THÈSE

pour obtenir le grade de

Docteur des Universités Mohammed Premier et Sorbonne Paris Nord

Discipline : "Ingénierie Informatique"

présentée et soutenue publiquement par

Mohammed CHAHBAR

le 20 Août 2021

Mécanismes d'Orchestration des Slices Réseaux dans un Contexte SDN/NFV

Directeurs de thèse : **Mme. Gladys Diaz, Mr. Kamal Ghoumid**

Co-Directeur de thèse : **Mr. Christophe Cérin**

JURY

Abdulhalim Dandoush,	Professeur associé, ESME-Sudria	Examinateur
Michelle Sibilla,	Professeur, Université Toulouse III - Paul Sabatier	Rapporteur
Abdellatif Kobbane,	Professeur, Université Mohammed V de Rabat	Rapporteur
Ahmed Hammade,	MCF-HDR, Université Bourgogne-Franche Comté	Rapporteur
Francine Krief,	Professeur, ENSEIRB/LaBRI	Examinateur
Gladys Diaz,	MCF-HDR, Université Sorbonne Paris Nord	Directrice de thèse, USPN
Kamal Ghoumid,	Professeur, Université Mohammed Premier	Directeur de thèse, UMP
Christophe Cérin,	Professeur, Université Sorbonne Paris Nord.	Co-directeur de thèse

Thesis

Submitted for the degree of Doctor of Philosophy of

Universities Mohammed Premier and Sorbonne Paris Nord

Specialization : "Computer Engineering"

presented and defended by

Mohammed CHAHBAR

20 August 2021

Orchestration Mechanisms of Network Slices in a SDN/NFV Context

Supervisors : **Mrs. Gladys Diaz, Mr. Kamal Ghoumid**

co-supervisor : **Mr. Christophe Cérin**

Committee

Abdulhalim Dandoush,	Associate professor, ESME-Sudria	Examiner
Michelle Sibilla,	Professor, Université Toulouse III - Paul Sabatier	Reviewer
Abdellatif Kobbane,	Professor, Université Mohammed V de Rabat	Reviewer
Ahmed Hammade,	MCF-HDR, Université Bourgogne-Franche Comté	Reviewer
Francine Krief ,	Professor, ENSEIRB/LaBRI	Examiner
Gladys Diaz,	MCF-HDR, Université Sorbonne Paris Nord	Supervisor, USPN
Kamal Ghoumid,	professor, Université Mohammed Premier	Supervisor, UMP
Christophe Cérin,	Professor, Université Sorbonne Paris Nord.	Co-Supervisor

Abstract

The development and the emergence of the Fifth Generation of Mobile Networks (5G) use cases has led to the adoption of a new approach of network management and service instantiation called Network Slicing. This technology allows the dynamic selection and lifecycle management of the appropriate cloud and network resources and their arrangement in isolated logical networks sharing the same physical infrastructure and called Network Slices (NSs). A NS is particularly instantiated to fulfill one or several predefined use cases or services with similar Quality of Service (QoS) requirements.

For example, the remote surgery use case requires ultra-high reliability and at the same time ultra-low latency communications. These same network characteristics are required by the vehicle-to-everything (V2X) use case for autonomous/assisted driving in addition to a high speed communication for entertainment purposes such as accessing the internet from the vehicule. Because of such conflicting and extremely diversified requirements of the new emerging services, the traditional ”one-size-fits-all” network approach is no longer efficiently feasible.

Network Slicing is supported by innovative technologies such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV). Together they are enabling network programmability and accelerating the deployment of new services. In this context, the objectives of our thesis are threefold. First, bringing forth an end-to-end 5G network slicing model that serves as a reference to researchers and developers for the realization of their NS solutions. Second, analysing the requirements and the design of the resource-facing solution responsible for the NS subnet management. Last, based on the aforementioned works, we propose the inter-slice bandwidth resource sharing, a novel mechanism to address the resource optimization challenge in slice-based infrastructures. The benefits of this concept is demonstrated by the implementation and the evaluation of our solution called InterS.

Résumé

Le développement et l'émergence des cas d'usage des réseaux mobiles de cinquième génération (5G) ont conduit à l'adoption d'une nouvelle approche de gestion de réseau et d'instanciation de services appelée "Network Slicing". Cette technologie permet d'une façon dynamique la sélection et la gestion du cycle de vie des ressources cloud et réseau appropriées et leur organisation en réseaux logiques isolés et partageant la même infrastructure physique appelés "Network Slices" (NSs). Un NS est particulièrement instancié pour répondre à un ou plusieurs cas d'usages ou services prédéfinis avec des exigences similaires en qualité de service (QoS) .

Par exemple, la chirurgie à distance est un cas d'usage qui nécessite une fiabilité ultra-élevée et en même temps des communications à latence ultra-faible. Ces mêmes caractéristiques de réseau sont requises par le cas d'usage des véhicules connectés (V2X) pour la conduite autonome/assistée en plus d'une communication haut débit à des fins de divertissement telles que l'accès à Internet depuis le véhicule. En raison de ces exigences conflictuelles et extrêmement diversifiées des nouveaux services émergents, l'approche traditionnelle du réseau "one-size-fits-all" n'est plus réalisable de manière efficace.

la technologie "Network Slicing" est supportée par des technologies innovantes telles que les réseaux définis par logiciel (SDN) et la virtualisation des fonctions réseau (NFV). Ensemble, ils permettent la programmabilité du réseau et accélèrent le déploiement de nouveaux services. Dans ce contexte, les objectifs de notre thèse sont triples. Premièrement, faire émerger un modèle de "Network Slicing" de 5G de bout en bout qui sert de référence aux chercheurs et développeurs pour la réalisation de leurs solutions de NS. Deuxièmement, analyser les exigences et la conception de la solution orientée ressources responsable de la gestion des "NS Subnets". Enfin, sur la base des travaux susmentionnés, nous proposons le partage de ressources de bande passante entre "Network Slices", un nouveau mécanisme pour relever le défi d'optimisation des ressources dans

les infrastructures à bases de NS. Les avantages de ce concept sont démontrés par la mise en œuvre et l'évaluation de notre solution appelée InterS.

Acknowledgments

I WOULD like to express my deepest gratitude to Mrs. Gladys Diaz , Mr. Christophe Cérin from Sorbonne Paris Nord University (France) and Mr. Kamal Ghoumid from Mohammed Premier University (Maroc) for agreeing to supervise my PhD.

I sincerely thank you for all your contributions, recommendations, priceless advice, and encouragement. I greatly appreciate the support of Mr Abdulhalim Dandoush. I am grateful for his help and valuable support.

Besides my advisors, I would like to thank the rest of my thesis committee for accepting to evaluate my work. I am grateful endlessly for the encouragement and the support of my parents. I am thankful for their patience and belief on me. This accomplishment would not have been possible without them.

I thank each and every person from whom I received help, without whom this work would have not have been possible.

Contents

List of Tables

List of Figures

Abbreviations

5G	Fifth Generation of Mobile Networks.
V2X	Vehicule-to-Everything.
NGMN	Next Generation Mobile Networks.
NS	Network Slice.
SDN	Software-Defined Networking.
NFV	Network Function Virtualization.
OPEX	OPerational EXpenditure.
CAPEX	CApital EXpenditure.
E2E	End-to-End.
ONOS	Open Network Operating System.
VM	Virtual Machine.
ETSI	European Telecommunications Standards Institute.
ISG	Industry Specification Group.
VNF	Virtual Network Function.
PNF	Physical Network Function.
NFVI	NFV Infrastructure.
MANO	Management and Orchestration.
VIM	Virtual Infrastructure Manager.
VNFM	VNF Manager.
NFVO	NFV orchestrator.
SFC	Service Function chaining.
COP	Control Orchestration Protocol.
SDO	Standards Developing Organization.
RAN	Radio Access Network.
CN	Core Network.

TN	Transport Network.
IETF	Internet Engineering Task Force.
3GPP	3rd Generation Partnership Project.
GSMA	Global System for Mobile Communications Association.
ACTN	Abstraction and Control of Traffic Engineered Networks.
NSSMF	Network Slice Subnet Management Function.
NSMF	Network Slice Management Function.
PoC	Proof of Concept.
COMS	Common Operations and Management on network Slices.
NSaaS	Network Slice as a Service.
NSP	Network Slice Provider.
NSA	Network Slice Agent.
NSI	Network Slice Instance.
NSSI	Network Slice Subnet Instance.
PCE	Path Computation Element.
MPLS	Multiprotocol Label Switching .
VXLAN	Virtual Extensible Local Area Network.
QoS	Quality of Service.
GST	Generic Slice Template.
NEST	Network Slice Type.
NSC	Network Slice Customer.
SST	Satandardized Slice Type.
NST	Network Slice Template.
TNS	Transport Network Slice.
SLA	Service Level Agreement.
CSC	Communication Service Customer.
CSP	Communication Service Provider.
NOP	Network Operator.
CSMF	Communication Service Management Function.
PLMN	Public Land Mobile Network.
NSD	Network Service Descriptor.
eMBB	Enhanced Mobile Broadband.
URLLC	Ultra Reliable Low Latency Communications.

mIoT	Massive Internet of Things.
UE	User Equipement.
SD	Slice Differentiator.
S-NSAAI	Single-Slice Selection Assistance Information.
C-RAN	Centralized RAN.
BBU	Base-Band Unit.
RU	Radio Unit.
CU	Centralized Unit.
DU	Distributed Unit.
CNC	Customer Network Controller.
MDSC	Multi-Domain Service Controller.
PNC	Provisioning Network Controller.
CSM	Customer Service Model.
SDM	Service Delivery Model.
NCM	Network Configuration Model.
DevCM	Device Configuration Model.
VN	Virtual Network.
VNS	Virtual Network Service.
VNAP	Virtual Network Access Point.
TE	Traffic Engineering.
KPI	Key Performance Indicator.
REST	Representational state transfer.
CPNC	Connectivity Provisioning Negotiation Protocol.
TNSI	Transport Network Slice Identifier.
TNSII	Transport Network Slice Interworking Identifier.
NESSMA	Network Slice Subnet Management Framework.
MDO	Multi-Domain Orchestrator.
DO	Domain Orchestrator.
CFS	Customer-Facing Service.
RFS	Resource-Facing Service.
SDF	Subnet Discovery Function.
WIM	WAN Infrastructure Manager.
TED	Traffic Engineering Database.

SAP	Slice Access Point.
DSCP	Differentiated Services Code Point.
TSBUR	Total Slice Bandwidth Usage Rate.
TSFAR	Total Slice Flow Acceptance Rate.

CHAPTER 1

Introduction

Chapter content

1.0.1 Problem Statement

NETWORK Slicing is a common-sense approach to cope with the explosion of services, business and customer demands of the Fifth Generation of Mobile Networks (5G) era and beyond. In fact, NGMN [1] has already developed a non-exhaustive list of 25 prominent mobile broadband use cases with their highly diversified requirements in order to support its vision for the 5G networks. "Remote surgery" is an example of a customer use case that requires ultra-high reliability and at the same time ultra-low latency communications. Because of the high variety of new emerging services, their conflicting and their extremely diversified requirements, the traditional ”one-size-fits-all” network approach is no longer efficiently feasible. Therefore, network slicing is proposed as a key feature that allows multiple dedicated virtual networks, called network slices (NSs) to operate on a common physical network infrastructure. a Network Slice (NS) is a logical isolated network instantiated over multi-domain, multi-technology physical networks that provides resource guarantees.

Built on top of emerging technologies such as software defined networking (SDN) and network function virtualisation (NFV), network slicing should select the appropriate compute, storage and network resources and manage them such as to meet the customer service requirements [2]. In contrast to the traditionnel model where all types of services are provided through a single network, NSs are specifically designed and instantiated to target one or several predefined use cases with similar quality of service requirements.

Likewise, it is possible to accommodate the new 5G services in a traditional network model but, at the expense of the operator cost in terms of Operational and Capital Expenditures (OPEX and CAPEX) and the complexity of network management. The main reason behind such high cost is the underuse of network resources. As an example, it is common for an operator to serve high resource demanding applications and reserve excessive amounts of resources that may be slightly solicited. Therefore, shifting from the traditional “one size fits all” to the Network Slicing model opens up opportunities for service providers and operators to increase their revenue while improving network scalability, flexibility, efficiency and management [3]. However, to fulfil its promises, the NSs implementation is constrained by overcoming many challenges such as isolation, modelization, security and the efficient share of compute and network resources [3].

1.0.2 Research Challenges

The works of our thesis are particularly oriented towards the network slicing technology modelization as well as the resource optimization challenge in slice-based infrastructures (Figure 1.1). Our objective is to bring forth an up-to-date end-to-end (E2E) 5G network slicing model that guides researchers and developers in the building of their future NS implementations. We also aim to develop and integrate a novel mechanism at the post-deployement phase of NSs that provides them with the charachteristics of elasticity and self-adaptability. Hence, a NS is automatically scaled up/down during its operation time with free resources of its neighbor slices in order to optimally accommodate difficult network conditions.

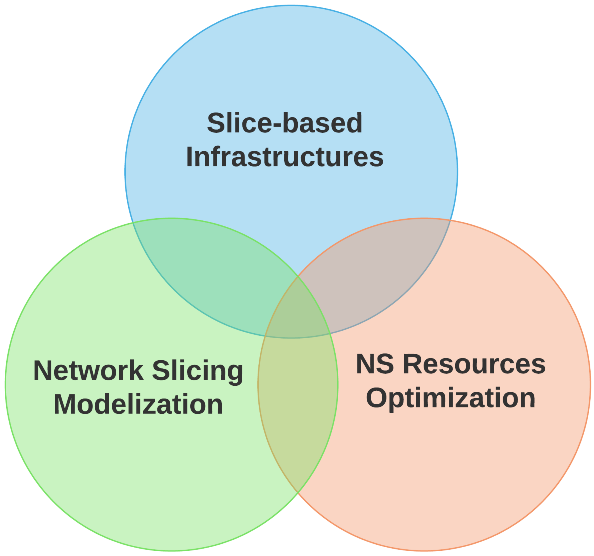


Figure 1.1: *Thesis Challenges*

This thesis address two main challenges of slice-based infrastructures as illustrated in Figure 1.1:

- **`Network Slicing Modelization`**: This a crucial phase that must be carried on any technology before its realization. The state of art in the network slicing field is lacking for an explicit 5G end-to-end network slicig model that can be refered to by researchers and developers for the management of NSs under their NS solutions. Furthermore, network slicing standardization is led by different organizations having each a different vision on network slicing and targeting only a specific part of the network infrastructure. Therefore, our first and second contributions answer respectively to the following research questions:

- **RQ-1**: *How to define an end-to-end network slicing model that provides a unified vision of the life-cycle management process of NSs ?*
- **RQ-2**: *How to build SDN/NFV-based solutions enabling the management and the orchestration of NSs ?*

- **Optimization of NS Resources**: resource optimization is one of the top priorities of network operators since it increases their revenue while improving the network scalability. The shift to the network slicing model will require the adoption of new mechanisms of network resource optimization. NS reconfiguration is actualy proposed to augment NS resources or to simply improve the network usage rate. However these operations follow an expensive two-phase strategy that should be frequently repeated all along the operation time of the network. Our third contribution answers the following interrogation by providing a novel mechanism to optimize bandwidth resources of NSs:
  - **RQ-3**: *How to optimize netwok resources in slice-based environnements ?*

1.1 Context

SDN, NFV and network orchestration technologies are the key enablers of network slicing. In this section we briefly define each of them and explain how they relate to each other in the fast delivery of new services.

1.1.1 Software-Defined Networking (SDN) [4]

The main motivation for the adoption of the SDN technology is the complexity of traditional networks. In fact, adding new features is very difficult. It can be achieved only through the installation of new firmware or updates to the operating system of the network equipment. Moreover, these functionalities are introduced in specialized, expensive and very difficult to configure equipments (middleboxes) such as firewalls, loadbalancers, intrusion detection system.

To cope with these challenges, the SDN approach separate the control plane of network devices from data plane and centralize it in a SDN controller (Figure 1.2). On the one hand, the controller, acting as a network operating system opens the network to applications and enables network programmability. On the other hand, it creates flow rules in the data plane via a southbound protocol such as openflow [5] in order to

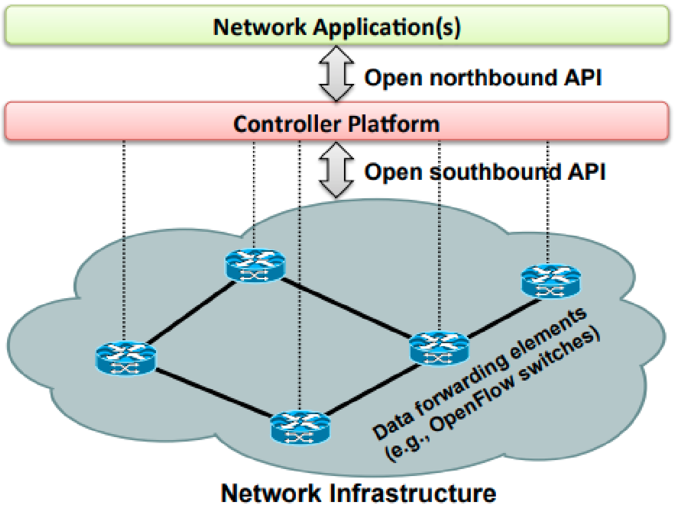


Figure 1.2: *SDN architecture: Simplified view [4]*

achieve the desired network behaviour. The most popular SDN controllers are ONOS, OpenDaylight, Ryu and Floodlight.

1.1.2 Network Function Virtualization (NFV) [6]

The NFV technology makes it possible to provide network services in virtual machines (VMs) running in cloud infrastructures. Each VM performs different network operations (Firewall, intrusion detection, Deep Packet Inspection, Load balancing).

The standardization of NFV technology has been led by the European Telecommunications Standards Institute (ETSI) through its NFV ISG (NFV Industry Specification Group) research group. Figure 1.3 illustrates a high level view of the NFV architecture. It is mainly composed of 3 parts:

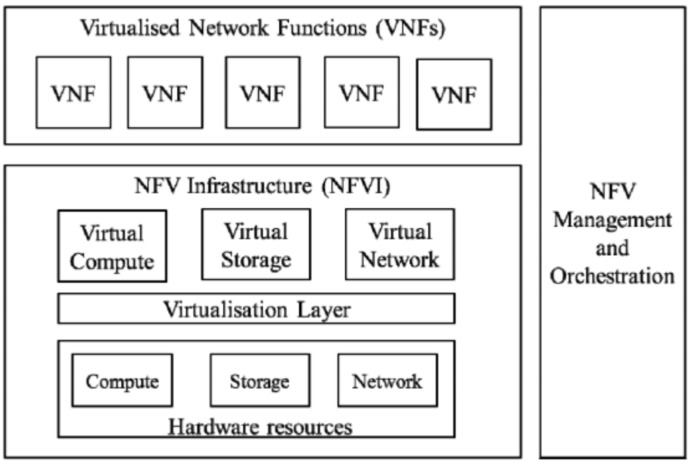


Figure 1.3: *High-level NFV framework [6]*

- **Virtual Network Functions (VNFs)**: It is the virtualization of a certain network function in one or more virtual machines. Each VNF operates independently from the others and can be divided into several sub-functions called VNF components (VNFCs). VNFs are supervised by Elemental Management Systems (EMs).

- **NFV Infrastructure (NFVI)**: this part includes all the necessary software and hardware resources for the deployment, operation and supervision of the VNFs. The virtualization layer is vital here since it allows the abstraction of hardware resources (processing, storage, network connectivity). This layer also ensures the independence of the VNF software from the hardware.

- **NFV Management and Orchestration (NFV MANO)**: It is made up of 3 components. First, the Virtualized Infrastructure Manager (VIM) which manages the interaction of VNFs with the physical resources under its control (allocation, release, inventory). Second, the VNF Manager (VNFM) which is responsible for the life cycle management (initialization, suspension and termination) of VNFs. Finally the NFV Orchestrator (NFVO), the element responsible for the realization of network services in the NFVI.

1.1.3 SDN/NFV [6]

SDN and NFV technologies are often discussed as being a single technology (SDN/NFV), this is due to the fact that they are complementary and able to provide a single network solution allowing a shift to opensource hardware and software. SDN can provide connectivity between VNFs in a flexible and automated manner while NFV makes use of SDN as part of Service Function Chaining (SFC). In this case, SDN controllers and management applications can run as VNFs in a scalable environment and therefore benefit from essential features such as availability, reliability and elasticity.

1.1.4 Network Resources and Service Orchestration [7]

Orchestration is the automation of tasks associated with the arrangement, management and coordination of services deployed across different technology/administrative domains with the objective of exposing them as a single service instance. It also consists in guaranteeing an adequate service performance throughout the duration of the delivery despite the concurrent use of resources between users and breakdowns of service. This is performed by the monitoring function of orchestration. Specifically, the orchestrator, based on feedback from monitoring tools, is also responsible for managing exceptions or deviations from workflows and thus adjusting the resources allocated to recover from failures or degradation of services [8].

An end-to-end NS solution cannot be achieved without end-to-end orchestration and management. As it will be explained later on this thesis, NS life cycle management involves a set of workflows between the service provider and operators. The delivery of the network services depends highly on the successful automation and coordination of these workflows. Therefore, in the context of the European 5G exchange project [9], an End-to-End Orchestration and Management (E2E MANO) reference architecture [10] was proposed (Figure 1.4).

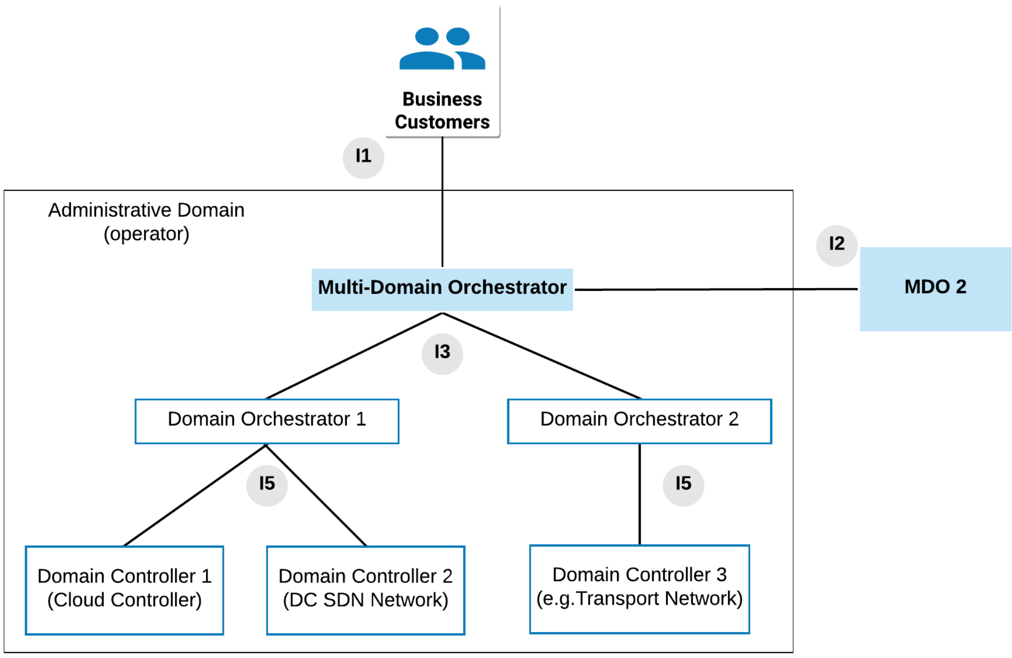


Figure 1.4: *End-to-End Management and Orchestration reference architecture [10]*

The architecture distinguishes three layers:

- **The resources layer**: includes different domains (eg. SDN networks, Data center) exposing an abstraction of their resources on interface 5.

- **The single-domain orchestration layer**: compromises multiple domain orchestrators (DOs) performing resource and/or service orchestration based on the abstraction exposed on interface 5.

- **The multi-domain orchestration layer**: composed of several multi-domain orchestrators (MDOs) that interact with DOs via interface 3, thus performing each resource and/or service orchestration in an administrative domain (eg.operator). Interface 2 is a business-to-business interface allowing provisioning and orchestration of resources across administrative domains. Lastly, Interface 1 that allows the interaction with business customers and the specification of their service requirements.

An example of a resource/service orchestrator is the NFVO defined in the aforementioned NFV framework. The functionality of the NFVO may be split into two main functional blocks. First, a resources orchestrator allocating NFVI resources (compute, storage, network) to be consumed by VNFs and network services. Second, a service orchestrator responsible for the life cycle management of network services as well as the instantiation of VNFs in coordination with the VNFM.

The NFVO is able to perform both resource and service orchestration and can be placed in the aforementioned E2E MANO reference architecture either as a DO or an MDO. Furthermore, two options related to the split of the NFVO's main functionalities (service and resource orchestration) are proposed in the ETSI NFV-IFA group specification [11]. The first use case address a network operator offering its infrastructure to different administrative domains (within the same operator). This leads to a separate implementation of the service orchestrator (NFVO) from resource orchestrators placed in each administrative domain. By comparison to the first option, the second one brings the operator services to the different administrative domains. Hence, a top service orchestrator called *Umbrella NFVO* is introduced interacting with NFVOs in each domain and taking in charge the composition of network services.

One of the challenges of orchestration is the ability to coordinate and manage resources and services in different technology/administrative domains, also called multi-domain orchestration. For this end, the Control Orchestration Protocol (COP) has been proposed and developed as part of the STRAUSS project [12]. The latter offers a set of transport services (topology service, connectivity provisioning, path calculation) shared between several control/orchestration systems.

The COP allows the interworking of heterogeneous control plane paradigms (OpenFlow, GMPLS / PCE). It provides a common north/south interface for SDN controllers allowing multiple implementations to be orchestrated by a single common protocol.

1.2 Our Contributions

The objective of our thesis is to tackle the resources optimization issue in slice-based infrastructures at the post deployement of NSs. This will be achieved by the design and the development of a NS solution that adopt a novel optimization mechanism. However, before any implementation, we require the definition of an network slicing model that details the life cycle management process of NSs from the service order request to the NS deployement. This model is meant to serve particularly as a reference for our implementation and eventually for others as well. We note that our vision on the network slicing model is based on the standards developped by the leading standards Developing Organizations (SDOs) in the field.

A NS is composed of several units within the network infrastructure called NS subnets. The management of a NS comes down to the coordinated management of all its subnets. Since the NS subnet management function is resource-facing and is directly related to our solution, we study the design and the requirements to be satisfied by a NS subnet management framework. Indeed, each component of the proposed framework relies on our vision on the end-to-end 5G network slicing model. Therefore, our contributions comes in a logical sequence to reach the objective of the thesis as depicted in Figure 1.5.

