transition towards next generation mobile networks implies new players in the network ecosystem (e.g., vertical industries), with different knowledge background with respect to current network stakeholders. In addition to the shift and the revolution that the mobile network transformation implies to already established stakeholders, the integration of such new players in the mobile ecosystem requires of further mutual efforts to advance in the service-based view of next generation mobile networks based on network softwarization techniques/concepts and the definition of open interfaces to achieve the desired and required network automation.

12.2 Future work

The way towards next generation mobile networks is never ending, and it involves multiple agents in the research, industry, and standardization communities exploring and defining it. With our research, we do not only expect that the contributions presented in this thesis represent a new step in this path, but also inspire new research directions to continue paving the way towards more dynamic and automated networks with a smooth integration among current network stakeholders, the ones incorporating at present and those yet to come. Next, we summarize some of these research lines to be investigated in the future.

Regarding the integration of network stakeholders within a single administrative domain, in Chapter 7, we presented two operational modes derived from the application of different strategies for the transport network resource abstraction (Infrastructure Abstraction (InA) and the Connectivity Service Abstraction (CSA)). The adoption of one or other mode may depend on the business relation established between the different network stakeholders operating the network infrastructure and orchestration processes, and may impact in the NSI deployment time and the degree of use of the available transport resources. An extensive analysis can be performed to assess this impact based on several aspects like the size of the NFVI infrastructure, the degree of distribution of both the NFVI-PoPs and the deployed network services. This last aspect, the creation of models describing the statistical arrival and departure of network service requests of different characteristics obeying to the needs of the network users is an important element in this proposed evaluation and which needs dedicated study because there is a lack in the literature. In front of the revolution introduced in the next generation mobile network with the dynamic/flexible deployment of network services, this may help operators to investigate suitable designs and to better size their NFVI infrastructures.

With respect to the relation between multiple administrative domains (ADs), related to the NSF process, several aspects can be researched to evolve the proposed procedure by including a more dynamic inter-domain service catalogue sharing scheme and exploring options adding more intelligence in the selection of the provider domain serving a nested NFV-NS than the practical one considered in Chapter 10. However, we can not omit potential security threads involved in such operations derived from the constant communications between consumer/provider domains. We consider that the use of Distributed Ledger Technologies (DLT) and the application of smart contracts can provide solutions to these aspects. Basically, a DLT, like Blockchain, is a digital system that records asset transactions -i.e., money,resources, information- by saving the transactions and their detail in different places at the same moment. When a transaction is done, its information and related metadata are saved in all nodes, making them all aware of that information and making it impossible to modify it without the others nodes knowing it. The smart contract is a set of binary code, similar to a computer application that runs on top of blockchain, immutable and operating independently (from its creator). To perform operations, users send transactions to the smart contract, which executes itself to generate an output data stored permanently on the Blockchain. This approach can be used by peering domains to securely share their service catalogue, and during the instantiation process, it can be used by a consumer domain to announce and receive offers from different provider domains to deploy the required nested NFV-NSs during the NSF process. It is worth noting that the application of DLT for network service orchestration purposes is starting to be considered in the literature. For instance, the work in [155] considers the use of DLT to deploy NSIs in multiple ADs following the delegation concept explained in Chapter 10, that is, without considering interconnection between NSSIs. Thus, another interesting point to research of the addition of DLT mechanisms in the NSF procedure is how it may affect to the procedure of interconnecting the different nested NFV-NSs at the different ADs and the way different ADs establish the data plane connectivity. As previously mentioned in Chapter 10, this is a fundamental aspect to effectively deploy E2E network slices instances. Some additional considerations to explore when adding DLT procedures within the proposed architecture are scalability aspects related with the amount of ADs and its impact on the signalling load and the introduced delay in the system's procedures (e.g., instantiation).

Finally, we cannot conclude this section without talking about the next big challenge in network automation, the *real* integration of Artificial Intelligence (AI) and Machine Learning (ML) techniques. AI/ML allows the introduction of smart close-loop operations to fully benefit next-generation mobile networks with automated service provision, operation, and assurance, as well as optimized slice management, fault management and resource orchestration. Indeed, this broad range of aspects are started to being analysed by diverse working groups of different SDOs, like ETSI ZSM [104], ETSI ENI [105], or O-RAN [106]. We mention the idea of *real* because during the last two decades, the idea of cognitive/autonomous networking has been present in the research community. However, there are multiple aspects that are preventing the success of AI/ML in networking: the complexity/heterogeneity/ever-increasing volume of the data to handle, the lack of ground-truth, standardized and representative datasets and the data dynamics. Additionally, another aspect to consider is the new dynamic network slice/service management procedures addressed in this work, which are in its early infancy and not yet fully applied to production environments. This contributes to add more complexity to this gigantic mission because it will continue transforming the network. Furthermore, the success in this mission will also depend on the network's ability to collect, aggregate and analyse relevant performance metrics at different levels (i.e., application, service, and infrastructure levels), and integrating all this mix in the available orchestration architectures. In this sense, in [156][1] and [157], we propose an initial drop to fill this ocean with our work on the architectural extensions of the SO module to include a data engineering pipeline interacting with an external AIML platform providing AI/ML as a service capabilities. These works provide an example of the initial integration of such elements into the orchestration architecture, focusing on the SO workflows to derive scaling operations based on the performance of a deployed network service and its service requirements. Further details of such architectural work are included in Appendix B. The next drop that we would like to contribute with in this process is on the research of architectural extensions required at the MTP level and the generation of suitable AI/ML models to identify anomalies and detect the root cause of problems affecting transport network resources to be able to trigger corrective actions.

[1] This article received the Best Fast Track paper award in the IEEE NFV-SDN'20 conference.

Bibliography

[1] ONF, “SDN architecture TR-502,” Technical Report 1.0, Open Networking Foundation (ONF), 2014.

[2] “ETSI ISG NFV Group.” Available online at: `https://www.etsi.org/technologies-clusters/technologies/nfv`, Accessed: April 2021.

[3] NGMN Alliance, “Description of Network Slicing Concept,” 5G White Paper 1, Next Generation Mobile Network (NGMN) Alliance, 2015.

[4] ETSI ISG NFV, “ETSI GR NFV-EVE 012 V3.1.1 (2017-12), "Network Functions Virtualisation (NFV) Release 3; Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architecture Framework,” Group Report (GR) 1, European Telecommunications Standards Institute (ETSI), 2017.

[5] 3GPP, “3GPP TS 23.501 (2017.12)-System Architecture for the 5G System, Release 15,” Technical Specification TS 23.501, Third Generation Partnership Project (3GPP), 2017.

[6] Patrick Marsch, Ömer Bulakci, Olav Queseth, Mauro Boldi, *5G System Design: Architectural and Functional Considerations and Long Term Research*. Wiley, May 2018.

[7] 3GPP, “3GPP TS 38.801 (2016.03)-Study on new radio access technology: Radio access architecture and interfaces",” Technical Specification TS 38.801, Third Generation Partnership Project (3GPP), 2016.

[8] “EU H2020 project 5G-Crosshaul, The integrated fronthaul/backhaul.” Available at: `https://5g-ppp.eu/xhaul/`, Accessed: April 2021.

[9] ETSI ISG NFV, “ETSI GR NFV-IFA 028 V3.1.1 (2018-01) Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on architecture options to support multiple administrative domains ,” Group Report (GR) 1, European Telecommunications Standards Institute (ETSI), 2018.

[10] J.Baranda, G. Avino, J. Mangues-Bafalluy, L. Vettori, R. Martínez, C.F. Chiasserini, C. Casetti, P. Bande, M. Giordanino, M. Zanzola, “Automated deployment and scaling of automotive safety services in 5G-Transformer,” in *IEEE Conference on Network Function Virtualization and Software Defined Networking, NFV-SDN 2019, Dallas, TX, USA, November 12-14, 2019*, pp. 1–2, 2019.

[11] J. Baranda, J. Mangues-Bafalluy, L. Vettori, R. Martínez, G. Avino, C. F. Chiasserini, C. Puligheddu, C. Casetti, J. Brenes, G. Landi, K. Kondepu, F. Paolucci, S. Fichera, L. Valcarenghi, “Arbitrating Network Services in 5G Networks for Automotive Vertical Industry,” in *Proceedings of the IEEE International Conference on Computer Communications (IEEE INFOCOM’20)*, July 2020.

[12] J. Baranda, J. Mangues-Bafalluy, L. Vettori, R. Martínez, K. Antevski, L. Girletti, C.J. Bernardos, K. Tomakh, D. Kucherenko, G. Landi, J. Brenes, X. Li, X. Costa-Pérez, F. Ubaldi, G. Imbarlina, M. Gharbaoui, “NFV Service Federation: enabling Multi-Provider eHealth Emergency Services,” in *Proceedings of the IEEE International Conference on Computer Communications (IEEE INFOCOM’20)*, July 2020.

[13] N. McKeown, “Software-Defined Networking, INFOCOM KeyNote talk.” Available online at: `http://yuba.stanford.edu/~nickm/talks/infocom_brazil_2009_v1-1.pdf`, 2009, Accessed: April 2021.

[14] “The Open Networking Foundation.” Available online at: `https://www.opennetworking.org/`, Accessed: April 2021.

[15] ONF, “SDN Architecture TR-521,” Technical Report 1.1, Open Networking Foundation (ONF), 2016.

[16] EU H2020 project 5G-Crosshaul, “D2.1, Study and assessment of physical and link layer technologies for 5G-Crosshaul,” Technical Report, 2016.

[17] ONF, “Openflow switch specification version 1.3.0,” Technical Report TS-006, Open Networking Foundation (ONF), 2012.

[18] L. Foundation, “Open Network Operating System (ONOS) Project.” Available online at: `https://onosproject.org"`, Accessed: April 2021.

[19] L. Foundation, “OpenDayLight SDN controller .” Available online at: `https://www.opendaylight.org/`, Accessed: April 2021.

[20] “RYU SDN Controller.” Available online at: `https://ryu-sdn.org/`, Accessed: April 2021.

[21] L. Foundation, “Open vSwitch: Production Quality, Multilayer Open Virtual Switch.” Available online at: `"https://www.openvswitch.org/"`, Accessed: April 2021.

[22] OpenStack Foundation, “The OpenStack project.” Available online at: `https://www.openstack.org/`, Accessed: April 2021.

[23] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, “P4: programming protocol-independent packet processors,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 87–95, 2014.

[24] ETSI, “NFV-Introductory White Paper at the SDN and OpenFlow World Congress,” White Paper, European Telecommunications Standards Institute (ETSI), October 22-24, 2012, Darmstadt-Germany.

[25] ETSI ISG NFV, “ETSI GS NFV 002 v1.1.1 (2013-10), Network functions virtualisation (NFV); Architectural framework,” Group Specification (GS) 1, European Telecommunications Standards Institute (ETSI), 2013.

[26] 3GPP, "3GPP TR 28.801 (2017-09)- Study on management and orchestration of network slicing for next generation network, Release 15)," Technical Report TR 28.801, Third Generation Partnership Project (3GPP), 2017.

[27] 5G-PPP, "5G Vision: The 5G Infrastructure Public Private Partnership: the next generation of communication networks and services." Available online at: `https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf`, March 2015.

[28] R. T. Fielding, *Architectural styles and the design of network-based software architectures.* PhD thesis, University of California, Irvine, 2000.

[29] "CPRI, Industry Initiative for a Common Public Radio Interface." Available at: `"http://www.cpri.info"`, Accessed: April 2021.

