Dynamic Control and Optimization of Wireless Virtual Networks

Quang Trung Luu

université
PARIS-SACLAY

Dynamic Control and Optimization of Wireless Virtual Networks

Contrôle et optimisation des réseaux virtuels sans fil

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Réseaux, Information et Communications
Unité de recherche: Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des Signaux et Systèmes, 91190, Gif-sur-Yvette, France
Référent: Faculté des Sciences d'Orsay

Thèse présentée et soutenue à Paris-Saclay, le 07/06/2021, par

Quang Trung LUU

Composition du jury

Adlen KSENTINI Professeur, EURECOM Sophia Antipolis	Président
Toufik AHMED Professeur, Université de Bordeaux	Rapporteur & Examinateur
Stefano SECCI Professeur, Conservatoire National des Arts et Métiers (CNAM)	Rapporteur & Examinateur
Roberto RIGGIO Chercheur senior, RISE Research Institutes of Sweden AB	Examinateur
Véronique VÈQUE Professeure, Université Paris-Saclay	Examinatrice

Direction de la thèse

Michel KIEFFER Professeur, Université Paris-Saclay	Directeur
Sylvaine KERBOEUF Ingénieur de recherche senior, Nokia Bell Labs	Coencadrante

Thèse de doctorat

NNT: 2021UPASG039

Dynamic Control and Optimization of Wireless Virtual Networks

Quang-Trung LUU

A thesis submitted in partial fulfillment

requirements for the degree of

Doctor of Philosophy

at Paris-Saclay University

Paris, June 7, 2021

To my family, especially my late grandfather—
one of the most resourceful man I've ever known.

"*Ad meliora et ad maiora semper.*"

—Latin phrase

"*You drown not by falling into a river, but by staying submerged in it.*"

—Paulo Coelho

"*But you see, I have, let's say, sixty years to live. Most of that time will be spent working. I've chosen the work I want to do. If I find no joy in it, then I'm only condemning myself to sixty years of torture. And I can find the joy only if I do my work in the best way possible to me.*"

—Ayn Rand

Acknowledgements

This thesis marks the end of three wonderful years at Bell Labs and the Laboratoire des Signaux et Systèmes (L2S), CentraleSupélec/Université Paris-Saclay. I thoroughly enjoyed my PhD and I am deeply grateful to have been able to work with and learn from so many people. I would like to express my gratitude to:

... my advisors

I would like to express my deepest appreciation to Profs. Michel Kieffer and Alexandre Mouradian, my advisors at L2S, and Dr. Sylvaine Kerboeuf, my supervisor at Bell Labs, who have been incredibly supportive throughout my PhD. This milestone would not have been possible without your guidance. Without Michel's keen eye for detail, many of my publications would have looked differently.

... my PhD dissertation committee

I would like to extend my sincere gratitude to Profs. Stefano Secci, Toufik Ahmed, Véronique Vèque, Adlen Ksentini, and Dr. Roberto Riggio, for serving in my committee and reviewing this thesis.

... my colleagues at Bell Labs and L2S – CentraleSupélec

It was a pleasure to have the chance to work with so many cool colleagues at Bell Labs and L2S. The atmosphere at these places was very special, professional yet enjoyable. Thank you all for sharing with me wonderful moments, in particular: Illyyne Saffar, Frédéric Faucheux, Bogdan Uscumlic, Soumeya Kaada, Fred Aklamanu, El Hocine Bouzidi, Ejder Bastug, Dominique Chiaroni, Barbara Orlandi, Dominique Verchère, Andrea Enrici, Marie-Line Alberi-Morel, Alberto Conte, Jean-Luc Lafragette, and many others at ENSA–Bell Labs; and Khac-Hoang Ngo, Viet-Dung Nguyen, Mourad Aklouf, Violeta Roizman, Oumaima El Joubari, Hirah Malik, and many others at L2S. With Fred, Hocine, and Quan, I have fond memories of conference and summer school trips in UAE and Italy.

... my former advisors and professors

There is no way I could have gotten where I am today without the support of my former advisors and professors, and I owe them a debt of gratitude. Special thanks to my former bachelor and master's advisors, Profs. Cao-Minh Ta, Yem Vu-Van, Antoine Diet, Yann Le Bihan, Stavros Koulouridis, Anthony Busson, and

Isabelle Guérin-Lassous. Profs. Xavier Checoury and Bernard Journet were also helpful throughout my application for the Paris-Saclay master's scholarship.

... my friends

My gratitude is extended to my Vietnamese friends both in and outside of Vietnam, including Phuong-Anh Nguyen, Minh-Hoa Nguyen, Thanh-Trung Nguyen, Quan Pham Van, Quang-Huy Tran, Ngoc-Duy Nguyen, Duc-Hung Luong, Ba-Hai Nguyen, Hoa Le, and many others. They have made my life in France colorful and pleasant. I have had great moments of joyfulness with them, with coffee breaks, lunches, and a couple of beers along the Seine!

... and, last, but not least, my family

Most of all, I would like to thank my whole family. They always back me up in any decision I have ever made. This thesis is dedicated to them, especially my parents and my late grandfather.

Trung Luu, Paris, April 2021.

Abstract

Network slicing is a key enabler for 5G networks. With network slicing, Mobile Network Operators (MNO) create various slices for Service Providers (SP) to accommodate customized services. As network slices are operated on a common network infrastructure owned by some Infrastructure Provider (InP), efficiently sharing the resources across various slices is very important.

In this thesis, taking the InP perspective, we propose several methods for provisioning resources for network slices. Previous best-effort approaches deploy the various Service Function Chains (SFCs) of a given slice sequentially in the infrastructure network. In this thesis, we provision aggregate resources to accommodate slice demands. Once provisioning is successful, the SFCs of the slice are ensured to get enough resources to be properly operated. This facilitates the satisfaction of the slice quality of service requirements. The proposed provisioning solutions also yield a reduction of the computational resources needed to deploy the SFCs.

In the first part, we consider deterministic slice resource demands. Time is slotted and the provisioning requests are processed independently in each time slot, over which resource demands are assumed constant. Resource provisioning is cast in the framework of mixed integer linear programming. Optimal and suboptimal reduced-complexity provision approaches are proposed. We further extend the provisioning framework by considering slices to be deployed over some specific geographical areas. In this situation, the coverage as well as minimum per-user rate constraints are taken into account.

In the second part, we address uncertain slice resource demands *(e.g.*, partly unknown number of users, fluctuations of user resource demands). Robust resource provisioning is formulated as a nonlinear constrained optimization problem. Several reduced-complexity robust provision approaches are proposed to provide a probabilistic guarantee that the slice resource demands are fulfilled, while limiting the impact on low-priority background services. Next, we consider the slice provisioning problem over all the time slots of the slice life-time and account for the dynamic nature of slice requests (*i.e.*, arrivals, slice start and stop times). The provisioning scheme is then cast in a max-min optimization problem. The aim is to maximize the amount of slices for which infrastructure resources can be granted while minimizing the provisioning costs. Several reduced-complexity strategies are proposed for the admission control and resource provisioning of prioritized slice requests with uncertain resource demands.

Keywords: network slicing, resource provisioning, resource allocation, coverage constraints, wireless network virtualization, 5G, dynamic provisioning, slice admission control, uncertainty, linear programming.

Résumé

Le découpage du réseau est une technologie clé des réseaux 5G, grâce à laquelle les opérateurs de réseaux mobiles peuvent créer des *tranches* de réseau indépendantes. Chaque tranche permet à des fournisseurs d'offrir des services personnalisés. Comme les tranches sont opérées sur une infrastructure de réseau commune gérée par un fournisseur d'infrastructure, il est essentiel de développer des méthodes de partage efficace des ressources.

Cette thèse adopte le point de vue du fournisseur d'infrastructure et propose plusieurs méthodes de réservation de ressources pour les tranches de réseau. Actuellement, les chaines de fonctions appartenant à une tranche sont déployées séquentielle-ment sur l'infrastructure, sans avoir de garantie quant à la disponibilité des ressources. Afin d'aller au-delà de cette approche, nous considérons dans cette thèse des approches de réservation des ressources pour les tranches en considérant les besoins agrégés des chaines de fonctions avant le déploiement effectif des chaines de fonctions. Lorsque la réservation a abouti, les chaines de fonctions ont l'assurance de disposer de suffisamment de ressources lors de leur déploiement et de leur mise en service afin de satisfaire les exigences de qualité de service de la tranche. La réservation de ressources permet également d'accélérer la phase d'allocation de ressources des chaines de fonctions.

Dans une première partie, nous considérons des demandes de ressources déterministes pour les tranches. Le temps est découpé en intervalles et les demandes des tranches (supposées constantes sur chaque intervalle) sont traitées indépendamment dans chaque intervalle. La réservation de ressources est traitée à l'aide d'outils de programmation linéaire en nombres entiers mixtes. Des approches de réservation optimales et sous-optimales à complexité réduite sont proposées. Nous avons égale-ment étendu la réservation de ressources a des situations où les tranches doivent être déployées sur des zones géographiques spécifiques. Dans cette situation, la couverture ainsi que la contrainte d'un débit minimum par utilisateur sont prises en compte.