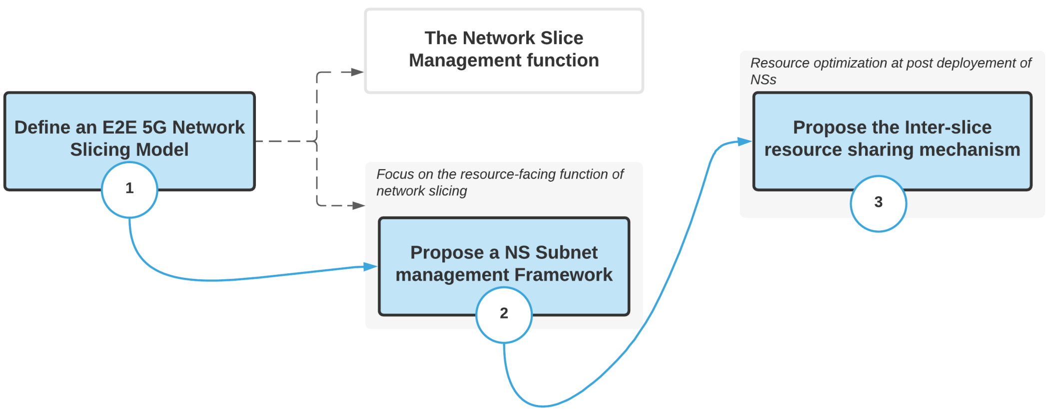


Figure 1.5: *Our contributions*

1.2.1 An E2E 5G Network Slicing Model

Today, the Network Slicing technology is massively addressed by the research community. However, Network Slice (NS) modelling details from Standards Developing Organizations (SDOs) are not yet well considered for E2E NS implementations. In addition, each SDO develops standards targeting only a specific part of the NS architecture. Therefore, based on a profound analysis of the major existing works, we propose an E2E network slicing model derived from the NS modelling works in Radio Access Networks (RAN), Core Networks (CN) and also Transport Networks (TN). The end goal is to clarify the E2E Network Slicing process from the service order request to the NS deployment and life-cycle management. Last, as there is no consensus on a specific information model in the Transport network domains (iii) we provide our vision on how several data models, developed by the Internet Engineering Task Force (IETF) working groups, can be integrated together in the context of the Abstraction and Control of Traffic Engineered Networks (ACTN) architecture in order to provision and manage Transport NSs [13].

1.2.2 A NS Subnet Management Framework

Current NS demonstrations are not fully aligned with the ongoing standardization activities, particularly with the Network Slice Subnet Management Function (NSSMF) defined by the 3rd Generation Partnership Project (3GPP). Our contribution provides a novel Framework for NS Subnet Management, called NESSMA, that satisfies a list of requirements derived from our vision on the end-to-end network slicing model detailed in chapter 2. The Framework is designed to jointly manage NS Subnets and their supported services having in mind its integration with NFV-MANO [14].

1.2.3 The Inter-Slice Bandwidth Resource Sharing Mechanism

Our third contribution presents the inter-slice bandwidth resources sharing, a novel mechanism for NS resource optimization based on the IETF Common Operations and Management on network Slices (COMS) information model. A proof of concept (PoC) is realized to demonstrate the benefit of the proposed approach. To the best of our knowledge, our PoC, so-called **InterS**, is the first to implement the COMS data model and evaluate the performance of the concept via realistic intensive experiments in a Software Defined Network (SDN). Thus, InterS allows a congested NS to temporarily acquire all or a part of the free bandwidth resources of neighbor slices and use them to serve its own traffic. Experiments have shown that InterS can significantly improves the operator's network bandwidth usage as well as the flow acceptance rates.[15].

1.3 Thesis outline

Our thesis is organized as follows:

- **Chapter 2:** we provide an E2E 5G network slicing model based on an in-depth analysis of the modelization works published by the leading SDOs such ETSI, Global System for Mobile Communications Association (GSMA), 3GPP and IETF. First, we start by providing a general architecture of network slicing common to the different SDOs. The latter serves as a stepping-stone to set the primary foundation of network slicing before diving deep into domain-specific visions. Second, we explain the two different visions of ordering a NS among SDOs. Third, we detail exhaustively the network slicing process in each of the RAN, the Core and the transport networks. Last, we address some of the open standardization issues as well as the future research directions in this area.

- **Chapter 3:** we provide a list of requirements that should be satisfied by a NS subnet management framework. We derive the requirements as well as the design of the framework called NESSMA from an exhaustive analysis of up-to-date standards from 3GPP, ETSI and IETF SDOs. NESSMA is designed to manage NS Subnets as well as their supported services' life cycles taking into account its integration with NFV-MANO

- **Chapter 4:** we realize a proof of concept of the inter-slice bandwidth resource sharing approach based on the COMS NS information model. Our implementation is based on the modelization works presented in previous contributions. Our PoC, so called InterS, allows the instantiation of a NS either as exclusive or shared. Our PoC includes two main implementations. The first one allows a congested NS to acquire free bandwidth resources from at most one neighbor NS. The second implementation, allows the congested NS to acquire and aggregate fragments of free resources from multiple neighbor NSs.

- **Chapter 5:** we conclude the thesis and discuss the perspective of our work. It summarizes the contributions, discuss the obtained results and highlights the remaining gaps that are subject to future investigations.

CHAPTER 2

An E2E 5G Network Slicing Model

Chapter content

2.1 Introduction

Network slicing is an ongoing standardization work performed by different SDOs targeting each a specific part of the network slicing architecture. E2E NS implementations have to be aligned with the SDOs' standards particularly with the NS information models. These models describe all the NS management entities as well as their relationships from the perspective of operators and service providers. Currently, an E2E network slicing model is not yet explicitly provided but need to be derived based on an exhaustive analysis of all the SDOs' contributions particularly their proposed NS information/data models to this day.

In our work presented in [16] we observed that the issued contributions may potentially be unified and represented in a common architecture and terminologies. Today, we go beyond this simplified vision. To the best of our knowledge, we are the first to provide by this work a comprehensive survey stitching together the different NS modeling works in RAN, CN and TN domains in order to bring forth an up-to-date E2E network slicing model. Prior works [17] [18] [19] [20] [21] addressed the network slicing concept in an abstract manner providing the research community with a general understanding on the topic. Instead, the work developed in this chapter will provide researchers and developers with an in-depth technology-independent NS definition necessary to build their complete NS solutions.

Our chapter is organised as follows. Section 2.2 provides a common and general vision of network slicing among the leading SDOs. It serves as a stepping-stone to set the primary NS foundations in terms of actors, architectures, functions and workflows. Section 2.3, introduces the NS template in order to request NS services. In section 2.4, we present a domain-specific vision of network slicing diving deep into the NS modeling details of each of the RAN, the CN and the TN domains. Then, our vision on the use of the available transport network data models for achieving network slicing in the TN domains is explained in section 2.6. We address also in this section some of the open standardization issues as well as future research directions. Section 5 concludes the chapter.

2.2 Network Slicing: A General Vision

This section provides the main NS workflows, functions, interfaces and the managed objects that are supposed to be common, to a large extent, to different SDOs, administrative and technology domains. The goal is to provide a general architecture that clarifies the basic E2E network slicing functionality before diving deep into domain-specific visions. The need for such a general vision was first identified by ETSI NGP workgroup and then led to the adoption of the reference network slicing architecture [22] depicted in Figure 2.1. This architecture comprises three actors:

- Tenants that consume the service supported by a NS. The service can itself be a NS, commonly known by the term NS as a Service (NSaaS).
- NS Providers (NSPs) that provide access to the NS instances.
- NS Agents (NSAs) that have the complete view and control of their own network domain.

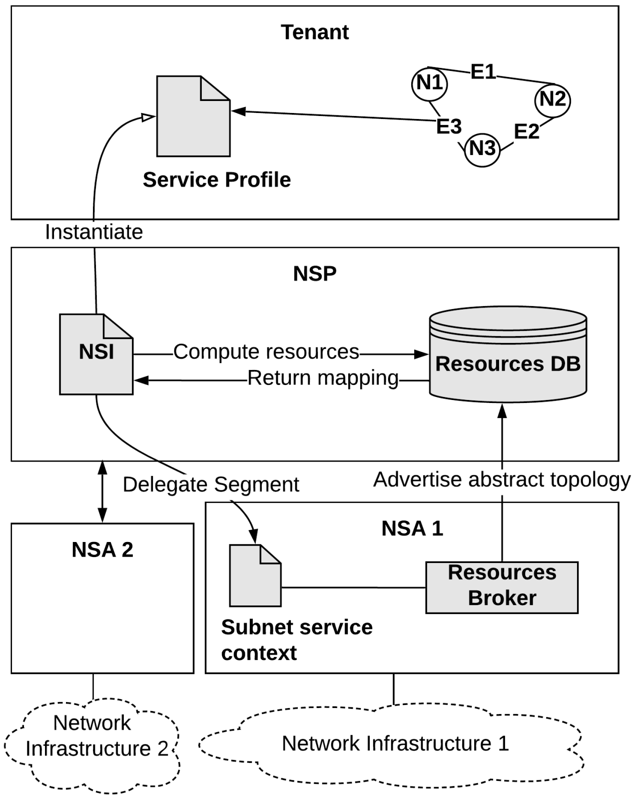


Figure 2.1: *ETSI NGP NS Framework. Introduction of the NS managed objects, NS workflows and functions in the NS architecture.*

Firstly, a tenant prepares a *service profile* that describes its desired NS. It comprises a *service graph* and some additional service attributes such as the service type, service profile identifier and the service subscribing entity. The *service graph*, represents the tenant's view of the NS so that on the one hand it describes the nodes in terms of compute, storage and service instance type, e.g., firewall and load balancer. On the other hand it defines the edges as the slice constraints, also called slice characteristics such as link bandwidth and packet loss. *Service endpoints* are additional attributes of the service profile that describe the entry and termination point of a service. The tenant uses then that service profile to order the creation of a NS.

Within the NSP, the *resource database* (DB) aggregates the exposed topological information from several domains in the underlying infrastructure along with their constraints and their belonging domain as shown in figure 2.1. Therefore, the resources DB provides the NSP with the full view of the abstracted underlying physical infrastructure from many NSAs. At the NSA level, a *resources broker* is in charge of gathering the domain-specific topology information (links, nodes) along with the constraints such as links cost, bandwidth, and latency. Then, for security reason these topological data are abstracted and exported to the resources DB at the NSP domain. This is done through the *subnet discovery* function located in each NSA.

The NSP creates the *NS Instance (NSI)* as an instantiation of the service profile once received from the tenant. The new created object will represents the NSP's own view of the NS. In fact, the aim of the NSI is to compute the mappings of the service profile to the abstracted infrastructure elements stored in the resources DB taking into consideration the tenant constraints. This is done using the following two functions defined in the ETSI slicing architecture: the *resources computation* and *NS mapping* functions. Table 2.1 overviews the high level network functions performed by NSP and NSA of this architecture.

The bindings of the service graph to the physical resources are then stored in the *runtime service context object* within the NSI. The usage of the term NSI here is relative to ETSI NGP and is not aligned with the other SDOs. To clear up this confusion, it is necessary here to differentiate the term NS from NSI. The former is a definition of a logical network in terms of a managed set of resources and network functions. An NSI is an activated NS, a set of resources and network functions forming a deployed logical network that satisfies the tenant's service requirements [23].

After computation and mapping functions are successful, the final step, at the NSP level, towards the deployment of the NS is to distribute the so-called *segments* (also called *subnets* by other SDOs) to NSAs. This is done through the NS *delegation function*.

Table 2.1: *NS functions*

NS functions	***Brief description***
Subnet Discovery	Exports an abstracted topology of the NSA domain to the resource DB.
Resource Computation	Determines the adequate resources for a specific service graph.
NS Mapping	Associates infrastructure elements with the logical ones.
NS delegation	Delegates segments to NSAs for mapping, encapsulation and segment interconnection.
Report aggregation	Monitor characteristics from different slice subnets for accounting and performance for business model.
Service assurance	Monitors each flow in a NS by NSP and takes decisions in order to provide service assurance.
Subnet augment	Allows the tenant to request an increase or a decrease of its allocated resources.

Providing that a NS may span across several NSA domains, a segment is the set of paths and nodes a NSI is allocating to a specific NSA. After a NSA receives the segment associated to a NS, it first computes the mapping of abstract to concrete resources from the subnet resource broker with the help of a *child path compute element (PCE)* and stores them on the *subnet service context run-time object.* NS segments are not meant to be deployed separately, they are all part of the same logical E2E NS. Hence, a *NS Gateway* is needed to perform segments interconnection. Also, its is exported as a node to the NSP during the discovery process with the next hop information pointing to the adjacent network domain.

The possible modes that describes the flow of tenant traffic within its assigned NS, as defined by the ETSI architecture, are threefold. First, *E2E encapsulation* where the tenant flow undergo the same encapsulation type such as MPLS, VXLAN, etc, along all the segments. Although, this mode appears to be simple, it does not promote the technology agnostic approach and leads to scale related issues. In contrast to the first mode, *segmented encapsulation* allow each NSA to independently choose its encapsulation technique. For example, the same tenant flow is encapsulated using VXLAN in some segment and MPLS in another one thanks to some of the NS Gateway functions [24].

Once a segment on a NSA's domain is successfully deployed, the resource broker inside the NSA updates the NSP information using the *subnet discovery function*. Furthermore, service characteristics should be collected at each segment throughout the slice life-cycle, and then reported to the NSP. Collecting statistics and aggregating performance metrics about both the slice subnets and also the flows in the context of a NSI is the role of the *report aggregation and the service assurance* functions respectively. They provide a complete E2E monitoring framework for accounting, performance, Quality of Service (QoS) of the business model and service assurance.

Network slicing includes also exposing the capability to the tenant in order to scale up or down its allocated resources at run-time. This resources augmentation is performed by the *subnet augment function*. A tenant *augment request* may be a change on the latency constraint from 10 ms to 5 ms for instance. This may lead to some additional operations that could affect the system stability. In particular, if the current path does not support the new latency constraint, the operations include setting up a new path to meet the new requirements, then moving flows from the older path to the new one and finally deprovisioning of the participating nodes in the abandoned path.

The subnet augment function requires a set of operations and corresponding interface to the underlying infrastructure. These are covered by the *tenant operated network service function*. For example, the tenant uses the "augment" operation to modify its constraints associated with a path or a node in the service graph.

To conclude, the ETSI network slicing architecture, following the NGMN conceptual outline [25], includes three layers: a service layer where the service graph is built, then the NS instance layer where the service graph to abstract resources mappings happens and lastly the resource layer where segments are delegated and deployed on the underlying infrastructure. Concerning the NS life-cycle, we can refer to the 3GPP specification that considers 4 phases: Preparation phase including design and pre-provisioning, an “instantiation, configuration and activation” phase, a run-time phase including supervision and reporting, as well as upgrade, reconfiguration and scaling, and a decommissioning phase. These phases are similar for all the SDOs.

2.3 Ordering a NS: Service Graph or NS Template ?

In the ETSI NGP architecture, the tenant is not necessarily an end customer, it may play the role of a service provider that offer its own services on top of the ordered NS instance (NSaaS model). This explains why the tenant is ordering a NS and playing in some sort the same role as the NSP. However, an end customer is rather interested in the fulfilment of a service whether it is supported by a NS in the underlying infrastructure or not.

While the ETSI NGP uses a *service graph* to order a NS, GSMA has entirely a different vision. GSMA is investigating all the potential service attributes, also called *NS characteristics*, from several enterprises and vertical industries, and gathering them in a template called *Generic Slice Template (GST)*. The goal is to define a template (see table 2.2) that helps both the NS customer (NSC) and NSP to identify a NS [26]. The GST will be used as a reference by vendors, operators, providers and customers in order to deploy a NS that accommodate a certain use case. Thereafter, to order the deployment of a NS, a NSC needs to fill some or all the GST template attributes with its desired values and/or ranges depending on the NS use case. The obtained template is then called *NEST (NS Type)*.

Table 2.2: *GSMA's NS templates*

NS templates	***Brief description***
GST	A template containing a set of attributes that characterize any slice.
NEST	A GST filled with values to define and identify slices.
S-NEST	A NEST template that defines a standardized slice.
P-NEST	Operator-specific NEST created based on negotiation between an operator and its customers.

In fact, the GSMA approach has several advantages. First, it gives the mean for a NSC to express its NS service requirements and at the same time allows operators and providers to fulfill any possible use case. Second, *Standardized slice types (SSTs)* are defined for the most popular use cases in standard NEST templates (S-NEST) and diffused to all operators around the world. Third, service capabilities offered to a user inside its home operator are conserved when it roams to a visited network. This could simply happens by exporting the NEST templates from the home network to the visited one. That is, NEST is a generic template that aims at describing the service requirements of a use case that will later help an operator to select an appropriate *NS template (NST)* [26] [27] also referred to as NS blueprint in [25].

The NST contains the network functions, their interconnections and the necessary configurations to meet the service requirements described in the NEST. As it is depicted in figure 2.2, a NST may correspond to one or multiple NESTs and is further instantiated to realize a NSI.

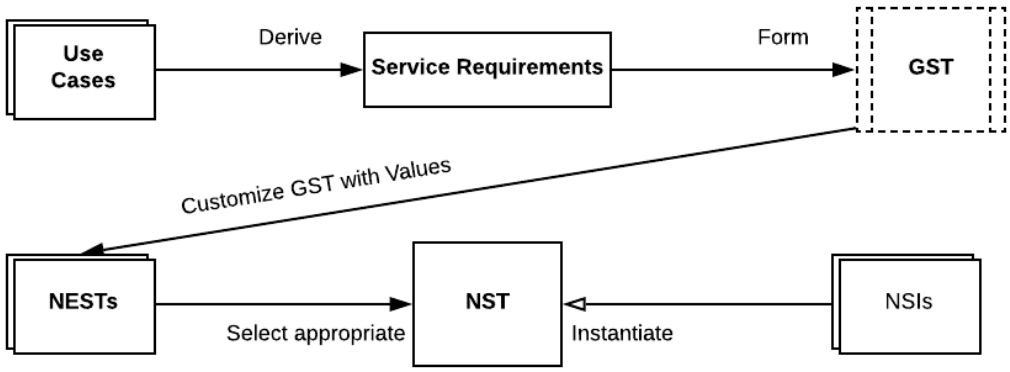


Figure 2.2: *Involving the GSMA GST/NEST in the instantiation phase of a NS.*

The detailed list of all the GST attributes as well as their explanations are provided in [28].

After this introduction of the network slicing general vision and how ordering a NS can be done with the use of the service graph or the NS template. The next section, will provide an in-depth exploration of the E2E network slicing information models within the RAN, CN and TN specific domains.

2.4 Network Slicing: A Domain-Specific Vision

For a given 5G customer service, an E2E NS that spans across the RAN, the CN and the TN needs to be established. Following the set up of the RAN and core sub-slices (subnets) by the 3GPP orchestration system, one or many transport NSs (TNS) has or have to be provisioned within the TNS provider as a set of connections with Service Level Agreements (SLAs) that connects together RAN and core sub-slices [29].

2.4.1 RAN and CN

RAN and CN slicing is still under standardization by 3GPP. In fact, 3GPP is tremendously contributing to the NS standardization work through several working groups particularly SA1 (service requirements), SA2 (Architecture), SA3 (Security), SA5 (Network Management). In order to get deeper insights on NSs, their implementation and life-cycle management we focus in this section on the 3GPP NS information model [30].

The SA5 group has defined the entities that compose a NS and the way they interact with each other using the information model depicted on figure 2.3.

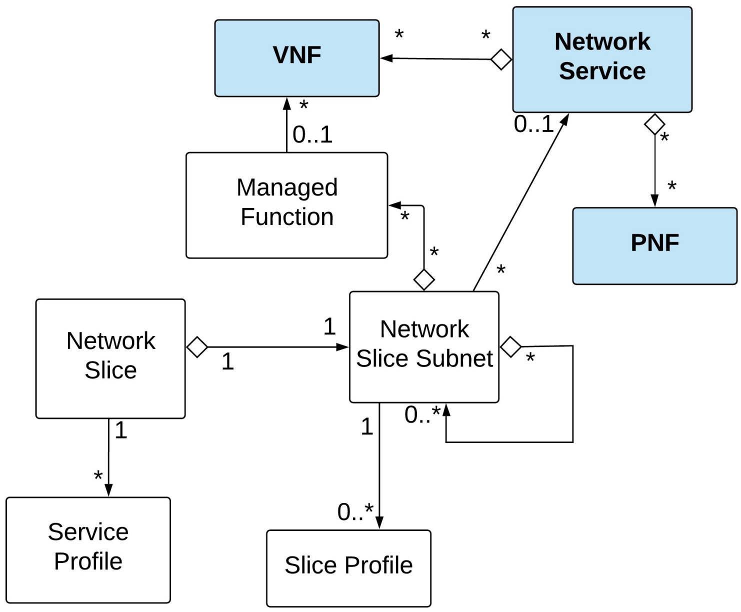


Figure 2.3: *3GPP NS Information Model*

Communication service is the term used by 3GPP to refer to the customer ordered service. In general, It is carried to the service provider as a set of service requirements that serve a certain business purpose. Based on those requirements the service provider derives the NS related requirements and orders the creation of the NS to the network operator. As indicated in the information model, the NS object supports a list of service profiles, each of them corresponds to a communication service and maintains the derived NS related requirements.

The NS also makes reference to its constituents subnets (this term is interchangeable with segments) which may recursively be composed of other subnets. Likewise, several slice profiles are associated to each subnet. The reason behind the modelling of a NS/NS subnet supporting multiple service/slice profiles is that multiple communication services may share the same NS. The data model describing all the entities in figure 2.3 except the ones in dark color are given in yang data modeling language in 3GPP TS 28.541 [30]. The service profile has many common attributes with the slice profile, some of them have reference in GSMA GST such as User Equipment (UE) mobility level.

A NS subnet, that aggregates a list of managed functions, is supported by at most one network service. A managed function is realized by one or many virtualized network functions (VNFs). A network service is a composition of the subnet's VNFs and PNFs (physical network functions) according to one or many forwarding graphs. The entities network service, VNF and PNF are out of the scope of 3GPP working groups. Thus, their descriptions are delegated to ETSI NFV-IFA VNF and network service information models [31] [32].

The 3GPP NS information model's entities are consumed, managed and provided by different actors. Each actor may play more than one role depending on the NS management use case. 3GPP has defined three main NS management roles (see table 2.3): Communication Service Customer (CSC), Communication Service Provider (CSP) and Network Operator (NOP).

Table 2.3: *NS management roles and actors [2]*

Actor	***Role***	***Description***
Basic customer	Communication Service Customer	Consumes a Communication Service but does not use the *Communication Service Management Function (CSMF)*.
Advanced customer	Communication Service Customer	Consumes a Communication Service. It has some capabilities limited by the CSP to manage the Communication Service via the CSMF.
NS Provider	Network Operator	Provides a NS and Consumes NS Subnet(s).
NS tenant	Communication Service Customer and Communication Service Provider	Consumes and manages a Communication Service and also NS(s) via the *NS Management Function (NSMF)*.
NS Subnet Provider	Network Operator	Provides a NS Subnet.
Communication Network Provider	Communication Service Provider and Network Operator	Provides Communication Service NS(s) and NS Subnet(s).
Network-as-a-service Provider	Network Operator	Provides NS and NS Subnet(s).

There are other business roles defined by 3GPP in [2] such as hardware supplier, NFVI supplier, network equipment provider, data center service provider, and virtualization infrastructure service provider.

A business agreement must be created between two actors in order to interact with each other. Commonly, real NS management use cases includes actors having multiple roles, therefore reducing the number of needed mutual relationships. For example an actor "NS Tenant" may simultaneously plays both the role of a Communication Service Customer (CSC) and a Communication Service provider (CSP).

The deployment and management of a NS in a NOP's infrastructure based on the customer's SLA are performed by three main 3GPP functions as illustrated in Figure 2.4. First the *Communication Service Management Function (CSMF)* receives the communication service related requirements from the CSC. Those requirements are converted to NS related requirements *(Service profile)* and then provided to the *NS Management Function (NSMF)*. Indeed, the CSMF manages the Communication Services provided by the Network Operator.

The NSMF creates and manages the NSIs based on the received NSs related requirements. Then, it derives the NS subnets related requirements *(Slice Profile)* and provides them to the *NS Subnet Management Function (NSSMF)* so that the NS subnet management is delegated to the NSSMF.

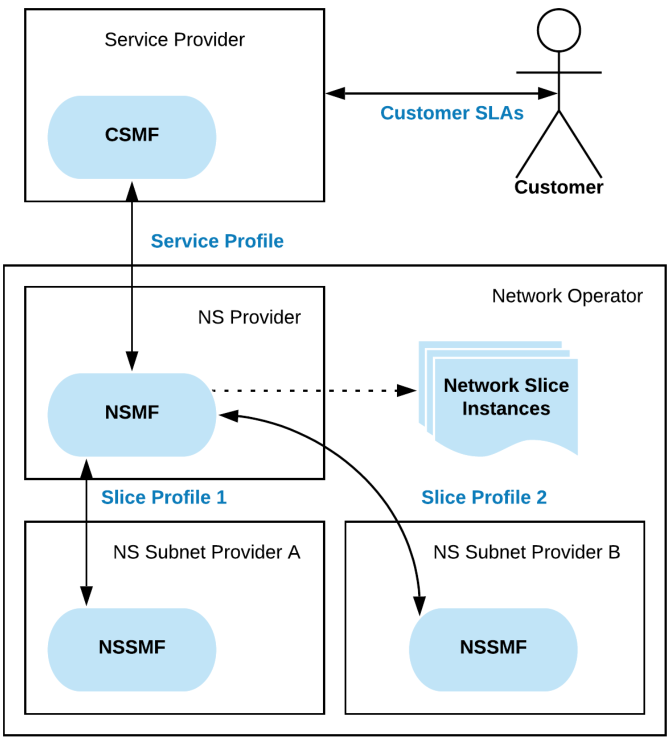


Figure 2.4: *3GPP NS Management Functions*

NS Subnets instances (NSSIs) are separated units associated to a NS instance (NSI). They can also be used (shared) simultaneously by another NSI if needed. Terminating a NSI doesn't necessarily affect those units, they are still existing and running but disassociated to the terminated NSI. This separation of the NSSMF from the NSMF offer a NSSI life-cycle management independent from the NSI life-cycle. Lastly, the NSSMF creates and manages the NSSIs based on the received NS Subnets related requirements.

NSs and NS Subnets are always provided by the NOP. Hence, the NSMF and the NSSMF are located inside it. Particularly, a NS is defined within a Public Land Mobile Network (PLMN). This is to say that a NSI is defined in one operator while its constituents NSSIs may be associated from different ones. Communication Services are generally managed by a CSP but might also be managed by the customer or the NOP so that there is a direct communication interface between them. Therefore, the CSMF might be placed at each of the CSP, the customer or the NOP depending respectively on the previously mentioned options.

The 3GPP NS information model defined in [30] lists all the attributes defining NSs. In the following of this section, we introduce the most relevant attributes for the understanding of the 3GPP network slicing vision. In the 3GPP data model, a NSI refers to a unique NSSI through the *NetworkSliceSubnetRef* attribute instead of multiples NSSIs. This is due to the modelling of each of the NS and the NS Subnet entities. As shown in figure 2.3, a NS is modelled as being composed of a unique NS Subnet.

A NSI may be well deployed and working but administratively configured by the operator to not serve any user [33]. Therefore the state of a NSI/NSSI is described by two attributes, the *operational state* and the *administrative state*.

A NS Subnet refers to its constituents subnets and supported managed functions respectively through the *NetworkSliceSubnetRef* and *ManagedFunctionRef* attributes.

The *NSInfo* attribute provides real-time information of a NSI that corresponds to a NSSI. This attribute contains many information such as the reference to the flavour of the *network service descriptor (NSD)* used to instantiate the network service and the reference to all information on network service's constituents VNFs and PNFs. An exhaustive definition of all the information provided by the NSInfo attribute is given by ETSI GS NFV-IFA 013 [34].

A same NS service may be provided by different operators so a user may roams from its home PLMN to one or multiple serving PLMNs without any service interruption. The PLMNs that support a NS or a NS subnet are identified respectively in each of the service profile and the slice profile as *PLMNIdList*.