[30] "O-RAN Alliance, Operator Defined Open and Intelligent Radio Access Networks." Available at:`"https://www.o-ran.org/"`, Accessed: April 2021.

[31] ABI research, "Wireless Backhaul Evolution, Delivering next-generation connectivity," Technical Report 1.0, GSMA, 2021.

[32] F. Cavaliere, P. Iovanna, J. Mangues-Bafalluy, J. Baranda and J. Núñez, K. Lin, H. Chang, P. Chanclou, P. Farkas, J. Gomes, L. Cominardi, A. Mourad, A. de La Oliva, J. Alberto Hernández, D. Larrabeiti, A. Di Giglio, A. Paolicelli, P. Ödling, "Towards a unified fronthaul-backhaul data plane for 5G, The 5G-Crosshaul project approach," *Elsevier Computer Standards and Interfaces*, vol. 51, pp. 56–62, Mar. 2017.

[33] L. Li, Z. Mao, and J. Rexford, "Toward Software-Defined Cellular Networks," in *European Workshop on Software Defined Networking (EWSDN), Darmstadt, Germany, Oct. 25-26*, pp. 7–12, IEEE, 2012.

[34] K. Pentikousis, Y. Wang, and W. Hu, "Mobileflow: Toward software-defined mobile networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 44–53, 2013.

[35] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "Softcell: Scalable and flexible cellular core network architecture," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT), New Yourk, NY, USA*, pp. 163–174, 2013.

[36] C. J. Bernardos, A. de la Oliva, P. Serrano, A. Banchs, L. M. Contreras, H. Jin, and J. C. Zuniga, "An architecture for software defined wireless networking," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 52–61, 2014.

[37] Z. R. Zaidi, V. Friderikos, Z. Yousaf, S. Fletcher, M. Dohler, and H. Aghvami, "Will SDN be part of 5G?," *IEEE Communications Surveys and Tutorials*, vol. 20, pp. 3220–3258, 2018.

[38] V.-G. Nguyen, T.-X. Do, and Y. Kim, "SDN and Virtualization-based LTE Mobile Network architectures: A comprehensive survey," *Springer Wireless Personal Communications*, vol. 86, no. 3, pp. 1401–1438, 2016.

[39] A. Hurtado-Borras, J. Pala-Sole, D. Camps-Mur, and S. Sallent-Ribes, "SDN wireless backhauling for small cells," in *2015 IEEE International Conference on Communications, ICC 2015, London, United Kingdom, June 8-12, 2015*, pp. 3897–3902, IEEE, 2015.

[40] R. Santos and A. Kassler, "A SDN Controller Architecture for small cell Wireless Backhaul using a LTE control channel," in *Proceedings of the 17th. International Symposium on A World of wireless, Mobile and Multimedia Networks (WoWMoM), Coimbra, Portugal*, pp. 1–3, 2016.

[41] J. Núñez-Martínez, J. Baranda, I. Pascual, and J. Mangues-Bafalluy, "WiseHAUL: An SDN-empowered Wireless Small Cell Backhaul testbed," in *17th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks, WoWMoM 2016, Coimbra, Portugal, June 21-24*, pp. 1–3, IEEE, 2016.

[42] "EU H2020 project 5G-Xhaul, Dynamically Reconfigurable Optical-Wireless Backhaul/Fronthaul with Cognitive Control Plane for Small Cells and Cloud-RANs." Available at: `"https://www.5g-xhaul-project.eu/"`, Accessed: April 2021.

[43] I. F. Akyildiz, P. Wang, and S.-C. Lin, "Softair: A software defined networking architecture for 5g wireless systems," *Elsevier Computer Networks*, vol. 85, pp. 1–18, 2015.

[44] X. Costa-Perez, A. Garcia-Saavedra, X. Li, T. Deiss, A. de la Oliva, A. di Giglio, P. Iovanna, and A. Mourad, "5G-crosshaul: An SDN/NFV integrated fronthaul/backhaul transport network architecture," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 38–45, 2017.

[45] R. Casellas, R. Muñoz, R. Martínez, R. Vilalta, L. Liu, T. Tsuritani, I. Morita, V. López, O. González de Dios, J.P. Fernández, "SDN orchestration of OpenFlow and GMPLS Flexi-grid Networks with a Stateful Hierarchical PCE," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 1, pp. 106–117, 2015.

[46] R. Vilalta, A. Mayoral, R. Muñoz, R. Casellas, R. Martínez, "Hierarchical SDN Orchestration for Multi-technology Multi-domain Networks with ierarchical ABNO," in *Proceedings of the 41st. European Conference on Optical Communications (ECOC), Valencia, Spain, 27 Sept. - 1 Oct.*, 2015.

[47] D. King, A. Farrell, "A PCE-Based Architecture for Application-Based Network Operations," Technical Report IETF RFC 7491, Internet Engineering Task Force (IETF), 2015.

[48] K.Katsalis, B. Rofoee, G. Landi, J.F Riera, K. Kousias, M. Anastasopoulos, L. Kiraly, A. Tzanakaki,T. Korakis, "Implementation experience in multi-domain SDN: Challenges, consolidation and future directions," *Elsevier Computer Networks*, vol. 129, no. 1, pp. 142–158, 2017.

[49] Jin Seek Choi , "Hierarchical Distributed Orchestration Framework for Multi-Domain SDTNs," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 12, pp. 1125–1135, 2017.

[50] D. Giatsios, K.Choumas, P. Flegkas, T. Korakis, and D. Camps-Mur, "SDN implementation of slicing and fast failover in 5g transport networks," in *2017 European Conference on Networks and Communications, EuCNC 2017, Oulu, Finland, June 12-15, 2017*, pp. 1–6, IEEE, 2017.

[51] R. Vilalta, A. Mayoral, J. Baranda, J. Núñez, R. Casellas, R. Martínez, J. Mangues-Bafalluy, R. Muñoz, "Hierarchical SDN Orchestration of Wireless and Optical Networks with E2E Provisioning and Recovery for Future 5G Networks," in *Proceedings of the 39th. Optical Fiber Communications Conference and Exhibition (OFC), Anaheim, CA, USA ,20-24 March*, pp. 1–3, IEEE, 2016.

[52] J. Mangues-Bafalluy, J. Núñez, R. Casellas, A. Mayoral, J. Baranda, J.X. Salvat, A. Garcia-Saavedra, R. Vilalta*, I. Pascual, Xi Li, R. Martínez, R. Muñoz, "Experimental Evaluation of Hierarchical Control over Multi-Domain Wireless/Optical Networks," in *Proceedings of the 26th European Conference on Networks and Communications (EuCNC), Oulu, Finland, June 12-15*, pp. 1–5, IEEE, 2017.

[53] R. Vilalta, V. López, A. Mayoral, N. Noshikane, M. Ruffini, D. Siracusa, R. Muñoz, " The Need for a Control Orchestration Protocol in Research Projects on Optical Networking," in *Proceedings of the 24th. European Conference on Networks and Communications (EuCNC), Paris, France, 29 June-2 July*, pp. 340–344, IEEE, 2015.

[54] "EU/Japan Project STRAUSS, Scalable and efficient orchestration of Ethernet services using software-defined and flexible optical networks." Available at: `https://cordis.europa.eu/project/id/608528"`, Accessed: April 2021.

[55] D. Giatsios, K. Choumas, P. Flegkas, T. Korakis, J. J. Aleixendri Cruelles, and D. Camps-Mur, "Design and evaluation of a hierarchical SDN control plane for 5g transport networks," in *2019 IEEE International Conference on Communications, ICC 2019, Shanghai, China, May 20-24, 2019*, pp. 1–6, IEEE, 2019.

[56] ONF, "Transport API (TAPI) 2.0 Overview," Technical Report Version 0.0., Open Networking Foundation (ONF), 2017.

[57] OIF, "SDN Transport API Interoperability Demonstration," White Paper, Optical Internetworking Forum(OIF), 2018.

[58] ETSI ISG NFV, "Mapping ETSI-NFV onto Multi-Vendor, Multi-Domain Transport SDN," Technical Report PoC 42, European Telecommunications Standards Institute (ETSI), 2017.

[59] M. Garrich Alabarce, and A. Bravalheri, "Overview of South-Bound Interfaces for Software-Defined Optical Networks," in *Proceedings of the 20th International Conference on Transparent Optical Networks, Bucarest, Romania, 1-5 July*, IEEE, 2018.

[60] OpenConfig, "Vendor neutral, model-driven network management designed by users." Available at: `"http://www.openconfig.net/"`, Accessed: April 2021.

[61] O. F. Yilmaz et al., "Automated management and control of a multi-vendor disaggregated network at the L0 layer," in *Proceedings of the 41th. Optical Fiber Communications Conference and Exhibition (OFC), San Diego, CA, USA, March 11-15*, pp. 1–3, IEEE, 2018.

[62] R. Enns, Martin Björklund, A. Bierman, J. Schönwälder, "Network configuration protocol (NETCONF)," Technical Report IETF RFC 6241, Internet Engineering Task Force (IETF), 2011.

[63] A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol," Technical Report IETF RFC 8040, Internet Engineering Task Force (IETF), 2017.

[64] M. Björklund, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)," Technical Report IETF RFC 6020, Internet Engineering Task Force (IETF), 2010.

[65] M. Channegowda , R. Nejabati , M.R. Fard , S. Peng , N. Amaya , G. Zervas , D. Sime- onidou , R. Vilalta , R. Casellas , R. Martínez , et al., "Experimental demonstration of an openflow based software-defined optical network employing packet fixed and flexible dwdm grid technologies on an international multi-domain testbed," *OSA Optics Express*, vol. 21, no. 5, pp. 5487–5498, 2013.

[66] R. Santos and A. Kassler, "Small Cell Wireless Backhaul Reconfiguration using Software-Defined Networking," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, (San Francisco, CA, USA), pp. 1–6, IEEE, March 2017.

[67] ONF, "Microwave Information Model TR-532," Technical Report 1.0, Open Networking Foundation (ONF), 12 2016.

[68] Nathan F. Saraiva de Sousa, Danny A. Lachos Perez, Raphael V. Rosa, Mateus A. S. Santos, and Christian Esteve Rothenberg, "Network Service Orchestration: A Survey," *Elsevier Computer Communications*, vol. 142-143, pp. 69 – 94, 2019.

[69] ETSI ISG NFV, "ETSI GS NFV-MAN 001 V1.1.1 (2014-12), Network Functions Virtualisation (NFV); Management and Orchestration," Group Specification (GS) 1, European Telecommunications Standards Institute (ETSI), 2014.

[70] P. Trakadas, P. Karkazis, H. C. Leligou, T. B. Zahariadis, F. Vicens, A. Zurita, P. Alemany, T. Soenen, C. Parada, J. Bonnet, E. Fotopoulou, A. Zafeiropoulos, E. Kapassa, M. Touloupou and D. Kyriazis, "Comparison of Management and Orchestration Solutions for the 5G Era," *MDPI Journal of Sensors and Actuator Networks*, vol. 9, no. 1, p. 4, 2020.

[71] Girma M. Yilma, Zarrar F. Yousaf, V. Sciancalepore, X. Costa-Perez, "Benchmarking Open Source NFV MANO Systems: OSM and ONAP," *Elsevier Computer Communications*, vol. 161, pp. 86 – 98, 2020.