Dans une deuxième partie, nous considérons des incertitudes dans la demande de ressources des tranches (nombre d'utilisateurs, fluctuations des besoins par utilisateur). La réservation de ressources robuste est alors formulée comme un problème d'optimisation non linéaire sous contraintes. Plusieurs approches de réservation robuste à complexité réduite sont proposées. Elles fournissent une garantie probabiliste que les demandes de ressources des tranches sont satisfaites, tout en

limitant l'impact sur les autres services opérées en tâches de fond sur l'infrastructure partagée. Nous abordons ensuite le problème de réservation de ressources en tenant compte de tout le cycle de vie d'une tranche. La nature dynamique des demandes de ressources d'une tranche est considérée (instants d'arrivée des demandes, démarrage et fin d'opération). La réservation de ressources est formulée comme un problème d'optimisation max-min. L'objectif est de maximiser la quantité de tranches pour lesquelles des ressources d'infrastructure peuvent être accordées tout en minimisant les coûts de des ressources réservées. Plusieurs stratégies à complexité réduite sont proposées et nécessitent la mise en œuvre d'un mécanisme conjoint de contrôle d'admission et de réservation de ressources pour des tranches de réseaux présentant différent niveaux de priorité. Les approches restent robustes à des demandes de ressources incertaines.

Mots clés : tranche de réseau, réservation de ressources, allocation de ressources, contraintes de couverture, virtualisation des réseaux sans fil, 5G, contrôle d'admission de tranches, incertitudes, programme linéaire.

Contents

List of Figures

List of Tables

Nomenclature

The meaning of an acronym is usually pointed out once on its first appearance in the text.

BBU	Base Band Unit
BS	Base Station
CDF	Cumulative Distribution Function
CG	Column Generation
C-RAN	Cloud Radio Access Network
CU	Central Unit
DL	Downlink
DU	Distributed Unit
FW	Firewall
GW	Gateway
HD	High Definition
IDPS	Intrusion Detection Prevention System
ILP	Integer Linear Program
ImP	Impact Probability
InP	Infrastructure Provider
JP	Joint Provisioning approach
LP	Linear Program
MILP	Mixed Integer Linear Program
MI-SLA	SLA between an MNO and an InP
MNO	Mobile Network Operator
MP	Master Problem
NFV	Network Function Virtualization
NP	Network resource Provisioning problem
PDF	Probability Density Function
pmf	probability mass function
PP	Pricing Problem
QIP	Quadratic Integer Program
QoE	Quality of Experience
QoS	Quality of Service
RB	Resource Block

RD	Resource Demand
RP	Radio resource Provisioning problem
RRH	Remote Radio Head
RU	Radio Unit
SDN	Software Defined Networking
SFC	Service Function Chain
SFC-RD	SFC Resource Demand
SLA	Service Level Agreement
SM-SLA	SLA between an SP and an MNO
SP	Service Provider or Sequential Provisioning approach
S-RD	Slice Resource Demand
SSP	Service Satisfaction Probability
TM	Traffic Monitor
UHD	Ultra High Definition
UL	Uplink
U-RD	User Resource Demand
VNF	Virtual Network Function
VOC	Video Optimizer Controller
WGMP	Weighted Graph Matching Problem

Notation

In general, normal-font letters (*e.g.*, x or X) denote scalars and bold-font letters (*e.g.*, $\boldsymbol{x}$ or $\boldsymbol{X}$) denote vectors. Random quantities are usually denoted by lower-case letters (*e.g.*, x for a scalar and $\boldsymbol{x}$ for a vector), whereas deterministic quantities are denoted by upper-case letters (*e.g.*, X for a scalar and $\boldsymbol{X}$ for a vector). The column convention is adopted for the representation of vectors.

Mathematical notation

$\mathbb{N}$, $\mathbb{N}^+$	set of natural numbers with and without zero ($\mathbb{N}^+ = \mathbb{N} \backslash \{0\}$)
$\mathbb{R}$	set of real numbers
$\mathbb{Z}$, $\mathbb{Z}^+$	set of integers with and without zero ($\mathbb{Z}^+ = \mathbb{Z} \backslash \{0\}$)
$\boldsymbol{X}^\top$	transpose of $\boldsymbol{X}$
$[x_1, \cdots, x_n]^\top$	representation of an n-dimensional column vector $\boldsymbol{x}$
$\lvert\mathcal{S}\rvert$ or $\operatorname{card}(\mathcal{S})$	cardinality of the set $\mathcal{S}$
$\operatorname{diag}(x_1, \cdots, x_n)$	diagonal matrix with diagonal entries $x_1, \cdots, x_n$
$\lvert x\rvert$	absolute value of scalar x
$f^{-1}(.)$	the inverse of a function $f(.)$
$\ln(x)$	natural logarithm of x
$\log_a(x)$	base-a logarithm of x. a is usually omitted if the based is not important
$A \triangleq B$	A is defined by B
$\sim$	distributed as
$\Pr(\mathcal{A})$	the probability of an event $\mathcal{A}$
$\mathbb{E}[x]$	the expected value of random variable x
$\operatorname{Var}(x)$	the variance of random variable x
$\boldsymbol{\Gamma}$	covariance matrix
$\mathcal{N}(\mu, \sigma^2)$	normal distribution of mean μ and variance σ^2
$\mathcal{B}(m, p)$	binomial distribution of m Bernoulli trials, each of which has a successful probability of p
$\operatorname{Pois}(\mu)$	Poisson distribution of rate μ

Commonly used symbols

$\mathcal{S}$	set of slices
s	slice index
N_s	number of users associated with slice s
$\mathcal{G} = (\mathcal{N}, \mathcal{E})$	infrastructure graph, $\mathcal{N}$ and $\mathcal{E}$ are respectively the sets of infrastructure nodes and links
$\mathcal{G}_s = (\mathcal{N}_s, \mathcal{E}_s)$	slice graph, $\mathcal{N}_s$ and $\mathcal{E}_s$ are respectively the sets of slice nodes and links
$a_n(i)$	available resources of type n on infrastructure node i
$a_{\mathrm{b}}(ij)$	available bandwidth on infrastructure link ij
$c_n(i)$	per-unit cost of resource of type n for node i
$c_{\mathrm{b}}(ij)$	per-unit cost of bandwidth for link ij
$r_{s,n}(v)$	resource demand of type n of a VNF instance v of an SFC of slice s
$r_{s,\mathrm{b}}(vw)$	bandwidth demand of a virtual link vw between two VNF instances v and w of an SFC of slice s
$R_{s,n}(v)$	resource demand of type n of a node v of slice s
$R_{s,\mathrm{b}}(vw)$	bandwidth demand of a slice link vw between two nodes v and w of slice s
$B_n(i)$	amount of resources of type n on infrastructure node i consumed by background services
$B_{\mathrm{b}}(ij)$	amount of bandwidth on infrastructure link ij consumed by background services
G_{tx}	antenna gain of the transmitter
G_{rx}	antenna gain of the receiver
P_{n}	noise power, given by $P_n = W_i N_0$
P_{tx}	transmitted signal power
P_{rx}	received signal power
PL	Path Loss
W	bandwidth (in Hz)

Chapter 1

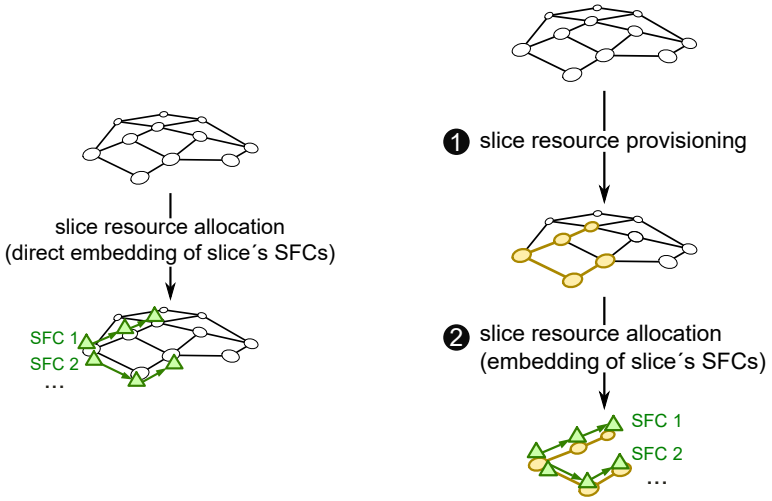
Introduction

1.1 Context

Beyond the connectivity access, the fifth generation (5G) mobile network offers operators unique opportunities to address new business models for consumers, enterprises, verticals, and third-party partners. 5G networks target different industry sectors to facilitate automation and monitoring. Dedicated services for vertical markets, *e.g.*, energy, e-health, smart city, connected cars, *etc.*, will be more easily deployed [Li et al., 2017]. The 5G architecture brings the required flexibility to support many services with different stringent requirements in terms of latency, throughput, and availability [Kaloxylos, 2018].

To increase flexibility and allow improved dynamicity, mobile networks are evolving towards systems consisting of virtual resources that can be instantiated and released on demand to timely meet demands of customers. To do that, technologies like Software-Defined Networking (SDN) and Network Function Virtualization (NFV) play an important role of increasing importance to provide such mobile network flexibility [Basta et al., 2014].

Leveraging SDN and NFV, network slicing appeared as a key enabling technology [5G Americas, 2016, IETF, 2017, Barakabitze et al., 2020]. Network slicing reduces overall equipment and management costs [Liang and Yu, 2014] by increasing flexibility in the way the network is operated [Rost et al., 2017]. Multiple dedicated end-to-end virtual networks or *slices* can be managed in parallel over a given infrastructure network. With network slicing, vertical markets can be addressed: Customers can manage their own applications by exploiting built-in network slices tailored to their needs [GSM Alliance, 2017]. As emphasized in [Weldon, 2015], the networking industry has begun a massive transformation toward network virtualization and cloud technology, as evidenced by the increasing number of outputs in filed patents, demonstrations, proofs-of-concept, field trials, and commercial deals. Innovative technologies like network slicing take an important role in realizing additional values for enterprises.



(a) Direct SFC embedding.

(b) Provisioned SFC embedding.