The service profile includes the attributes of the slice profile in addition to the the *SST* and the *NS availability* attributes. The SST refers to the expected NS behaviour in terms of features and services [35]. 3GPP defines three standardized SSTs as described by table 2.4: eMBB (Enhanced Mobile Broadband), URLLC (Ultra Reliable Low Latency Communication) and mIoT (massive Internet of Things). At the time of writing, standardized mIoT characteristics are not yet included in the 3GPP slicing data model published in TS 28.541 [30].

Table 2.4: *Standardised SST values*

SST	***SST Value***	***Bref Description***
eMBB	1	Slice suitable for the handling of 5G enhanced Mobile Broadband.
URLLC	2	Slice suitable for the handling of ultra-reliable low latency communications.
mIoT	3	Slice suitable for the handling of massive IoT.

The identification of NSs in control plane is done using the *Network Single-Slice Selection Assistance Information* (S-NSSAIs). It is based on the SST and an optional complementary information, called *Slice Differentiator* (SD), that allows the definition of different NSs types around the same SST value.

3GPP distinguishes three options for the definition of a S-NSSAI that is comprised of:

- A standardized SST value and no SD.
- A standardized SST value and a SD.
- A non-standardized SST value and no SD.

The service and slice profiles are comprised of a collection of S-NSSAIs values also referred to as a NSSAI. The later indicates the different NS behaviours that can be accommodated by the NS/NS subnet. An NSSAI may be configured, requested or allowed.

A given user can have at most eight S-NSSAIs. During the registration procedure, the UE includes an NSSAI in its request to the PLMN. The later verifies the requested NSSAI against the subscription information. If the match is successful, the UE obtains the allowed NSSAI from the PLMN which may include one or many S-NSSAIs. In addition, the latter can be used simultaneously by the UE. Finally, a NS and a NSI can now be selected for that UE. Also a service/slice profile includes a list of performance requirements that are defined for each standardized SST value (e.g., low latency and high reliability) [36]. In addition, they comprises other characteristics that can be considered as NS/Subnet-level attributes such as Maximum Number of UEs, UE mobility level, latency [37].

2.4.2 Transport Network Slicing

2.4.2.1 Definition

A transport NS provides a set of connections between a group of VNFs or/and PNFs from both the RAN and the CN. Each connection has its own deterministic SLAs and is implemented in the underlying transport network (TN) using any technology type (IP, optics, microwave, etc), any tunnel type such as IP/MPLS and any Layer service type (L0, L1, L2, L3).

Depending on the RAN deployment, there might be multiple transport sub-slices per a single E2E NS:

- In the case of the distributed RAN deployment, in addition to the TN connecting together the RAN and the CN, another TN (i.e., Internet) connects the users' mobile applications to the CN. This is a common deployment in 4G and 5G networks.

- An additional transport network is introduced in the case of the centralized RAN (C-RAN) between the RAN's two functional units, the base-band unit (BBU) and the radio unit (RU), called Front-haul network. The BBU is further divided in the case of cloud RAN to include two separated units called the centralized and the distributed units (CU and DU). The former is hosted in the cloud while the latter remains close to the antenna. An additional transport network is then introduced in this case called Mid-haul in order to connect the DU to the CU.

In total, there might be a maximum of four transport sub-slices (Figure 2.5) per an E2E NS depending on the aforementioned deployment options [38].

Network slicing standardization in the transport domain is under the IETF realm. Actually, continuous efforts are being made to model the network slicing technology and

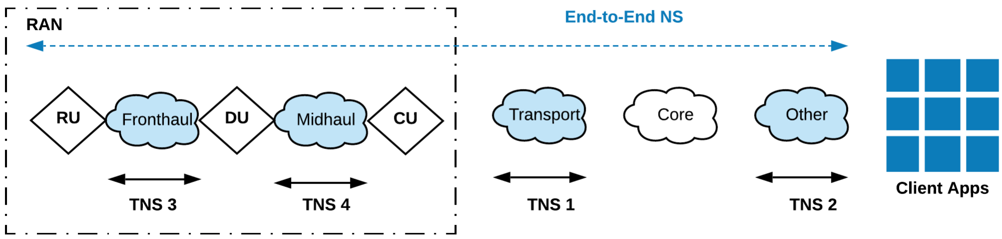


Figure 2.5: *The 4 types of a Transport NS: Fronthaul, Midhaul, Transport and an other Network separating the core to client applications.*

consequently to define a standardized architecture in the TN domains. In fact there are multiple mechanisms and data models covering different parts of the transport domain architecture that may be applied to the network slicing use case. However, there is no consensus within IETF on the use of this model or that one. In addition, it is not clear how they can be applied to network slicing. In the remaining of this section, we will list in a technology-independent manner all the candidate works within IETF that have the potential to accommodate network slicing in the the TN domain. Then, in the next section we will present our vision in this context.

2.4.2.2 ACTN Architecture

The most likely candidate architecture to accommodate network slicing is the one of the Abstraction and Control of Traffic Engineered (TE) Networks (ACTN) defined by the IETF TEAS (Traffic Engineering Architecture and Signaling) workgroup in [39]. ACTN is a set of management and control functions that facilitates the presentation of virtual networks built from abstractions [40] of the underlying TE networks to customers. TE networks or TE topologies mechanisms are key enablers of network slicing since they allow the dynamic provisioning of the E2E connectivity. The TE topology (or TE Network) model will be further discussed in this section.

The ACTN architecture (Figure 2.6) includes three controllers: the Customer Network Controller (CNC), the Multi-Domain Service Coordinator (MDSC) and the Provisioning Network Controller (PNC). Three interfaces are then introduced by this three-tier architecture: CNC-MDSC interface (CMI), MDSC-PNC interface (MPI) and Southbound interface (SBI).

The information exchanged in each of the cited interfaces are governed by a set of data models [41] structured by IETF into yang language modules. These models are

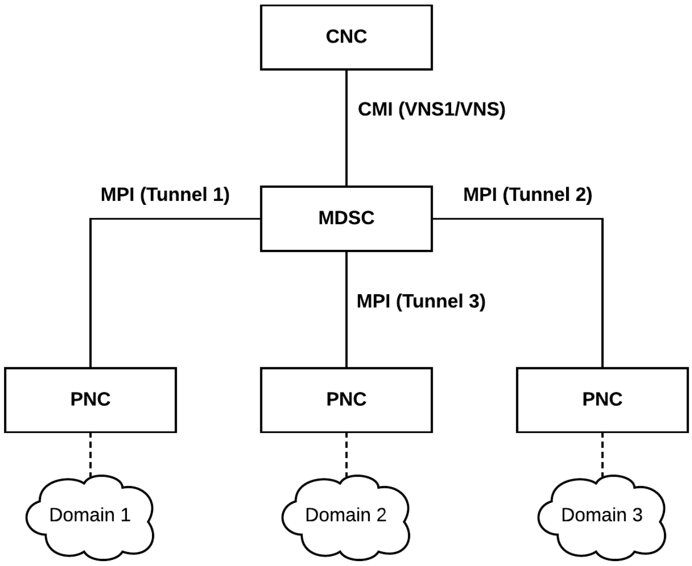


Figure 2.6: *The ACTN Architecture: CNC, MDSCs and PNCs.*

classified in general to 4 categories: *Customer service model (CSM), service delivery model (SDM), network configuration model (NCM) and device configuration model (DevCM)* [42] [43]. DevCMs are located in the SBI and are technology specific. While in this chapter the network slicing concept is surveyed in a technology-independent way, those models will not be covered in the following of this section. However, we will address all the other data models that we consider as enablers of the network slicing technology located in either the CMI or the MPI. Then, in section 2.6 we will explain our vision on how they may relate to each other in order to instantiate and manage NSs.

The CNC request the instantiation of a Virtual Network Service (VNS) to the MDSC via the CMI interface. The MDSC is at the core of the ACTN model obviating the need of the underlying technologies for network and service control while helping the E2E NS consumer to express the desired TNS by the mean of SLAs. This is achieved in the MDSC by the implementation of network-related functions such as multi-domain coordination, abstraction and service-related functions like service mapping/translation and virtual network service coordination.

From a bottom-up perspective, each PNC exposes an abstracted topology to the MDSC. The MDSC merges then all the advertised topologies at the MPI into its native network topology. Therefore, the MDSC provides an abstracted topology based on its view to the customer. It is worth mentioning that a hierarchy of MDSCs is very likely to be envisaged in real world deployments. In that case lower-level MDSCs (MDSC-Ls) are connected to higher-level ones (MDSC-Hs) using the MPI interface. Likewise, each MDSC-L advertises an abstract topology to the MDSC-H. The PNC may be implemented as an SDN controller, e.g., an active PCE-based or P4 controller to dynamically manage a network domain. We note that the ACTN Framework resumes most of the NS functions defined in section 2.2 by ETSI NGP with more focus and details on the abstraction functions.

2.4.2.3 TE Topology Model [44]

The TE topology model describes the provider's data store of abstract TE topologies provided to customers independently to any specific underlying technology. Indeed, a TE topology may be hierarchical so each network element of an overly topology is mapped to the next underlay topology until reaching the provider's native one. Within the data store, a TE topology comes with a set of TE informations related to nodes and links that are used for the selection of a TE path. Certain TE information along with the provider's policy or a negotiated customer-provider policy (bandwidth, shortest path, etc) are given as inputs to the abstraction process in order to produce selective information representing the potential ability to connect across the domain [40].

2.4.2.4 The Virtual Network (VN) Model [45]

In the ACTN architecture, a type 1 and type 2 Customer VNSs allow respectively the creation of type 1 and type 2 VNs defined by the VN model. A type 1 VN is a set of edge-to-edge abstract links (also called VN members) modelled as an abstract node along with a set of Virtual Network Access Points (VNAPs). In a type 2 VN, the provider disclose a detailed abstract topology as an underlay to the one abstract node, allowing customers to configure explicit path for some of the VN members.

2.4.2.5 The COMS Data Model [46]

An other data model outside the IETF TEAS working group called COMS [47] [48] is proposed for the technology-independent management of NSs. Similar to the TE topology model, COMS augments also the data model for network topologies [49] but with the essential entities facilitating configuration of QoS, reporting of statistics, QoS threshold monitoring, services instances, and description of compute and storage resources. Figure 2.7 shows our vision of the IETF NS data model with the augmenting entities (dark color).

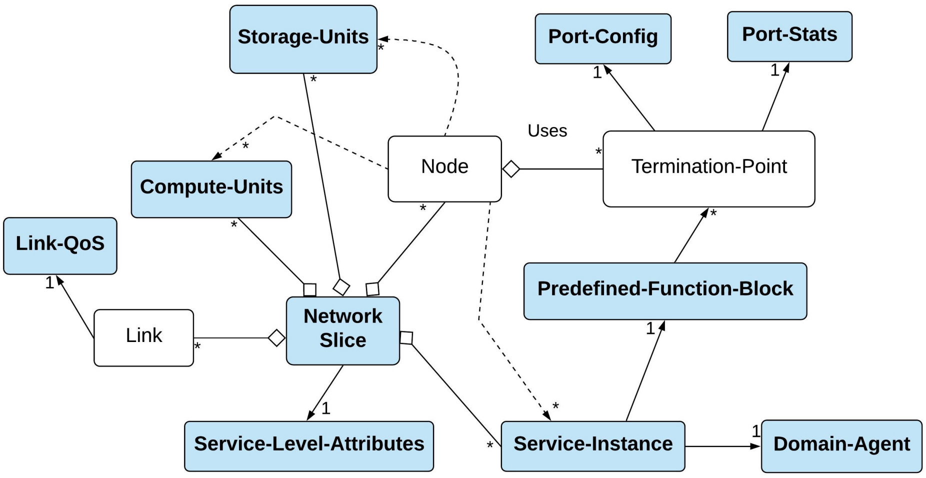


Figure 2.7: *IETF COMS Information Model*

As a result of the last augmentation, a node will make references to a list of compute units, storage units and service instances (e.g., firewall and load-balancer). This is illustrated in figure 2.7 with dashed arrows. Also, a *termination-poin* entity is described with configuration and statistic attributes. Lastly, a link is augmented with QoS attributes. The content of the augmenting entities is listed in table 2.5.

Table 2.5: *COMS NS augmenting entities*

NS Entities	***NS characteristics***
Port-Config	packet rate, packet loss probability, packet loss threshold.
Port-Stats	received packets, sent packets.
Link-QoS	link bandwidth agreement, link throughput, link throughput threshold, link-latency agreement, link-latency, link-jitter agreement, link-jitter, link-jitter threshold, path restrictions: list of mandatory nodes, list of mandatory links, list of excluded nodes, list of excluded links.
Compute-Unit	compute unit ID, number of Cores, ram, access mode, location, unit type.
Storage-Unit	storage unit ID, size, access rate, access mode, read-write mode type, redundancy type, location.
Service-Instance	service instance ID, Pre-defined Function Block: Domain-agent name, southbound(sb) IP address, sb port, northbound (nb) IP address, nb port, Function Id, Function name, IP address, port, list of termination points.
Slice-Level-Attributes	service time start, service time end, life-cycle status, reliability level, resource reservation level, availability, availability-threshold, Access-control: match, action, priority, counter.

2.4.2.6 The NS Stitching Data Model [50]

Furthermore, the COMS model is augmented to show up NS subnets and their interconnections within the E2E transport NSs. The new model will not only isolate the management of subnets but also will allow the description of the interconnection instances (implemented as gateways in the data plane) to the upper layers for management operations such as subnet sharing or substituting. The model introduces the mechanism of anchor node and anchor termination point in which all open-ended links within a subnet are linked to an anchor node at a anchor termination point (Figure 2.8).

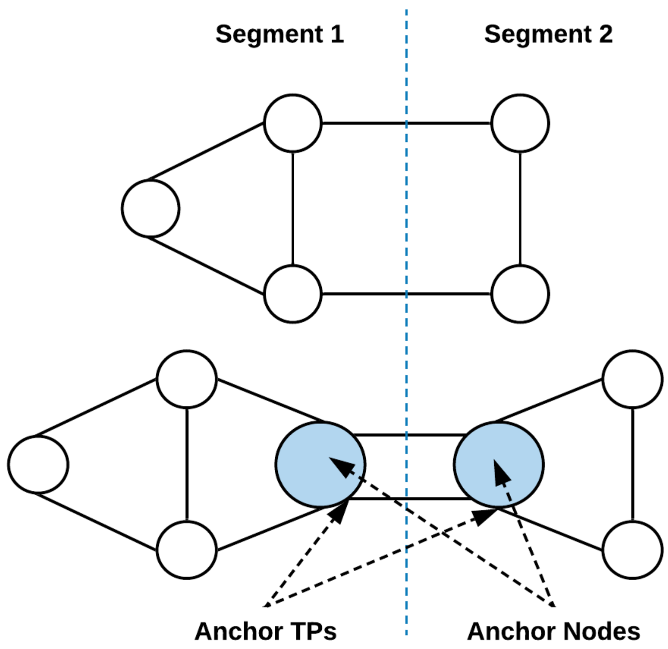


Figure 2.8: *The NS Subnet Stitching Mechanism*

A similar mechanism is used in TE networks to identify adjacent domains while merging multiple provider TE topologies into the client native one by the help of the *inter-connect plug id* attribute. However, this mechanism is specific to providers' domains interconnection and does not serve the objectives of subnet interconnection instances.

2.4.2.7 The TE Tunnel Data Model [51]

Categorized as a network configuration model, it describes the PNC's TE tunnel data store as it is seen and influenced by the MDSC. The later computes a VNS against the abstract view provided by the PNC then request the instantiaion of the computed TE links as TE tunnels in the PNC's domain.

2.4.2.8 The Performance Telemetry Data Model (PTISA) [52]

The transport NS consumer may want to subscribe to the monitoring of a set of Key Performance Indicators (KPIs) of its interest at the NS level (VN 1 type). For this aim the PTISA model can be used. Performance Telemetry data in the PTISA is divided into VN-level and tunnel-level data. Hence, The VN and the tunnel models are separately augmented with KPI telemetry data.

2.4.2.9 The ACTN Common Interfaces Information Model (ACTN C2IM) [53]

Finally, the ACTN C2IM models the CMI and MPI by providing respectively their required VN and TE tunnel primitives.

2.5 Our Vision on the E2E 5G Network Slicing Model

In this section, we look at the aforementioned studies described in section 2.4.2 as foundations of the network slicing technology in the transport domains. Therefore, we provide our vision on the network slicing architecture in the TN domain by analysing and explaining, in a technology-independent manner, how the different models and techniques work together within the ACTN framework in order to provide TNS services. Then, our vision on the TNS customer service model is provided. Finally, a unified E2E 5G network slicing architecture is proposed based on the different outputs of the SDOs.

2.5.1 Enable Network Slicing Within the ACTN Framework

Figure 2.9 depicts our vision. The customer in our case is the 3GPP system orchestrator (the operator). Therefore the CNC hosts the aforementionned 3GPP NSMF. By analogy to the RAN and the CN, the CNC sends the TNS related requirement over the CMI. At the latter, a service customer model encompasses the technology-agnostic TNS profile and the 3GPP RAN and CN access points. At the time of writing this article, neither a TNS profile nor a clear NS customer service model are defined within the IETF. Examples of customer service models include L2SM [54] and L3SM [55] which allow respectively the presentation of L2VPN [56] and L3VPN [57] services to a customer by an operator.

In alignment with the service model architecture provided in the context of SDN in [43], we adopt the functional split of the MDSC between a service orchestrator (SO)

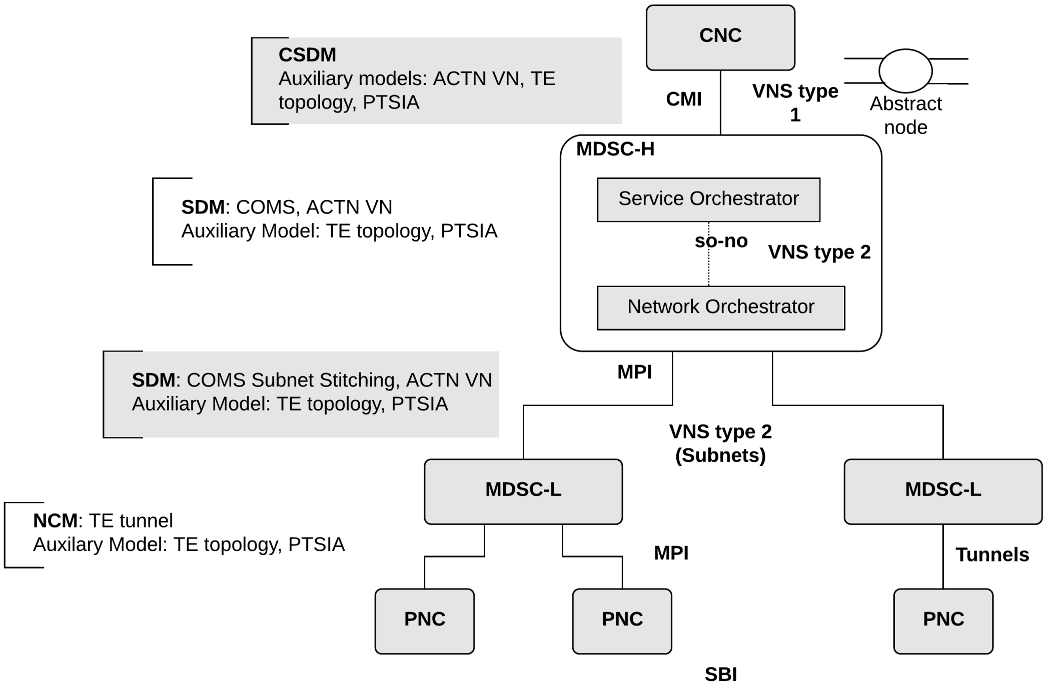


Figure 2.9: *Enabling network slicing within the ACTN architecture*

and a network orchestrator (NO). In this implementation option, the SO and the NO realize respectively the service-related and the network-related functions. As a result, a new interface (SO-NO) is exposed within the ACTN architecture controlled by a new data model type called the service delivery model.

Customer service models and service delivery models are sub-types of the service model. In general, the latter describes the service in a technology-agnostic way so that the customer has no knowledge of how the service is engineered in the underlying technologies. However, the level of service abstraction at the SO-NO interface is lower in comparison to the CMI. Particularly, a service delivery model is network centric and may be aware of many features such as topology, technology and operator policy [43].

The service orchestrator communicates the computed TNS (based on its abstract view) to the network orchestrator as a VN over the SO-NO interface. Recursively, the network orchestrator segments the virtual network service received on its northbound interface then requests the instantiation of the resulting virtual services to the concerned PNCs over the MPI. This is done using network configurations models. The PNC configures and manages the resources under its control domain. It establishes the tunnels in the network infrastructure as part of the virtual service segment received at the MPI.

It is important to note that the transport slice service received by the MDSC at the CMI is progressively mapped and translated at each layer until its realization as connectivity tunnels in the underlying infrastructure. The multi-layer/multi-domain service mapping function addressed here is an essential brick of the network and service automation architecture [58]. Indeed, the later is to be bound to the ACTN framework for the sake of network slicing fulfillment.

The data models needed to support the ACTN architecture in order to enable network slicing can be as follows:

i. The yang data model for TE topologies [44] to support the ACTN abstraction function.

ii. The yang data model for VN operation to request the instantiation of the TNS and its constituent subnets [45].

iii. The technology independent information model for common operation and management of network slicing (COMS) [46].

iv. The NS Subnet stitching data model [50].

v. The yang data model for TE tunnels and interfaces [51].

vi. The yang data models for VN and TE-tunnel performance telemetry and scaling intent autonomics (PTSIA) [52].

vii. The ACTN common interfaces information model (ACTN C2IM) [53].

In the context of network slicing, the TE topology model is used at MPIs, the SO-NO interface as well as the CMI. We consider the TE topology as an auxiliary model for the service delivery and the customer service models.

According to both 3GPP and GSMA, the NSP orders a NS service using a set of Service Level Specifications (SLSs), also called *NS related requirements*, without specifying any service graph. It comes to the NOP to map those SLSs, using the NSMF, to RAN/CN/TN slice profiles. The NSSMF of each domain is then the responsible for translating those profiles to topology and network service models to be instantiated in the underlying infrastructure.

Following the same modeling, a transport slice profile should be defined in terms of a set of parameters, metrics and thresholds, along with the customer (operator) access points of the transport slice service. Therefore we believe that there is no need for a *service graph* in the customer service model but a type 1 VN. This is referred to as a type 1 virtual network service (VNS) in the ACTN framework.

A type 1 VN is the easiest way for customers to express E2E connectivity so they do not deal with service graphs or any technical detail. Consequently, in addition to the slice profile, the customer service model has to refer to an abstract node with the different VNAPs along with a connectivity matrix. The use of the VN model in the ACTN is addressed only at the CMI. However, we propose to use it also at the SO-NO interface allowing the service orchestrator to request the instantiation of a VN to the network orchestrator. In addition, the VN model needs to be used in an eventual implementation of layered MDSCs at the MPI interface between a MDSC-H and MDSC-Ls in order to segment the VNS to NS subnets. Clearly, the VNS used outside the CMI is of type 2 (more concrete than Type 1). We consider the VN model as a service delivery model. Indeed, it relies on the TE topology model to refer to a VN abstract topology. This is achieved in the yang module using the attributes: *vn-topology-id* and *abstract-node*.

Since we categorize the COMS data model as service delivery model, it becomes necessary to explain the specificities of this model in comparison with the VN and the TE topology ones. We show in figure 2.10 how all of them can be integrated together in the whole network slicing process. The COMS model comes with a new data store which provides a complete and unified view of NSs. Without COMS, a NS is stored in the TE topology data store and there is no way to recognize it as slice only through the VN data store. However, the VN model provides only a mechanism for VN instantiation and does not include NS provisioning and management operations. Furthermore, managing a NS as TE topology is a heavy task because of its numerous contained TE information necessary to compute a NS but useless for its maintenance and life-cycle management.

Another strong point about COMS is its integration with the ETSI NFV MANO (management and orchestration) framework so that it describes compute, storage and network resources relayed on by the VNFs purchased with a NS [47] [48].

For all these reasons, we propose to use the TE topology model only for the provider's domain abstractions and the computation of a NS based on the received customer service

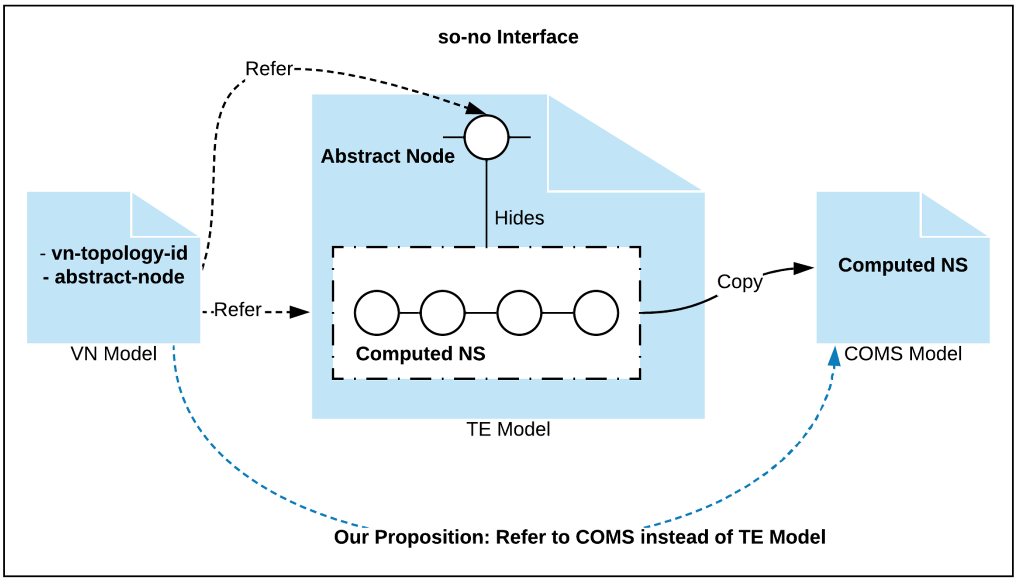


Figure 2.10: *Integration of VN, TE and COMS models*

model. The TE computed network can be then stored in the COMS data store with the aforementioned augmenting entities as a NS. This how the COMS service delivery model is derived from the customer service one. In the network slicing case, instead of pointing out to a TE topology, the VN model refers to the COMS topology network at the SO-NO interface for the provisioning and management of a NS.

In this context, an ongoing project is recently lunched under the opensource Open Network Operating System (ONOS) controller in partnership with several telecom operators (e.g., Huawei and SK Telecom) [59]. It aims at the implementation of the ACTN framework. Since the project is not intended in a first place for TNS implementation it is actually involving only the TE topology and the TE tunnel yang data models. Nevertheless, the framework if extended by a NS engine and an implementation of the COMS and the VN models, can serve as a reference platform for the TNS experimentation.

Afterwards, the VNS (type 2) received by each low level MDSC is mapped to the tunnel network configuration model [60] so that one or multiple tunnels are sent to the concerned PNCs in order to be deployed in the underlying domains.

Finally, the MDSC-H requests the subscription for subnets level (VN 2 type) telemetry data from multiple MDSC-Ls. Recursively, the MDSC-L maps the received VN KPI telemetry subscription request to multiple tunnel KPI telemetry subscription requests sent to one or many concerned PNCs. Besides, tunnel telemetry data is derived from low-level data collected via performance monitoring counters in network elements and pulled up to the PNC using the device configuration model. Moreover, the PTISA model provides a mechanism for the CNC, the MDSC-H and the MDSC-L to respectively configure automatic scale-in and scale-out of a NS (VN type 1), a NS subnet (VN type 2) and a TE tunnel.