[72] M. Peuster, M. Marchetti, G. García-de-Blas and H. Karl, "Automated testing of NFV orchestrators against carrier-grade multi-PoP scenarios using emulation-based smoke testing," *EURASIP Journal on Wireless Communications Networks*, vol. 2019, p. 172, 2019.

[73] ETSI, "The Open Source MANO (OSM) project." Available online at: `https://osm.etsi.org/`, Accessed: April 2021.

[74] Kubernetes, "Production-Grade Container Orchestration." Available online at: `https://kubernetes.io/es/`, Accessed: April 2021.

[75] ETSI ISG NFV, "ETSI GS NFV-IFA009 V1.1.1 (2016-07), Network Functions Virtualisation (NFV); Management and Orchestration; Report on Architectural Options," Group Specification (GS) 1, European Telecommunications Standards Institute (ETSI), 2016.

[76] Canonical, "Juju." Available online at: `https://docs.jujucharms.com/2.4/en/about-juju`, Accessed: April 2021.

[77] Linux Foundation, "ONAP – Open Network Automation Platform." Available online at: `https://www.onap.org`, Accessed: April 2021.

[78] OASIS, "Topology and Orchestration Specification for Cloud Applications (TOSCA) Simple Profile in YAML Version 1.2." Available online at: `http://docs.oasis-open.org/tosca/`, Accessed: April 2021.

[79] "The Docker Website." https://www.docker.com/.

[80] Gigaspace, "Cloudify, Cloud and NFV orchestration platform." Available online at: `https://cloudify.co/`, Accessed: April 2021.

[81] OpenStack Foundation, "Tacker - OpenStack." Available online at: `https://wiki.openstack.org/wiki/Tacker`, Accessed: April 2021.

[82] Fraunhofer and Technical University of Berlin, "Open Baton: An open source reference implementation of the ETSI Network Function Virtualization MANO specification." Available online at: `http://openbaton.github.io/`, Accessed: April 2021.

[83] Linux Foundation, "OPNFV - Open Platform for NFV." Available Online at: `https://www.opnfv.org/`, Accessed: April 2021.

[84] "EU H2020 project 5G-TRANSFORMER, 5G-Mobile Transport Platform for Verticals." Available at: `"http://5g-transformer.eu/"`, Accessed: April 2021.

[85] M. Kourtis, M. J. McGrath, G. Gardikis, G. Xilouris, V. Riccobene, P. Papadimitriou, E. Trouva, F. Liberati, M. Trubian, J. Batalle, H. Koumaras, D. Dietrich, A. Ramos, J. F. Riera, J. Bonnet, A. Pietrabissa, A. Ceselli, and A. Petrini, "T-NOVA: an open-source MANO stack for NFV infrastructures," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 586–602, 2017.

[86] "EU H2020 project SONATA, Agile Service Development and Orchestration in 5G Virtualized Networks." Available at: `"http://www.sonata-nfv.eu/"`, Accessed: April 2021.

[87] "EU H2020 project 5G-TANGO, 5G Development and Validation Platform for global Industry-specific Network Services and Apps." Available at: `"https://5gtango.eu/"`, Accessed: April 2021.

[88] D. Dietrich, A. Abujoda, A. Rizk, and P. Papadimitriou, "Multi-Provider Service Chain Embedding With Nestor," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 91–105, 2017.

[89] G. Sun, Y. Li, V. Chang, "Service function chain orchestration across multiple domains: A full mesh aggregation approach," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1175–1191, 2018.

[90] Q. Zhang, X. Wang, I. Kim, P. Palacharla, and T. Ikeuchi, "Vertex-centric computation of service function chains in multi-domain networks," in *IEEE NetSoft Conference and Workshops (NetSoft 2016), Seoul, South Korea, June 6-10*, pp. 211–218, IEEE, 2016.

[91] A. Abujoda and P. Papadimitriou, "Distnse: Distributed network service embedding across multiple providers," in *Proceedings of the 8th International Conference on Communication Systems and Networks, (COMSNETS), Bangalore, India, January 5-10*, pp. 1–8, IEEE, 2016.

[92] MEF, "Lifecycle Service Orchestration (LSO): Reference Architecture and Framework," Service Operations Specification MEF 55, Metro Ethernet Forum (MEF), 3 2016.

[93] ETSI ISG NFV, "ETSI GS NFV-IFA 030 V3.2.2 (2019-05) Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Multiple Administrative Domain Aspect Interfaces Specification ," Group Specification (GS) 1, European Telecommunications Standards Institute (ETSI), 2019.

[94] ETSI ISG NFV, "ETSI GS NFV-IFA013 V2.4.1 (2018-02), Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification ," Group Specification (GS) 1, European Telecommunications Standards Institute (ETSI), 2018.

[95] "EU H2020 project 5G-Ex, 5G-Exchange." Available at: `"https://5g-ppp.eu/5gex/"`, Accessed: April 2021.

[96] "EU H2020 project Vital, VIrtualized hybrid satellite-TerrestriAl systems for resilient and fLexible future networks." Available at: `"https://cordis.europa.eu/project/id/644843"`, Accessed: April 2021.

[97] A. Sgambelluri, F. Tusa, M. Gharbaoui, E. Maini, L. Toka, J. Martín-Pérez, F. Paolucci, B. Martini, W. Y. Poe, J. M. Hernandes, A. Muhammed, A. Ramos, Ó. G. de Dios, B. Sonkoly, P. Monti, I. Vaishnavi, C. J. Bernardos, and R. Szabó, "Orchestration of network services across multiple

operators: The 5g exchange prototype," in *2017 European Conference on Networks and Communications, EuCNC 2017, Oulu, Finland, June 12-15, 2017*, pp. 1–5, IEEE, 2017.

[98] A. Francescon, G. Baggio, R. Fedrizzi, R. Ferrús, I. Grida Ben Yahia and Roberto Riggio, "X-MANO: Cross-domain management and orchestration of network services," in *2017 IEEE Conference on Network Softwarization (NetSoft 2017), Bologna, Italy, July 3-7*, pp. 1–5, IEEE, 2017.

[99] "EU H2020 project 5G-EVE, 5G European Validation Platform for Extensive Trials." Available at: `"https://www.5g-eve.eu/"`, Accessed: April 2021.

[100] "EU H2020 project 5G-VINNI, 5G Verticals Innovation Infrastructure." Available at: `"https://www.5g-vinni.eu/"`, Accessed: April 2021.

[101] "EU H2020 project 5GENESIS, 5th Generation End-to-end Network, Experimentation, System Integration, and Showcasing ." Available at: `"https://5genesis.eu/"`, Accessed: April 2021.

[102] N. Uniyal, A. Siddique Muqaddas, D. Gkounis, A. Bravalheri, S. Moazzeni, F. Sardis, M. Dohler, R. Nejabati, D. Simeonidou, "5GUK Exchange: Towards Sustainable End-to-End Multi-Domain Orchestration of Softwarized 5G Networks," *Elsevier Computer Networks and ISDN Systems*, vol. 178, p. 107297, May 2020.

[103] A. Mestres, A. Rodriguez-Natal, J. Carner, P. Barlet-Ros, E. Alarcón, M. Solé, V. Muntés-Mulero, D. Meyer, S. Barkai, M. J. Hibbett, G. Estrada, K. Ma'ruf, F. Coras, V. Ermagan, H. Latapie, C. Cassar, J. Evans, F. Maino, J. Walrand, and A. Cabellos, "Knowledge-defined networking," *ACM SIGCOMM Computer Communication Review*, vol. 47, pp. 2–10, Sept. 2017.

[104] ETSI, "Zero Touch Network and Service Management (ZSM)." Available online at: `https://www.etsi.org/committee/zsm`, Accessed: April 2021.

[105] ETSI, "Experiential Networked Intelligence (ENI)." Available online at: `https://www.etsi.org/committee/eni`, Accessed: April 2021.

[106] O-RAN Working Group 2, "AI/ML workflow description and requirements," Technical Report O-RAN.WG2.AIML-v01.01, ORAN-Alliance, 2020.

[107] W. Mohr, "The 5G Infrastructure Association, ITU-R Workshop on IMT-2020 terrestrial radio interfaces." Available online at: `https://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2020/Documents/S03-1_5GPPP%20EU_Rev.pd`, 2017, Accessed: April 2021.

[108] Centre Tecnològic de Telecomunications de Catalunya (CTTC/CERCA), "EXTREME Testbed®." Available at: `http://networks.cttc.es/mobile-networks/extreme_testbed/`, Accessed: April 2021.

[109] Centre Tecnològic de Telecomunications de Catalunya (CTTC/CERCA), "ADRENALINE Testbed®." Available at: `http://networks.cttc.es/ons/adrenaline/`, Accessed: April 2021.

[110] IEEE, "IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz.," technical report, Institute of Electrical and Electronics Engineers (IEEE), 2013.

[111] Candela Technologies, "Ath10k CT 10.1 Firmware." Available online at: `http://www.candelatech.com/ath10k-10.1.php`, Accessed: April 2021.

[112] IEEE, "Standard for Information technology–Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band.," technical report, Institute of Electrical and Electronics Engineers (IEEE), 2012.

[113] "The Graphical Network Simulator-3 (GNS3)." Available online at: `https://www.gns3.com/`, Accessed: April 2021.

[114] OpenStack Foundation, "Devstack: All-in-one single machine." Available online at: `https://docs.openstack.org/devstack/latest/`, Accessed: April 2021.

[115] J. Baranda, J. Mangues-Bafalluy, I. Pascual, J. Núñez-Martínez, J. L. de la Cruz, R. Casellas, R. Vilalta, J. X. Salvat, and C. Turyagyenda, "Orchestration of End-to-End Network Services in the 5G-Crosshaul Multi-Domain Multi-Technology Transport Network," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 184–191, 2018.

[116] J. Baranda, I. Pascual, J. Mangues-Bafalluy and J. Núñez-Martínez, "Wireless Interface Agent for SDN mmWave Multi-hop Networks: Design and Experimental Evaluation," in *Proceedings of the 2nd ACM Workshop on Millimeter Wave Networks and Sensing Systems (mmNets), co-located with ACM MobiCom 2018, New Delhi, India, October 29*, pp. 9–14, ACM, 2018.

[117] EU H2020 project 5G-Crosshaul, "D3.1, XFE/XCI design at year 1, specification of southbound and northbound interface," Technical Report, 2016.

[118] A. Mayoral López de Lerma, *Integrated IT and SDN Orchestration of multi-domain multi-layer transport networks.* PhD thesis, Politechnical University of Catalonia (UPC), Spain, 2019.

[119] Core Network Dynamics GmbH (Twilio), "The OpenEPC project." Available online at: `https://sites.google.com/a/corenetdynamics.com/openepc/home`, Accessed: April 2021.

[120] J. Vestin and A. Kassler, "Low frequency assist for mmWave backhaul - the case for SDN resiliency mechanisms," in *2017 IEEE International Conference on Communications Workshops, ICC Workshops 2017, Paris, France, May 21-25*, pp. 205–210, IEEE, 2017.

[121] N.L.M., van Adrichem, B.J. Van Asten, and F.A. Kuipers, "Fast recovery in software-defined networks," in *Third European Workshop on Software Defined Networks (EWSDN), Budapest, Hungary, September 1-3*, pp. 61–66, IEEE Computer Society, 2014.

[122] A. Tirumala, F. Qin, J. Dugan, and K. Gibbs, *Iperf: The tcp/udp bandwidth measurement tool*, 2005. Available at: `http://dast.nlanr.net/Projects`.