Figure 1.1: Illustration of (a) a direct SFC embedding, and (b) the proposed two-phase approach, where slice resource provisioning is performed before the SFCs deployment within the provisioned resources.

1.2 Slice Resource Provisioning

Contrary to previous best-effort approaches where various Service Function Chains (SFCs) of a slice are deployed sequentially in the infrastructure network, in this thesis, we propose solutions that provision resources for slices to accommodate slice resource demands, *i.e.*, *reserve* infrastructure resources *in advance* for the future deployment of slice SFCs.

Once provisioning is performed for a given slice, the SFCs of that slice are ensured to get the provisioned resources. This facilitates the satisfaction of the contracted service requirements with desired quality. In addition, as will be shown in the next chapters, our provisioning solutions yield a reduction of the computational resources needed to deploy the SFCs.

Figure 1.1 illustrates the SFC embedding approach considered in prior arts (Figure 1.1a). compared with the slice resource provisioning approach (Figure 1.1b). In Figure 1.1b), the SFC embedding (or deployment) process is split into two phases: first, resource provisioning is performed for a given slice and second, the SFCs of that slice are deployed within the provisioned resources resulting from the first phase.

Taking the perspective of an Infrastructure network Provider (InP), several provisioning frameworks are investigated to account for various use cases. We first propose a slice resource provisioning framework addressing multiple slice demands in terms of computing, memory, and wireless capacity. We further extend the provisioning framework by considering the situation where slices have to be deployed over different specific geographical areas. In such situation, the coverage as well as minimum per-user rate constraints have to be taken into account. Finally, slice

resource provisioning and admission control are combined to cope with (*i*) the uncertainties related to the slice resource demands (*e.g.*, fluctuations of user resource demands, time-varying number of users of the slice, *etc.*); and (*ii*) the dynamic nature of slice requests, *i.e.*, slice arrivals and departures.

The proposed approach is fully consistent with the 3GPP view of the management aspects of network slicing [3GPP, 2020]. The proposed slice resource provisioning methods may take place in the *network environment preparation* task of the *preparation* phase, see Figure 1.2. In this phase, the design and capacity planning of network slice, the on-boarding and evaluation of required network functions, and the provisioning of infrastructure resources have to be done before the creation and activation of network slice instances. The management aspects of network slicing will be discussed in detail in Chapter 2.

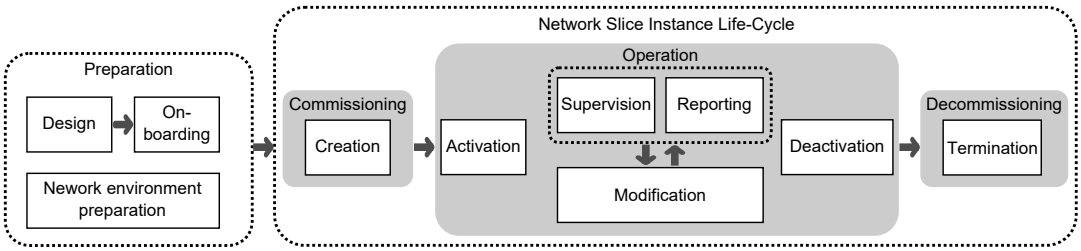


Figure 1.2: 3GPP view on network slicing managements aspects [3GPP, 2020].

1.3 Research Challenges

One of the main problems to solve in this context is provisioning each network slice with the right amount of physical resources (computing, storage, and network) to accommodate slice resource demands and satisfying predefined service requirements. The amount of resources provisioned for a slice depends on the services attached to it, their Quality of Service (QoS) requirements expressed in terms of latency, bandwidth, computing, and storage requirements. Such requirements depend on the demand for the consumption of services in the slice. It is expected that a limited number of types of network slices will co-exist driven by the business sustainability (*e.g.*, ultra-HD video, e-health, sensor network, intelligent transportation systems, gaming, tactile internet, *etc.*). Better identifying slice characteristics will facilitate resource provisioning. Having all that said, one defines the following Challenge 1.

> **Challenge 1.** Enough infrastructure resources should be provisioned to accommodate slice resource demands, so as the desired service requirements are satisfied. The amount of resources provisioned to a slice depends on the characteristics of the service it provides, its QoS requirements expressed, *e.g.*, in terms of bandwidth, computing, and storage requirements.

Many research challenges remain when network slicing incorporates the wireless part of legacy or 5G networks [Li et al., 2017, Kaloxylos, 2018], where the radio

access has to be considered. For instance, in [Chatterjee et al., 2018], the service characteristics required by an SP are: the minimum data rate, minimum rate coverage probability, the density of user equipments (UEs), and the geographical zone to be covered by the slice. In this thesis, we also tackle the problem of slice resource provisioning with some coverage constraints. Challenge 2 summarizes such issues.

> **Challenge 2.** In a wireless slicing context, *e.g.*, RAN slicing, some constraints related to the coverage area of the slice as well as the user location also have to be taken into account.

In the survey [Barakabitze et al., 2020] on 5G network slicing, the authors provide a taxonomy of network slicing, architectures and future challenges. One of the open questions is how to meet the slice requirements of different verticals, where multiple network segments including the radio access, transport, and core networks, have to be considered. Infrastructure networks on which slices are operated must support high-quality services with increasing resource consumption (video streaming, telepresence, augmented reality, remote vehicle operation, gaming, *etc.*). Moreover, the number of users of each slice, their location (usually difficult to predict [Richart et al., 2016]), and resource demands may fluctuate with time. These uncertainties may impact significantly the resources consumed by each slice and make the slice resource provisioning problem more challenging. Enough infrastructure resources should be dedicated to a given slice to ensure an appropriate QoS despite the uncertainties in the number of slice users and their demands. Over-provisioning should also be avoided, to limit the infrastructure leasing costs and leave resources to concurrent slices. This leads to Challenge 3.

> **Challenge 3.** An efficient slice resource provisioning mechanism should be robust against the uncertainties related to slice resource demands. Moreover, the proposed provisioning approach has to be performed so as to limit its impact on low-priority background services, which may co-exist with slices in the infrastructure network.

In addition to the uncertainty issue, it is also necessary to account the dynamic nature of slice provisioning requests: Slice request arrive at different time instants, with various activation delays, life durations, and time-variant resource demands. These parameters significantly impact the aggregate resource demands of network slices. The variety of services supported by slices induces very different QoS requirements [Li et al., 2018]. In traditional slice resource allocation approaches [Huin et al., 2017, Wang et al., 2017, Su et al., 2019, Barakabitze et al., 2020], resources are allocated to slices just before its required activation time. With such a *just-in-time* slice management, it is difficult to guarantee the availability of enough infrastructure resources at the deployment time and during the life-time of a slice. In such case, slice demands may be rejected. Therefore, a novel slice resource provisioning approach should be introduced, providing *anticipated* slice admission control.

Slices are admitted, possibly largely before their activation time when enough infrastructure resources are available to meet their QoS requirements. This leads to Challenge 4.

> **Challenge 4.** Slice provisioning requests should be processed in an anticipated way, largely before their activation time, to guarantee the availability of infrastructure resources at the deployment time and during the life-time of the slices. The resulting slice admission control mechanism should take into account the dynamic nature of slice provisioning requests and the priority level of slice requests.

1.4 Thesis Description

This thesis is a contribution to network slicing in 5G communication systems and beyond. The main contribution is to propose efficient slice resource provisioning methods that can adapt to different situations raised by the above-posed challenges.

1.4.1 Thesis Outline and Contributions

In Part I, some background on thenetwork slicing paradigm and related work are presented. Assumptions and hypotheses widely used throughout the thesis are also described. The main contributions of this thesis follow in Parts II and III. A section highlighting related works and the main contribution is provided at the beginning of each chapter.

Part I (*Background and Assumptions*) introduces some background on the paradigm of network slicing and highlight some network slicing-related research directions with relevant prior arts.

- **Chapter 2** (*Network Slicing in 5G*) presents a brief history of network slicing, highlights the main enabling technologies of network slicing, *e.g.*, software-defined networking and network function virtualization. This chapter also describes a conceptual architecture of a typical network slicing system, and discusses different aspects such as the life-cycle management of network slices;
- **Chapter 3** (*Related Works*) highlights some studies related to various aspects of network virtualization and network slicing. Specifically, we present a literature review on (*i*) SFC embedding and slice resource allocation, (*ii*) slice resource allocation with coverage constraints, (*iii*) uncertainty-aware slice resource allocation, and (*iv*) dynamic slice resource allocation;
- **Chapter 4** (*Hypotheses and Assumptions*) presents the notations, assumptions and hypotheses that are used throughout the thesis. A typical network slicing system is described, with all the involved entities. The relation

and interactions between these entities, *e.g.*, the exchange of the characteristics of user demand, slice demand, and the dedicated service are also detailed.

Part II (*Resource Provisioning for Deterministic Demands*) proposes novel methods of slice resource provisioning for deterministic demands.

Chapter 5 (*Resource Provisioning for the Core Network*) addresses Challenge 1, in which the problem of slice resource provisioning in the core network is considered. In this chapter, the available resources in the infrastructure and the slice resource demands are considered to be deterministic. We discuss how a slice resource demand is formed and the way we address the problem of slice resource provisioning;

Chapter 6 (*Coverage-Constrained Resource Provisioning*) addresses Challenge 2. It extends the study in Chapter 5 by considering the problem of joint core and RAN network resource provisioning. To that end, the slice resource provisioning consists in finding (*i*) a set of Base Stations (BS) that provides sufficient radio resources to mobile users so as to satisfy coverage constraints; (*ii*) the placement of the VNFs on the data center nodes; and (*iii*) the routing of data flows between the VNFs, while respecting the structure of SFCs and optimizing a given objective (*e.g.*, minimizing the infrastructure and software costs).