2.5.2 TNS Customer Service Model

Section 2.4 rises an important question about the customer service model that should be used to order NSs: Does it carry a service graph as stated by ETSI NGP or rather a NS template as affirmed by GSMA? This is the major difference between SDOs' vision. 3GPP is aligned with the GSMA vision and is already considering many of the GST attributes in the definition of its service/slice profile. Table 2.6 lists those commonly used attributes and their reference to GSMA GST fields.

Table 2.6: *3GPP and GSMA commun NS characteristics*

3GPP field	***Value***	***Equivalent GSMA attribute***
Connection Density	Integer (UE/sq km)	Terminal Density
E2E Latency	integer (ms)	slice quality of service parameters
UE Mobility Level	Enumeration (stationary, nomadic, restricted mobility, fully mobility)	Supported device velocity
Resources Sharing Level	Enumeration (shared, non shared)	Simultaneous use of the network slice
Reliability	Float	Reliability
Communication Service Availability	Float	Availability
UE Speed	integer (Km/h)	Supported device velocity

Nevertheless, the GSMA GST template extends the 3GPP service/slice profile with standardized slice characteristics. As for IETF, it is not yet clear if the customer service model supports a service graph or a transport slice profile following the same modelling of 3GPP within the RAN and CN. In fact, authors in [38] proposed an initial version of the NS customer service model at the CMI interface also called Transport Slice

Connectivity Interface (TSCI) information model. This model supports the vision of the ETSI NGP and represents a NS service as a set of transport networks (subnets) including nodes and their interconnection links. The nodes may represent either the endpoints of the transport slice connections (RAN and CN network functions) or the endpoints of the transport service (border routers). This in progress work claims to be flexible in that matter.

Another transport slice data model is proposed in [61] in order to support network slicing as a service between a customer and a provider. The later augments the data model of network topologies. Truly, after the request of a NS service and that later is implemented in the underlying network infrastructure, the TNS provider is required to expose the NSI management capability to the customer for operational state retrieval and any augmentation in the NSI resources. The transport slice data model provide the means to serve that goal. The model operates at the CMI, and may be considered as service customer model to be activated after the creation of the NSI for management capability exposure purposes.

Furthermore, an analysis was performed on the GST's 35 attributes in order to decide on their impact on the transport network since some of them target either the RAN or the CN [62]. The end goal is to define the functionality required on the northbound interface of the MDSC (CMI). As result, GSMA GST attributes were categorized as directly impactive, indirectly impactive and non-impactive attributes. This work is definitely an important step toward the definition of a NS customer model.

We believe that the TNS provisioning request at the CMI is an abstract intent-based request following the same modeling of 3GPP in the provisioning of RAN and CN sub-slices. Thus, the NS customer service model would be based on GST SLAs, most likely the ones with direct impact on the transport network, along with customer endpoints as explained in section 2.4.2. Further, this model may be augmented with the TE service mapping model [63] as proposed in [64]. The goal is to record the mapping between a requested NS service and its instantiation in the underlying infrastructure in terms of either a VN, a TE topology or tunnels. This will allow to view the NS service instantiations outside the MDSC from either an operator for diagnostics purposes or to give an idea to customers on how their services are instantiated in the transport domains.

Another point, we propose to use the TE topology model only for the provider's domain abstractions and the computation of a NS based on the received customer service model. The TE computed network can be then stored in the COMS data store with the augmenting entities resumed in section 2.4.2.5 as a NS. This how the COMS service delivery model is derived from the customer service one. In the network slicing case, instead of pointing out to a TE topology, the VN model refers to the COMS topology network at the SO-NO interface for the provisioning and management of a NS.

Finally, the discussion on the TNS provision model is an open topic within IETF and requires more feedback and reviews from standardization bodies and vertical customers. Currently, three NS provision models are envisaged: The SaaS-like, the PaaS-like and the IaaS-like models [65]. The former is the model adopted by GSMA and 3GPP to respectively request and provide a NS service. In the PaaS-like model the tenant's request carries a representation of the NS as a set of nodes, their interconnections along with connectivity configurations. This is aligned with the ETSI NGP vision. Tenants in the IaaS-like model have direct control over the underlying infrastructure so they order NSs with concrete resource configurations. The latter model is not aligned with the common network slicing vision. Yet, no use case was cited in the related work in order to prove its usefulness. Table 2.7 summarizes the aforementioned similarities and differences between the different SDOs'visions. Regarding the service order management, the REST open API specification provided by the TM Forum can be used to place an order at each of the CMI as well as the interface between the E2E NSC and the NSP (operator) [66]. Also, authors in [67] have provided a specification of Connectivity Provisioning Negotiation Protocol (CPNP) which is meant for exchanging and negotiating dynamically connectivity provisioning parameters between a customer and a provider (or only between providers). By comparison to the TM Forum REST API, this protocol introduces the concept of "Offer" that allows the service provider to accommodate received orders beyond monolithic yes/no answers. CPNP can also be used at the aforementioned interfaces to order a NS service.

Table 2.7: *Summary on the similarities and the differences between SDOs'visions*

SDOs modelling aspects	**Provision models [65]**			**NS Service request**			**Network domain**				**Main contribution**
	SaaS-like	PaaS-like	IaaS-like	Service graph	Service/ Slice profile	VN Type (1,2)	RAN	Core	Transport	General	
ETSI NGP		✓		✓						✓	Generalized NS architecture and its associated workflows.
3GPP	✓				✓		✓	✓			Information/data model for NS provisioning and management on RAN and Core network.
GSMA	✓				✓		✓	✓	✓		GST/NEST
IETF		✓				✓			✓		The ACTN architecture and several data models(cited in section 2.4.2).
Proposed vision		✓			✓	✓ VN type 1 + TNS profile			✓		Enable Network Slicing within the ACTN architecture by integrating together multiple IETF data models.

2.5.3 Towards a Unified Network Slicing Model

We have observed that the issued standards are generally completing each other and may potentially be unified as presented in our work published in [16]. In this context, ETSI ZSM workgroup is building an E2E NS federated solution [68] by making use of all the aforementioned outputs from ETSI, 3GPP, IETF and GSMA and stitching them to the ZSM architecture [69]. For instance, 3GPP has introduced interesting concepts for the RAN and the CN that could be mapped and applied to the transport network such as NSMF/NSSMF, service/slice profile, NSI/NSSI. All those terminologies could be mapped to the TN so that we define the transport-NSSMF, transport slice profile, and transport NSSI as illustrated in figure 2.11.

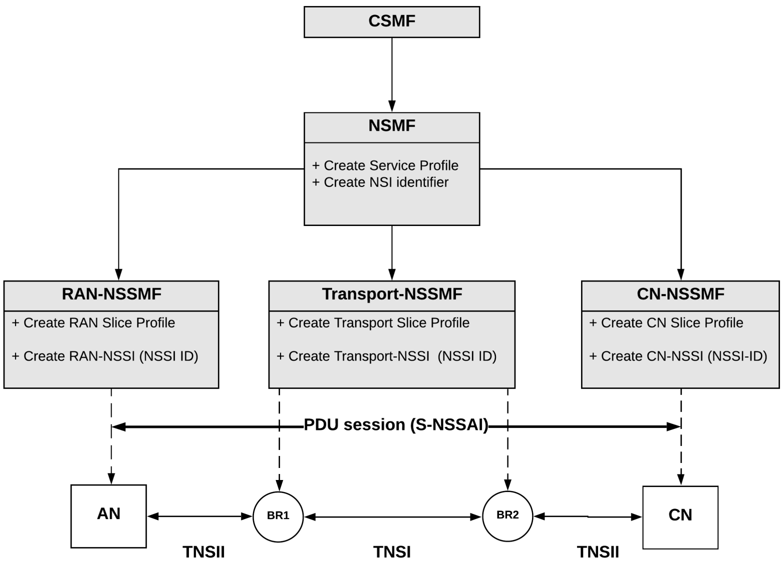


Figure 2.11: *Mapping of 3GPP NS functions to the Transport Network*

Besides, the NS functions provided by ETSI NGP in table 2.1 are all included in the 3GPP NSMF/NSSMF [2]. As described in [70], the mapping of 3GPP network slicing to the transport network happens in each of the management plane, the control plane and the data plane. In the former, a binding between the NSI with the Transport-NSSI is maintained in each of the Transport-NSSMF and the NSMF. In the control plane, the S-NSSAI is used during the UE registration and the PDU session setup to identify a NS from the UE perspective thus the selection of a NSI. However, there is no explicit mapping of the S-NSSAI to the TN. The S-NSSAI is normally mapped to the NSI in the management plane and this will help in the selection of the Transport-NSSI. Regarding the data plane, the Transport Network Slice Interworking Identifier (TNSII) and the

Transport Network Slice Identifier (TNSI) are proposed [70] to respectively identify transport NSs in the operator wide network and locally in the transport domain. The mapping between the TNSII and the TNSI is maintained within the transport network border nodes (br1 and br2). The mapping between the S-NSSAI or the NSI/Transport-NSSI and the TNSII is maintained in the RAN/CN nodes so that packets of an uplink flow going to the transport network from the RAN/CN is encapsulated with TNSII. Finally, a NS life-cycle management REST API [71] is developed by the 5G riders on the storm catalyst [72] and provided as a proposal to the TM Forum (API-1284). The latter binds together the data model work from 3GPP SA5, a JSON version of the GST and TM Forum API guideline.

Since orchestration is a key enabler of Network Slicing, we map the major SDOs'aforementioned contributions to the End-to-End Management and Orchestration reference architecture (MANO) [10], proposed in the context of the 5G Exchange project, as depicted in Figure 2.12.

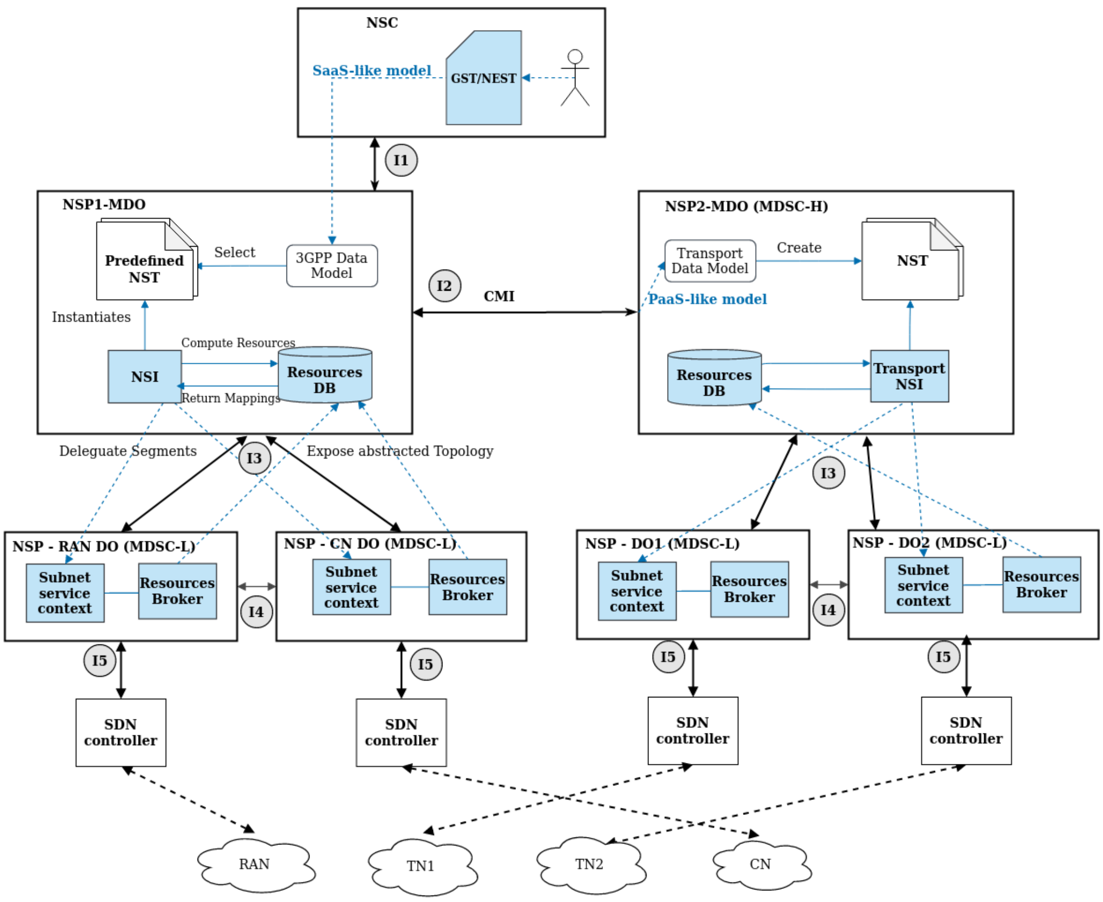


Figure 2.12: *Mapping of the ETSI NGP NS information model objects, the 3GPP NS data model, the GST/NEST and the ACTN Framework on the End-to-End Management and Orchestration reference architecture (MANO)*

We propose by this figure an architecture that binds together all the standardization works on network slicing including NS informations models. The end goal of this first proposal is to clarify our vision on the E2E network slicing process from the service order request to the NS deployement and management.

2.6 Challenges and Future Research Directions

In this section, we discuss some open issues related to the GSMA GST template as well as we highlight the security and isolation problem brought by the network slicing technology.

2.6.1 Enhancing the GST

The 5G riders on the storm catalyst project promoted by the TM Forum [72] have a different vision on how the GST should be structured based on its ongoing demonstrations as well as the Huawei's operational experience with NS management. The project focuses on an operational NS use case that support emergency services provided to first responders during extreme weather events such as flooding or storms. As a result a reviewed GST [73] is proposed that includes the following contributions:

i. Preparing the flat GST to automation and programmatic use by transforming it to a JSON template validated by a JSON schema.

ii. Making the GST customer facing by separating the Customer Facing Service (CFS) attributes from Resource Facing Service (RFS) ones so that most customers (non-technicals) deal only with what/where/when questions.

iii. Many of the GST attributes are ignored in the reviewed template on the pretext that their values can be derived from other populated parameters. After comparison of the two templates, we notice that the following GST attributes are missing: Downlink throughput per network slice, Energy efficiency, Isolation level, Location based message delivery, Maximum supported packet size, NB-IoT Support, Uplink throughput per UE.

iv. The catalyst is also enhancing the GST with new attributes from NGMN and 3GPP such as the use case type and activity factor.

v. All the GST attributes are reorganized into profiles.

2.6.2 NS Security and Isolation

Nevertheless, the network slicing concept brings new security issues. In particular, providing and demonstrating the adequate NS isolation level is one of the biggest challenges faced by the 5G operators [74]. In fact, 3GPP architecture demonstrates a requirement for authenticating users of network slice resources. There is however a need for separate per-slice security policies, e.g. having different authentication requirements between IoT and broadband. Inter-operation between NSs is a major issue related to VNFs isolation at L2 or L3 levels. Moreover, the IETF COMS model already presented in details in section 2.4.2.5 includes a set of isolation attributes covering the following areas: traffic, bandwidth, processing, storage. However, COMS is not a customer-facing model.

According to GSMA, a NS instance may be fully or partly, logically and/or physically, isolated from another network slice instance. These properties are included in the GSMA GST under the "isolation level" attribute. Finally, the NSP has to assure the NSC with evidence that confirms the NS isolation existence. This should be achieved by the measurements of the isolation properties in the operator's infrastructure as well as NS orchestration functions such as the report aggregation and service assurance functions proposed by ETSI NGP. The IETF PTSIA model may also be used to verify NS isolation since it allows a NSC (operator) to subscribe to transport NSs and NS subnets telemetry data. This is an interesting and complicated research direction that should be coupled with the frameworks designs in the future.

2.7 Conclusion

In this chapter, current NS modellings works in RAN, CN and transport network domains were bound together in order to bring forth the implicitly expressed 5G E2E network slicing model from the SDOs' technical specifications.

We started by introducing the network slicing general vision and how ordering a NS can be done with the use of the service graph. An in-depth exploration of the E2E network slicing information models within the RAN, CN and TN specific domains is then provided. We have analyzed and resumed most important models, their attributes and functions that are required for maintaining NSs at all its possible life-cycle phases. We tried to simplify the comprehension of all the complicated operations and the relationship between the elements of different layers or different domains via tables, figures and diagrams accompanied by brief descriptions. Our vision on how all these models and frameworks can be merged, completed and augmented together for an efficient E2E network slicing management has been proposed that is the real benefit of this work.

CHAPTER 3

Network Slice Subnet Management Framework

Chapter content

3.1 Introduction

According to the standardization activities mainly driven by 3GPP, ETSI and IETF, an end-to-end NS instance may span across multiple network domains. In that case, the end-to-end NS is composed of multiples NS instances running each on a specified domain. The single domain NS instances called "NS Subnets" cannot run separately from each other and are interconnected using NS gateways [24] in order to form the end-to-end NS.

In addition, the aforementioned SDOs'activities [75], [2], [76] on Network Slicing state that the NS Subnet Management Function (NSSMF) is performed separately from the NS Management Function (NSMF). However, prior works do not include any demonstration for the NSSMF as well as current experimentations of NSs do not highlight this separation neither clarify the scope of the two functions on their architectures [77]. Furthermore, understading the NSSMF as well as the implementation of its internal components is necessary to build a resource-facing NS solution.

Therefore, the aim of this chapter is to realize the NSSMF by providing a novel framework founded on a set of requirements derived from an exhaustive analysis of up-to-date standards from 3GPP, ETSI and IETF (Section 3.2). The framework is designed to manage NS Subnets as well as their supported services'life cycles taking into account its integration with NFV-MANO [78] (Section 3.3).

This chapter is organized as follows. Section 3.2 list all the requirements that need to be fulfilled by the NESSMA framework. Section 3.3 explains the NESSMA overall architecture then provides its system design. Section 3.4 concludes the chapter.

3.2 Requirements on the NESSMA Framework

The NESSMA framework is based on the E2E MANO reference architecture as well as the ETSI umbrella NFVO implementation option both explained in section 1.1.4. The NSSMF is realized by the resources and the single-domain orchestration layers of the E2E MANO architecture. Its scope is highlighted in green color in figure 3.1.

In this section, we discuss all the requirements related to the NSSMF that need to be satisfied by our framework. They are derived from our vision of the E2E network slicing model presented in the previous chapter.

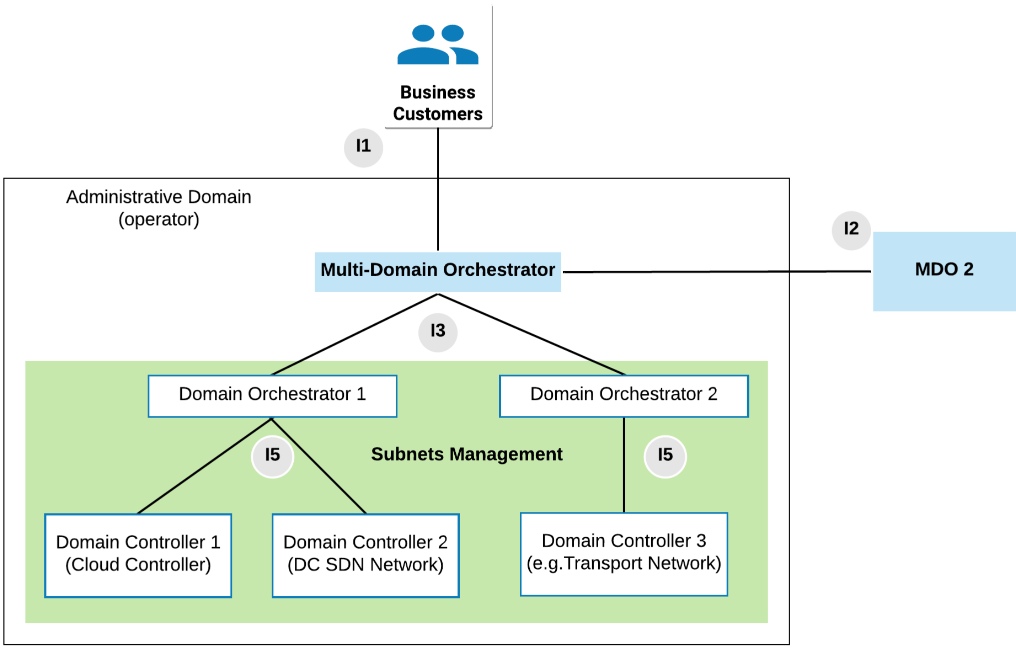


Figure 3.1: *The position of the NSSMF in the E2E MANO architecture*

1. **`Subnet discovery function`** [22]: The Subnet discovery function (SDF) is responsible for abstracting a domain's physical topology as well as exposing it to the NS provider (NSP). Therefore, the latter constitute the complete view of the underlying physical infrastructure by aggregating topology informations (node, links, constraints and their belonging domain) from different domains. The SDF is necessary for the end-to-end NS instantiation and it is achieved by a resource broker.

2. **`Subnet provisioning`** [22]: A NS Subnet is the part of the end-to-end NS delegated to a NS agent (NSA) for deployment as well as life cycle management. Once a NSA receives the Subnet associated to a NS, it first computes the mappings to concrete resources from the resource broker with the help of a child Path Computation Element (PCE) then stores them on the Subnet service context run-time object for deployment. After a Subnet is successfully deployed on a certain domain, the resource broker inside the NSA informs the NSP by sending an update (using the SDF) to the resource database about the newly allocated resources.

3. **`NS Subnet Stitching`** [22], [50]: Concatenating Subnets with each other is

inevitable in order to form the end-to-end slice. NS Subnets are not meant to be deployed separately, they are all part of the same logical network (NS). Hence, the ETSI as well as other SDOs have defined the term Network Slice Gateway to refer to the logical node that mainly performs Subnets interconnection.

4. **`Subnet Encapsulation`** [22]: In order to achieve service isolation between slices (thus Subnets), an encapsulation (e.g., MPLS, VXLAN, etc) need to be set for each tenant by the NSA. Two possible encapsulation modes are envisaged: End-to-end encapsulation and Segmented encapsulation. Traffic flow may also be forwarded without any encapsulation across all the Subnets with only egress and ingress port mappings with consideration for slice constraints.

5. **`Subnet monitoring`** [22], [46]: The goal of monitoring is to support accounting and charging operations as well as provide service assurance by continuously reporting any violation of an already set threshold parameter within the Subnet, such as the packet loss, the availability (of NS), the link throughput and jitter associated to a NS.

6. **`Subnet configuration and statistics`** [46]: ports in each node (e.g., switch, router) participating in the Subnet are configured with packet rate, packet loss probability. Statistics include the number of received and sent packets in a period of time for each port participating in the Subnet. Also, it covers the current throughput, latency and jitter of the links contributing to the Subnet.

7. **`Subnet augmentation`** [22]: Network Slicing includes exposing the capability to the tenant in order to increase or decrease its allocated resources at runtime. Typically, NS augmentation is performed in two steps. First, an augment request is sent from the tenant to the NSP. Then, the latter identifies the Subnets concerned with the resources augmentation and order the responsible NSAs to apply the new changes.

8. **`Subnet sharing`** [76]: A NS creation event may trigger either the instantiation of a new Subnet or the use of an existing one with an eventual modification. Accordingly, Subnets support NSs and are shared between them. Consequently, the life cycle of a NS instance (NSI) and its contained NS Subnet instances (NSSIs) are not necessarily the same. For example, when a NSI is deleted, one or many of

its constituents NSSIs still exist because they are belonging to other NSIs.

9. **`Managing NS Subnet services`** [79]: The 3GPP NS data model indicates clearly that a NS Subnet support multiple services. More specifically, the *Network Service* entity describes the interconnection between the Subnet defined network functions (VNFs: Virtualized Network Functions and PNFs: Physical Network Functions) following one or many service topologies called *Forwarding Graphs* (FGs). Also, a NS Subnet is independently managed from its supported services'life cycles. We highlight that the NSSMF includes the management of the NS Subnet as well as its supported services [75].

10. **`Integration with NFV-MANO`**: While the NFV-MANO framework deals with orchestration of NFV Network services, Network Slicing requires a high-level service orchestration capable of managing NSs considered as customer-facing services. Therefore, a new NS abstraction layer need to be introduced in the NFVI allowing the integration of NSs'life cycles management in the NFV-MANO. Two possible opportunities of this integration are envisageable. The first consists of extending the NFVO's service orchestrator functional block with a NS management engine. The second consists of performing NS orchestration in an external entity and interfaces it with NFV-MANO. In this last context, efforts of the IETF NFV research group (NFVRG) [80] to integrate NSs in the NFV-MANO architecture result in a revisited ETSI NFV framework (Figure 3.2).

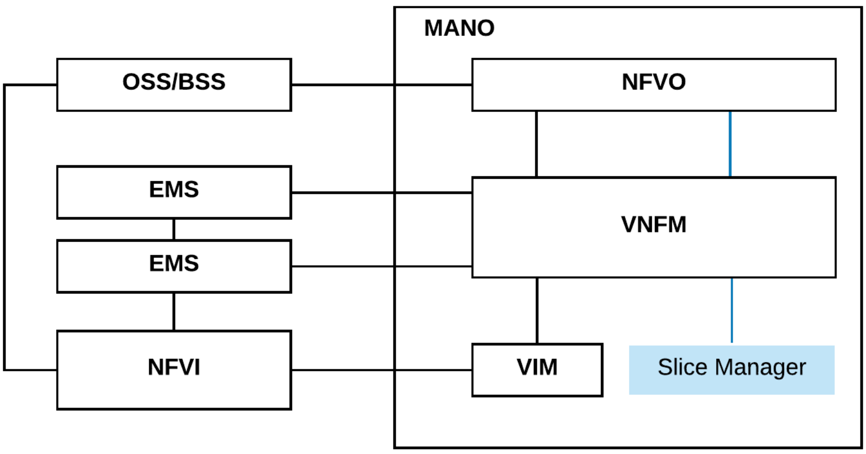


Figure 3.2: *Revisited NFV-MANO Framework [81]*

3.3 The NESSMA Framework

3.3.1 NESSMA Architecture

The NESSMA Architecture (Figure 3.3) follows the NFVRG vision but further implement the NS Subnet Manager/NS Manager (NSSM/NSM) as a specialized VIM. In addition, it is designed with the respect of both the ETSI NGP NS [22] and the End-to-End Management and Orchestration (Figure 1.4) reference architectures. We map the NSSM and the NSM to respectively the NSA and NSP of the ETSI NS architecture.