[123] R. Santos, H. Ogawa, G. Khanh Tran, K. Sakaguchi, A. Kassler, "Turning the Knobs on OpenFlow-Based Resiliency in mmWave Small Cell Meshed Networks," in *2017 IEEE Globecom Workshops, Singapore, December 4-8*, pp. 1–5, IEEE, 2017.

[124] D. Steinmetzer, A. Loch, A. García-García, J. Widmer, and M. Hollic, "Mitigating Lateral Interference: Adaptive Beam Switching for Robust Millimeter-Wave Networks," in *Proceedings of the 1st ACM Workshop on Millimeter-Wave Networks and Sensing Systems (mmNets@MobiCom 2017), Snowbird, UT, USA, October 16 - 20,*, pp. 29–34, ACM, 2017.

[125] K. Nguyen, M. Golam, K. Ishizu, and F. Kojima, "Empirical Investigation of IEEE 802.11ad Network," in *2017 IEEE International Conference on Communications Workshops, ICC Workshops 2017, Paris, France, May 21-25*, pp. 192–197, IEEE, 2017.

[126] J. Baranda, L. Vettori, R. Martínez, J. Mangues-Bafalluy, “A Mobile Transport Platform Interconnecting VNFs over a Multi-Domain Optical/Wireless Network: Design and Implementation ,” in *24th International Conference on Optical Network Design and Modelling (ONDM’20)*, IEEE, May 2020.

[127] R. Martínez, L. Vettori, J. Baranda, J. Mangues-Bafalluy, E. Zeydan, “Experimental Validation of Compute and Network Resource Abstraction and Allocation Mechanisms within a NFV Infrastructure ,” in *Procs of the 2021 IFIP/IEEE International Symposium on Integrated Network Management (IEEE IM 2021)*, IEEE, May 2021.

[128] ETSI ISG NFV, “ETSI GS NFV-IFA005 V3.1.1 (2018-08) Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification ,” Group Specification (GS) 1, European Telecommunications Standards Institute (ETSI), 2018.

[129] ETSI ISG NFV, “ETSI GR NFV-IFA022 v3.1.1 (2018-04) , Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on Management and Connectivity for Multi-Site Services,” Group Report (GR) 1, European Telecommunications Standards Institute (ETSI), 2018.

[130] CTTC/CERCA, “mobilE tranport pLatform for multi-tEChnology neTwoRk infrAstructure (ELECTRA).” Available online at: `https://gitlab.cttc.es/cttc-cnd/ELECTRA`, Accessed: April 2021.

[131] J. Yen, “Finding the k shortest loopless paths in a network,” *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[132] A. Bravalheri, A. Siddique Muqaddas, N. Uniyal, R. Casellas, R. Nejabati and D. Simeonidou, “VNF Chaining Across Multi-PoPs in OSM using Transport API,” in *Optical Fiber Communications Conference and Exhibition, OFC 2019, San Diego, CA, USA, March 3-7*, pp. 1–3, IEEE, 2019.

[133] J. Baranda, I. Pascual, M. Requena, and J. Mangues-Bafalluy, “Deploying a containerized ns-3/LENA-based LTE mobile Network Service through the 5G-TRANSFORMER platform,” in *Proceedings of the 4th IEEE Conference on Network Functions Virtualization and Software Defined Networking (IEEE NFV-SDN 2018), 27-29 November, Verona (Italy)*, pp. 1–2, IEEE, 2018.

[134] J. Mangues-Bafalluy, J. Baranda, I. Pascual, R. Martínez, L. Vettori, G. Landi, A. Zurita, D. Salama, K. Antevski, J. Martin-Perez, D. Andrushko, K. Tomakh, B. Martini, X. Li, J. X. Salvat, “5G-TRANSFORMER Service Orchestrator: Design Implementation and Evaluation,” in *Proceedings of the 28th European Conference on Networks and Communications (EuCNC), Valencia, Spain, June 18-21*, pp. 31 –36, 2019.

[135] X. Li, T. Deiss, J. Mangues-Bafalluy, J. Baranda, X. Costa-Pérez, G. Landi, C. J. Bernardos, P. Iovanna, A. Zurita, P. Bertin, “Automating Vertical Services Deployments over the 5GT Platform,” *IEEE Communications Magazine*, vol. 58, no. 7, pp. 44–50, 2020.

[136] X. Li, C.F.Chiasserini, J.Mangues-Bafalluy, J.Baranda, G.Landi, B.Martini, X.Costa-Perez, C.Puligheddu, L.Valcarenghi, “Automated Service Provisioning and Hierarchical SLA Management in 5G Systems,” *IEEE Transactions on Service and Network Management*, 2021.

[137] K. Antevski, J. Martín-Pérez, N. Molner, C. F. Chiasserini, F. Malandrino, P. Frangoudis, A. Ksentini, X. Li, J. X. Salvat Lozano, R. Martínez, I. Pascual, J. Mangues-Bafalluy, J. Baranda, B. Martini, and M. Gharbaoui, “Resource Orchestration of 5G Transport Networks for Vertical

Industries," in *29th IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2018, Bologna, Italy, September 9-12*, pp. 158–163, IEEE, 2018.

[138] ETSI ISG NFV, "ETSI GS NFV-IFA014 V3.3.1 (2019-09), Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Network Service Templates Specification," Group Specification (GR) 1, European Telecommunications Standards Institute (ETSI), 2019.

[139] ETSI ISG NFV, "ETSI GS NFV-IFA011 V3.2.1 (2019-04), Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; VNF Descriptor and Packaging Specification," Group Specification (GS) 1, European Telecommunications Standards Institute (ETSI), 2019.

[140] CTTC/CERCA, "Multi-nfvo fEderated ServiCe orchestrAtion pLatform (MESCAL)." Available online at: `https://gitlab.cttc.es/cttc-cnd/MESCAL`, Accessed: April 2021.

[141] ETSI, "ETSI 302 637-2–v1.3.1, Intelligent Transport Systems (ITS); Vehicular Communications; Basic set of applications; part 2: Specification of cooperative awareness basic service," tech. report, European Telecommunications Standards Institute (ETSI), 2014.

[142] ETSI, "ETSI 302 637-3–v1.3.1, Intelligent Transport Systems (ITS); Vehicular Communications; Basic set of applications; part 3: Specifications of decentralized environmental notification basic service," tech. report, European Telecommunications Standards Institute (ETSI), 2019.

[143] G. Avino, M. Malinverno, C. Casetti, C.F. Chiasserini, F. Malandrino, M. Rapelli, G. Zennaro, "Support of Safety Services through Vehicular Communications: The Intersection Collision Avoidance Use Case," in *2018 IEEE International Conference of Electrical and Electronic Technologies for Automotive, Milan, Italy, 9-11 July*, IEEE, 2018.

[144] Claudio Casetti, Carla-Fabiana Chiasserini, Nuria Molner, Jorge Martín-Pérez, Thomas Deiß, Cao-Thanh Phan, Farouk Messaoudi, Giada Landi, Juan Brenes Baranzano, "Arbitration among vertical services," in *Proceedings of the 29th IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications, (PIMRC), Bologna, Italy, September 9-12*, pp. 153–157, IEEE, 2018.

[145] J. Baranda, J. Mangues-Bafalluy, L. Vettori, R. Martínez, G. Landi, K. Antevski, "Demo: Composing Services in 5G-TRANSFORMER," in *Proceedings of the 20th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Catania (ACM MobiHoc'19), Italy, July 2-5*, pp. 407–408, 2019.

[146] P. Alemany, T. Soenen, C. Parada, J. Bonnet, R. Casellas, R. Martínez, R. Vilalta, R. Muñoz, J. Luis de la Cruz, A. Pol, A. Roman, P. Trakadas, P. Karkazis, M. Touloupou, E. Kapassa, D. Kyriazis, "Network Slicing Over A Packet/Optical Network For Vertical Applications Applied To Multimedia Real-Time Communications," in *IEEE Conference on Network Function Virtualization and Software Defined Networks (IEEE NFV-SDN), Dallas, TX, USA, November 12-14*, pp. 1–2, IEEE, 2019.

[147] J. Baranda, J. Mangues-Bafalluy, R. Martínez, L. Vettori, K. Antevski, C. J. Bernardos, X. Li, "Realizing the network service federation vision: Enabling automated multidomain orchestration of network services," *IEEE Vehicular Technology Magazine*, vol. 15, no. 2, pp. 48–57, 2020.

[148] J. Baranda, J. Mangues-Bafalluy, R. Martínez, L. Vettori, K. Antevski, C. J. Bernardos, X. Li, "5G-TRANSFORMER meets Network Service Federation: design, implementation and evaluation ," in *6th IEEE Conference on Network Softwarization (NetSoft 2020) Ghent, Belgium, June 29 - July 3*, pp. 175–179, IEEE, 2020.

[149] 5G PPP Infrastructure Association, "5G PPP Infrastructure -Trials and Pilots Brochure (N°2)." Available online at: `https://5g-ppp.eu/wp-content/uploads/2020/12/5GInfraPPP_10TPs_Brochure2.pdf/`, December 2020, Accessed: April 2021.

[150] "5TONIC Open Innovation Lab." Available at: `https://www.5tonic.org/`, Accessed: April 2021.

[151] J. Baranda, J. Mangues-Bafalluy, L. Vettori, R. Martínez, "Demo: Scaling Composite NFV-Network Services," in *Procs of the 21st ACM International Symposium on Theory, Algorithmic, Foundations, and Protocol Design for Mobile Networks and Mobile Computing (ACM MobiHoc 2020), October 11-14 2020, virtual event*, pp. 307–308, ACM, 2020.

[152] J. Baranda, J. Mangues-Bafalluy, L. Vettori , R. Martínez, E. Zeydan, "Scaling Federated Network Services: Managing SLAs in Multi-Provider Industry 4.0 Scenarios," in *Procs of the IEEE International Conference on Computer Communications (IEEE INFOCOM'21), May 10-13 2021, virtual event*, IEEE, 2021.

[153] J. Baranda, J. Mangues-Bafalluy, L. Vettori , R. Martínez, E. Zeydan, "Enabling SLA Management of Federated Network Services through Scaling Operations," in *submitted for publication in the IEEE/IFIP Network Operations and Management Symposium (IEEE NOMS'22), April 2022, Budapest, Hungary*, IEEE, 2022.

[154] 5G ACIA, "5G Non-Public Networks for Industrial Scenarios White Paper," white paper, 5G Alliance for Connected Industries and Automation (5G ACIA), July 2019.

[155] P. Alemany, R. Vilalta, R. Muñoz, R. Casellas, R. Martínez, "Peer-to-Peer Blockchain-based NFV Service Platform for End-to-End Network Slice Orchestration Across Multiple NFVI Domains," in *3rd IEEE 5G World Forum (5GWF 2020), Bangalore, India, September 10-12*, pp. 151–156, IEEE, 2020.

[156] J. Baranda, J. Mangues-Bafalluy, E. Zeydan, L. Vettori, R. Martínez, X. Li, A. Garcia-Saavedra, C. Fabiana Chiasserini, C. Casetti, K. Tomakh, O. Kolodiazhnyi, C. Jesus Bernardos, "On the Integration of AI/ML-based scaling operations in the 5Growth platform," in *IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2020, Leganes - Madrid, Spain, November 10-12*, pp. 105–109, IEEE, 2020.