Part III (*Resource Provisioning for Slice Requests with Uncertainties*) presents some methods of slice resource provisioning with uncertainties and dynamic demands.

Chapter 7 (*Uncertainty-Aware Resource Provisioning*) investigates a method to provision infrastructure resources for network slices, while being robust to a partly unknown number of users of the slice leading to a partly unknown usage of the slice resources. Moreover, since some parts of the infrastructure network on which slices should be deployed are often already employed by low-priority background services, the provisioning approach will be performed so as to limit its impact on these services. The approach proposed in this chapter is an answer to Challenge 3;

Chapter 8 (*Admission Control and Resource Provisioning for Prioritized Slice Requests with Uncertainties*) addresses Challenge 4. It extends the study introduced in Chapter 7 by considering the resource provisioning for concurrent slices. It accounts for the dynamicity of slice requests, which refers to the fact that (*i*) the resource demand of these slices may evolve during their lifetime, (*ii*) the requests are submitted somewhat in advance, and (*iii*) different slice requests may belong to different priority classes. Several reduced-complexity provisioning strategies are considered to solve the problem of slice resource provisioning, accounting

for the life-cycles of slice requests (*i.e.*, arrival, activation, and departure time), while being robust against the uncertainties of slice resource demands. In addition, the proposed methods account for slice priority level and provide a differentiated acceptance rate for prioritized slice requests.

Chapter 9 (*Conclusion and Perspective*) draws some conclusions and perspectives. This chapter discusses some aspects that deserve more developments and some potential further research directions.

The characteristics of variables used in each chapter are summarized in Table 1.1.

Table 1.1: Considered characteristic of variables in each chapter.

	Part II		Part III	
Characteristic	Chap. 5	Chap. 6	Chap. 7	Chap. 8
Deterministic slice resource demands	✓	✓	–	–
Uncertain slice resource demands	–	–	✓	✓
Prioritized slice requests with uncertainties	–	–	–	✓

1.4.2 Publications

This thesis is based on the following publications:

Journal Papers

(J_1) [Luu et al., 2021a] Q.-T. Luu, S. Kerboeuf, and M. Kieffer, "Admission Control and Resource Provisioning for Prioritized Slice Requests with Uncertainties," *submitted to IEEE Transactions on Network and Service Management*, 2021.

(J_2) [Luu et al., 2021c] Q.-T. Luu, S. Kerboeuf, and M. Kieffer, "Uncertainty-Aware Resource Provisioning for Network Slicing," in *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 79-93, Mar. 2021.

(J_3) [Luu et al., 2020a] Q.-T. Luu, M. Kieffer, A. Mouradian, and S. Kerboeuf, "Coverage-Aware Resource Provisioning Method for Network Slicing," in *IEEE/ ACM Transactions on Networking*, vol. 28, no. 6, pp. 2393-2406, Dec. 2020

Conference/Workshop Papers

(C_1) [Luu et al., 2021b] Q.-T. Luu, S. Kerboeuf, and M. Kieffer, "Foresighted Resource Provisioning for Network Slicing," *IEEE International Conference on High Performance Switching and Routing (HPSR)*, Paris, June 2021, pp. 1-8.

(C_2) [Luu et al., 2020b] Q.-T. Luu, S. Kerboeuf, A. Mouradian, and M. Kieffer, "Radio Resource Provisioning for Network Slicing with Coverage Constraints," in *Proc. IEEE International Conference on Communications (ICC)*, Dublin, Ireland, June, 2020, pp. 1-6.

(C_3) [Luu et al., 2018] Q.-T. Luu, M. Kieffer, and A. Mouradian, and S. Kerboeuf, "Aggregated Resource Provisioning for Network Slices," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, Dec. 2018, pp. 1-6.

Project Deliverable

(Pd_1) [Perrot et al., 2020] N. Perrot, M. Antonia, S. Kerboeuf, Q.-T. Luu, *et al.*, "Virtual Network Orchestration Framework and Algorithms," *ANR MAESTRO-5G Project Deliverable D3.1*, 2020.

Demonstrations

(D_1) [Orlandi et al., 2018] B. Orlandi, S. Kerboeuf, F. Faucheux, J.-L. Lafragette, A. Boubendir, and Q.-T. Luu, "Network Slicing Made Easy! From Graph-based Design to Automated Deployment of Network Slices in 5G," *Nokia 5G Smart Campus Event*, Nozay, 2018 (in partnership with Orange Labs).

Talks

(T_1) [Luu et al., 2020c] Q.-T. Luu, S. Kerboeuf, A. Mouradian, and M. Kieffer, "Resource Provisioning for Network Slices with Coverage Constraints," *ANR MAESTRO-5G Workshop on Orchestration of 5G Networks and Beyond*, CentraleSupélec, Gif-sur-Yvette, Dec. 2020.

Patents

(Pa_1) [Kerboeuf et al., 2018] S. Kerboeuf, Q.-T. Luu, M. Kieffer, and A. Mouradian, Method and Apparatus for Mapping Network Slices Onto Network Infrastructures With SLA Guarantee, *WIPO Patent No. WO2020114608A1* (filed on Dec. 07, 2018 by Nokia Solutions and Networks).

Part I

Background and Assumptions

Chapter 2

Network Slicing in 5G

This chapter provides an overview of the network slicing paradigm in the 5G communication systems. We first provide a general introduction of the 5G systems and its main use cases. The concept, history, enabling technologies, principles, conceptual architecture, and the life-cycle management of network slicing are then presented.

2.1 5G Systems

The fifth generation mobile network is envisioned as a novel paradigm of smart world that can empower machine-to-machine and machine-to-human type applications for making our life safer and easier. Diverse industrial sectors are targeted, with the aim of facilitating automation and monitoring processes. With the help of 5G, dedicated applications such as autonomous vehicles, augmented reality, smart industries, *etc.*, will be more easily deployed [3GPP, 2019].

Compared to the former mobile network systems, 5G aims to provide services with higher capacity, higher speed, and lower latency. Table 2.1 summarizes the main characteristic requirements of 5G [Yang et al., 2018].

Table 2.1: Characteristic requirements of 5G [Yang et al., 2018].

Characteristics	*Value*
Peak data rate	$>$ 10 Gbps
User experienced data rate	$>$ 0.1 Gbps
Connection density	million connections/km^2
Service density	million connections/km^2
End-to-end latency	millisecond order

The use cases of 5G networks can be mainly divided into three types: enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communication (URLLC), and massive Machine Type Communication (mMTC) as shown in Figure 2.1 [IMT, 2015].

eMBB: Mobile broadband addresses the use cases for personal accesses to multimedia services. The increase of mobile broadband demand leads to the advent of eMBB, which outperforms the existing mobile broadband technologies in terms of QoS and seamless user experience. Different usage scenarios with different requirements are covered by eMBB, including wide-area coverage and hotspot. For wide-area coverage case, it is desired to have a large and seamless coverage supporting high user mobility, with sufficiently high data rates compared to those of the existing MBB technologies. The hotspot case, on the other hand, supports high user density and high traffic load. This use case requires a higher demand for data rate, but a lower mobility support than the wide-area coverage use case [Kazmi et al., 2019].

URLLC: This use case requires stringent capabilities, especially in terms of latency (order of milliseconds), availability, and throughput. Numerous specific examples can be listed including remote medical surgery, autonomous driving, industrial processes in manufactures, *etc*.

mMTC: This use case is characterized by a massive number of connected devices that typically exchange a relatively low data traffic. This use case does not require a strict requirement on latency.

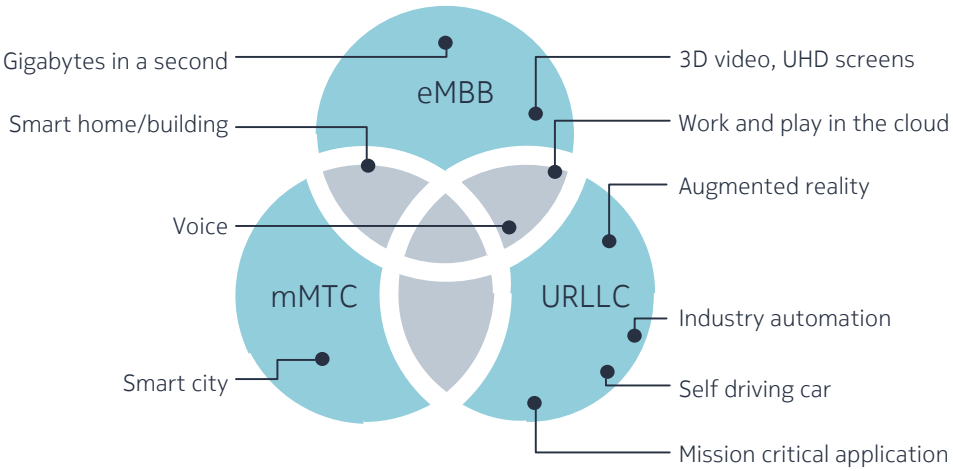


Figure 2.1: 5G use cases [IMT, 2015].

Figure 2.2 illustrates the IMT-2020 recommendation for supported capabilities of 5G use cases. Eight main parameters are considered in this recommendation, including peak data rate, user experienced data rate, latency, mobility, connection density, energy efficiency, spectrum efficiency, and area traffic capacity [IMT, 2015]. It can be seen that, each of these capabilities has different importance in different use case. For instance, the importance of connection density and network energy efficiency in the mMTC use case is highest compared to other capabilities.