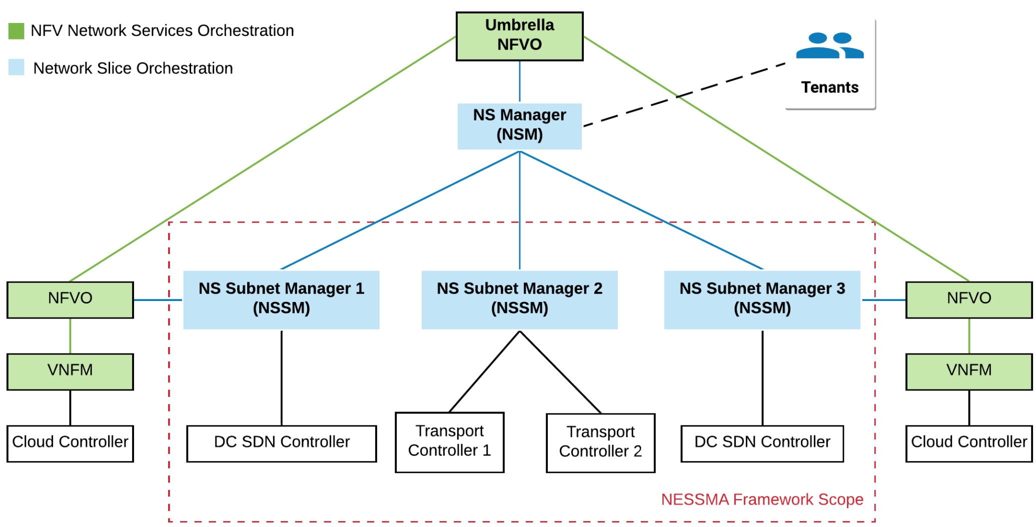


Figure 3.3: *The NESSMA architecture*

The architecture also adopts the Umbrella NFVO approach [11] since the latter offers NFV service composition which is aligned with the Network Slicing use case. For example, a tenant service would consist of connecting multiple VNFs running in different Data centers (DCs) connected across different transport domains. Following the ETSI NS architecture, the NSM constitutes the end-to-end NS view exploiting the abstractions of the underlying physical infrastructures exposed by different NSSMs. Next, the end-to-end NFV Network Service is separated from the NS, Segmented by the Umbrella NFVO and distributed to the underlying NFVOs for deployment (instantiation of VNFs/creation of the network service in each DC). The remaining NS is also Segmented to Subnets and delegated to the transport NSSM for deployment in each transport domain. Finally, each NFVO requests to the NSSM the installation of the required programmable flow rules through the SDN controller.

In the NESSMA Architecture, the NFVO cannot communicate directly with the

SDN controller to install flow rules on the forwarding devices as it is the case in the NFV-MANO, instead all the network related operations are requested to the NSSM. The latter will contextualize the NS Subnets in the forwarding devices by the mean of a flow table partitioning virtualization. Moreover, a NSSM can act as a multi-SDN orchestrator (NSSM 2) or what is called the WAN Infrastructure manager (WIM) enabling a unified NS orchestration of different transport domains. However, traditional WIMs [82] allows only for the provisioning of end-to-end services (connections) and do not deal with the NS orchestration and management. Lastly, we mention that the NSM is out of scope of the proposed NESSMA framework described in the Following section.

3.3.2 NESSMA's System Design

According to the requirements identified in section 3.2, we define the necessary system components of the NESSMA framework within each of the NSSM (Figure 3.4) and the SDN controller. We highlight that one of the main feature (resulting from the 9th requirement) of our framework is the joint management (provisioning, augmentation, monitoring, etc) of NS Subnets as well as their supported services'life cycles. The NESSMA system components are as follow:

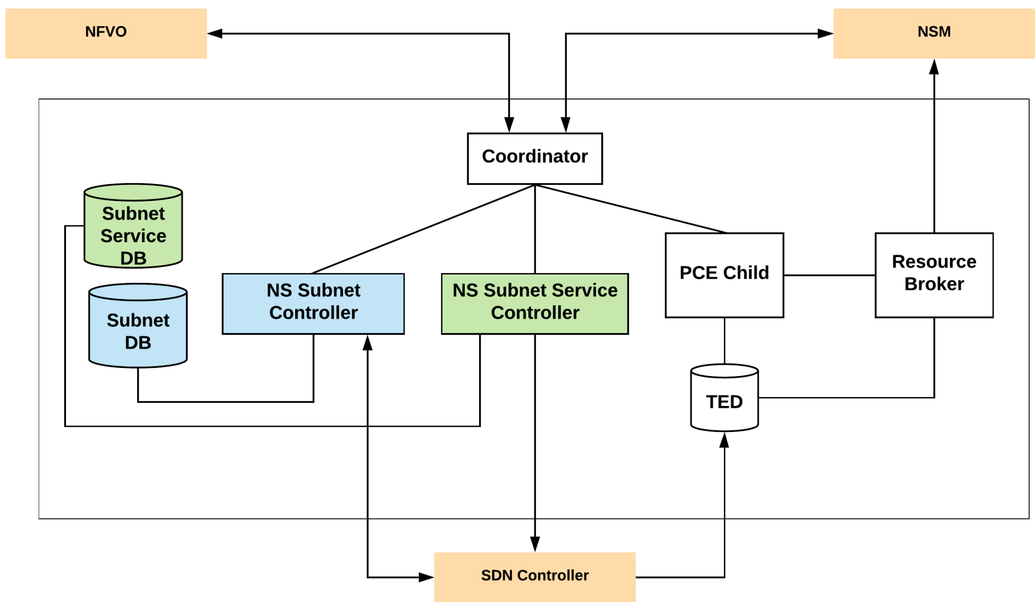


Figure 3.4: *NSSM's System components*

- **`Subnet controller`**: responsible for allocating the Subnet logical topology with the required capabilities (e.g. latency, bandwidth). It communicates with the

PCE through the coordinator in order to compute the received Subnets on the Traffic Engineering Database (TED). In addition to resources provisioning it is also responsible for any Subnet related augmentation, statistics reports and monitoring. To perform these functions it communicates with the underlying SDN controller via Rest APIs and also websockets to handle monitoring events.

- **`Subnet Service controller`**: responsible for the realization of the FGs (described in the Network Service entity) received by either the NFVO (e.g. VNF interconnection) or the NSM (e.g. transport connectivity) ordering the installation of the appropriate flow rules in the appropriate Subnet. Similarly to the Subnet controller it also performs service's life cycle management.

- **`Path subsystem`**: provides path computation services to any form of networks. It may be a generic path finder algorithm based on some optimization techniques or a path Computation Element (PCE) [83] where the shortest path is one particular case. In the NESSMA framework it coordinates with a parent path element, e.g., PCE, located at the NSM using a northbound protocol. It also operates on the TED to compute the mappings of the delegated Subnets (from the NSM) to the physical infrastructure.

- **`Resource Broker`**: defined in ETSI NS architecture, it allows the exposing of an abstracted network domain's topology to the NSM based on the TED and with the help of a path computation algorithm.

- **`Subnet database`**: it stores all the running Subnets as virtual networks along with their mapping to the physical infrastructure. This virtualization is performed by the SDN controller once it receives the Subnet provisioning request. The controller acknowledge the creation of the Subnet to the NSSM by sending back the elements'identifiers (IDs) of the virtual network.

- **`Subnet Service database`**: stores informations about the running services received either by the NFVO or the NSM, thus enabling to track their corresponding installed flow rules for an eventual augmentation or even a termination.

- **`Traffic Engineering database (TED)`**: stores topology informations along with their constraints (e.g.bandwidth, latency) and status of resources (e.g.up/down, residual/unreserved) [84]. It is populated and continuously updated by the SDN controller.

- **Coordinator**: coordinates the aforementioned system components allowing the NSM and the NFVO to jointly provision NS Subnets on the physical infrastructure.

NS Subnet Managment requires the introduction of a NS Subnet layer in the SDN controller on top of the resource abstraction layer, extending the controller framework with 3 blocks:

1. **APIs**: The NESMA framework defines a set of Restconf APIs (provisioning, augmentation and statistics) that allows the NSSM to manage the life cycle of NS Subnets as well as the services running inside them. More specifically, Subnet and service APIs are accessed respectively by the Subnet and the Service controllers using the Restconf protocol. In addition, Subnet and service event handlers sends out periodically notifications to respectively the Subnet and Service controllers using websockets.

2. **Virtualization**: Allows to contextualize NS Subnets on the physical infrastructure enabling their management by the upper-layer (NSSM) in an abstract and a flexible manner. In our framework, it is achieved by a *Subnets-views data store and a Flow-table partitioner*.

   First, a Subnet provisioning request from the NSSM carries within the concrete forwarding elements participating in the Subnet (including the NS gateways) along with their endpoints and the necessary configurations. This results in the creation of a tenant context (e.g.virtual switch) in each participating element by allocating flow tables and setting queues. Next, the newly created virtual network along with its mapping to the physical network is maintained in the Subnets-views data store.

3. **Authentication**: A tenant authentication application (e.g.Extensible Authentication over LAN) is necessary to control the access of end-users to NSs. If the authentication is successful, the application installs a flow rule in the edge physical switch context that sets a tenant ID in the tenant end users traffic. An other flow rule that match that ID will redirect the traffic to a virtual switch context.

Finally, for the sake of implementation all the NSSM's functional blocks can be developed from scratch except for the PCE and the resource broker. One of the available open source implementations can be chosen. The resources broker function is performed by an ALTO server [85]. In addition, the Open Baton framework [86] and the ONOS [87]/Ryu [88] SDN controllers are the most suitable projects for the NESSMA framework's design. While ONOS/Ryu provides all the needed abstractions and features to develop the controller's blocks mentioned above, Open Baton is fully aligned with the NFV-MANO and provides the necessary Software Development Kits (SDKs) to interface the NSSM as a VIM with the NFVO.

3.4 Conclusion

In this chapter we focused on the design and the realization of the NS subnet management function responsible for the management of the NS subnets. The proposed framework architecture as well as its internal system components are designed based on a list of requirements derived from an analysis of the main NS standardization activities detailed in the previous chapter. Three main features are distinguished for the proposed framework: 1) Alignment with standardization activities, 2) the joint management of NS Subnets and their supported services, 3) integration with NFV-MANO.

CHAPTER 4

Inter-Slice Bandwidth Resource Sharing

Chapter content

4.1 Introduction

In this chapter we proposes the inter-slice bandwidth resources sharing, a novel mechanism for bandwidth resources optimization in slice-based infrastructures. The approach is based on the IETF COMS technology-independent information model [46] which introduces four sharing levels to further optimize the network cost (in terms of bandwidth) of slice-based infrastructures, and thus increasing the operator revenue. We provide as well a proof of concept that implements and evaluates the mechanism in a SDN-based network. In practice, a network operator provisions a slice to be deployed according to the user request with a specified E2E bandwidth capacity. Note that a user may have several flow types (e.g., a video streaming and on-line gaming) circulating inside his reserved slice. Furthermore, many users may share the same slice. However, in particular NS conditions such as increase of users or peak times a slice may become congested so that the total bandwidth of the flows soliciting the slice at a specific point in time is greater than the slice bandwidth capacity. Surely, the capacity of a network is not usually set to the sum of its served flows since the latter are not simultaneously present within the network.

Therefore, our solution InterS, allows a congested slice to temporarily acquire all or a part of the free bandwidth resources of its neighbor slices and make use of them to serve its own traffic. To the best of our knowledge, InterS is the first solution demonstrating the inter-slice resources sharing concept based on the IETF COMS NS data model. Besides operator profitability, we believe that this concept may help to reduce NS reconfigurations that may occur periodically in a slice-based infrastructure. The goal of NS reconfiguration is to augment NS resources or to simply guarantee better network utilization rate. This is referred to as the NS augmentation function in standardization bodies such as ETSI NGP [89].

The augment operation is triggered by an operator/service provider to either increase or decrease the actual NS resources at runtime. In case of a NS scale up update, if the new required resources are not supported on the slice's actual route, the operations includes 1)the creation of a new path with the required resources, 2)rerouting the flows as well as migrating the VNFs to a new path and 3)releasing the resources in the older route. This two-phase strategy is also referred to as make-before-break [90]. By reducing the frequency of such costly operations, InterS is providing more scalability, efficiency and profitability.

Indeed, isolation is one of the key expectations of network Slicing. However, the strength and level of isolation may vary depending on the contracted requirements between service providers and operators as well as NS usage scenarios [74]. Thus, in InterS, shared slices does not conserve the same level of isolation as exclusive ones since they are frequently allowing other slices' traffic on their free resources.

This chapter is organized as follows. Section 4.2 explains our proposed inter-slice bandwidth resource sharing approach. Section 4.3 presents the details of our proof of concept, called InterS, that includes the design of the solution as well as the experiments performed on the latter to evaluate the proposed concept. Section 4.3.3 and 4.3.4 evaluate respectively two different implementations of InterS with the shared non preemptive resource reservation level. Section 4.3.5 compares the two previous implementations. Section 4.3.6 highlights the limitation of the shared non preemptive mode of InterS. Finally, Section 4.3.7 and 4.3.8 cope with the latter limitation by implementing and evaluating the shared preemptive resource reservation level. Section 4.4 concludes the chapter.

4.2 The Inter-Slice Bandwidth Resource Sharing Approach

This section introduces the inter-slice bandwidth resource sharing approach and provides a concrete example of its operation.

4.2.1 Overview

The inter-slice bandwidth resources sharing concept demonstrated in this chapter is based on the *resource reservation level* attribute under the *slice-level attributes* entity of COMS (Figure 4.1). Four levels of resources reservation are defined for a NS:

- **`None:`** no specific resource reservation, for this level, a NS instance will share and compete for resources with other NS instances.
- **`Shared non preemptive:`** the free reserved resources could be used by other NS instances but they can not be retrieved if other NSs are still using them.
- **`Shared preemptive:`** the free reserved resources could be used by other slice instances, and will be retrieved if the NS needs them.
- **`Exclusive:`** the reserved resources can not be used by other NS instances, even if these resources are free.

COMS is a technology-independant information model for transport network slicing. It is mainly meant for the management of NSs as well as NS subnets. As discussed in section 2.5.1, COMS is an essential brick of the ACTN architecture. The COMS NS data

```
slice-level-attributes

   +--rw netslice:slice-level-attributes
      +--rw netslice:service-time-start? yang:date-and-time
      +--rw netslice:service-time-end?   yang:date-and-time
      +--rw netslice:lifecycle-status?   lifecycle-status-type
      +--rw netslice:access-control
      |  +--rw netslice:match?      string
      |  +--rw netslice:action?     string
      |  +--rw netslice:priority?   string
      |  +--rw netslice:counter?    int64
      +--rw netslice:reliability-level?  reliability-level-type
      +--rw netslice:resource-reservation-level?
                          resource-reservation-level-type
      +--rw netslice:availability?                  int64
      +--rw netslice:availability-threshold?        string
```

Figure 4.1: *COMS'Slice-level attributes entity*

stores are located in each of the transport multi-domain orchestrator and its domain orchestrators respectively called MDSC-H and MDSC-L by the IETF standardization body (Figure 4.2). The 3GPP system orchestrator responsible for the orchestration and the management of the RAN and the CORE domains starts the request of the provisioning of a transport NS using a transport slice profile. The later is processed by the MDSC-H and translated to a NS definition based on the COMS data model and saved in the COMS data store. The definition of the subnets associated to the NS is then provided to the MDSC-Ls during the NS subnet delegation process and are stored in the COMS NS subnet data stores.

Our solution InterS particularly invokes the inter-slice bandwidth resource sharing mechanism in order to optimally deal with the NS congestion issue in slice-based infrastructures. Therefore, InterS allows a congested slice to temporary acquire all or a part of the free bandwidth resources of its non exclusive neighbor slices and make use of them to serve its own traffic. We define neighbor slices as a set of slices that share at least one physcial link of the physical infrastructure, ideally a slice path.

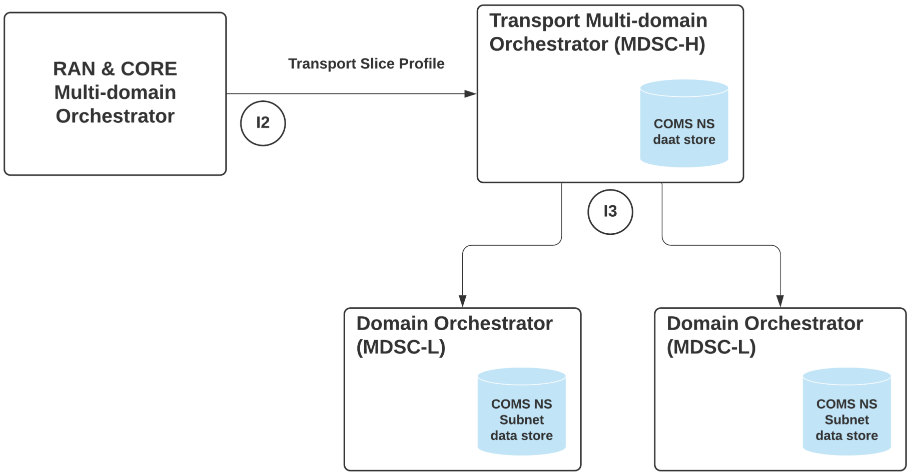


Figure 4.2: *COMS data store*

4.2.2 An Example of InterS Operation

The goal of this section is to explain the inter-slice bandwidth resource sharing mechanism by realizing the following experiment. Particularly, the shared non preemptive mode is used for this matter. The scenario of the experiment was chosen to reflect the different behaviors of InterS in a simple and clear manner. We note that the realized experiment serves only as an example of the proposed approach. The experiments of our proof of concept that we relyed on for the evaluation of the concept are detailed in section 4.3.2.

The experiment was performed on a network topology that consists of three Open-VSwitch in a linear topology (S1-S2-S3) and 16 hosts, eight source hosts (H1 to H8) connected to Switch 1 and and eight destination hosts (H9 to H16) connected to Switch 3. The maximum capacity of the path S1-S2-S3 is set to 150 Mb/s.

InterS was configured to support four QoS flow types *A, B, C, D* respectively characterized by a guaranteed data rate of *10, 20, 30, 40 Mb/s.* Also, three NS instances (NSIs) were created in the network with the characteristics described in table 4.1. In addition eight flows of different types were generated from hosts H1 to H8 with the transmission data rates and the durations described in table 4.2.

Figure 4.3, shows the result of the experiment. In particular, graph a), shows the transmission rate of the 8 generated flows during the whole experiment. Graphs, b), c) and d) illustrate respectively the flows that were served by NSI 1, NSI 2 and NSI 3 resources as well as their instantaneous throughput at the receiving hosts. In the

Table 4.1: *NS Instances (NSIs)*

NSIs	**NS Id**	**Bandwidth (MB/s)**	**Path & Endpoints**	**Allowed flow types**	**Bandwidth reservation level**
NSI1	1	70	S1-S2-S3: H2-H10, H4-H12, H5-H13	A, B, C, D	Shared non pre-emptive
NSI2	2	50	S1-S2-S3: H3-H12, H6-H14	A, B, C, D	Shared non pre-emptive
NSI3	3	30	S1-S2-S3: H1-H9, H7-H15, H8-H16	A, B, C	Exclusive

Table 4.2: *Flows generated in the Network*

Flows	**Transmission data rate Mb/s**	**End-points**	**Type**	**NS**	**Starting time (s)**	**Duration (s)**
Flow1	40	H1-H9	D	NS3	0	4
Flow2	45	H2-H10	D	NS1	0	24
Flow3	10	H3-H11	A	NS2	5	19
Flow4	40	H4-H12	D	NS1	10	30
Flow5	30	H5-H13	C	NS1	15	25
Flow6	20	H6-H14	B	NS2	20	20
Flow7	30	H7-H15	C	NS3	25	15
Flow8	10	H8-H16	A	NS3	30	10

following the aforementioned graphs will be analyzed.

- At time t=0, Flow 1 of type D is sent from source host H1 to destination host H9 at the rate of 40 Mb/s for 4 seconds. Referring to table 4.1 (path and end-points), Flow 1 is belonging to NSI 3. Knowing that the latter does not support the flow type D, Flow 3 is dropped. This explains the absence of the flow's associated trace in all of the b), c) and d) graphs. At the same instant, the Flow 2 belonging to NSI 1 is introduced in the network with a transmission rate of 45 Mb/s for the duration of 24 seconds. At this moment, NSI 1's available bandwidth is 70 Mb/s, thus the flow is rate-limited to 40 Mb/s and served in NSI 1. In our QoS model, we do not allow a flow to send more than its guaranteed rate, all the excess traffic is dropped.
- At time t=5, the Flow 3 belonging to NSI 2 is sent to the network at a rate of 10

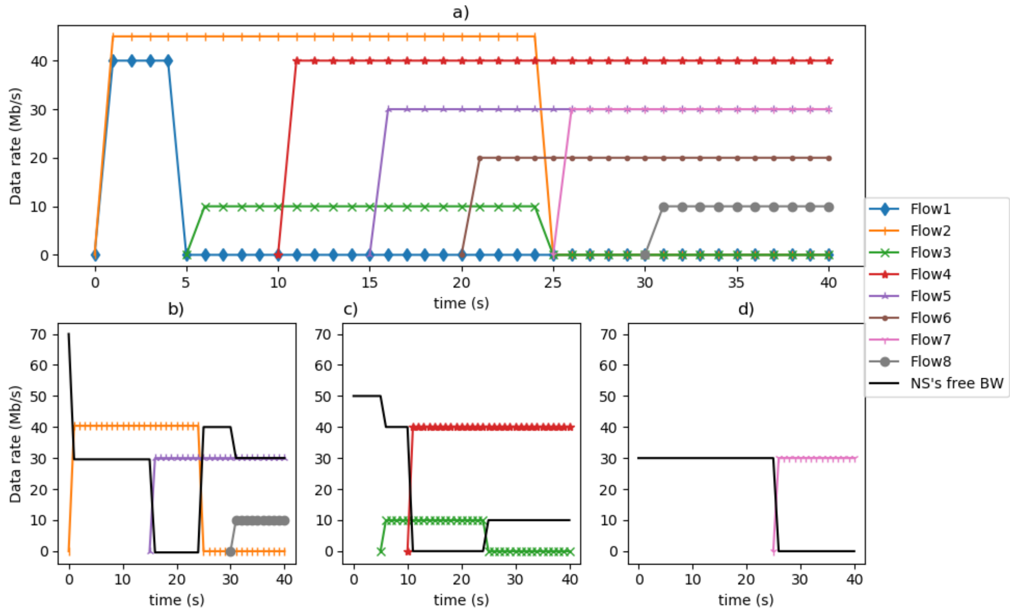


Figure 4.3: *a) Flows data rates at sender hosts. b) Instantaneous throughput of flows (served in NSI 1) at receiving hosts. c) Instantaneous throughput of flows (served in NSI 2) at receiving hosts. d) Instantaneous throughput of flows (served in NSI 3) at receiving hosts.*

Mb/s for a period of 19 seconds. Since NSI 2's full capacity is free at this instant, Flow 3 is well served.

- At time t=10, Flow 4, another flow of type D belonging to NSI 1 is arriving to the network. NSI 1's available bandwidth is only 30Mb/s at the moment, hence NSI 1 select NSI 2 resources (40Mb/s) to accommodate this traffic. After this operation, NSI 2's capacity is fully used.

- At time t = 15, Flow 5 is admitted in NSI 1 occupying all its free remaining bandwidth.

- At time t=20, Flow 6 of type B is arriving at NSI 2, however, NSI 2's resources are actually accommodating two flows (Flows 3 and 4) with a total throughput of 50Mb/s which is equal to NSI 2's bandwidth capacity. Because NSI 3 is "exclusive", the only remaining option to check is NSI 1. Likewise, the latter is actually serving two flows (Flows 2 and 5) which are consuming its whole bandwidth. As a result, Flow 6 is dropped. This explains the absence of the flow's associated trace in the graphs b), c) and d).

- At time t = 25, Flow 7 of type C is well admitted in NSI 3. At this moment also, transmission of Flows 2 and 3 has ended, thus releasing their used resources in respectively NSI 1 and NSI 2.

- At time t =30, Flow 8 of type A arrives at NSI 3. No more resources can actually be allocated in this slice, therefore NSI 3 will try to acquire resources from an neighbor NSI with the maximum free bandwidth resources. Actually, NSI 2 and NSI 1 have respectively an unused bandwidth of 10 and 30 MB/s, thus both of them are candiates for resources sharing. Finally, NSI 1 bandwidth resources (10Mb/s) are acquired by NSI 3 to serve Flow 8.

4.3 Proof of Concept

This section presents a Proof of Concept of the inter-slice bandwidth resource sharing approach called InterS. It describes the design of our implementation as well as the experiments realized to evaluate the concept.

4.3.1 InterS's General Design

This section explains the general design of InterS that covers only the **non preemptive** sharing mode. An extended version of InterS and an augmentation with the **preemptive** sharing mode are described in sections 4.3.4 and 4.3.7. Our InterS solution is particularly demonstrating the above reservation levels for NS bandwidth resources. InterS is not implementing the None level since it is not compatible with our NS QoS model.

In fact, according to the overall vision on Network Slicing provided by different standards associations, a NS is all about the allocation of isolated resources with certain guarantees [13]. Hence, prior instantiation, a NS has to be defined with a specific bandwidth capacity to be maintained during the whole NS lifecycle. Thus, A NS is capable of forwarding a specific network flow with its contracted rate guarantee from its source to its destination. Furthermore, we believe that the "None" level refers to best effort traffic that does not belong to any NS but uses the free resources of NSs in a specific path.

In our proof of concept, a NS is instantiated using a Rest API by defining the values of the following attributes: **bandwidth, path, NS Identifier (Id), a list of allowed flow types and the bandwidth reservation level**. Different flow types are also defined in the system with different rate guarentees. All the flows supported in the system are transmitted in the slice-based network at their guaranteed data rate with a reserve of NS bandwidth availability. Figure 4.4 shows the initial design of InterS under the Ryu SDN framework [88].

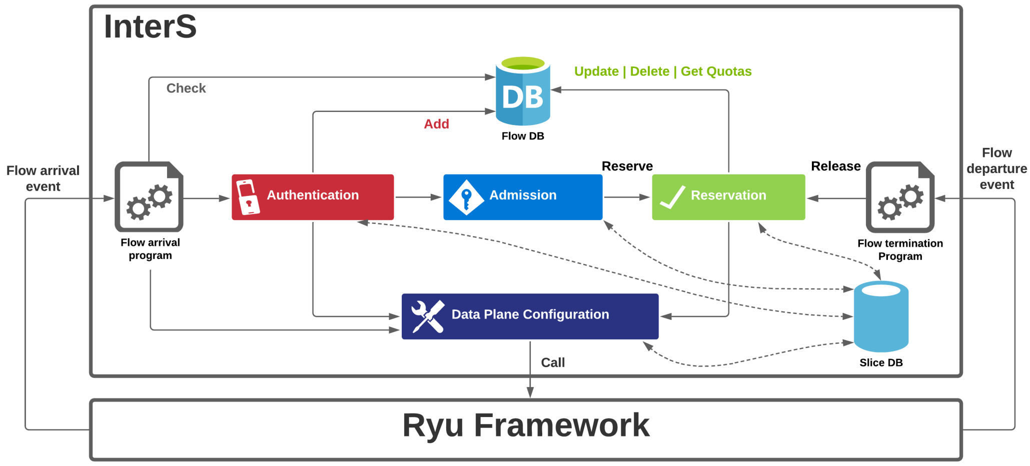


Figure 4.4: *InterS's General design*

InterS mainly listens to two events: **`Flow arrival and termination (departure) events`**.

- **`Flow arrival event:`** is captured by the *flow arrival program (FAP)* which triggers the authentication, admission, reservation and data plane (DP) configuration systems and coordinates their eventual interactions with the flow and slice databases (DB) in order to admit new flows in the network and reserve the required NS bandwidth resources.

- **`Flow departure event:`** is captured by the *flow termination program (FTP)* which trigger the reservation system and coordinates its interaction with the flow DB and the slice DB to respectively release NS bandwidth resources and delete flows from the flow DB.

Following the arrival of each new flow, the FAP get notified and launches the following InterS components:

1. **`Authentication:`** After the verification that the received notification is about the arrival of a new flow, the FAP launches the flow authentication process. The latter will verify that 1)the source host is belonging to an already established NS, 2) the flow type indicated in the flow is supported in the system and 3) the switch triggering the flow arrival notification is the Slice Access Point (SAP). Following

a successful authentication, the flow is saved in the flow DB and the FAP proceed the flow to the admission system.