[157] J. Baranda, J. Mangues-Bafalluy, E. Zeydan, C. Casetti, C. Fabiana Chiasserini, M. Malinverno, C. Puligheddu, M. Groshev, C. Guimarães, K. Tomakh, D. Kucherenko, O. Kolodiaznnyi, "Demo: AIML-as-a-Service for SLA management of a Digital Twin Virtual Network Service," in *Procs of the IEEE International Conference on Computer Communications (IEEE INFOCOM'21), May 10-13 2021, virtual event*, IEEE, 2021.

[158] "EU H2020 project 5Growth, 5G-enabled Growth in Vertical Industries." Available at: `"https://5growth.eu/"`, Accessed: April 2021.

[159] D. de Vleeschauwer, J. Baranda, J. Mangues-Bafalluy, C. F. Chiasserini, M. Malinverno, C. Puligheddu, L. Magoula, J. Martín-Pérez, S. Barmpounakis, K. Kondepu, L. Valcarenghi, X. Li, C. Papagianni, A. García-Saavedra, "5Growth Data-Driven AI-Based Scaling," in *Procs of the 30th European Conference on Networks and Communications (EUCNC'21) and 6G Summit, 8-11 June 2021, Porto (Portugal), virtual event*, 2021.

[160] Apache Kafka, "A distributed streaming platform." Available online at: `https://kafka.apache.org/`, Accessed: April 2021.

[161] Apache Kafka, "Unified analytics engine for big data." Available online at: `https://spark.apache.org/`, Accessed: April 2021.

Appendix A

Additional NFV-NS lifecycle management workflows of the Service Orchestrator

This appendix presents the workflows defined at the SO module to perform other relevant NFV-NS lifecycle management operations, namely onboarding and termination, complementing those presented in Chapter 8.

A.1 NFV-NS Service Onboarding

Figure A.1 shows the operational workflow the SO follows to onboard the descriptors associated to an NFV-NS. In this case, first the VNF descriptors (VNFDs) are onboarded and then finally, the descriptor of the NFV-NS is onboarded. Three observations to be considered regarding the presented workflow are: (i) the onboarding process is triggered by the Vertical Slicer (VS), (ii) OSM MANO platform is used as Core MANO platform[1] , and (iii) although following the ETSI-NFV IFA013 [94] procedures, in the case of VNFs, the received VNF package contains only the VNFD, assuming that the software images for the VNFs have been already uploaded to the VIMs registered at the MTP.

Initially, the VS sends an *OnboardVnfPackage* request (step 1) for each of the constituent VNFs in an NFV-NS to the SO. The SOE parses the request to get the URL from where to download the VNF package and after downloading it, the SOE unpacks its content (steps 2 and 3). As mentioned before, this package only contains the VNFD, which is parsed (step 4) to create the corresponding entry in the VNFD Catalogue DB (step 5). This entry in the DB contains the content of the VNFD descriptor and some information associated to the VNFD like its identifier, name and version. Once the entry in the DB is created, the SOE contacts the ROOE to onboard this descriptor at the Core MANO platform

[1] A similar approach is followed if another Core MANO platform is used, e.g., Cloudify.

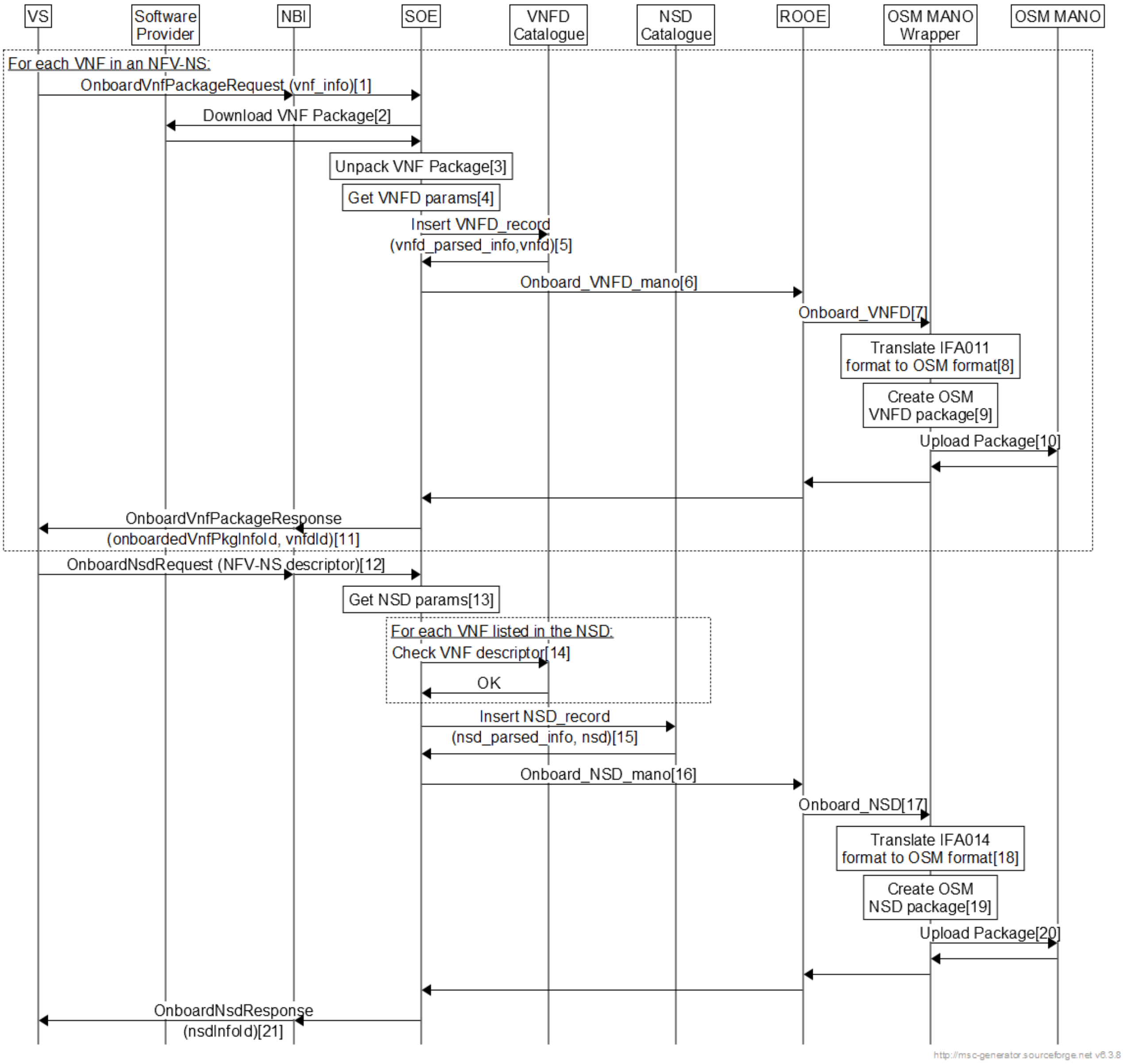


Figure A.1: Onboarding workflow for NFV-Network Services (step 6), in this case OSM. The ROOE contacts the OSM MANO wrapper (step 7), who translates the received JSON file following the ETSI-NFV IFA011 [139] format to the YAML information models defined by OSM, creates the corresponding OSM package, and uploads it to the OSM MANO platform (steps 8, 9 and 10). After the package has been uploaded and acknowledged, the SOE returns an *OnboardVnfPackage* response (step 11) to the VS validating the VNFD onboarding.

After all VNFDs are onboarded, the VS sends an *OnboardNsd* request to the SO to proceed with the NFV-NS descriptor (NSD) onboarding (step 12). The SOE parses the descriptor, which is embedded in the received request, checks with the VNFD catalogue DB that the descriptors of all listed VNFs are available, and creates an entry in the NSD Catalogue DB (steps 13, 14, and 15). This entry contains the NSD and some information associated to the NSD like the NFV-NS identifier, name and version. Once the entry in the DB is created, the SOE contacts the ROOE to onboard this descriptor at the Core MANO platform (step 16), in this case OSM. The ROOE contacts the OSM MANO wrapper (step 17), who translates the received JSON file following the ETSI-NFV IFA014 [138] format to the YAML information models defined by OSM, creates the corresponding OSM package and uploads it to the OSM MANO platform (steps 18, 19, and 20). After the package has been uploaded and acknowledged, the SOE returns an *OnboardNsd* response (step 21) to the VS validating the NSD onboarding.

To conclude with the onboarding operation, we include the following remark. In the case of a composite NFV-NS, the composite NSD is stored in the NSD Catalogue DB and there is no need of further interaction with the MANO platform because the different nested NFV-NSs will be instantiated one by one as explained in Section 10.3.1. Of course, prior to store the composite NSD in the DB, the SO checks that the NSDs of the different nested NFV-NSs are available in the NSD Catalogue DB, otherwise the composite NFV-NS could not be instantiated.

A.2 NFV-NS Service Termination

A.2.1 Regular NFV-NS Service Termination

Figure A.2 presents the operational workflow followed by the SO to perform the termination of an instantiated regular NFV-NS. In this workflow, operations are performed in reverse order with respect to the ones defined in the instantiation workflow presented in Section 8.3.

Initially, the VS module requests the termination of the instantiated NFV-NS (step 1) issuing an ETSI-NFV IFA013 [94] request to the SO. Then, the SOE checks with the help of the NS Instance DB if the status of the specified NFV-NS Identifier is correct (step 2), i.e., the identifier exists in the database and it is not in *"instantiating"*, *"terminating"*, *"terminated"* or *"not instantiated"* status. In case of a correct check, the SOE returns an operation identifier to the VS and proceeds with the termination procedure.

First, it contacts with the SLA manager to delete possible configured alerts (step 3). The SLA manager retrieves this info from the NS Instance DB (step 4), and in case of configured alerts, the SLA manager contacts the MON Platform to remove them (step 5). After that, the SOE repeats the same procedure with the configured performance monitoring jobs and dashboards, but contacting with the Monitoring Manager (steps 6 to 9).