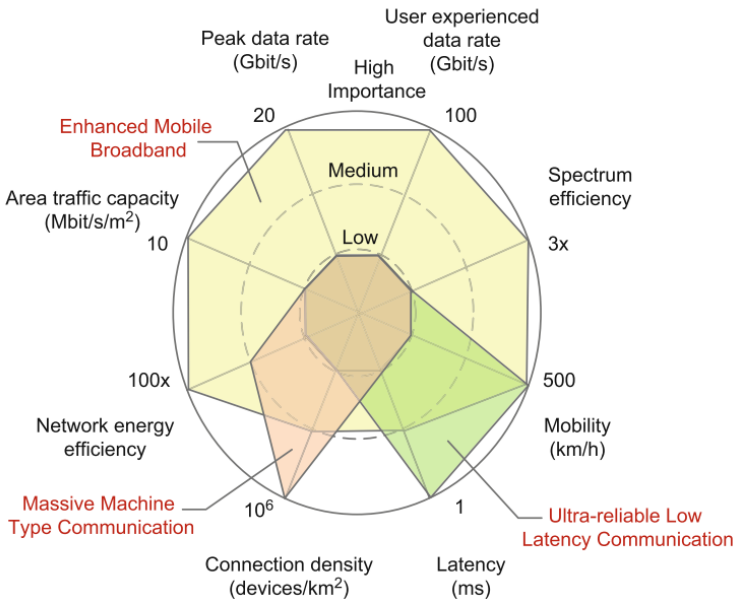


Figure 2.2: IMT-2020 recommendation for supported capabilities of 5G use cases [IMT, 2015, Kazmi et al., 2019].

2.2 Network Slicing: Concept and History

Several definitions of network slicing have been introduced by different research bodies, *e.g.*, 3GPP [3GPP, 2016a], GSMA [GSM Alliance, 2017], *etc*. These definitions refer to the same concept for network slices: a network slice is an end-to-end logical network running on a common underlying (physical or virtual) infrastructure, mutually isolated with other slices, and with independent control and management [Galis and Makhijani, 2018].

Several organizations are currently active in the different activities (*e.g.*, standardization, telecommunications providers, *etc*.) of network slicing. Figure 2.3 provides a snapshot of various global organizations involving in different network slicing activities [GSMA, 2018]. In the bottom of Figure 2.3, we have the Standardization Developing Organizations such as 3GPP, IEEE, ETSI, IETF, which involve in providing the unified standards for network slicing. The Telecom Industry Organizations such as GSMA, NGMN are presented in the middle. These organizations contribute to various pertinent technologies to realize network slicing. Finally, on top of Figure 2.3, one can find the Vertical Industry Organizations such as 5GAA and the Industrial Internet Consortium. These organizations concentrate on different industrial network slicing specifications. For instance, the 5GAA alliance is devoted to the realization of network slicing in autonomous vehicle industry, while the ZVEI contributes to network slicing applied to the electronics industry.

The idea of resource slicing can be traced back to the 1960s, with the introduction of the concept of virtualization, when the first operating system (CP-40) developed by IBM [Lindquist et al., 1966]. CP-40 supported time-sharing and virtual memory and allowed multiple users to simultaneously work on a complete

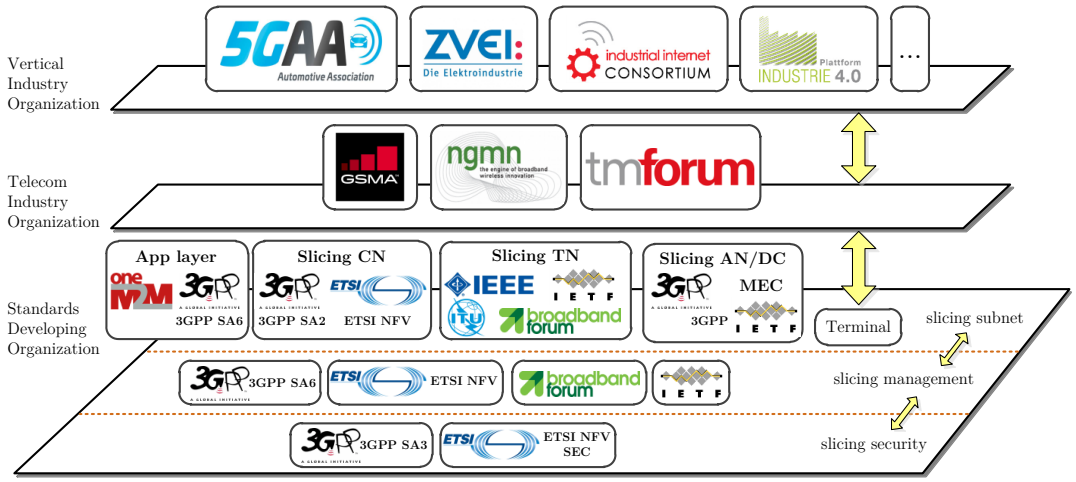


Figure 2.3: Network slicing relevant industry groups and SDOs [GSMA, 2018].

set of hardware and software. Virtualization technologies then continued to offer unprecedented advantages to communication systems. For instance, datacenters today use virtualization to make abstraction of of the physical resources (*e.g.*, CPUs, memory, file storage) and create aggregate logical resource pools. Virtualization of datacenters facilitates the management and allows administrators to share hardware resources across a wide-range network. This leads to a lower operational cost and higher resource utilization efficiency and data center functionality.

In the late 80s, the concept of "overlay networks," commonly known as an early form of network slicing [Srinivasan et al., 1989], was presented. Overlay networks provided the first form of network slicing since heterogeneous network resources were combined together to create virtual networks over a common infrastructure. Nevertheless, this kind of network architecture lacked a mechanism that could enable its programmability.

Several research efforts in the early 2000s, *e.g.*, PlanetLab [Chun et al., 2003] and GENI (Global Environment for Network Innovations) [Elliott, 2008], have been conducted on designing a testbed to evaluate and verify new network protocols. PlanetLab introduced a virtualization framework allowing multiple users to program network functions so as isolated and application-specific slices could be obtained.

The advent of SDN in 2008 [McKeown et al., 2008] and NFV in 2012 [ETSI, 2012] further extended the programmability capability of overlay networks and made the network to be agile and flexibly controlled. Finally, the advance of virtual machine (VM) and recent technologies such as dockers/containers facilitated the realization of network slicing. While VMs may provide full logical isolation for the operation of VNFs within a network slice, containers offers a flexible functionality for network slicing, thus can efficiently support 5G network slices with highly mobile users [Barakabitze et al., 2020].

Table 2.2 summarizes some important events related to the advent of network slicing.

Table 2.2: The road to network slicing.

Period	Concept	Description
1960s-1980s	Virtualization	Virtualization on a machine supports time-sharing and virtual memory on personal computers to allow simultaneous users working on the same machine [Lindquist et al., 1966].
Late 1980s	Overlay networks	Early form of network slicing: Inter-connected nodes over logical links [Srinivasan et al., 1989]
2000s	Network testbeds	Evaluation and verification testbeds based on overlay networks for new network protocols: PlanetLab [Chun et al., 2003], GENI [Elliott, 2008]. These testbeds aimed to run multiple experiments at the same time.
2008	SDN	OpenFlow v1: first attempt to decouple control and data plane using open-source software [McKeown et al., 2008]
2012	NFV	Decoupling network functions from physical network hardware [ETSI, 2012]
2015	Network slicing	Introduced in the 5G white paper of NGMN Alliance [NGMN Alliance, 2015]
2016		Support of network slicing, 3GPP technical report TR23.799 (Release 14) [3GPP, 2016b]
...		Open Networking Foundation (ONF) technical recommendation TR-526 "Applying SDN Architecture to Network Slicing" [ONF, 2016]

2.3 Network Slicing Enablers

In this section, the fundamental enabling technologies that shape the network slicing paradigm, Software-Defined Networking and Network Function Virtualization, will be described.

2.3.1 Software-Defined Networking

SDN technology enables the realization of fully configurable and scalable network slices [Afolabi et al., 2018, Nakao et al., 2017]. A reference architecture of SDN is defined by the Open Networking Foundation (ONF) technical recommendation TR-502 [ONF, 2014], in which SDN is based on the three following cornerstones:

(1) decoupling control plane from data plane (traffic forwarding and processing);

(2) (logically) centralized control;

(3) network service programmability.

The architecture of SDN allows a common infrastructure network to efficiently support numerous client network instances, which are customized, tailored, and optimized for services having diversified requirements. In Figure 2.4 one depicts the basic components of SDN [ONF, 2014]. A typical SDN architecture has three layers: infrastructure layer, control layer, and application layer. These layers are also referred to as the data, controller, and application planes.

The *infrastructure layer* (data plane) consists of network elements, which expose their capabilities toward the control layer (controller plane) via the Data-Controller Plane Interface (D-CPI, also called southbound interface) [ONF, 2014];

The *application layer* (application plane) communicates the network requirements toward the controller plane via the Application-Controller Plane Interface (A-CPI, also called the northbound interface;

The *control layer* (controller plane), where the SDN controller locates, translates the application requirements and employs low-level control over the network elements, while providing relevant information up to the SDN applications. An SDN controller is essentially the “brain" of a typical software-define network. It acts as a strategic point, managing flow control to the network elements (*e.g.*, switches and routers) in the infrastructure layer and to the applications located in the application layer to deploy intelligent networks.