2. `Admission:` Its main function is to check if the arriving flow could be actually served in its proper NS at its defined guaranteed rate. If this is not the case, the admission system will repeat the last step for all the other non exclusives neighbor NSs. Resources of the eligible neighbor NS with the maximum available bandwidth are then temporarily provided to the congested NS to serve the flow. This is the part where the inter-slice bandwidth resource sharing happens.

3. `Reservation:` for each slice registred in the slice DB, the reservation system keeps track of the actual resources being reserved by the NS's proprietary flows, the actual resources acquired by neighbor NSs and the free resources. It is also responsible for the release of NS resources following the termination of a flow as well as its removal from the flow DB. In the case of flow admission, the entry corresponding to the flow already added by the authentication process in the flow DB is updated by the reservation system to keep track of the eventual acquired resources used to accomodate the flow.

4. `DP configuration:` The admission of a specific flow is translated within the DP configuration system to a set of flow rules and meters configured in the data plane nodes all along the path of the selected NS. Following a failed admission or authentication, the flow is rejected and only the SAP is configured to the drop this flow. This is achieved by calling the Ryu Framework which in turn send the appropriate openflow messages to the nodes.

5. `Flow DB:` an inventory of all the flows existing in the network at a specific point of time, either being served in a specific NS or dropped. The interaction of the above mentioned components with the flow DB is governed by a set of operations described in table 4.3.

6. `Slice DB:` an inventory of all the NSs requested for instantiation to InterS using Rest API by the network operator. All the components of InterS are interacting with this database in read-mode only.

Table 4.3: *Flow DB operations*

Operation	Description
`Add`	used to add a flow in the database after a successful authentication.
`Update`	after a successful admission of a flow, its updated in the flow DB with a new attribute indicating the eventual usage of resources from neighbor NSs. This update allows also to differentiate between flows admitted in the network and the ones being processed or rejected.
`Check`	used by the FAP to check if the arriving flow is already existing in the Flow DB. This prevents the FAP from proceeding a currently admitted/dropped flow to the next process.
`Delete`	remove a flow from the database following its termination in the data plane.
`Get`	retrieve some specific informations about the flow such as the quotas of resources acquired from neighbor NSs which leaded to the admission of the flow. This operation is used in the implementation option detailed in section 4.3.4.

4.3.2 Experiments

The experiments were performed on a network topology that consists of three OpenVSwitch in a linear topology (S1-S2-S3) and 100 hosts, 50 source hosts (H1 to H50) connected to Switch 1 and and 50 destination hosts (H51 to H100) connected to Switch 3. The maximum capacity of the path S1-S2-S3 is set to 500 Mb/s (Figure 4.5).

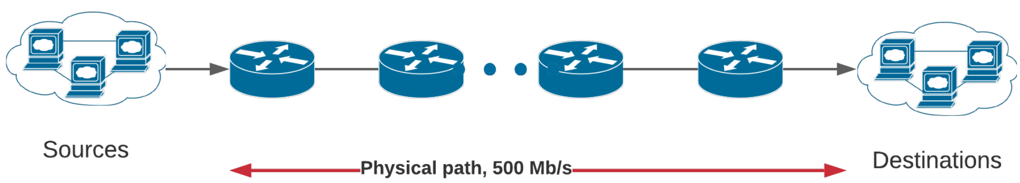


Figure 4.5: *Network topology*

InterS was configured to support four QoS flow types *A, B, C, D* respectively characterized by a guaranteed data rate of *10, 20, 30, 40 Mb/s.* In fact, the type of a specific flow is recognized in InterS by parsing the IP Differentiated Services Code Point (DSCP) field in the flow packets.

In order to run intensive experiments and evaluate the sharing approach of InterS, we have developed "FlowEvents", a traffic generator framework over Mininet [91] and OpenVSwitch [92] automating the generation of random flows towards the network at different arrival rates.

We have defined 3 scenarios with 5 network slices that share the same S1-S2-S3 path. Each slice has 100 Mb/s initial bandwidth capacity out of the 500 mb/s available in the physical path. In scenario 1 all the slices are instantiated as exclusives. In scenario 2, only 2 slices are exclusives and the 3 remaining ones are shared. Lastly, scenario 3 defines all the slices as shared. Table 4.4 specify the sharing mode that will be implemented and evaluated in each of the following sections.

Table 4.4: *the NS sharing mode evaluated by section*

Section	Scenario 1	Scenario 2	Scenario 3
4.3.3 (implementation 1)	All Exclusives	2 Exclusives, 3 Shared non preemptives	All Shared non preemptives
4.3.4 (implementation 2)	All Exclusives	2 Exclusives, 3 Shared non preemptives	All Shared non preemptives
4.3.7	All Exclusives	2 Exclusives, 3 Preemptives	All Shared preemptives

For a fair comparison between the 3 distinct scenarios, FlowEvents allows the creation of unique pseudo-random "Experience" each characterized by a different seed number. We consider 5 different flow arrival rates: 1, 1/2, 1/4, 1/8, 1/16, and thus different load in the network. Our experiments includes 15 Experiences (Table 4.5); for each scenario and for every arrival rate, we run 15 iterations (from seed 1 to seed 15).

Table 4.5: *The number of experiences executed per flow arrival rate in each scenario*

Scenarios	tested flow arrival rates	number of experiences per flow arrival rate
Scenario 1	1, 1/2, 1/4, 1/8, 1/16	15
Scenario 2	1, 1/2, 1/4, 1/8, 1/16	15
Scenario 3	1, 1/2, 1/4, 1/8, 1/16	15

In all the iterations of all the experiments, we generate a random number of flows uniformly in the interval [5, 50]. All the values of the parameters related to each flow are also generated randomly such as the source host, the destination, the slice identifier, the arrival time, the transmission duration, and the flow type. The Flow rate is a summation of the flow's guaranteed rate and a number picked randomly from the [1, 10] interval. The later simulates a flow excess rate that will be dropped later by InterS. For each experiment, we vary the flow duration between 40 and 180 seconds. We note that all the supported flow types previously defined are allowed in each of the 5 slices.

The execution of the experiment was simultaneously happening on a set of VMs allocated on the Grid 5000 public cloud (Figure 4.6). One VM is allocated per scenario and flow arrival rate to run the 15 experiences. By this separation we simply aim to reduce the time of our lenthy experiment. A pyhton script is used to automatically gather the results obtained on the different VMs and output them to files as datasets. All

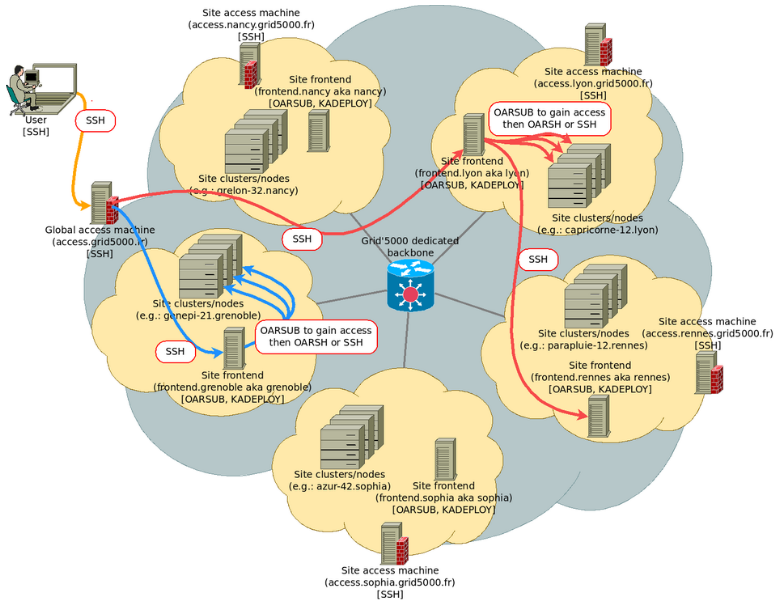


Figure 4.6: *Access to Grid5000 [93]*

the graphs presenting the results of the experiment are then generated from the produced datasets. The main softwares installed in each VM are: 1)The Ryu SDN Framework [88], 2)our solution InterS [15], 3)OpenVSwitch [92], 4)Mininet [91] and 5)"FlowEvents". All the VMs are requested from the "Nancy" site of Grid5000, particularly from both the "Grimoire" and "Grisou" clusters. The resources capacities of those VMs are described in table 4.6.

Table 4.6: *VMs resources capacities*

Cluster	CPU	Cores	Memory	Storage
Grimoire	2 x Intel Xeon E5-2630 v3	8 cores/CPU	128 GiB	600 GB HDD + 4 x 600 GB HDD + 200 GB SSD
Grisou	2 x Intel Xeon E5-2630 v3	8 cores/CPU	128 GiB	1 x 600 GB HDD + 1 x 600 GB HDD

The goal of the experiments is to evaluate the impact of the InterS approach combined to the flow arrival rate on both the total slice flow acceptance (TSFAR) and the bandwidth usage (TSBUR) rates. Thus, the performance metrics considered for this study is the the TSFAR and TSBUR. TSFAR and TSBUR are respectively calculated using the equations 4.1 and 4.2 below. The terms of both equations are described in table 4.7.

$$TSFAR = \frac{1}{458} \sum_{i=1}^{5} \sum_{j=1}^{15} n_{ij} \tag{4.1}$$

$$TSBUR = \frac{1}{500} \sum_{i=1}^{5} \left(\frac{100}{15} \sum_{j=1}^{15} \sum_{k=1}^{n_{ij}} \frac{RF_{ij}^{(k)} * DF_{ij}^{(k)}}{BW_i * D_j} \right) \tag{4.2}$$

Table 4.7: *Equations terms and meanings*

Terms	Meaning
$n_i j$	the number of admitted flows within slice i in experience j.
$RF_{ij}^{(k)}$	the achieved rate of the kth flow admitted within slice i in experience j.
$DF_{ij}^{(k)}$	the duration of the kth flow admitted within slice i in experience j.
BW_i	the bandwidth capacity of slice i.
D_j	the duration of experience j.
458	the total number of flow events in each experience.
500	the total bandwidth capacity of the 5 slices.

4.3.3 Implementation and Evaluation of the Shared Non Preemptive Resource Reservation Level

In this section, a first implementation of the shared non preemptive reservation level is evaluated. The implementation allows a congested NS to acquire free resources from at most one neighbor NS.

4.3.3.1 Algorithm

The algorithm of the admission system for the non preemptive sharing mode is generally described below (Algorithm 1).

The $freeBW()$ is a method of the Slice Class. When called by a NS instance it return the free resources of the instance at the moment of the call.

Algorithm 1: Non preemptive sharing mode

```
   Input  : The new arriving Flow, Flow_F
   Output : Accept or Reject Flow_F
1  i ← Get the NS identifiant to which Flow_F is belonging to
2  if Slice_i.freeBW() >= Flow_F's guarenteed data rate (FGR)
3  |   Pass
   |                                  // Flow_F Accepted, Go to line 10
4  else
5  |   n ← Get the neighbor Slice identifiant with the Maximum free bandwidth
   |     resources(If it exists)
6  |   if Slice_n.freeBW() >= FGR
   |   |                                                 // Flow_F Accepted
7  |   |   Scale up Slice_i with Slice_n resources for Flow_F (Scale down to normal
   |   |     after Flow_F termination)
8  |   else
9  |   |   Reject Flow_F
10 if Flow_F is not Rejected
11 |   Reserve Slice_i resources for Flow_F
12 |   Configure data plane for Flow_F
13 |   Forward Flow_F to Slice_i
```

4.3.3.2 Numercial Results

The obtained results, shown in Table 4.8, are clearly indicating the increase in TSFAR and TSBUR from scenario 1 to scenario 3 for all the flow arrival rate values. Table 4.9 precises the values of improvement in percentage from scenario 1 to sceanrio 3 as well as from scenario 1 to scenario 2 per each event rate. As detailed by table 4.9, scenario 3 has clearly achieved the best amelioration in TSFAR and TSBUR and this for each flow events rate.

Table 4.8: *the TSFAR (/458) and TSBUR (/500) obtained from each scenario for different flow arrival rates.*

	rate:1		rate:1/2		rate:1/4		rate:1/8		rate:1/16	
	TSBUR	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**
SC1	242.82	289	222.031	301	194.869	335	169.407	393	125.02	437
SC2	254.108	299	237.359	320	218.033	366	188.148	432	131.748	454
SC3	261.344	306	245.6	331	228.764	385	198.008	446	132.765	458

Table 4.9: *Improvement values (%) in TSFAR and TSBUR per event rate from SC1 to SC3 and SC1 to SC2*

	rate:1		rate:1/2		rate:1/4		rate:1/8		rate:1/16	
	TSBUR	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**
SC1−>SC3	7.63	5.88	10.62	9.97	**17.39**	**14.93**	16.88	13.49	6.2	4.81
SC1−>SC2	4.65	3.46	6.9	6.31	11.89	9.25	11.06	9.92	5.38	3.89

We are particularly interested by analyzing the improvement in TSBUR and TSFAR between the extreme scenarios 3 and 1 for each events rate. We observe from Figure 4.7 that the minimum augmentation in TSBUR and TSFAR is achieved by an events rate equal to 1/16. In this case TSBUR and TSFAR are respectively improved in scenario 3 comparing to scenario 1 by 6.2% and 4.8%. The maximum increase in TSBUR and TSFAR is achieved by an events rate equal to 1/4. In this case TSBUR and TSFAR are respectively improved by 17.4% and almost 15%. Likewise, we observe roughly this same pattern in the improvement from scenario 1 to scneario 2.

We conclude from the results illustrated in Figure 4.7 as well as table 4.9 that InterS performs the best at moderate flow arrival rates such as 1/4 and 1/8.

We think that this a logical conclusion since at high events rates the flows' inter-arrival times converge to 0. This means that they all arrive in approximately the same moment so that the total slices' bandwidth capacities are insufficient to serve them all. As a results a significant number of flows will be rejected in both the "Exclusives" and the "Shared" scenarios. Likewise, at a low events rate setup, flows' arrival are sufficiently separated in time so that a significant number of flows is accepted in both the "Exclusive" and the "Shared" scenarios.

We observe also, from table 4.8, that in the whole realized experiments, the flow acceptance rate has reached 100% only at scenario 3 with the 1/16 events rate setup.

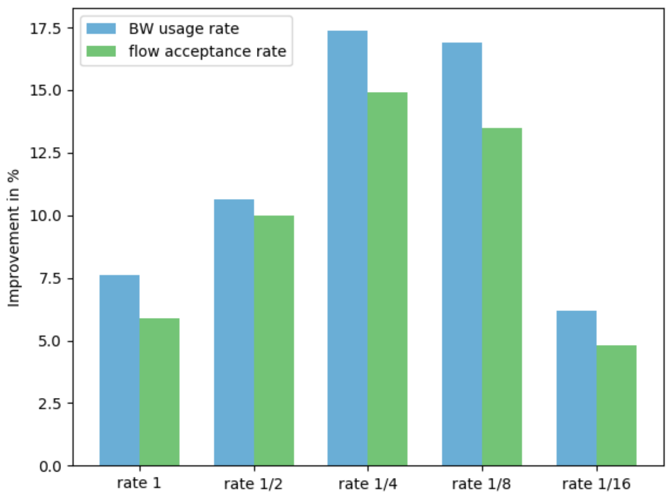


Figure 4.7: *Improvement percentage of TSBUR and TSFAR per events rate*

Figure 4.8 and 4.9 illustrates respectively the bandwidth usage rate of each slice averaged on the 15 experiences and the per slice flow acceptance rate during the 15 experiences. Both figures' results are obtained from the 1/4 events rate configuration.

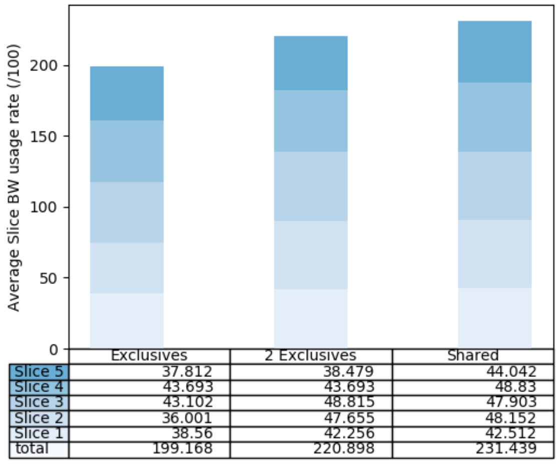


Figure 4.8: *Per slice bandwidth usage rate in a 1/4 events rate setup.*

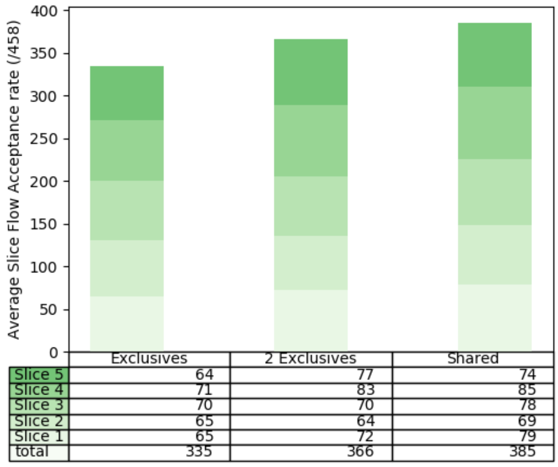


Figure 4.9: *per Slice flow acceptance rate during the 15 experiences in a 1/4 events rate setup.*

Table 4.8 shows that the TSFAR keeps increasing from the left scenario to the right one as well as from the highest events rate configuration to the lowest. However, while TSFAR increases TSBUR keeps decreasing from the highest events rate setup to the lowest. The later is due to the fact that TSBUR is calculated on the basis of the experience duration D_j. Figure 4.10 illustrates the impact of the events arrival rate configuration on the 15 experiences' durations in minutes. It is obvious that the experience duration gets bigger in low events rate configurations leading to lower TSBUR. Naturally, experience 5 with its 49 flow events has recorded a duration of roughly 20 minutes.

Lastly, we point out to an inaccuracy in the metering function of OpenVSwitch, that we call the meter error, encountered in all the experiences of the above experiments. To

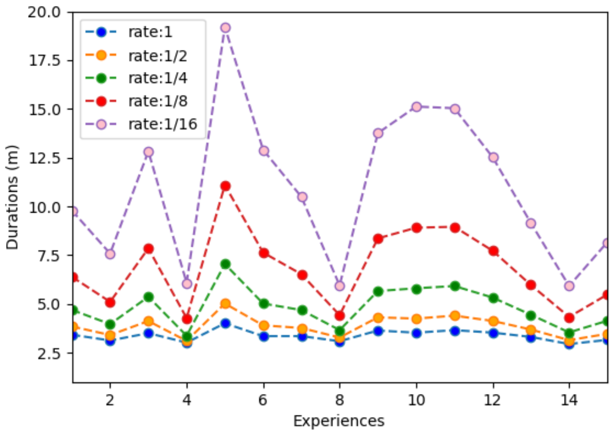


Figure 4.10: *Experience duration per events arrival rate*

highlight the error, two graphs of respectively the ideal and the achieved flow rates as reported at the destinations are traced in Figure 4.11. Moreover, The Flows belongs to the third scenario of the first experience configured with a 1/4 events rate.

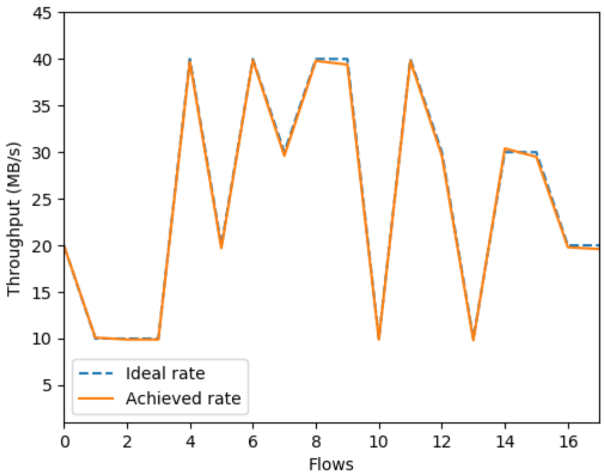


Figure 4.11: *The meter error observed in the OpenVSwitch metering Function*

For instance, we observe that Flow 9's received rate is 39,4 Mb/s while the flow has a guaranteed rate of 40Mb/s. This behavior is occurring in almost every flow of the experience. For the experience shown in the figure above, the average meter error is 0.2 Mb/s leading to an average loss rate of 0.8%. This rate is approximately the same for the other experiences since the observed behavior is the same. Although this value seems negligible for this experience, its impact may cause relevant loss in the achieved TSBUR as indicated by the ideal TSBUR values in table 4.10. TSBUR values in the current section (table 4.8) are calcultaed based on the achieved throupuut of flows.

However, in the following sections and for a matter of accuracy in comparisons between results of different implementations, we will ignore the meter error in the calculus of the TSBUR (ideal TSBUR).

Table 4.10: *the ideal TSBUR (/500) obtained from each scenario for different flow arrival rates.*

	rate:1	rate:1/2	rate:1/4	rate:1/8	rate:1/16
SC1	246.846	225.391	199.168	172.853	126.881
SC2	258.117	240.075	220.898	191.355	133.081
SC3	264.862	248.25	231.439	201.333	135.109

4.3.4 Aggregation of Resources From Multiple Neighbor Slices

Until now, InterS allowed any congested slice to acquire the free bandwidth resources of at most one neighbor slice. Thus a flow is rejected if there is no alternative slice capable of providing the required resources.

In this implementation, we observe that flows are rejected even though the total free bandwidth of all the neighbor slices are superior to the flows' guaranteed rates. This limitation has caused us to think about a new implementation option where a congested slice is able to aggregate free fragments of resources from different neighbor slices and acquire them to admit a flow that would be dropped on the previous implementation.

The new extension is used as a last resort when the required resources can not be wholly obtained from one neighbor slice only. This new behavior is integrated in InterS as an implementation option that can be activated or deactivated by the operator using Rest API. In the following of this section, we aim to discover the impact of the second implementation on both TSBUR and TSFAR and compare it to the first one in the exact same conditions.

4.3.4.1 Algorithm

The algorithm of the extended admission system for the non preemptive sharing mode is generally described below (Algorithm 2).

$\sum_k^{all} Slice_k.freeBW()$ is the total free resources of all the neighbor slices including $Slice_i$. As detailled in alogrithm 2 from line 11, a set of resources fragments are acquired from different neighbor slices to scale up a congested slice ($Slice_i$) and therefore allowing it to admit and forward $Flow_F$. These resources are immediately released and returned back to their proprietary slices after $Flow_F$ termination. For this end, the "Get Quotas" operation introduced in Figure 4.4 is used by the reservation system at the flow termination to obtain the resource contribution of each neighbor slice in the given congested slice.

Algorithm 2: Resource aggregation from multiple neighbor slices

```
   Input  : The new arriving Flow, Flow_F
   Output : Accept or Reject Flow_F
1  i ← Get the NS identifiant to which Flow_F is belonging to
2  if Slice_i.freeBW() >= Flow_F's guarenteed data rate (FGR)
3      Pass
                                      // Flow_F Accepted, Go to line 20
4  else
5      n ← Get the neighbor Slice identifiant with the Maximum free bandwidth
        resources(If it exists)
6      if Slice_n.freeBW() >= FGR
                                                          // Flow_F Accepted
7          Scale up Slice_i with Slice_n resources for Flow_F (Scale down to normal
            after Flow_F termination)
8      else if 2d implementation is enabled and FGR <= ∑_k^all Slice_k.freeBW()
                                                          // Flow_F Accepted
9          RR ← FGR − Slice_i.freeBW()
10         Sort neighbor Slices descendingly based on free resource criteria (From
            highest to lowest)
                   // Scale up Slice_i with the required resources RR from
            neighbor Slices
11         foreach Slice_k in neighbor Slices
12             if Slice_k.freeBW() >= RR
13                 Scale up Slice_i with RR from Slice_k
14                 break
15             else
16                 Scale up Slice_i with all Slice_k.freeBW()
17                 RR ← RR − Slice_k.freeBW()
18     else
19         Reject Flow_F
20 if Flow_F is not Rejected
21     Reserve Slice_i resources for Flow_F
22     Configure data plane for Flow_F
23     Forward Flow_F to Slice_i
```

4.3.4.2 Numerical Results

The obtained results, shown in Table 4.11, are clearly indicating the increase in TSFAR and TSBUR from scenario 1 to scenario 3 for all the flow arrival rate values. Table 4.12 precises the values of improvement in percentage from scenario 1 to sceanrio 3 as well as from scenario 1 to scenario 2 per each event rate. As detailed by table 4.12, sceanrio 3 has clearly achieved the best amelioration in TSFAR and TSBUR and this for each flow events rate.

Table 4.11: *the TSFAR (/458) and TSBUR (/500) obtained from each scenario for different flow arrival rates.*

	rate:1		**rate:1/2**		**rate:1/4**		**rate:1/8**		**rate:1/16**	
	TSBUR	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**
SC1	246.846	289	225.391	301	199.168	335	172.853	393	126.881	437
SC2	259.918	293	243.868	315	226.185	368	195.079	436	134.005	456
SC3	267.321	294	251.453	320	236.212	384	204.793	449	135.01	458

Table 4.12: *Improvement values (%) in TSFAR and TSBUR per event rate from SC1 to SC3 and SC1 to SC2*

	rate:1		**rate:1/2**		**rate:1/4**		**rate:1/8**		**rate:1/16**	
	TSBUR	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**
SC1–>SC3	8.29	1.73	11.56	6.31	**18.6**	**14.63**	18.48	14.25	6.41	4.81
SC1–>SC2	5.3	1.38	8.2	4.65	**13.56**	9.85	12.86	**10.94**	5.6	4.35

We are particularly interested by analyzing the improvement in TSBUR and TSFAR between the extreme scenarios 3 and 1 for each events rate. We observe from Figure 4.12 that the minimum augmentation in TSBUR and TSFAR is respectively achieved by the 1/16 and the 1 events rates. In this case TSBUR and TSFAR are respectively improved in scenario 3 comparing to scenario 1 by 6.41% and 1.73%. The maximum increase in TSBUR and TSFAR is achieved by an events rate equal to 1/4. In this case TSBUR and TSFAR are respectively improved by 18.6% and 14.63%.

Regarding the amelioration from scenario 1 to scenario 2, the maximum increase in TSBUR and TSFAR is respectively achieved by the 1/4 and 1/8 events rates.

We conclude from the results illustrated in Figure 4.12 as well as table 4.12 that this implementation of InterS performs the best at moderate events arrival rates such as 1/4 and 1/8.

Figure 4.13 and 4.14 details respectively the bandwidth usage rate of each slice averaged on the 15 experiences and the per slice flow acceptance rate during the 15 experiences. Both figures' results are obtained from the 1/4 events rate configuration.

We observe also, from table 4.11, that in the whole realized experiments, the flow acceptance rate has reached 100% only at scenario 3 with the 1/16 events rate setup.