After that, the SOE centers the termination operation in releasing the allocated NFVI resources. For that, it contacts the ROOE (step 10). Then, the ROOE contacts the Core MANO wrapper (step 11), who first retrieves the instance resource information from the DB module (step 12). Next, the Core MANO wrapper contacts the Core MANO platform to terminate the computing resources allocated for the VNFs (steps 13 and 14). Once, the computing resources are released, the wrapper contacts the

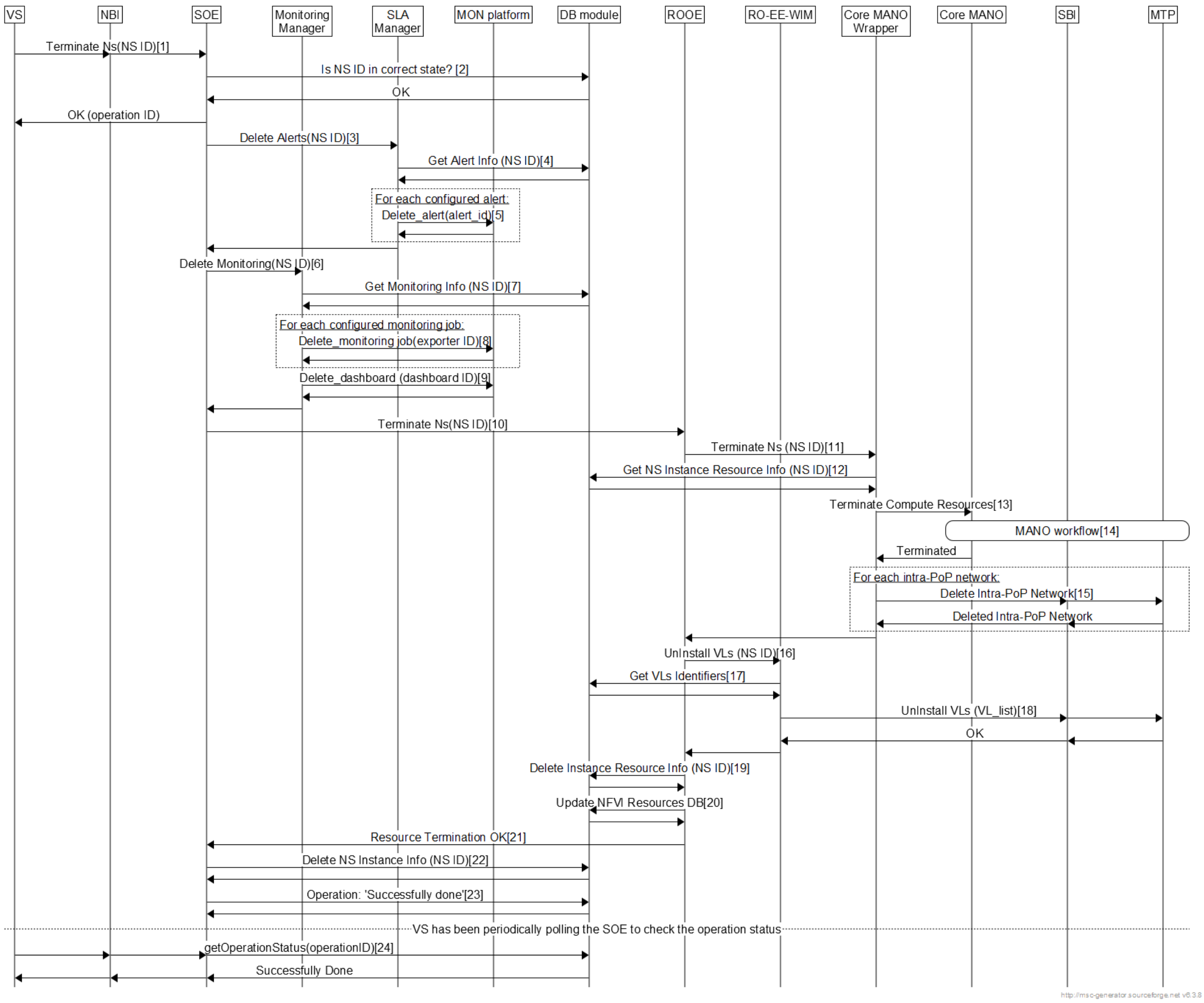


Figure A.2: Termination workflow for regular NFV-Network Services

MTP, through the SBI, to delete the associated intra-PoP networks (step 15). After releasing resources at the VIM level, the ROOE contacts the RO-EE-WIM entity to release the allocated networking resources interconnecting NFVI-PoPs (step 16). For that, the RO-EE-WIM retrieves the associated connectivity identifiers from the NFV-NS Instance resource DB and issues a request to the MTP, through the SBI, with the list of them (steps 17 and 18). Once all the NFVI resources have been released, the ROOE deletes the associated entry for this NFV-NS in the Instance Resource DB module (step 19), updates the NFVI Resources DB (step 20) and confirms the SOE that NFVI resources have been released (step 21). Finally, the SOE deletes the NFV-NS instance info entry of the DB module and sets the operation as successfully done (step 22 and 23). While the termination operation progressed, the VS polled periodically its status. This polling finishes when the VS receives confirmation that the termination operation has been successfully done (step 24).

A.2.2 Composite/Federated NFV-NS Service Termination

Figure A.3 presents the operational workflow followed by the SO to perform the termination of an instantiated composite NFV-NS deployed in multiple administrative domains (ADs). This termination operation must be launched from the consumer domain (CD), which coordinated the instantiation process.

When the VS module requests the termination of the composite NFV-NS (step 1) issuing an ETSI-NFV IFA013 [94] request, the SOEp of the CD checks in the NS Instance DB the status of the specified NFV-NS Identifier (step 2). In case of a correct check, it returns an operation Identifier to the VS (step 3) and proceeds with the termination procedure.

First, the SOEp gets the entry of the composite instantiation information from the NS Instance dB (step 4). This entry contains details like the deployment flavour, instantiation level, among others, of each nested NFV-NS. A relevant detail in this entry is the instantiation order of the nested NFV-NS. Nested NFV-NSs will be terminated in inverse order to which they were created, avoiding potential conflicts in resource deallocation operations like the termination of computing resources or intra-PoP networks at the VIM level. Physical resources (CPU, RAM, Storage, intra-PoP networks) were allocated as needed during instantiation time. For those resources used by multiple nested NFV-NSs, like the case of intra-PoP networks within a VIM in charge of supporting the common Virtual Links defined at the composite level, their creation is triggered by the MTP by the first nested NFV-NS requiring it. During the instantiation of the following nested NFV-NSs, the SO realises that the required resource is not created because the MTP will not report the creation of additional resources. Then, to avoid the mentioned resource deallocation problems, the resource is needed to be terminated when all the rest of nested NFV-NSs using this resource have been previously deallocated.

According to that mentioned order, first, the nested NFV-NSs instantiated in the provider domains (PDs) will be terminated, in case they were involved during the instantiation process. For that, the SOEp sends a termination request through the *SO-SO* interface following the ETSI-NFV IFA030/013 specification (step 5). At the PD, the SOEp checks the status of the requested nested NFV-NS identifier (step 6) and upon a correct check, it returns the corresponding operation identifier to the SOEp at the CD (step 7). In the meantime, the SOEp at the PD proceeds with the termination of the requested NFV-NS following the steps mentioned in Section A.2.1 (steps 8 and 9). Note that during the termination process of the nested NFV-NS at the PD, the inter-nested connections from the PD to the CD will also be terminated because during the instantiation process, their identifiers were stored in the same DB entry associated to the nested NFV-NS. While the termination process progresses at the PD, the SOEp of the CD polls periodically the status of the provided operation (step 10). Once the SOEp at the CD acknowledges the operation as *"Successfully Done"*, it proceeds with the termination of the following nested NFV-NS.

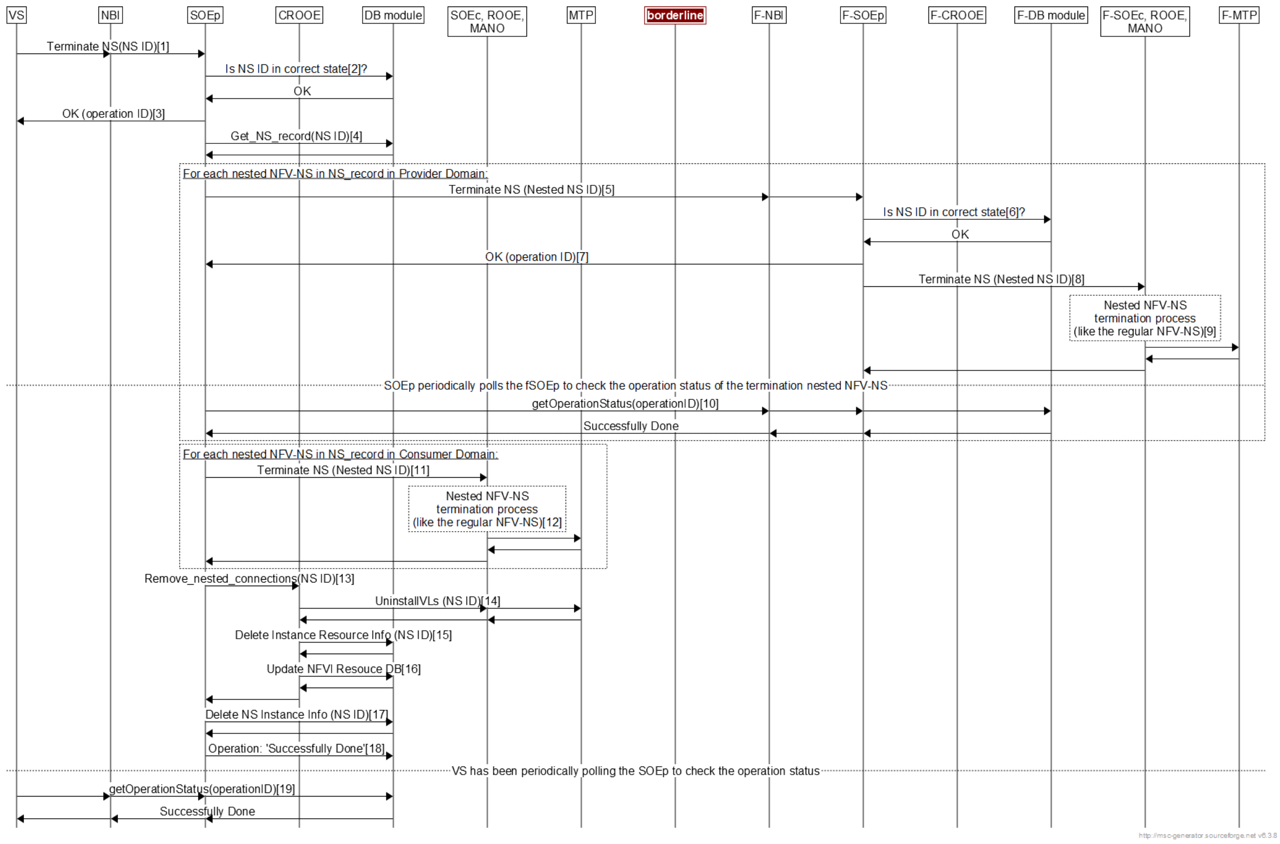


Figure A.3: Termination workflow for composite NFV-Network Services deployed in multiple ADs

After all the nested NFV-NSs deployed at the PD/s have been terminated, the SOEp of the CD repeats the same operation with those nested NFV-NSs instantiated in the CD (steps 11 and 12), which were the first nested NFV-NSs deployed during the instantiation process[2]. Once all nested NFV-NSs have been terminated, the SOEp proceeds to remove the inter-nested connections from the CD to the PD by contacting the CROOE (step 13). Differently to the case of the NFV-NSs deployed at the PD/s, where the inter-nested connections identifiers are stored in the same DB entry that the NFV-NS "intra-nested" connections identifiers, in the case of the CD, all the inter-nested connections identifiers from the CD to the PD are stored in an DB entry of the Network Service Instance Resource DB corresponding to the composite deployment. The CROOE contacts the MTP through the ROOE (RO-EE-WIM) to uninstall these interconnections (step 14). After that, the CROOE removes the associated information from the Network Instance Resource DB module and updates the NFVI Resources DB (steps 15 and 16). Then, the CROOE acknowledges the correct connection termination to the SOEp, which deletes the NFV-NS info entry of the NS Instance DB module and sets the operation as successfully done (step 17 and 18). While the termination operation progressed, the VS polled the SOEp periodically about its status. This polling finishes when the VS receives confirmation that the termination operation has been successfully done (step 19).

To conclude with the termination operation of a composite NFV-NS, it is worth mentioning the case of a deployment using a reference to a regular NFV-NS. While there are composite NFV-NSs using a regular NFV-NS, the SO will not allow the termination of this regular NFV-NS. Additionally, when terminating a composite NFV-NS using a reference regular NFV-NS, the SO will proceed to deallocate the resources that were allocated during the instantiation of the composite NFV-NS to avoid disrupting

[2]Note that in the case nested NFV-NSs are not deployed in PDs, the SO would skip steps 5 to 10. This would correspond to the case of the complete deployment of the composite NFV-NS in the CD, as explained in Chapter 9.

the operation of the initially instantiated reference regular NFV-NS.