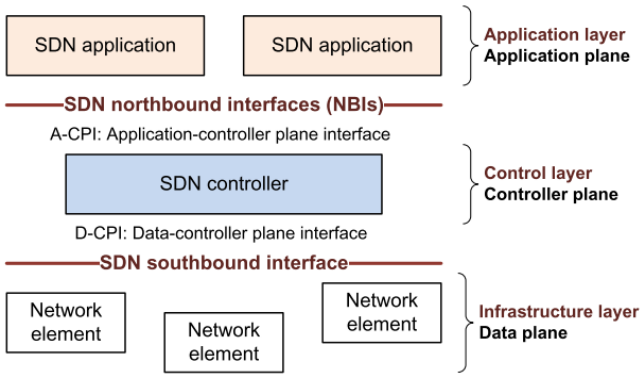


Figure 2.4: Fundamental components of SDN [ONF, 2014].

2.3.2 Network Function Virtualization

The concept of NFV has officially been introduced in 2012 by numerous world’s leading telecommunication service providers [ETSI, 2012]. NFV allows networks to be agile, flexibly controlled, and capable to automatically respond to the traffic and service requirements.

A typical NFV framework defined by ETSI consists of the following main components [ETSI, 2014], see also Figure 2.5.

(1) **VNFs** (Virtualized Network Functions) are *softwarized* network functions that can be deployed on an NFVI (NFV Infrastructure);

(2) **NFVI** (NFV Infrastructure) is the collection of software and hardware components upon which NFV services are deployed. NFVI consists of

   (a) physical hardware, *e.g.*, servers, switches, routers;

   (b) virtualization layer: in charge of abstracting hardware resources and decoupling VNFs from the underlying hardware on which they are running;

   (c) virtual infrastructure: including virtualized resources, *e.g.*, virtual compute (virtual machines, containers), virtual storage, and virtual (overlay) networks;

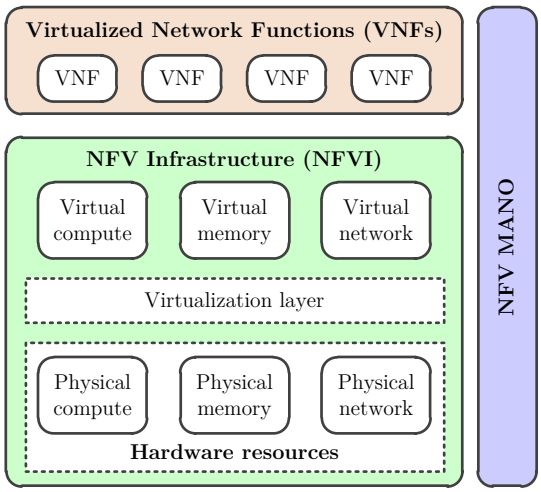


Figure 2.5: Three main components of NFV [ETSI, 2014].

(3) **NFV-MANO** (NFV Management and Orchestration) architectural framework is the collection of all functional blocks, data repositories used by these blocks, and reference points and interfaces through which these functional blocks exchange information for the purpose of managing and orchestrating NFVI and VNFs.

Fundamentally, the SDN and NFV technologies are complementary but increasingly co-dependent. While the former provides the means to dynamically control the network and providing networks as a service, the latter offers the capability to manage and orchestrate the virtualization of resources for the provisioning of network functions and their composition into higher-layer network services [ETSI, 2020].

2.4 Network Slicing Principles

In what follows, we describe the main principles of network slicing

Automation of network operation Automation enables a dynamic slice life-cycle management, in which on-demand configuration of slice instances (*e.g.*, instantiating, activating, deactivating). It also enables the optimization the use of network resources by reconfiguring (*e.g.*, VNF auto-scaling, migrating) [Barakabitze et al., 2020];

Slice isolation Isolation is a fundamental principle of network slicing that ensures the simultaneous coexistence of multiple slices running on a common physical infrastructure. This concept is to avoid any interference of one slice on the other slices. In addition, isolation enhances the network slice architecture in the security aspect of network slicing: any cyber-attacks or technical failures, if it occurs on one slice, would have no or limited impact on the life-cycle of the other slices that are

simultaneously running. Different degrees of isolation are required depending on the type of service (eMBB, URLLC, or mMTC) [3GPP, 2020]. For instance, URLLC slices have stringent requirements for the isolation of radio spectrum, due to latency and security reasons that can only be guaranteed when employing a hard spectrum slicing [Afolabi et al., 2018].

Slice customization Customization guarantees an efficient utilization of resources allocated to an SP so as the related service requirements are met. Using SDN, slice customization can be realized at different levels [Afolabi et al., 2018, Barakabitze et al., 2020]: (*i*) in all layers of the abstracted network topology, (*ii*) on the decoupled data plane and control plane using NFV that provides service-customized network functions, data forwarding mechanism, and programmable policies. In the latter level, value-added services can be enabled with the help of, for example, artificial intelligence and data-driven methods.

Elasticity of network resources The elasticity of network resources allows the slices to adapt to the time-variant of service characteristics [Afolabi et al., 2018], *e.g.*, network condition, number of associated users, so as the contracted Service Level Agreement (SLA) is assured. Elasticity can be realized through: VNF scaling, re-provisioning of allocated network resources, *etc*. Nevertheless, such reconfiguration operation may impact the concurrent slices and other background traffics, negotiations between different entities (InPs, MNOs, and SPs) are therefore of necessity.

Programmability This feature allows third-parties to realize the provisioning of services by controlling the allocated slice resources via open APIs. This facilitates the above-mentioned elasticity of network resources and on-demand service-tailored customization [Barakabitze et al., 2020].

Hierarchical abstraction In network slicing, an additional abstraction layer is introduced by the physical or logical creation of decoupled groups of network resources and network functions. This property enriches the exploitation of network slices. For example, an SP, who acquired a network slice from an MNO, can enable verticals to partially or fully use the leased resources as part of their services [Afolabi et al., 2018, Barakabitze et al., 2020].

2.5 Network Slicing Conceptual Architecture

There are currently various proposals of architecture for network slicing in 5G, for example, the proposal of NGMN [NGMN Alliance, 2016], 5GPP [5GPPP, 2017], and Nokia [Nokia, 2016] It is possible to define a generic conceptual framework representing those different architectural proposals. A high-level architecture of a typical network slicing system is depicted in Figure 2.6 [NGMN Alliance, 2016],

which is composed of three main layers, namely the Service Instance layer, the Network Slice Instance layer, and the Resource layer.

The *Service Instance* layer includes the service instances that represent the dedicated end-user services supported by the network slices.

The *Network Slice Instance* layer is where the network slice instances take place. Each network slice instance consists of a set of VNFs and the allocated resources, forming a logical network to meet certain characteristics required by the service instances. Examples of characteristics are ultra reliability or ultra low latency. A network slice instance can be formed by multiple sub-network slice instances. Each sub-network slice instance could be shared among different network slice instances.

The *Resource* layer is where the physical resources (*e.g.*, storage, computing), logical resources (*e.g.*, partitions of physical resources), and the network functions (*e.g.*, functionalities of telecom nodes such as gateway or remote radio head) locate.

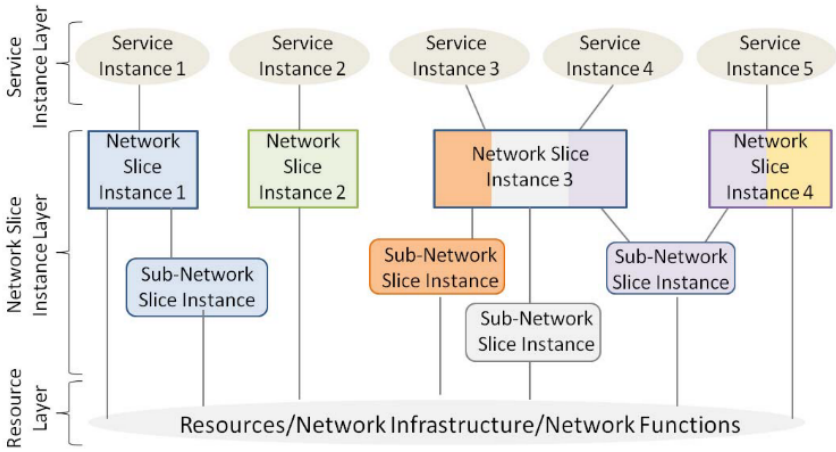


Figure 2.6: Network slicing conceptual architecture introduced by NGMN [NGMN Alliance, 2016].

2.6 Network Slice Life-Cycle Management

As discussed in Chapter 1, the process of slice resource provisioning is fully consistent with the 3GPP views of the management aspects of network slicing [3GPP, 2020], see Figure 2.7. In Figure 2.7, the management of network slicing is split into four phases: preparation, commissioning, operation, and decommissioning. The life-cycle of a given network slice instance begins from the *commissioning* phase, where it is created, and ends at the *decommissioning* phase, where it is terminated.

The *preparation* phase includes (*i*) the design of network slice template, (*ii*) the planning of network slice capacity, (*iii*) the on-boarding and evaluation of network slice requirements, and finally, (*iv*) the preparation of network environment and essential preparation operations, where the process of slice resource provisioning takes place.

The *commissioning* phase is where the network slice instances are created with

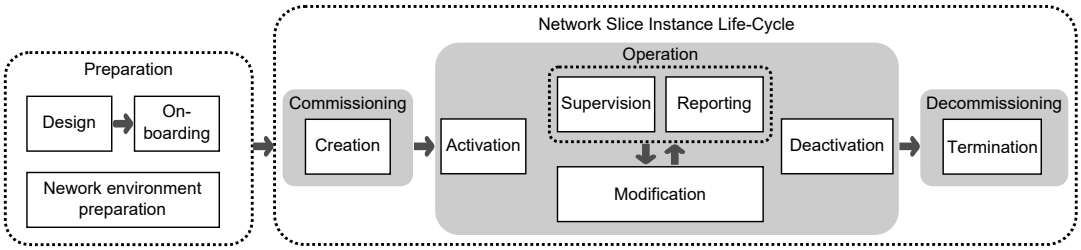


Figure 2.7: 3GPP view on network slicing managements aspects [3GPP, 2020].

the prepared template and provisioned network resources that have already been determined from the *preparation* phase. During the creation of network slice instance, the provisioned resources are allocated and well configured so as the network slice requirements are satisfied.