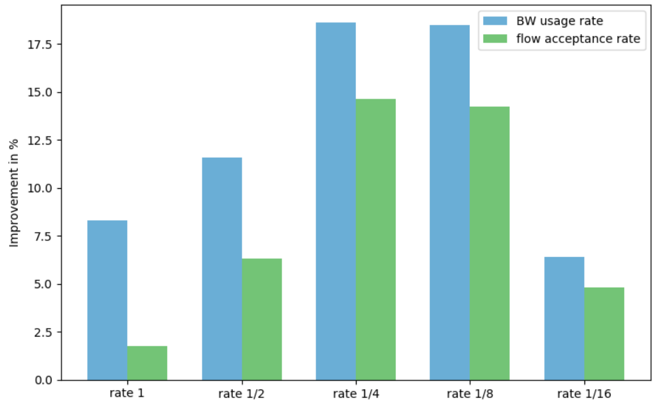
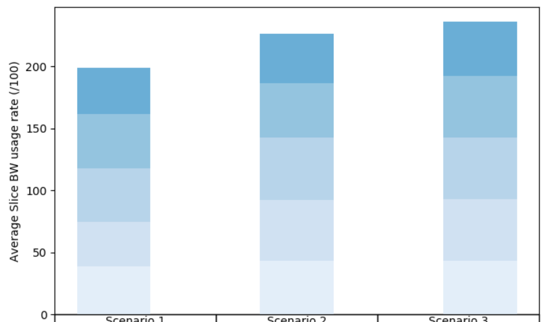
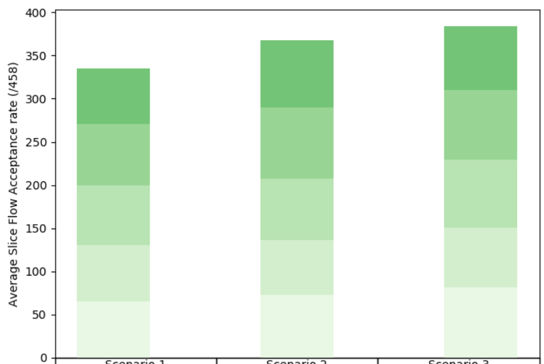


Figure 4.12: *Improvement percentage of TSBUR and TSFAR per events rate*



	Scenario 1	Scenario 2	Scenario 3
Slice 5	37.812	40.013	43.629
Slice 4	43.693	43.738	49.971
Slice 3	43.102	49.809	49.614
Slice 2	36.001	49.078	49.814
Slice 1	38.56	43.547	43.184
total	199.168	226.185	236.212

Figure 4.13: *Per slice bandwidth usage rate in a 1/4 events rate setup.*



	Scenario 1	Scenario 2	Scenario 3
Slice 5	64	78	74
Slice 4	71	83	81
Slice 3	70	71	78
Slice 2	65	63	70
Slice 1	65	73	81
total	335	368	384

Figure 4.14: *per Slice flow acceptance rate during the 15 experiences in a 1/4 events rate setup.*

4.3.5 Comparison of the Two Implementations of InterS

The aim of this section is compare the two implementations in terms of improvement in TSFAR and TSBUR achieved between extreme scenarios as well as from scenario 1 to scenario 2. We recall that table 4.9 and table 4.12 precise respectively the different values of amelioration measured in the afforementioned scenarios.

As concluded in sections 4.3.3 and 4.3.4, the two implementation of InterS (with the shared non preemptive mode) perform the best at moderate flow arrival rates such as 1/4 and 1/8.

Regarding the improvement in TSBUR, table 4.9 and table 4.12 are clearly indicating that implementation option 2 has slightly outperformed the first one in the amelioration between extreeme scenarios as well as between scenario 1 and scenario 2 and this for all the defined flow arrival rates.

Concerning the improvement in TSFAR between extreeme scenarios, Figure 4.15 illustrates the outperformance of implementation 1 over implementation 2 and this for all the tested flow arrival rates except 1/8. This is interpreted at flow events rate 1/8 by the admission of 3 additional flows in implementation 2.

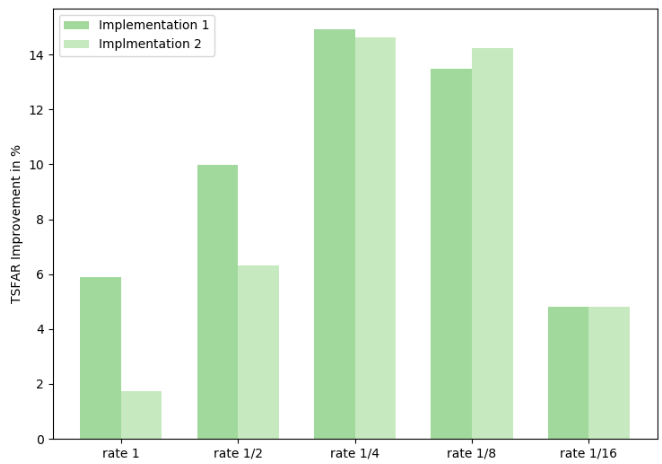


Figure 4.15: *Comparison of the TSFAR amelioration between implemenations from scenario 1 to scenario 3.*

We also observe that the difference between the two implementations shown in Figure 4.15 for flow events superior or equal to 1/4 is not as relevent as the difference measured in the 1 and 1/2 flow events rate. Precisely, implementation 2 has lead to the loss of respectively 12 and 11 flows in flow events rates 1 and 1/2 in comparison to implementation 1. From the latter, it is obvious to conclude that the second implementation of InterS should not be adopted in the 1 and 1/2 network flow rate conditions.

Regarding the TSFAR amelioration between sceanrio 1 and scenario 2, Figure 4.16 shows the outperformance of implementaiton 2 over implementation 1 for all the tested flow arrival rates that are superior or equat to 1/4. The latter is interpreted by the admission of a total of 8 additional flows in all of 1/4, 1/18 and 1/16 configurations. Similarly to the improvement measured between extreeme scenarios, implementation 1 is performing better than the second one at events rates 1 and 1/2. Precisely, implementation 2 has lead to the loss of respectively 6 and 5 flows in flow events rates 1 and 1/2 in comparison to implementation 1.

From the comparison of Figures 4.15 and 4.16, we conclude that implmentation 1 performs more better than implementation 2 at flow events rates 1 and 1/2. Thus, it is inappropriate to enable the second implementation at such network conditions.

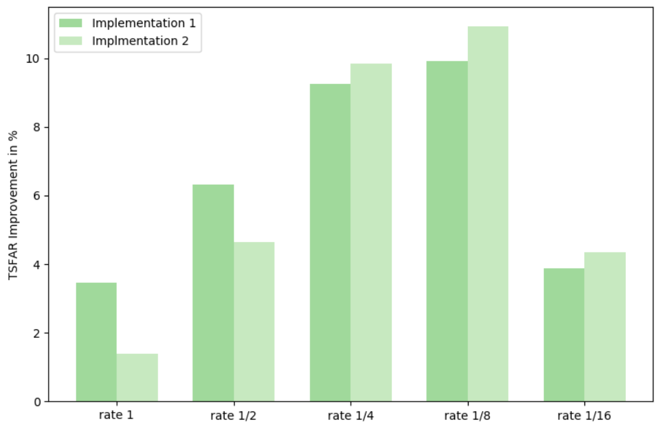


Figure 4.16: *Comparison of the TSFAR amelioration between implemenations from scenario 1 to scenario 2.*

4.3.6 Limitation of the Shared Non Preemptive Mode

The limitation of the shared non preemptive reservation level manifests in the innability of a slice to retrieve immediately its used resources by other slices when it needs them. This issue can be identified in the per slice flow acceptance rate figure either in the first implementation (Figure 4.9) or the second one (Figure 4.14). Scenario 1 is realizing the normal behaviour of Slices and thus providing a relevant information about the normal capacities of operational slices in terms of the flow acceptance rate.

We observe from the results presented in Figures 4.9 and 4.14 that the total number of accepted flows increase significantly for some slices and decrease for others when moving from scenario 1 to sceanrio 2 and 3. This is happening because the shared non preemptive reservation level does not allow a Slice to retrieve its resouces used by other slices until the later realases them. Therefore, this sharing mode is benefiting some slices on the expense of penalizing others. This limitation is even more important in high flow arrival rates as shown by table 4.13.

Table 4.13: *The total number of flows accepted by Slice 2 obtained in each scenario for flow arrival rates 1 and 1/4.*

	rate:1			rate:1/4		
	SC1	**SC2**	**SC3**	**SC1**	**SC2**	**SC3**
Implementation 1	60	50	56	65	64	69
Implementation 2	60	48	53	65	63	70

In fact, Slice 2 is significantly penalized in scenario 2 as well as 3 for "rate:1" but slightly penalized in "rate:1/4" and only in sceanrio 2.

4.3.7 Implementation and Evaluation of the Shared Preemptive Mode

The goal of this section is to extend InterS with the implementation of the shared preemptive resource reservation level defined in the COMS information model. Instantiating a NS as shared preemptive means that its free resources can be used by neighbor NS instances but are retrieved immediately if the NS needs them. The new sharing mode is then integrated to InterS and evaluated.

4.3.7.1 InterS'Design Extension

In order to realize the shared preemptive feature, InterS is extended with a new component that manages the retrieval of resources and the re-admission of flows as shown in Figure 4.17.

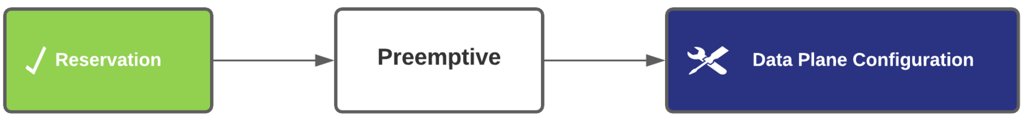


Figure 4.17: *Introduction of the preemptive component in InterS's design.*

In contrary to the shared non preemptive mode, a shared preemptive Slice has the priority and the authority over its resources during the whole operation time. Thus, it can retrieve its resources from neighbor slices immediately after a flow arriving event if the available resources are not sufficient to support the flow. Because of the instantaneous resource retieval characteristic of the shared preemptive slice, one or several flows are stopped during their transmission within the neighbor slices.

The preemptive component is responsible to stop the transmission of the well chosen flows and instruct the data plane configuration system to present these flows again to InterS as new arriving ones.

All the flows that are admitted in a specific slice and supported by resources acquired from a shared preemptive slice are susceptible to be stopped and re-admitted many times until their transmission is finished or they are admitted and supported by resources of their proper slices or rejeted.

Instead of dropping a flow, our implementation allows to present it continiously each time it is stopped by the preemptive process to the InterS's admission sytem as a new flow. This will surely increase the chance of the flow to be admitted definitely or complete its data transmission to the destination host. In our implementation when a flow is re-admitted, it resumes the transmission of data from the point it stopped at previously without any loss in data. Figure 4.18 clarifies the details of the process of flow re-admission.

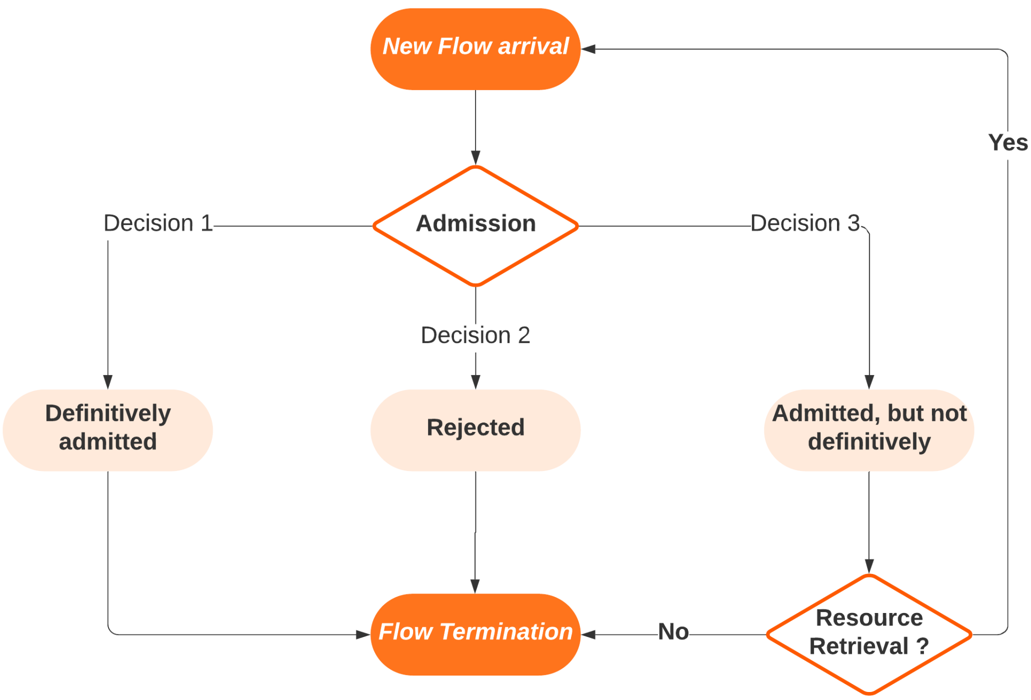


Figure 4.18: *Flowchart of the flow re-admission process.*

In the shared non preemptive implementation, we say that a flow is:

- **`"admitted definitively"`** if it is fully supported by the resources of the slice to which it is belonging to.

- **"admitted but not definitively"** if it is fully or partialy supported by resources of one or multiple neighbor slices. Thus, the resources can be retrieved any time during the flow transmission which causes the flow to be stopped and introduced again to the admission system.

- **rejected** if its proper slice does not possess the required resources and the latter are not available within neighbor slices to be acquired at the flow arrival time.

4.3.7.2 Algorithm

As observed from section 4.3.3, implementation option 2 of InterS is more complicated and have not shown a significant outperformance in comparison to option 1. Therefore, the InterS's first implementation of the shared non premeptive is choosen to be extended with the shared preemptive reservation level feature. Algorithm 3 describes the extended admission system for the shared preemptive sharing mode.

Algorithm 3: InterS's extended admission system

Input : The new arriving Flow, $Flow_F$
Output : Accept Definitively, Not Definitively Accept or Reject $Flow_F$

1 $i \leftarrow$ Get the NS identifiant to which $Flow_F$ is belonging to
2 **if** $Slice_i.freeBW() >= Flow_F$*'s guarenteed data rate (FGR)*
3   **Pass**
        `//` $Flow_F$ `Accepted Definitively, Go to line 10`
4 **else**
5     $n \leftarrow$ Get the neighbor Slice identifiant with the Maximum free bandwidth resources(If it exists)
6   **if** $Slice_n.freeBW() >= FGR$
        `//` $Flow_F$ `Not Definitively Accepted`
7   Scale up $Slice_i$ with $Slice_n$ resources for $Flow_F$ (Scale down to normal any time $Slice_n$ claims its resources or at $Flow_F$ termination)
8 **else**
9   Reject $Flow_F$
10 **if** $Flow_F$ *is not Rejected AND Definitively Accepted*
11   **if** $Slice_i$*'s resources reservation level is Shared Preemptive*
12   Call the preemptive method on $Slice_i$: $Slice_i$.Preemptive()
13   Reserve $Slice_i$ resources for $Flow_F$
14   Configure data plane for $Flow_F$
15   Forward $Flow_F$ to $Slice_i$

The new algorithm of the admission system has the same structure as the one of the first implementation of the shared non preemptive but with some key differences.

1. The admission of a flow may follow up with one of the aforementioned decisions explained in the previous section: accept definitively, not definitively accept and reject.

2. There is two different perceptions of "free resources" depending on the resource reservation level of a slice. From a shared non preemptive slice's perspective, "free resources" are the ones that are not used by neither the proprietary slice nor any neighbor slice. For a shared preemptive slice, "free resources" include the resources that are not used by the proprietary slice and are used or not by a neighbor slice. Therefore, the freeBW() method has different behaviors depending on the resource reservation level of the slice calling it.

3. The slice is scaled down not only after flow termination but also if an eventual supporting slice is retrieving its resources (line 7).

4. An aditional set of operations represented by the Preemptive() method are executed if the slice is of the shared preemptive type (line 11).

One of the main functions of the preemptive component is to select the appropriate flows of neighbor slices (supported by $Slice_i$ resources) to be stopped and sent for re-admission for the sake of the optimization of bandwidth resources. This is also a way for the preemptive component to ensure the obtention of the highest possible network usage rate.

The flows selection process is based on the following optimization formula:

$$\begin{aligned} \min_{R} \quad & \sum_{i=0}^{m} R_i \\ \text{s.t.} \quad & \sum_{i=1}^{m} R_i >= RR \end{aligned} \tag{4.3}$$

m is the minimum number of flows that realize the equation 4.3. R_i is the guarented data rate of flow F_i existing in the network and respecting the constraint. RR designate the resources required to admit the new arriving flow ($Flow_F$). RR is the substraction of the free resources of $Slice_i$ (to which $Flow_F$ belongs to) that are not used by any neighbor slice from the flow's guarented data rate (FGR). The idea of the equation 4.3 is to stop the least number of flows with a minimum summation of their guarented data rates being at least equal to RR.

The output of the optimization algorithm is a list of flows associated to their guarented data rates F_i/R_i with i in [1,m]. If there is multiple flows with the the same guarented data rate, the lastly arrived flow will be selected by the algorithm to be stopped and sent for re-admission. In this way, we are privileging old flows over the recent ones. This is a part of the optimization algorithm since the old flows have existed in the network for a longer time and have generally more chance to complete the data transmission than the recent ones. We believe this will contribute to an increase of the flow acceptance rate.

4.3.7.3 Numerical Results

The obtained results, shown in Table 4.14, are clearly indicating the increase in TSFAR and TSBUR from scenario 1 to scenario 3 for all the flow arrival rate values. Table 4.15 precises the values of improvement in percentage from scenario 1 to scenario 3 as well as from scenario 1 to scenario 2 per each event rate. As detailed by table 4.15, scenario 3 has clearly achieved the best amelioration in TSFAR and TSBUR and this for each flow events rate.

Table 4.14: *the TSFAR (/458) and TSBUR (/500) obtained from each scenario for different flow arrival rates.*

	rate:1		rate:1/2		rate:1/4		rate:1/8		rate:1/16	
	TSBUR	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**
SC1	246.846	289	225.391	301	199.168	335	172.853	393	126.881	437
SC2	257.959	302	238.452	319	223.212	371	192.036	431	133.712	455
SC3	258.253	303	243.481	326	230.235	381	201.894	446	134.776	458

Table 4.15: *Improvement values (%) in TSFAR and TSBUR per event rate from SC1 to SC3 and SC1 to SC2*

	rate:1		rate:1/2		rate:1/4		rate:1/8		rate:1/16	
	TSBUR	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**	**TSBUR**	**TSFAR**
SC1–>SC3	4.62	4.84	8.03	8.31	15.6	**13.73**	**16.8**	13.49	6.22	4.81
SC1–>SC2	4.5	4.5	5.79	5.98	**12.07**	**10.75**	11.1	9.67	5.38	4.12

We are particularly interested by analyzing the improvement in TSBUR and TSFAR between the extreme scenarios 3 and 1 for each events rate. We observe from Figure 4.19 that the minimum augmentation in TSBUR and TSFAR is respectively achieved by the 1 and the 1/16 events rates. In this case TSBUR and TSFAR are respectively improved in scenario 3 comparing to scenario 1 by 4.62% and 4.81%. The maximum increase in TSBUR and TSFAR is respectively achieved by the 1/8 and 1/4. In this case TSBUR and TSFAR are respectively improved by 16.8% and 13.73%.

Regarding the amelioration from scenario 1 to scenario 2, the maximum increase in TSBUR and TSFAR is achieved by the 1/4 events rate.

We conclude from the results illustrated in Figure 4.19 as well as table 4.15 that the shared preemptive implementation of InterS is also performing the best at moderate events arrival rates such as 1/4 and 1/8.

Figures 4.20 and 4.21 illustrate respectively the bandwidth usage rate of each slice averaged on the 15 experiences and the per slice flow acceptance rate during the 15 experiences.

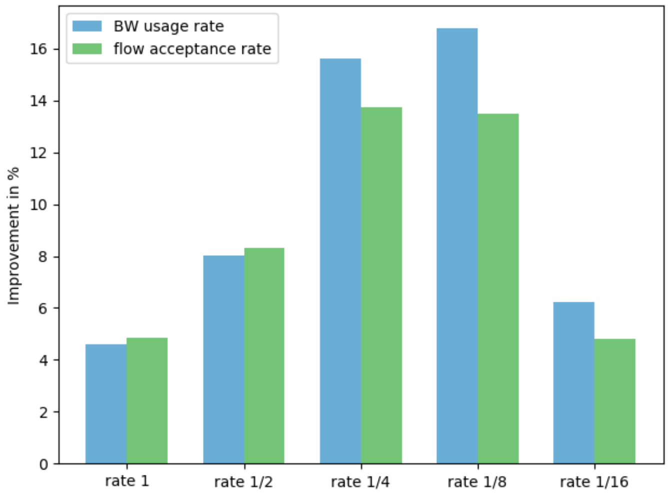
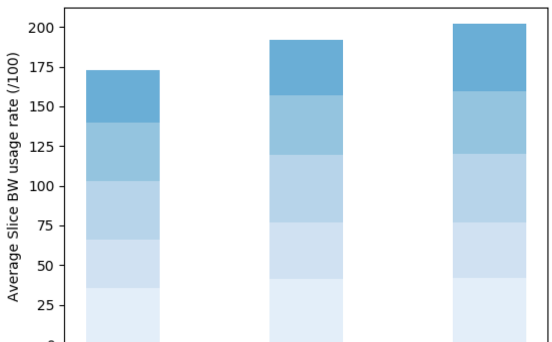


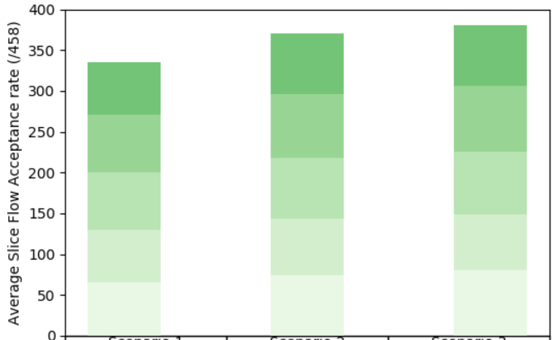
Figure 4.19: *Improvement percentage of TSBUR and TSFAR per events rate*



	Scenario 1	Scenario 2	Scenario 3
Slice 5	33.059	34.855	42.403
Slice 4	36.87	37.887	39.589
Slice 3	36.541	42.169	42.708
Slice 2	30.541	36.123	34.97
Slice 1	35.842	41.002	42.224
total	172.853	192.036	201.894

Figure 4.20: *Per slice bandwidth usage rate in a 1/8 events rate setup.*

In our shared preemptive implementation, flows are associated to admission codes that indicates how a flow was admitted or rejected and if it was iterrupted, stopped and re-addmitted. InterS defines 5 flow admission types:



	Scenario 1	Scenario 2	Scenario 3
Slice 5	64	75	75
Slice 4	71	78	80
Slice 3	70	74	77
Slice 2	65	70	69
Slice 1	65	74	80
total	335	371	381

Figure 4.21: *per Slice flow acceptance rate during the 15 experiences in a 1/4 events rate setup.*

- `Type 0`: flows are associated to an admission code equal to zero. It identifies the flows that got rejected the first time they were presented to the admission system.

- `Type 1`: flows are associated to an admission code equal to 1. It identifies the flows that got accepted the first time they were presented to the admission system.

- `Type 2`: flows are associated to an admission code equal to 2. It identifies the flows that were not admitted definitely (decision 3 in Figure 4.18) and they got the chance to complete the transmission of data before any interruption from a resource owner slice.

- `Type 3`: flows are associated to an admission code strictly superior than 2. It identifies the flows that were not admitted definitively and got stopped/re-admitted one or many time before they were definitively accepted or completed their transmission.

- `Type 4`: flows are associated to an admission code strictly inferior to 0. It identifies the flows that were not admitted definitively and got stopped/re-admitted one or many time before they were rejected.

Table 4.16 reveals the distribution of flows obtained in scenario 3 with a 1/4 events rate configuration at the end of the experiment (15 experiences).

Table 4.16: *Example of flows distribution*

Flow Types	Total Flows (/458)
Type 0	55
Type 1	331
Type 2	33
Type 3	17
Type 4	22

4.3.8 Comparison Between the Shared Non Preemptive and the Shared Preemptive Modes

Figure 4.22 compares the improvement in TSBUR and TSFAR between the shared preemptive and the shared non preemptive implementations of InterS. Particularly, upper graphs of the figure concerns the amelioration from scenario 1 to scenario 3 and lower graphs relate to the improvement from scenario 1 to scenario 2.

We observe that the shared non preemptive implementation has generally outperformed the shared preemptive one in the improvement of both TSFAR and TSBUR between extereme scenarios.

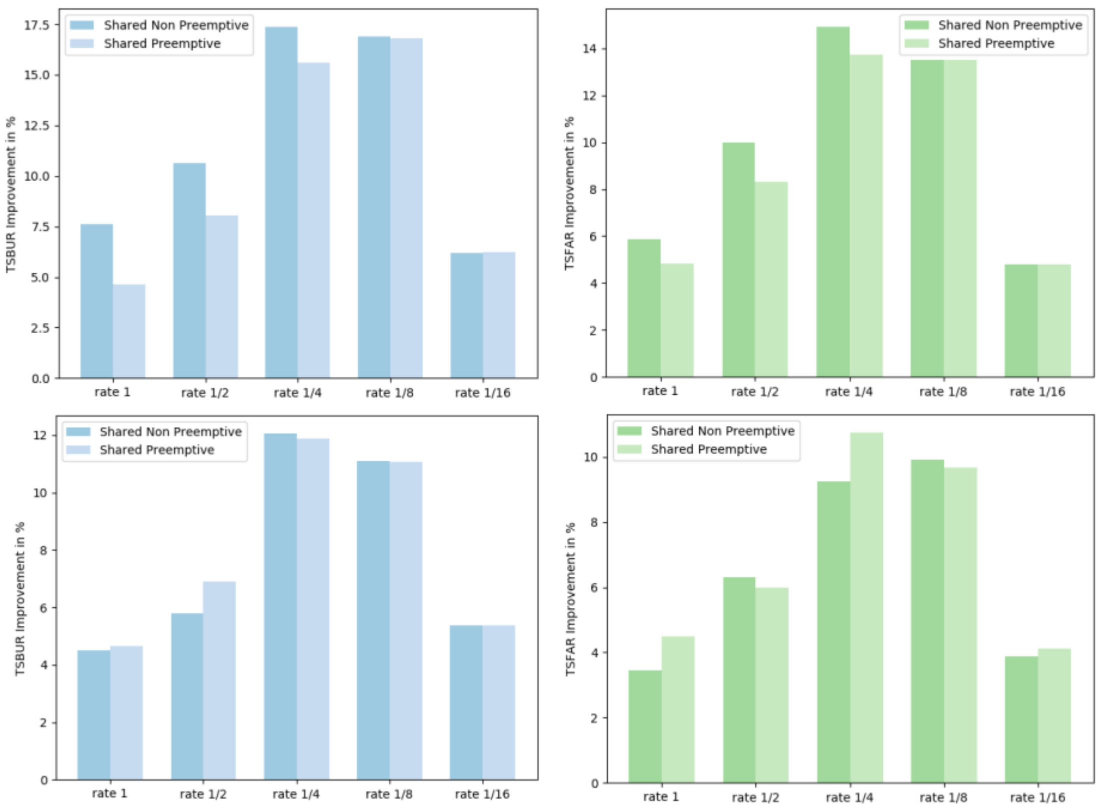


Figure 4.22: *Compare InterS'two sharing modes*

Regarding the improvement achieved in scenario 2, the shared preemptive implementation has outperformed at the 1 and the 1/2 events rates for TSBUR and the 1, 1/4 and 1/16 for TSFAR.

Generally, we conclude that the the difference in performance between the implmentation of the two sharing modes is not so relevant. We recall that the different values of TSFAR and TSBUR improvement corresponding to each of these two implementations are detailled in tables 4.9 and 4.15.