Appendix B

Integrating of AI/ML-based operations in the Service Orchestrator

This appendix presents the initial extensions of the orchestration architecture to include AI/ML-based operations, as stated in Chapter 12. More specifically, the focus of these extensions are set on the Service Orchestrator (SO) module and the procedure to perform AI/ML-driven scaling operations [156]. This work has also been demonstrated in [157].

B.1 Service Orchestrator Architectural extensions

The inclusion of AI/ML capabilities in the proposed management and orchestration architecture is based on the AIML as a Service paradigm. This approach has been also considered in other research and standardization initiatives, like O-RAN [106]. Based on this consideration, the idea is to evolve the proposed architecture to enable the possibility of requesting and using AI/ML-based models provided by an external block for performing different orchestration tasks during run-time. This external block has high computational capacities and can efficiently cope with the training of such models.

Based upon these architectural considerations, next, we present the extensions of the SO architecture to enable AI/ML-based scaling operations. It mainly consists of the update and the addition of new submodules to interact with an external block to download required AI/ML models and the support of a data engineering pipeline collecting, consuming and ingesting the required NFV-NS performance metrics to evaluate the downloaded AI/ML model. The SO obtains these required AI/ML models from the AIML platform designed within the context of the EU 5Growth project [158]. This AIML platform [159] becomes a new module of the E2E Network Service Orchestration platform presented in Chapter 5. It accommodates both supervised ML and reinforcement learning models, train them when needed, and create the files required during the inference phase at the SO for the model evaluation.

Prior to describing the SO architectural extensions, it is worth mentioning the proposal of a new information element (IE) in the NSD that serves the SO to enable the automation of the configuration

Listing 7 New defined NSD IE for AI/ML-based operations

```
"aimlRules": [
    { "ruleId": "aiml_rule1",
      "scope": "scaling",
      "nsMonitoringParamRef": ["mp1", "mp2"]} ]
```

of AIML-based operations. An example of this IE is presented in Listing 7 and it is based on the format of the ETSI-NFI IFA014 [138] descriptors used in our architecture. The presence of this new IE expresses the need of interaction with the AIML platform to configure AI/ML-based decisions for a given MANO scope ("scaling") and specifies the metrics out of the ones already defined for this kind of network service in the NSD field *"monitoredInfo"* required by this AI/ML problem (*"mp1"* and *"mp2"*) to perform its decisions.

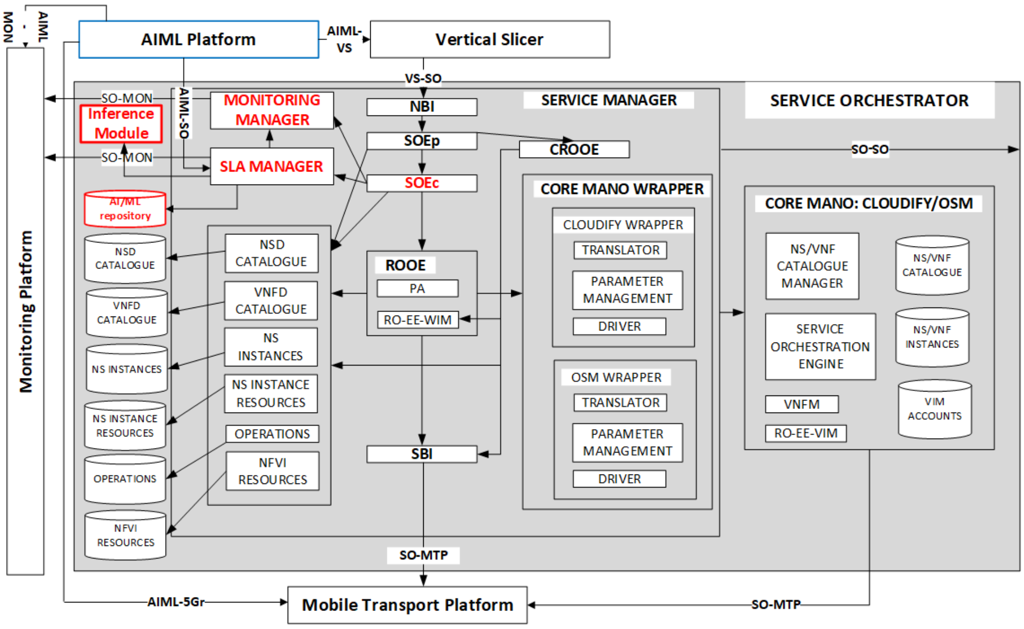


Figure B.1: Service Orchestrator Architecture supporting AIML-based scaling operations

Figure B.1 presents the architecture of the SO module, showing also the relation with the VS, the MTP and the MON building blocks of the E2E Network Service Orchestration architecture presented in Chapter 5 and the new included AIML platform module. The marked submodules in red are those that have been modified/added to support AI/ML-based scaling operations:
•*SLA Manager*: This submodule is responsible for handling the NFV-NS SLA compliance and triggering the scaling process in case of a detected SLA violation. As stated in Chapter 8, this detection was based upon an alerting system configured through the MON platform. Now, its logic has been extended to orchestrate all the operations to handle the AI/ML-based scaling operation when including the described IE in the NSD. For this purpose, it interacts mainly with the Monitoring Manager submodule, the AIML platform, and with the Inference module.
•*Monitoring Manager*: This submodule interacts with the MON platform to configure the collection of performance metrics expressed in the NSD and the configuration of visualization dashboards. Now, its interaction with the MON platform includes also the configuration of dedicated data objects (i.e., Apache Kafka topics [160]) to collect the monitoring data expressed in the AI/ML IE of the NSD. Then, this data is consumed and ingested by the Inference module.
•*Service Orchestrator Engine child (SOEc)*: this submodule of the Service Orchestrator Engine, handling the orchestration of regular (i.e., non-composite) NFV-NSs, has been extended to interact with

the new capabilities of the SLA Manager module.

Furthermore, the SO architecture adds the Inference Module (implemented with an instance of Apache Spark [161]) and the *AI/ML repository*, which is a new submodule where the SLA Manager stores the requested AI/ML models and processing routines obtained from the AIML platform. These models and processing routines are required by the Inference Module to run the jobs in charge of checking the SLA compliance by evaluating the downloaded AI/ML models, which decide on the appropriate instantiation level (IL) of the NFV-NS instance given current service demands. Finally, the MON platform required of extensions to support AI/ML-based operation. In addition, to host the Apache Kafka platform, the interface of the MON platform has been extended to handle the creation of *"data scraper"* elements. These elements, orchestrated by the SLA Manager, filter out the collected monitoring data for an NFV-NS stored in the MON platform and insert them in the requested Kafka topic to be ingested by the corresponding inference job.

Our proposed orchestration architecture could also accommodate other AI/ML-based problems if specified in the presented IE. For that, we would need to evolve the submodules in charge of orchestrating the corresponding instantiation operation to support the interaction with the AIML platform and launch the corresponding Inference processes, similarly to the approach herein described for the scaling operation use case.

B.2 AI/ML-Based NFV-NS Scaling Workflow

Figure B.2 presents the operational workflow followed by the SO to enable the AIML-based scaling operation of a regular NFV -NS. This workflow takes as starting point the last step of the instantiation process (after VNFs have been allocated and their interconnections and monitoring jobs have been configured), when the SOEc contacts the SLA Manager, as presented in Figure 8.3 of Chapter 8.

Initially, the SLA Manager checks the existence of an AI/ML IE in the NSD for the *scaling* problem (step 1). The next steps happen upon a positive confirmation. Then, the SLA Manager contacts the Monitoring Manager to configure a dedicated data topic in the Apache Kafka instance run by the MON platform (step 2). This topic is needed to insert the required monitoring information expressed in the NSD to handle the AI/ML-based scaling operation for the given NFV-NS instance (step 2). After that, the SLA Manager, through the Monitoring Manager, creates the *"data scrapers"* elements at the MON platform to filter out the monitoring data specified at the AI/ML IE (step 3). The SLA Manager contacts the AIML Platform to download the required model and its associated processing routine (step 4a) and stores them in the AI/ML repository (step 4b). In the last steps of the data engineering pipeline configuration, the SLA Manager configures an Apache Spark streaming job at the Inference module (step 5a). Additionally, the SLA manager publishes the current NFV-NS IL in the dedicated Apache Kafka topic created in step 2 to give the appropriate context to the inference process. Once the pipeline is configured, the SLA Manager saves the associated information (e.g., Apache Kafka topic, data scrapper, Apache Spark job references) in the NFV-NS instance database (step 6).

From this point on, periodically, the inference process running in Apache Spark ingests the data requested in step 3 from the Kafka topic (step 7a), performs online classification (step 7b), and notifies the result (i.e., the best IL given the current context) to the SLA Manager (step 7c). The SLA Manager checks the notification (step 8a), and if the received IL differs from the current IL, it stops the inference process (step 8b) and triggers the scaling operation through the northbound interface of the SO (step 8c). This scaling operation is carried out as explained in Section 8.4.1 of Chapter 8. Additionally, as the last steps of the scaling procedure, the SOEc contacts the SLA Manager, which retrieves the information from the NFV-NS instance database and repeats steps from 5) to 6) to start up again the inference job and close the loop.