The *operation* phase is where a network slice instance operates to provide its dedicated service. This phase involves the activation, the operation monitoring, the modification, and the deactivation of network slice instances.

- the activation is where the instance is created and is made ready to support the dedicated service;
- the operation monitoring of the network slice instance is where the supervision and performance reporting take place
- the modification includes any reconfiguration actions of the network slice instances, which could be the changes in the allocated network resources or in the topology of the network slice instance. The modification of network slice instance could be triggered as a result from the supervision and performance reporting processes (*e.g.*, when a certain KPI or QoS is not satisfied), or if a new network slice requirements are reported;
- the deactivation is where the network slice instances are made inactive and stop providing their dedicated services.

The *decommissioning* phase involves the complete elimination of the network slice instances from the system. After this phase, the network slice instances do not exist anymore.

2.7 Conclusion

In this chapter, we provided some background on the network slicing paradigm. A short history on the advent of network slicing was first introduced, then followed by the description of the two main enabling technologies that shape the form of network slicing: the software-defined networking and the network function virtualization technology. We finally presented the principles, the generic conceptual architecture, and the life-cycle management aspects of network slicing.

Chapter 3

Related Works

In this chapter, some prior research related to the problem of assigning physical resources to virtual networks will be presented. Specifically, we review, in what follows, some studies on the topic of (*i*) SFC embedding and slice resource allocation (Section 3.1), (*ii*) slice resource allocation with coverage constraints (Section 3.2), (*iii*) uncertainty-aware slice resource allocation (Section 3.3), and (*iv*) dynamic slice resource allocation (Section 3.4).

3.1 SFC Resource Allocation

A large and growing body of literature has investigated the SFC resource allocation problem. Since a slice can be seen as a collection of SFCs, allocating resources for a given slice means allocating resources for *all* SFCs constituting that slice.

The SFC resource allocation problem is usually represented as a mapping of elements (virtual nodes or VNFs and virtual links) of SFCs onto the physical infrastructure. The mapped infrastructure nodes and links must satisfy some specific requirements of the virtual nodes and virtual links of the SFCs, *e.g.*, in terms of resource demands, latency, or availability. In the literature, the problem of SFC resource allocation is also called the Virtual Network Embedding, VNF placement, or SFC embedding.

In [Riggio et al., 2016, Vizarreta et al., 2017], computing, memory, and aggregate wireless resource demands of SFCs are considered. The minimization of the SFC embedding cost is formulated either as an *Integer Linear Programming* (ILP) [Cohen et al., 2015, Riera et al., 2016, Vizarreta et al., 2017] or as a *Mixed Integer Linear Programming* (MILP) problem [Chowdhury et al., 2012, Kang et al., 2017], which are known to be NP-hard [Fischer et al., 2013]. In [Tajiki et al., 2018], the VNF placement problem is expressed as an *Integer Quadratic Programming* (IQP) problem with a set of energy consumption constraints, and then is transformed to a more amenable linear form.

To address the high computational complexity resulting from the ILPs or MILPs, various heuristics have been proposed, see, *e.g.*, [Riggio et al., 2016, Vizarreta et al., 2017, Cohen et al., 2015]. For example, [Riggio et al., 2016] introduced a heuristic

based on the search of shortest paths to sequentially embed the SFCs. In [Vizarreta et al., 2017], the candidate infrastructure nodes are sorted to find the best node, in terms of deployment cost, to host a given VNF. Its neighbors are then considered as candidates to deploy the next VNF.

The *Column Generation* (CG) technique has been widely studied to solve large ILP problems [Huin et al., 2017]. With CG, the original ILP is decomposed into a *Master Problem* (MP) and a *Pricing Problem* (PP). The MP is the original problem where only a subset of variables is considered. The PP is a new problem created to identify a new variable, *i.e.*, a column, to add to the MP to improve the current solution. In [Huin et al., 2017] or [Liu et al., 2017], CG has been used to relax ILP-based SFC embedding or reconfiguration problems. Specifically, in [Huin et al., 2017], the SFC embedding problem is addressed. Only core capacity and bandwidth resources for infrastructure nodes and links are considered.

In [Mechtri et al., 2016], the join VNF and virtual link placement is formulated as a *Weighted Graph Matching Problem* (WGMP), where the SFC graph and the infrastructure graph are modeled as weighted graphs, on which each node and each link have their own weight corresponding to their required resource (for the SFC graph), or their available resource (for the infrastructure graph). An *eigendecomposition*-based method is then proposed to solve the WGMP problem, whose aim is to find, with a reduced complexity, the optimum matching between the SFC graph and the infrastructure graph. In [Huin et al., 2017, Liu et al., 2017] and [Mechtri et al., 2016], a unique type of resource is considered at infrastructure nodes (processing) and at links (bandwidth).

The resource allocation problem among competing slices in a heterogeneous cloud infrastructure is addressed in [Halabian, 2019]. Slice resource demands are aggregated in a vector of VNF resource demands in the slice. These demands are multiplied by a coefficient linked to the number of services to be processed per time unit. The considered types of resource are CPU, memory, and bandwidth. The resource allocation among multiple slices is performed considering two different approaches. The first involves a centralized convex optimization problem, whose objective is to maximize the total slice utility. Nevertheless, as pointed out in [Halabian, 2019], such centralized solution lacks of scalability, is not robust to a failure of the central optimizer, and is prone to non-collaborative slice providers which may harm the system. For these reasons, a distributed method based on game theory is considered to improve robustness and scalability. Optimization is performed in a decentralized way among the data centers and slice providers. The results provided by all entities determine the final resource allocation for all slices. Nevertheless, the placement of VNFs in data centers is predetermined by the MNO and again, wireless resources are not considered.

3.2 Slice Resource Allocation with Coverage Constraints

The design of efficient allocation mechanisms for virtualized radio resources has been recently addressed in [Chatterjee et al., 2018]. This paper aims at minimizing the leasing cost of BSs so as to meet SP demands, while providing, with a given probability, a minimum data rate for any user located in their coverage area. The rate constraint is expressed as a linear function of the BS load (number of users served by the BS), of the distance from users to the nearest BS, and of the downlink interference. This linear approximation, however, requires some assumptions. For instance, a user of an SP is assumed to be served by its nearest BS among the set of BSs allocated to the SP. This reduces somehow the potentiality of achieving the optimal sharing of the radio resource.

In [Teague et al., 2019], a heterogeneous spatial user density is considered, and the joint BS selection and adaptive slicing are formulated as a two-stage stochastic optimization problem. The first stage aims at defining the set of BSs to activate. The second stage aims at allocating wireless resources of the BSs to each point of the region to be covered by the SP. Several random realizations of user locations are generated to get a reduced-complexity deterministic optimization problem. A genetic algorithm is then used for the optimization.

In [Lee et al., 2016], a network slicing framework for multi-tenant heterogeneous cloud radio access network is introduced. The sharing of radio resources in terms of data rate is considered, with some constraints related to the fronthaul capacity, the transmission power budget of RRHs, or the tolerable interference threshold of an RRH on a sub-channel. Slicing is formulated as a weighted throughput maximization problem, which aims at maximizing the total rate obtained by users connected to given RRHs on given sub-channels. Nevertheless, the proposed framework does not consider computing and memory resources associated to the processing within the BBUs. Such resources are assumed to be properly scaled so as to support the required service rate. Moreover, the proposed framework addresses only downlink data services.

A game theory-based distributed algorithm is proposed to solve the problem of wireless network slicing in [D'Oro et al., 2018]. The proposed algorithm accounts for the limited availability of wireless resources and considers different aspects such as congestion, deployment costs, and the RRH-user distance. The coverage area of RRH is considered, but the possible coverage constraints required by different slices are not taken into account.

3.3 Uncertainty-Aware Slice Resource Allocation

Several works on uncertainty-aware resource allocation for virtual networks can be found in the literature.

In many conventional approaches, enough network resources are allocated to make a service available to all users, all the time. In reality, many applications such

as e-mail and instant messaging do not require such exclusive service. To address this problem, in [Trinh et al., 2011], flexible service availability levels are defined. These flexible levels lead to cost savings for the infrastructure provider that can offer overbooked resources for users accepting a service with possibly degraded availability. In the context of network slicing, SPs can benefit from such an approach by providing services with reduced availability or degraded quality to some users ready to accept these conditions. Nevertheless, to evaluate the incidence on the QoS of such under-provisioning mechanism, it is necessary to introduce models of the number of users of a service and of the resource consumption. Such models have not been considered in [Trinh et al., 2011].

A worst-case allocation at peak traffic is considered in [Huin et al., 2017, Wang et al., 2017]. Nevertheless, this infrastructure resource overbooking is costly and most of the time unnecessary, as all individual slice resource demands are very unlikely peaking simultaneously. In [Coniglio et al., 2015], the virtual network embedding problem is solved considering uncertain traffic demands. An MILP formulation is considered, where some of the constraints are required to be satisfied with high probability. In [Mireslami et al., 2019], the total deployment costs for cloud computing applications are minimized, while satisfying some QoS constraints. To cope with the uncertain nature of the demands, a stochastic optimization approach is adopted by modeling user demands as random variables obeying normal distributions. Deployment is performed based on the mean demands increased by an integer amount of their standard deviations. This might lead to a conservative solution, requiring more allocated resources than needed. This also reduces somehow the possibility of having service-dependent satisfaction levels.