In the shared preemptive implementation, the inter-slice bandwidth resource sharing approach does not come with the limitation found in the previous resource reservation level and discussed in section 4.3.6. In fact, no slice is penalized in favor to another and this for all the flow events rate of the experiment. This can be observed firstly in Figure 4.21. The number of admitted flows in scenario 2 and 3 are greater than the one of scenario 1 for all the 5 slices.

As concluded in section 4.3.6, the limitation is even more important in higher events rates. Table 4.17 demonstrates the abscene of the issue in the shared preemptive implementation by providing the per slice flow acceptance rate at the highest events rate of the experiment (rate 1).

Table 4.17: *The number of accepted flows per slice for flow events rate 1.*

	Scenario 1	**Scenario 2**	**Scenario 3**
Slice 1	55	60	60
Slice 2	60	60	61
Slice 3	57	59	59
Slice 4	59	61	61
Slice 5	58	62	62

Finally, we conclude that the two sharing modes have achieved closer results in terms of performance, but the shared preemptive resource reservation level is more important and priviliged since its resolve the issue of slice penalization.

4.4 Conclusion

The Network Slicing paradigm promises operators to cope with the high variety of 5G services. The network optimization issues of slice-based infrastructures is still an interesting challenge. In this direction, this chapter proposed a novel approach we called inter-slice bandwidth resources sharing based on the COMS information model. We are also the first to realize a proof of concept that evaluated the approach.

Our solution InterS allows a congested Network Slice (NS) to temporarily acquire all or part of the free bandwidth resources of neighbor slices and use them to serve its own traffic. Therefore InterS allows the network operator's customers to instantiate a NS either as exclusive or shared. Two different sharing modes are implemented within InterS and evaluated.

Experiments have shown that the two sharing modes, also called resource reservation levels, can significantly improve the flow acceptance and the bandwidth usage rates of the network. However the use of the shared non preemptive mode results in penalizing some slices in favor to others regarding the per slice flow acceptance rate. The usage of the shared preemptive mode has resolved this issue.

CHAPTER 5

General Conclusion and Perspectives

Chapter content

5.1 Conclusion

Network Slicing is becoming a mature technology in recent years due to the continuous efforts conducted by the standards organizations, academic units and operators. In this thesis, our main contributions covered the modelization of the network slicing technology as well as the proposition of the inter-slice resource sharing mechanism to deal with the resource optimization challenge in slice-based infrastructures.

In our first contribution an E2E 5G network slicing model was provided based on an in-depth analysis of the modelization works published by the leading SDOs such ETSI, GSMA, 3GPP and IETF. We commenced by providing a general architecture of network slicing common to a large extent to the different SDOs. This part has served as a stepping-stone to set the primary foundation of network slicing before diving deep into domain-specific visions. Then we explained the two different visions of ordering a NS among SDOs. Next, the network slicing process in each of the RAN, the Core and the transport networks were exhaustively detailed. Last, we have addressed some open standardization issues as well as future research directions in this area.

Our second contribution focuses on providing a list of requirements that should be satisfied by a NS subnet management framework. The design as well as the requirements of the framework called NESSMA are derived from the E2E network slicing model provided by the first contribution. NESSMA is designed to manage NS Subnets as well as their supported services' life cycles taking into account its integration with NFV-MANO.

Our third contribution realizes a proof of concept of the inter-slice bandwidth resource sharing approach based on the COMS NS information model. Indeed, our implementation is based on the modelization works presented in previous contributions. Our PoC, so called InterS, allows the instantiation of NS with one of the following resource reservations levels: 1) Exclusive, 2) Shared non preemptive and 3) Shared preemptive. Our PoC includes two main implementations. The first one allows a congested NS to acquire free bandwidth resources from at most one neighbor NS. The second implementation, allows the congested NS to acquire and aggregate fragments of free resources from multiple neighbor NSs. Three main experiments were realized. The first evaluates the shared non preemptive mode within the first implementation. The second experiment evaluates the shared non preemptive mode within the second implementation. Finally, the third experiment evaluates the shared preemptive mode within the first implementation. The comparison of the two implementations has shown that they are slighly outperforming each other depending on different predefined network conditions. The results obtained from the realized experiments have shown that the shared non preemptive mode penalizes some slices in favor of others in the operation of the inter-slice bandwidth resource sharing. The usage of the shared preemptive resource reservation level has resolved the issue.

Finally, the results of all the experiments have shown that the inter-slice bandwidth resource sharing mechanism presented by the evaluated resource reservation levels can significantly improve the flow acceptance and the bandwidth usage rates of the network. Therefore, it presents a real resource optimization opportunity for slice-based infrastructures.

5.2 Perspectives

The perspectives of our work are summarized below. We group the futur works in short, medium and long terms research directions.

5.2.1 Short Term Research Directions

The inter-slice resource sharing approach presented in this thesis is a novel and an outstanding research direction. The short term research directions are as follow:

- **`Neighbor slices' selection criteria`**: regarding the algorithm of the inter-slice bandwidth resource sharing, the selection criteria of neighbor slices can be extended to other policies other than the maximum amount of free resources. A relevant criteria could be a measure of slice activity. Together, the two criterias will ensure the availibilty of free resources in the present and the futur so that the neighbor slice will not be penalized by the resources sharing mechanism. An other criteria that can be considered is the duration of resources availability in a neighbor slice. This will require the prediction of flows durations in the network so that the neighbor slice is selected only if its free resources are available for a duration at least equal to the flow duration of the congested slice. The slice activity as well as the flow duration can be predicted using machine learning algorithms based on the history of the NS.

- **`Extension to other network resources`**: we have particularly demonstrated the benefit of the inter-slice resource sharing for bandwidth resources. However, it can be applied to other resources such as compute or storage.

5.2.2 Medium and Long Terms Research Directions

We aim to achieve the following long term objectives:

- **`Realization of the ACTN Framework`**: NS modelization is still an important challenge in the standardization community. In the absence of a clear consensus within IETF on the modelization of network slicing in the transport domains, our vision was explained and provided in section 2.5.1. The implementation and the evaluation of such a vision will be possible in the future, especially with the launching of the ACTN project within the opensource ONOS controller. We believe that if the project is extended by a NS engine and an implementation of the COMS and the VN information models, it can serve as a reference platform for the transport network slice experimentations. This is also an opportunity to define and experiment with a transport customer service model based on our recommendations detailed in section 2.5.2.
- **`5G use case experiments`**: we aim to operate our proposed inter-slice resource sharing mechanism in a large environnement with different network topologies including different types of VNFs and experiment with a 5G use case such as “connected vehicules”.

Publications

Based on the research work presented in this thesis, some papers have been published in international conferences and journals:

i. **Conferences:**

1. Mohammed Chahbar, Gladys Diaz, and Abdulhalim Dandoush. "InterS: Towards Inter-Slice Bandwidth Resource Sharing". In: *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*. 2021, pp. 1–6. DOI: 10.1109/HPSR52026.2021.9481833
2. Mohammed Chahbar, Gladys Diaz, and Abdulhalim Dandoush. "Towards a Unified Network Slicing Model". In: *2019 15th International Conference on Network and Service Management (CNSM)*. 2019, pp. 1–5. DOI: 10.23919/CNSM46954.2019.9012745
3. Mohammed Chahbar et al. "NESSMA: Network Slice Subnet Management Framework". In: *2019 10th International Conference on Networks of the Future (NoF)*. 2019, pp. 54–57. DOI: 10.1109/NoF47743.2019.9015010
4. Tarek Menouer et al. "Scheduling Service Function Chains with Dependencies in the Cloud". In: *2020 IEEE 9th International Conference on Cloud Networking (CloudNet)*. 2020, pp. 1–3. DOI: 10.1109/CloudNet51028.2020.9335790

ii. **Journals:**

1. Mohammed Chahbar et al. "A Comprehensive Survey on the E2E 5G Network Slicing Model". In: *IEEE Transactions on Network and Service Management* 18.1 (2021), pp. 49–62. DOI: 10.1109/TNSM.2020.3044626

Bibliography

[1] NGMN Alliance. "Next generation mobile networks (NGMN) 5G whitepaper". In: (2015) *Cited on page 2.*

[2] *Telecommunication management; Study on management and orchestration of network slicing for next generation network.* Technical report TR 28.801. 3rd Generation Partnership Project (3GPP), 2018 *Cited on pages 2, 23, 42, 48.*

[3] Peter Rost et al. "Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks". In: *IEEE Communications Magazine* 55.5 (2017), pp. 72–79. DOI: 10.1109/MCOM.2017.1600920 *Cited on page 2.*

[4] Diego Kreutz et al. "Software-Defined Networking: A Comprehensive Survey". In: *Proceedings of the IEEE* 103.1 (2015), pp. 14–76. DOI: 10.1109/JPROC.2014.2371999 *Cited on pages 4, 5.*

[5] N. McKeown et al. "OpenFlow: enabling innovation in campus networks". In: *Comput. Commun. Rev.* 38 (2008), pp. 69–74 *Cited on page 4.*

[6] *Network Functions Virtualisation (NFV); Architectural Framework, V1.2.1.* Group specification GS NFV 002. ETSI, 2014 *Cited on pages 5, 6.*

[7] Riccardo Guerzoni et al. "Analysis of end-to-end multi-domain management and orchestration frameworks for software defined infrastructures: An architectural survey". In: *Transactions on Emerging Telecommunications Technologies* 28 (Sept. 2016). DOI: 10.1002/ett.3103 *Cited on page 6.*

[8] Barbara Martini, Molka Gharbaoui, and Piero Castoldi. "Cross-Functional resource orchestration in optical telco clouds". In: *2015 17th International Conference on Transparent Optical Networks (ICTON).* 2015, pp. 1–5. DOI: 10.1109/ICTON.2015.7193450 *Cited on page 7.*

[9] 5G Exchange Project. *Accessed on November 2021.* URL: https://5g-ppp.eu/5gex/ *Cited on page 7.*

[10] *View on 5G Architecture version 1.0*. Report GS NFV-IFA 009. 5G PPP Architecture Working Group, 2016 *Cited on pages 7, 43.*

[11] *Network Functions Virtualisation (NFV); Management and Orchestration; Report on Architectural Options*. Report GS NFV-IFA 009. ETSI, 2016 *Cited on pages 8, 52.*

[12] A. Mayoral et al. "Control orchestration protocol: Unified transport API for distributed cloud and network orchestration". In: *IEEE/OSA Journal of Optical Communications and Networking* 9 (2017), A216–A222 *Cited on page 8.*

[13] Mohammed Chahbar et al. "A Comprehensive Survey on the E2E 5G Network Slicing Model". In: *IEEE Transactions on Network and Service Management* 18.1 (2021), pp. 49–62. DOI: 10.1109/TNSM.2020.3044626 *Cited on pages 10, 65, 97.*

[14] Mohammed Chahbar et al. "NESSMA: Network Slice Subnet Management Framework". In: *2019 10th International Conference on Networks of the Future (NoF)*. 2019, pp. 54–57. DOI: 10.1109/NoF47743.2019.9015010 *Cited on pages 10, 97.*

[15] Mohammed Chahbar, Gladys Diaz, and Abdulhalim Dandoush. "InterS: Towards Inter-Slice Bandwidth Resource Sharing". In: *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*. 2021, pp. 1–6. DOI: 10.1109/HPSR52026.2021.9481833 *Cited on pages 10, 70, 97.*

[16] Mohammed Chahbar, Gladys Diaz, and Abdulhalim Dandoush. "Towards a Unified Network Slicing Model". In: *2019 15th International Conference on Network and Service Management (CNSM)*. 2019, pp. 1–5. DOI: 10.23919/CNSM46954.2019.9012745 *Cited on pages 15, 42, 97.*

[17] Alcardo Alex Barakabitze et al. "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges". In: *Computer Networks* 167 (2020), p. 106984. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2019.106984 *Cited on page 15.*

[18] Shunliang Zhang. "An Overview of Network Slicing for 5G". In: *IEEE Wireless Communications* 26.3 (2019), pp. 111–117. DOI: 10.1109/MWC.2019.1800234 *Cited on page 15.*

[19] Xenofon Foukas et al. “Network Slicing in 5G: Survey and Challenges”. In: *IEEE Communications Magazine* 55.5 (2017), pp. 94–100. DOI: 10.1109/MCOM.2017.1600951 *Cited on page 15.*

[20] Alexandros Kaloxylos. “A Survey and an Analysis of Network Slicing in 5G Networks”. In: *IEEE Communications Standards Magazine* 2.1 (2018), pp. 60–65. DOI: 10.1109/MCOMSTD.2018.1700072 *Cited on page 15.*

[21] Q. Chen and Caizhang Liu. “A Survey of Network Slicing in 5G”. In: *DEStech Transactions on Computer Science and Engineering* (2017) *Cited on page 15.*

[22] *Next Generation Protocols (NGP); E2E Network Slicing Reference Framework and Information Model*. Group report GR NGP 011. ETSI, 2018 *Cited on pages 16, 49, 50, 52.*

[23] Liang Geng et al. *Network Slicing Architecture*. Internet-Draft draft-geng-netslices-architecture-02. Work in Progress. Internet Engineering Task Force, July 2017. 27 pp. *Cited on page 17.*

[24] Shunsuke Homma et al. *Gateway Function for Network Slicing*. Internet-Draft draft-homma-rtgwg-slice-gateway-02. Work in Progress. Internet Engineering Task Force, Mar. 2020. 30 pp. *Cited on pages 18, 48.*

[25] *Description of network slicing Concept*. Tech. rep. NGMN Alliance, 2016 *Cited on pages 19, 20.*

[26] *5G network slicing Report: From Vertical Industry Requirements to NS Characteristics*. Tech. rep. GSMA, 2018 *Cited on page 20.*

[27] *5G; Management and orchestration; Provisioning, version 15.5.0 Release 15*. Technical specification TS 28.531. 3GPP, 2020 *Cited on page 20.*

[28] *Generic NS Template, Version 2.0*. Tech. rep. GSMA, 2019 *Cited on page 21.*

[29] Reza Rokui et al. *IETF Definition of Transport Slice*. Internet-Draft draft-nsdt-teas-transport-slice-definition-02. Work in Progress. Internet Engineering Task Force. 19 pp. *Cited on page 21.*

[30] *Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3, version 15.5.0 Release 15*. technical specification TS 28.541. 3GPP, 2020 *Cited on pages 21, 22, 25, 26.*

[31] *Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Report on NFV Information Model*. Tech. rep. GR NFV-IFA 015. ETSI, 2017 *Cited on page 23.*

[32] *Network Functions Virtualisation (NFV); Management and Orchestration; Network Service Templates Specification*. Tech. rep. GS NFV-IFA 014. ETSI, 2016 *Cited on page 23.*

[33] *Telecommunication management; State management data definition Integration Reference Point (IRP); Information Service (IS), Version 15.1*. Technical specification TS 28.625. 3GPP, 2019 *Cited on page 25.*

[34] *Network Functions Virtualisation (NFV); Management and Orchestration; Os-Ma-Nfvo reference point-Interface and Information Model Specification, Version 2.1.1*. Tech. rep. GS NFV IFA 013. ETSI, 2016 *Cited on page 25.*

[35] *System architecture for the 5G System (5GS), Version 15.9.0*. Technical specification TS 23.501. 3GPP, 2020 *Cited on page 26.*

[36] *Service requirements for the 5G system, Version 15.8*. Technical specification TS 22.261. 3GPP, 2019 *Cited on page 27.*

[37] *Management and orchestration; 5G end to end Key Performance Indicators (KPI), Version 15.4*. Technical specification TS 28.554. 3GPP, 2019 *Cited on page 27.*

[38] Reza Rokui et al. *5G Transport Slice Connectivity Interface*. Internet-Draft draft-rokui-5g-transport-slice-00. Work in Progress. Internet Engineering Task Force, July 2019. 28 pp. *Cited on pages 27, 38.*

[39] Daniele Ceccarelli and Young Lee. *Framework for Abstraction and Control of TE Networks (ACTN)*. RFC 8453. Aug. 2018. DOI: 10.17487/RFC8453 *Cited on page 28.*

[40] Adrian Farrel et al. *Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks*. RFC 7926. July 2016. DOI: 10.17487/RFC7926 *Cited on pages 28, 30.*

[41] Aiko Pras and Jürgen Schönwälder. *On the Difference between Information Models and Data Models*. RFC 3444. Jan. 2003. DOI: 10.17487/RFC3444 *Cited on page 28.*

[42] Dean Bogdanović, Benoît Claise, and Carl Moberg. *YANG Module Classification*. RFC 8199. July 2017. DOI: 10.17487/RFC8199 *Cited on page 29.*

[43] Qin Wu, Will (Shucheng) LIU, and Adrian Farrel. *Service Models Explained*. RFC 8309. Jan. 2018. DOI: 10.17487/RFC8309 *Cited on pages 29, 33, 34.*

[44] Xufeng Liu et al. *YANG Data Model for Traffic Engineering (TE) Topologies*. RFC 8795. Aug. 2020. DOI: 10.17487/RFC8795 *Cited on pages 30, 35.*

[45] Young Lee et al. *A YANG Data Model for VN Operation*. Internet-Draft draft-ietf-teas-actn-vn-yang-11. Work in Progress. Internet Engineering Task Force, Feb. 2021. 60 pp. *Cited on pages 30, 35.*

[46] Li Qiang et al. *Technology Independent Information Model for Network Slicing*. Internet-Draft draft-qiang-coms-netslicing-information-model-02. Work in Progress. Internet Engineering Task Force, Jan. 2018. 28 pp. *Cited on pages 30, 35, 50, 59.*

[47] Liang Geng et al. *COMS Architecture*. Internet-Draft draft-geng-coms-architecture-02. Work in Progress. Internet Engineering Task Force, Mar. 2018. 12 pp. *Cited on pages 30, 36.*

[48] Li Qiang et al. *The Use Cases of Common Operation and Management of Network Slicing*. Internet-Draft draft-qiang-coms-use-cases-00. Work in Progress. Internet Engineering Task Force, Feb. 2018. 12 pp. *Cited on pages 30, 36.*

[49] Alexander Clemm et al. *A YANG Data Model for Network Topologies*. RFC 8345. Mar. 2018. DOI: 10.17487/RFC8345 *Cited on page 30.*

[50] Xavier de Foy et al. *Interconnecting (or Stitching) Network Slice Subnets*. Internet-Draft draft-defoy-coms-subnet-interconnection-03. Work in Progress. Internet Engineering Task Force. 13 pp. *Cited on pages 32, 35, 49.*

[51] Tarek Saad et al. *A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces*. Internet-Draft draft-ietf-teas-yang-te-26. Work in Progress. Internet Engineering Task Force, Feb. 2021. 120 pp. *Cited on pages 32, 35.*

[52] Young Lee et al. *YANG models for VN/TE Performance Monitoring Telemetry and Scaling Intent Autonomics*. Internet-Draft draft-ietf-teas-actn-pm-telemetry-

autonomics-05. Work in Progress. Internet Engineering Task Force, Feb. 2021. 32 pp. *Cited on pages 33, 35.*

[53] Young Lee et al. *Information Model for Abstraction and Control of TE Networks (ACTN)*. RFC 8454. Sept. 2018. DOI: 10.17487/RFC8454 *Cited on pages 33, 35.*

[54] Bin Wen et al. *A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery*. RFC 8466. Oct. 2018. DOI: 10.17487/RFC8466 *Cited on page 33.*

[55] Qin Wu et al. *YANG Data Model for L3VPN Service Delivery*. RFC 8299. Jan. 2018. DOI: 10.17487/RFC8299 *Cited on page 33.*

[56] Himanshu C. Shah et al. *YANG Data Model for MPLS-based L2VPN*. Internet-Draft draft-ietf-bess-l2vpn-yang-10. Work in Progress. Internet Engineering Task Force, July 2019. 49 pp. *Cited on page 33.*

[57] Dhanendra Jain et al. *Yang Data Model for BGP/MPLS L3 VPNs*. Internet-Draft draft-ietf-bess-l3vpn-yang-04. Work in Progress. Internet Engineering Task Force. 23 pp. *Cited on page 33.*

[58] Qin Wu et al. *A Framework for Automating Service and Network Management with YANG*. RFC 8969. Jan. 2021. DOI: 10.17487/RFC8969 *Cited on page 35.*

[59] Open Network Foundation (ONF). *ONF's ACTN (Abstraction and Control of TE networks) project, Accessed on November 2021*. URL: https://wiki.onosproject.org/pages/viewpage.action?pageId=8424694 *Cited on page 37.*

[60] Igor Bryskin et al. *TE Topology and Tunnel Modeling for Transport Networks*. Internet-Draft draft-ietf-teas-te-topo-and-tunnel-modeling-06. Work in Progress. Internet Engineering Task Force, July 2020. 109 pp. *Cited on page 37.*

[61] Xufeng Liu et al. *IETF Network Slice YANG Data Model*. Internet-Draft draft-liu-teas-transport-network-slice-yang-03. Work in Progress. Internet Engineering Task Force, May 2021. 32 pp. *Cited on page 39.*

[62] Luis M. Contreras, Shunsuke Homma, and Jose A. Ordonez-Lucena. *IETF Network Slice Use Cases and Attributes for Northbound Interface of IETF Network Slice Controllers*. Internet-Draft draft-contreras-teas-slice-nbi-04. Work in Progress. Internet Engineering Task Force, Feb. 2021. 14 pp. *Cited on page 39.*

[63] Young Lee et al. *Traffic Engineering and Service Mapping Yang Model*. Internet-Draft draft-lee-teas-te-service-mapping-yang-13. Work in Progress. Internet Engineering Task Force, Dec. 2018. 28 pp. *Cited on page 39.*

[64] Daniel King et al. *Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Network Slicing*. Internet-Draft draft-king-teas-applicability-actn-slicing-10. Work in Progress. Internet Engineering Task Force, Mar. 2021. 22 pp. *Cited on page 39.*

[65] Shunsuke Homma et al. *Network Slice Provision Models*. Internet-Draft draft-homma-slice-provision-models-02. Work in Progress. Internet Engineering Task Force, Nov. 2019. 26 pp. *Cited on pages 40, 41.*

[66] *Service Ordering Management API REST Specification R18.5.1*. Tech. rep. TMF641. TM Forum (TMF), 2019 *Cited on page 40.*

[67] Mohamed Boucadair et al. *Dynamic Service Negotiation: The Connectivity Provisioning Negotiation Protocol (CPNP)*. RFC 8921. Oct. 2020. DOI: 10.17487/RFC8921 *Cited on page 40.*

[68] *Zero-touch Network and Service Management (ZSM); End to end management and orchestration of network slicing, Version 0.14.0*. Tech. rep. GS ZSM 003. ETSI, 2020 *Cited on page 42.*

[69] *Zero-touch network and Service Management (ZSM); Reference Architecture, Version 1.1.1*. Tech. rep. GS ZSM 002. ETSI, 2019 *Cited on page 42.*

[70] Xuesong Geng et al. *5G End-to-end Network Slice Mapping from the view of Transport Network*. Internet-Draft draft-geng-teas-network-slice-mapping-03. Work in Progress. Internet Engineering Task Force, Feb. 2021. 18 pp. *Cited on pages 42, 43.*

[71] 5G riders on the storm catalyst. *Network Slice Management Interface, Accessed on November 2021*. URL: https://github.com/5g-ridersonthestorm/gsma-gst *Cited on page 43.*

[72] TM Forum. *Catalyst projects, 5G riders on the storm catalyst, Accessed on November 2021*. URL: https://www.tmforum.org/5g-riders-storm/ *Cited on pages 43, 44.*

[73] 5G riders on the storm catalyst. *A JSON version of the GSMA GST, Accessed on November 2021*. URL: https://github.com/5g-ridersonthestorm/gsma-gst *Cited on page 44.*

[74] Zbigniew Kotulski et al. "On end-to-end approach for slice isolation in 5G networks. Fundamental challenges". In: *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*. 2017, pp. 783–792. DOI: 10.15439/2017F228 *Cited on pages 45, 60.*

[75] Hannu Flinck et al. *Network Slicing Management and Orchestration*. Internet-Draft draft-flinck-slicing-management-00. Work in Progress. Internet Engineering Task Force, July 2017. 10 pp. *Cited on pages 48, 51.*

[76] *5G; Management and orchestration; Concepts, use cases and requirements*. Technical specification TS 28.530. 3rd Generation Partnership Project (3GPP), 2018 *Cited on pages 48, 50.*

[77] Slawomir Kukliński et al. "A reference architecture for network slicing". In: *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. 2018, pp. 217–221. DOI: 10.1109/NETSOFT.2018.8460057 *Cited on page 48.*

[78] *Network Functions Virtualisation (NFV); Management and Orchestration*. Group specification GS NFV-MAN 001. ETSI, 2014 *Cited on page 48.*

[79] *Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3*. Technical specification TS 28.541. 3rd Generation Partnership Project (3GPP), 2019 *Cited on page 51.*

[80] *IETF NFVRG, Accessed on November 2021*. URL: https://datatracker.ietf.org/rg/nfvrg/about/ *Cited on page 51.*

[81] *Towards Integration of Slice Networking in NFV*. Meeting 98 NFVRG Meeting. Internet Engineering Task Force, 2017 *Cited on page 51.*

[82] Yuki Yoshida et al. "SDN-Based Network Orchestration of Variable-Capacity Optical Packet Switching Network Over Programmable Flexi-Grid Elastic Optical Path Network". In: *Journal of Lightwave Technology* 33.3 (2015), pp. 609–617. DOI: 10.1109/JLT.2014.2351852 *Cited on page 53.*

[83] Adrian Farrel et al. *An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control*. RFC 8283. Dec. 2017. DOI: 10.17487/RFC8283 *Cited on page 54.*

[84] Daniel King and Adrian Farrel. *A PCE-Based Architecture for Application-Based Network Operations*. RFC 7491. Mar. 2015. DOI: 10.17487/RFC7491 *Cited on page 55.*

[85] Jan Seedorf and Eric Burger. *Application-Layer Traffic Optimization (ALTO) Problem Statement*. RFC 5693. Oct. 2009. DOI: 10.17487/RFC5693 *Cited on page 56.*

[86] *Open Baton, Accessed on November 2021*. URL: https://openbaton.github.io/index.html *Cited on page 56.*

[87] *ONOS, Accessed on November 2021*. URL: https://onosproject.org *Cited on page 56.*

[88] *Ryu SDN Framework, Accessed on November 2021*. URL: https://ryu-sdn.org/ *Cited on pages 56, 66, 70.*

[89] ETSI GR NGP 011. "Next Generation Protocols (NGP); E2E network slicing Reference Framework and Information Model". In: (2018) *Cited on page 59.*

[90] A. Gausseran et al. "Be Scalable and Rescue My Slices During Reconfiguration". In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 2020, pp. 1–6. DOI: 10.1109/ICC40277.2020.9148871 *Cited on page 59.*

[91] Bob Lantz, Brandon Heller, and N. McKeown. "A network in a laptop: rapid prototyping for software-defined networks". In: *Hotnets-IX*. 2010 *Cited on pages 68, 70.*

[92] Ben Pfaff et al. "The Design and Implementation of Open vSwitch". In: *NSDI*. 2015 *Cited on pages 68, 70.*

[93] *Grid 5000 Cloud, Accessed on November 2021*. URL: https://www.grid5000.fr/w/SSH *Cited on page 70.*

[94] Tarek Menouer et al. "Scheduling Service Function Chains with Dependencies in the Cloud". In: *2020 IEEE 9th International Conference on Cloud Networking (CloudNet)*. 2020, pp. 1–3. DOI: 10.1109/CloudNet51028.2020.9335790 *Cited on page 97.*