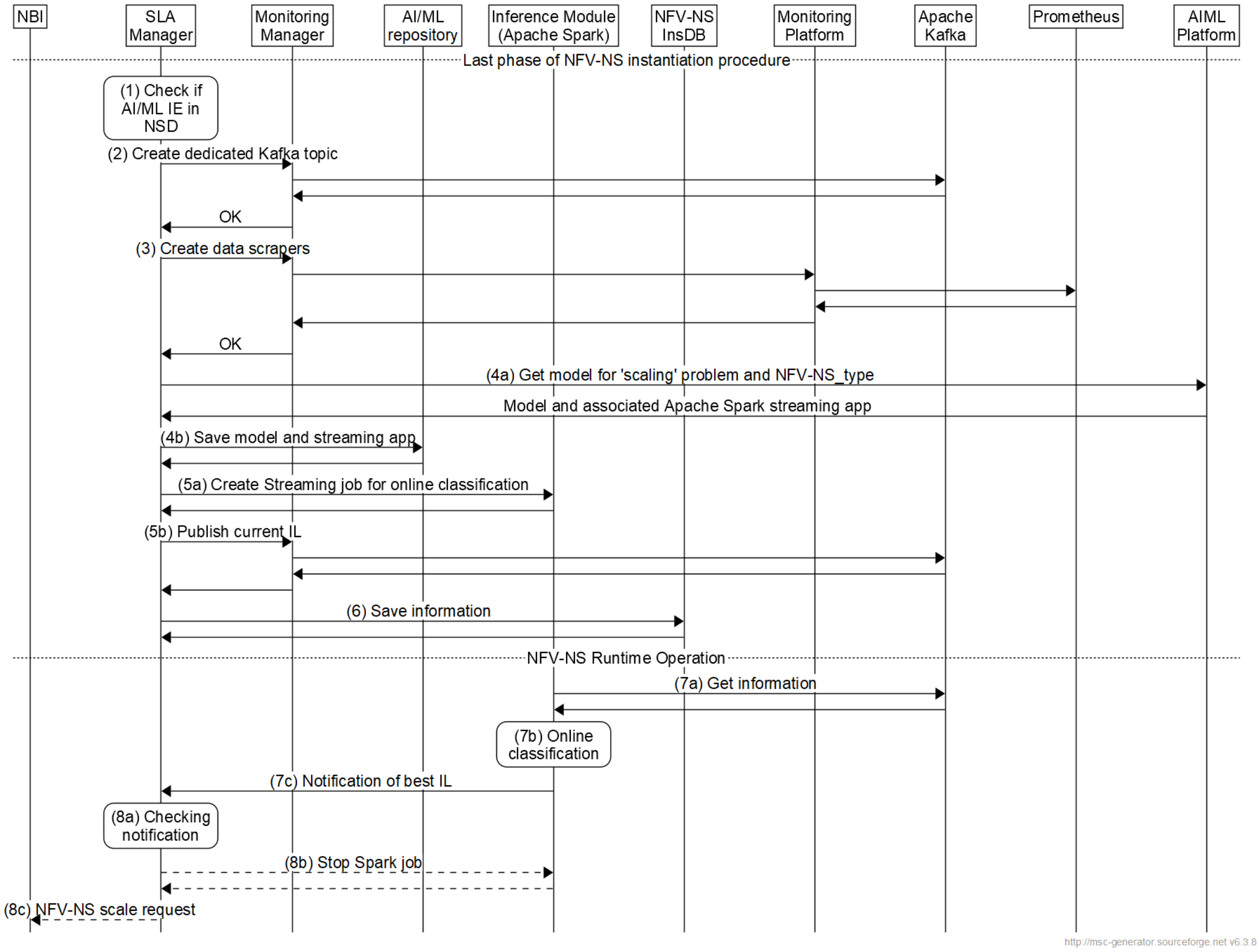


Figure B.2: Onboarding workflow for NFV-Network Services

Appendix C

List of publications

This appendix presents the list of published and submitted publications derived from the work presented in this thesis. Publications are classified in journals, international conferences, demo papers, and collaborations in inverse chronological order. *Collaborations* subsection includes bibliographical references where I have a light involvement but help refining some aspects of the proposed architectural design (e.g., interfaces) and disseminating PoC results to a wider (i.e., non-exclusive scientific) community.

C.1 Journal publications

1. **J. Baranda**, J. Mangues-Bafalluy, I. Pascual, J. Núñez-Martínez, J. L. de la Cruz, R. Casellas, R. Vilalta, J. X. Salvat, and C. Turyagyenda, "Orchestration of End-to-End Network Services in the 5G-Crosshaul Multi-Domain Multi-Technology Transport Network", in IEEE Communications Magazine, vol. 56, no. 7, pp. 184-191, July 2018.

2. **J. Baranda**, J. Mangues-Bafalluy, R. Martínez, L. Vettori, K. Antevski, C. J. Bernardos, X. Li, "Realizing the network service federation vision: Enabling automated multidomain orchestration of network services", in IEEE Vehicular Technology Magazine, vol. 15, no. 2, pp. 48-57, March 2020.

3. X. Li, T. Deiss, J. Mangues-Bafalluy, **J. Baranda**, X. Costa-Pérez, G. Landi, C. J. Bernardos, P. Iovanna, A. Zurita, P. Bertin, "Automating Vertical Services Deployments over the 5GT Platform", in IEEE Communications Magazine, vol. 58, no. 7, pp. 44-50, July 2020.

4. X. Li, C.F.Chiasserini, J.Mangues-Bafalluy, **J.Baranda**, G.Landi, B.Martini, X.Costa-Pérez, C.Puligheddu, L.Valcarenghi, "Automated Service Provisioning and Hierarchical SLA Management in 5G Systems", in IEEE Transactions on Service and Network Management, August 2021.

C.2 Conference papers

C.2.1 Accepted publications

1. **J. Baranda**, I. Pascual, J. Mangues-Bafalluy and J. Núñez-Martínez, "Wireless Interface Agent for SDN mmWave Multi-hop Networks: Design and Experimental Evaluation", in Proceedings of the 2nd ACM Workshop on Millimeter Wave Networks and Sensing Systems (mmNets), co-located with ACM MobiCom 2018, New Delhi, India, October 29, pp. 9-14, 2018.

2. J. Mangues-Bafalluy, **J. Baranda**, I. Pascual, R. Martínez, L. Vettori, G. Landi, A. Zurita, D. Salama, K. Antevski, J. Martín-Perez, D. Andrushko, K. Tomakh, B. Martini, X. Li, J. X. Salvat, "5G-TRANSFORMER Service Orchestrator: Design Implementation and Evaluation", in Proceedings of the 28th European Conference on Networks and Communications (EuCNC), Valencia, Spain, June 18-21, 2019.

3. **J. Baranda**, L. Vettori, R. Martínez, J. Mangues-Bafalluy, "A Mobile Transport Platform Interconnecting VNFs over a Multi-Domain Optical/Wireless Network: Design and Implementation", in Proceedings of the 24th International Conference on Optical Network Design and Modelling (ONDM'20), virtual event, May 2020.

4. **J. Baranda**, J. Mangues-Bafalluy, R. Martínez, L. Vettori, K. Antevski, C. J. Bernardos, X. Li, "5GTRANSFORMER meets Network Service Federation: design, implementation and evaluation", in Proceedings of the 6th IEEE International Conference on Network Softwarization (IEEE NetSoft'20), virtual event, June 2020.

5. **J. Baranda**, J. Mangues-Bafalluy, E. Zeydan, L. Vettori, R. Martínez, X. Li, A. Garcia-Saavedra, C. Fabiana Chiasserini, C. Casetti, K. Tomakh, O. Kolodiazhnyi, C. Jesus Bernardos,"On the Integration of AI/ML-based scaling operations in the 5Growth platform", in Proceedings of the 6th IEEE Conference on Network Function Virtualization and Software Defined (IEEE NFV-SDN'20), Leganes - Madrid (virtual event), Spain, November 10-12, pp. 105-109, 2020. This paper received the BEST Fast Track paper award.

6. R. Martínez, L. Vettori, **J. Baranda**, J. Mangues, E. Zeydan, "Experimental Validation of Compute and Network Resource Abstraction and Allocation Mechanisms within a NFV Infrastructure", in Proceedings of the 2021 IFIP/IEEE International Symposium on Integrated Network Management (IEEE IM'2021), virtual event, May 2021.

C.2.2 Submitted publications

7. **J. Baranda**, J. Mangues-Bafalluy, L. Vettori , R. Martínez, E. Zeydan, "Enabling SLA Management of Federated Network Services through Scaling Operations", submitted for publication in the IEEE/IFIP Network Operations and Management Symposium (IEEE NOMS'22), Budapest, Hungary, April 2022.

C.3 Demo papers

1. **J. Baranda**, I. Pascual, M. Requena, and J. Mangues-Bafalluy, "Deploying a containerized ns-3/LENA-based LTE mobile Network Service through the 5G-TRANSFORMER platform", in Proceedings of the 4th IEEE Conference on Network Functions Virtualization and Software Defined Networking (IEEE NFV-SDN 2018), 27-29 November, Verona (Italy), 2018.

2. **J. Baranda**, J. Mangues-Bafalluy, L. Vettori, R. Martínez, G. Landi, K. Antevski, "Demo: Composing Services in 5G-TRANSFORMER", in Proceedings of the 20th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc'19), July 2-5, Catania (Italy), pp. 407-408, 2019.

3. **J.Baranda**, G. Avino, J. Mangues-Bafalluy, L. Vettori, R. Martínez, C.F. Chiasserini, C. Casetti, P. Bande, M. Giordanino, M. Zanzola, "Automated deployment and scaling of automotive safety services in 5G-Transformer ", in Proceedings of the 5th IEEE Conference on Network Function Virtualization and Software Defined Networking (IEEE NFV-SDN 2019), November 12-14, Dallas, TX, (USA), pp. 1-2, 2019.

4. **J. Baranda**, J. Mangues-Bafalluy, L. Vettori, R. Martínez, G. Avino, C. F. Chiasserini, C. Puligheddu, C. Casetti, J. Brenes, G. Landi, K. Kondepu, F. Paolucci, S. Fichera, L. Valcarenghi, "Arbitrating Network Services in 5G Networks for Automotive Vertical Industry", in Proceedings of the IEEE International Conference on Computer Communications (IEEE INFOCOM'20), July 6-9, virtual event, 2020.

5. **J. Baranda**, J. Mangues-Bafalluy, L. Vettori, R. Martínez, K. Antevski, L. Girletti, C.J. Bernardos, K.Tomakh, D. Kucherenko, G. Landi, J. Brenes, X. Li, X. Costa-Pérez, F. Ubaldi, G. Imbarlina, M. Gharbaoui, "NFV Service Federation: enabling Multi-Provider eHealth Emergency Services", in Proceedings of the IEEE International Conference on Computer Communications (IEEE INFOCOM'20), July 6-9, virtual event, 2020.

6. **J. Baranda**, J. Mangues-Bafalluy, L. Vettori, R. Martínez, "Demo: Scaling Composite NFV-Network Services", in Proceedings of the 21st ACM International Symposium on Theory, Algorithmic, Foundations, and Protocol Design for Mobile Networks and Mobile Computing (ACM MobiHoc 2020), October 11-14, virtual event, pp. 307-308, 2020.

7. **J. Baranda**, J. Mangues-Bafalluy, L. Vettori , R. Martínez, E. Zeydan, "Scaling Federated Network Services: Managing SLAs in Multi-Provider Industry 4.0 Scenarios", in Proceedings of the IEEE International Conference on Computer Communications (IEEE INFOCOM'21), May 10-13, virtual event, 2021.

8. **J. Baranda**, J. Mangues-Bafalluy, E. Zeydan, C. Casetti, C. Fabiana Chiasserini, M. Malinverno, C. Puligheddu, M. Groshev, C. Guimarães, K. Tomakh, D. Kucherenko, O. Kolodiaznnyi, "Demo: AIML-as-a-Service for SLA management of a Digital Twin Virtual Network Service", in Proceedings of the IEEE International Conference on Computer Communications (IEEE INFOCOM'21), May 10-13, virtual event, 2021.

C.4 Collaborations

1. K. Antevski, J.Martín-Pérez, N. Molner, C.F.Chiasserini, F.Malandrino, P. Frangoudis, A. Ksentini, X. Li, J.X.Salvat, R. Martínez, I. Pascual, J. Mangues-Bafalluy, **J. Baranda**, B. Martini, M. Gharbaoui, "Resource Orchestration of 5G Transport Networks for Vertical Industries", in

Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 9-12 September 2018, Bologna (Italy).

2. 5G PPP Infrastructure Association, "5G-PPP Infrastructure -Trials and Pilots Brochure (N°2)". Available online at: https://5g-ppp.eu/ the-5g-ppp-infrastructure-trials-and-pilots-brochure-n2-is-out/, December 2020.

3. D. de Vleeschauwer, **J. Baranda**, J. Mangues-Bafalluy, C. F. Chiasserini, M. Malinverno, C. Puligheddu, L. Magoula, J. Martín-Pérez, S. Barmpounakis, K. Kondepu, L. Valcarenghi, X. Li, C. Papagianni, A. García-Saavedra, "5Growth Data-Driven AI-Based Scaling", in Proceedings of the 30th edition of the European Conference on Networks and Communications (EUCNC'21) & 6G Summit, 8-11 June 2021, Porto (Portugal), virtual event.