A network slice embedding problem is considered in [Fendt et al., 2019], where available resources and resource demands are assumed to be partly uncertain. They are described by normal distributions built upon the data history on mobile network resource availability as well as slice resource utilization. To control the probability that a slice embedding solution will benefit from enough infrastructure resource, despite the uncertainties, some adjustable safety factor γ is introduced. As in [Mireslami et al., 2019], enough resources are dedicated to a service so as to satisfy the mean plus γ times the standard deviation of the demands. In [Fendt et al., 2019], additionally, a similar approach is considered to account for the uncertainty in the available resources. A *probability of feasibility*, depending on γ, is then evaluated for the slice embedding to measure the risk of having a degraded service for some users. The proposed solution leads to a slice resource allocation solution robust to uncertainties. Nevertheless, the resource demands of the different components of the slice have been considered as independent. Moreover, the safety factor γ is chosen identical for resource demands and available resources. This again may lead to allocating more resources than strictly necessary, and increases the operation cost.

The network slice embedding problem with demand uncertainties is also ad-

dressed in [Baumgartner et al., 2018]. The minimization of deployment costs considering first static resource demands is formulated as an MILP. Two robust network slice design formulations are then proposed to (*i*) handle demand uncertainties, and (*ii*) additionally account for correlations among the uncertain demands. A tuning parameter Γ is introduced to control the trade-off between robustness to the demand uncertainties and the deployment costs. Uncertainties related to the background traffics on the infrastructure, which clearly affect the residual infrastructure resources, are not considered.

To reduce the computation effort required to solve the robust network slice embedding problem, [Bauschert and Reddy, 2019] proposes to use a genetic algorithm, shown to surpass the performance of state-of-the-art robust MILP solvers used, *e.g.*, in [Baumgartner et al., 2018]. Uncertainties in infrastructure link bandwidth are also considered in [Wen et al., 2019], where possible failures of infrastructure nodes or links are taken into account to propose a robust algorithm that minimizes the network resource consumption under uncertain demands, while remapping the network slice in case of infrastructure failures. Since [Baumgartner et al., 2018], [Bauschert and Reddy, 2019], and [Wen et al., 2019] assume that the distribution of the variable demands and available infrastructure resource are unknown, their optimization are relatively conservative. Furthermore, uncertainties in various types of resources such as computing, memory, or wireless are not addressed.

3.4 Admission Control with Dynamic Slice Requests

The topic of dynamic slice/SFC deployment has also received significant attention in recent literature, see, for example, [Liu et al., 2017, Fendt et al., 2019, Wang et al., 2019].

In [Liu et al., 2017], a dynamic resource allocation for SFCs is investigated. The deployment of newly arrived SFCs and readjustment of in-service SFCs are taken into account. An ILP formulation is used to address the dynamic deployment problem, aiming at minimizing the cost of VNF deployment and migration. A pre-calculation of all possible routing paths has to be performed in advance, which requires some computational effort before using the deployment algorithm. In [Sun et al., 2019], the adaptive adjustment of allocated resources of each slice is enabled after each decision time period (slicing time). An hybrid slice reconfiguration framework is introduced in [Wang et al., 2019]. The slice can be reconfigured either within small time intervals for individual slices, or within large time intervals to readjust resource allocation of multiple slices. A deep-learning approach is adopted in [Huynh et al., 2019] for dynamic slice resource allocation, with the aim to maximize the long-term revenue of the network provider. Uncertainties related to the slice allocation requests and occupation time are considered. Nevertheless, slices are regarded as a whole, *i.e.*, not made up of multiple elements (*e.g.*, VNFs), which somewhat over-simplifies the problem of slice resource allocation.

Slice admission control (SAC) mechanisms have been developed recently [Noroozi

et al., 2019, Ebrahimi et al., 2020, Han et al., 2020, Bega et al., 2017, Bega et al., 2020] to address issues related to the unavailability of enough resources to satisfy all slice requests. In [Noroozi et al., 2019], SAC is formulated as a boolean linear program and a two-step sub-optimal algorithm based on variants of the knapsack problem are proposed to alleviate the complexity. Admission is done for slices with the highest profit considering first the RAN and *aggregate* core network resources. In a second step, the core network resources are considered without any aggregation to determine whether a slice deployment is possible.

In [Ebrahimi et al., 2020], SAC and resource allocation are performed jointly, to minimize the power consumption of the cloud nodes and the network bandwidth of the infrastructure provider. Transmission delay is taken into account in the slice SLA. Some elastic variables are introduced in an ILP formulation to extend the bounds on some constraints. They help determining when resources may be lacking, in which case slices are rejected starting from those with the highest requirements in terms of resource. Nevertheless, the dynamics of slice requests (time of arrival, slice duration) and the variation of slice resource demands during their life time are not considered in [Noroozi et al., 2019] and [Ebrahimi et al., 2020].

The dynamics of slice requests is considered by [Han et al., 2020] in the SAC problem. If not accepted, a request is queued for being potentially served later. The case of impatient tenants, who may leave their queues before being served, is taken into account. Nevertheless, neither the dynamics of resource demands within each slice, nor the activation time of a slice are accounted for. Moreover, infrastructure resources of each type are fully aggregated. As opposed to [Ebrahimi et al., 2020] and to our work, none of the details about the structure of the slice and of the infrastructure are taken into account in the resource model. Consequently, the proposed mechanism does not allow to provision the slice in addition to admission control.

Online SAC is considered in [Bega et al., 2017] and [Bega et al., 2020] leveraging on machine learning approaches. The aim is to maximize the revenue of the InP while guaranteeing the SLAs of the admitted slices. Both papers focus on radio resources of base stations. In [Bega et al., 2017], two different types of slices are considered to account for elastic and inelastic traffic. An admissibility region is determined first, indicating the maximum number of slices that the system can support without breaking the SLAs. Both works formalize the admission control problem into a semi-Markov decision process and derive the optimal policy obtained when the request arrival parameters are known. The approach has a high computation cost and is off-line (requires system parameters to be known *a priori*). An alternative Q-learning approach is proposed in [Bega et al., 2017] to adapt to changing environments while achieving close to optimal performance. In [Bega et al., 2020], a deep reinforcement learning method is developed to overcome the scalability issue of the Q-learning approach. These works consider tenants submitting slice requests for an immediate deployment, contrary to our work, where

slice requests are assumed to be submitted for an immediate but also for future deployment, which permits the development of a resource provisioning strategy.

3.5 Slice Resource Provisioning

The topic of slice resource provisioning is relatively new to the area of network slicing and has thus a limited coverage in the literature at the time being. One may find, for instance, in [Xiong et al., 2019] and [Sun et al., 2019], a preliminary study on the problem of joint resource provisioning and resource allocation for network slicing. In these papers, different slice resource provisioning frameworks in a virtualized radio access network context are introduced, where the heterogeneity of service requirements is considered. Provisioning is performed at the resource block (RB) level. The problem of radio resource provisioning and allocation from base stations (BSs) to a slice, and the assignment of users within the slices to BSs are considered. The first problem (provisioning and allocation) is solved in [Xiong et al., 2019] via heuristics, while a deep reinforcement learning technique is considered in [Sun et al., 2019]. The second problem (user assignment) is cast in the framework of an NP-complete 0-1 multiple knapsack problem. In these papers, the slice resource provisioning problem is studied, but is limited to the radio resources.

3.6 Conclusion

This chapter highlights some studies related to various aspects of network virtualization and network slicing. A literature review has been conducted on various topics related to the thesis. In Chapter 4, the hypotheses, assumptions, and notations that are used throughout this thesis will be presented.

Chapter 4

Hypotheses and Assumptions

This chapter presents the notations, assumptions, and hypotheses that are used throughout the thesis. A typical network slicing system is described, with all the involved entities. The relation and interactions between these entities, *e.g.*, the exchange of the characteristics of user demand, slice demand, and the dedicated service are also detailed.

4.1 Network Slicing System Entities

A typical network slicing system involves several entities: one or many Infrastructure Providers (InPs), Mobile Network Operators (MNOs), and Service Providers (SPs), as depicted in Figure 4.1 [Liang and Yu, 2014]. In some architecture proposals, the InP and the MNO can be the same entity [Samdanis et al., 2016]. This can also apply for the MNO and the SP. In this thesis, one considers the MNOs, SPs, and InPs as distinct entities.

The InPs own and manage the wireless and wired infrastructure such as the cell sites, the fronthaul and backhaul networks, and cloud data centers. The MNOs lease resources from the InPs to setup and manage the slices. The SPs then exploit the slices supplied by the MNOs, and provide to their customers the required services running within the slices.

Various steps involved in the proposed provisioning approach are illustrated in Figure 4.1. To satisfy expected service demands of users: (1) the SP identifies the necessary service characteristics in terms of QoS, satisfaction probability, *etc*. These service characteristics are forwarded to the MNO within the SLA between the SP and the MNO (denoted as SM-SLA); (2) the MNO then translates these characteristics into constraints to be satisfied by the slice dedicated to the required service; (3) the slice characteristics and constraints form the SLA between the MNO and the InP (denoted as MI-SLA) and include the aggregate resource demands of all users, the successful provisioning probability that has to be guaranteed by the InP, *etc*.

Then slice resource provisioning is performed (4–5) by the InP based on the MI-SLA between the MNO and the InP. This step is followed by slice deployment

and activation; (6) the provisioned resource are leased by the MNO to deploy and activate the target slice. Finally, the slice is exploited (7–8) by an SP who assigns users to the SFCs supplied by the MNO.

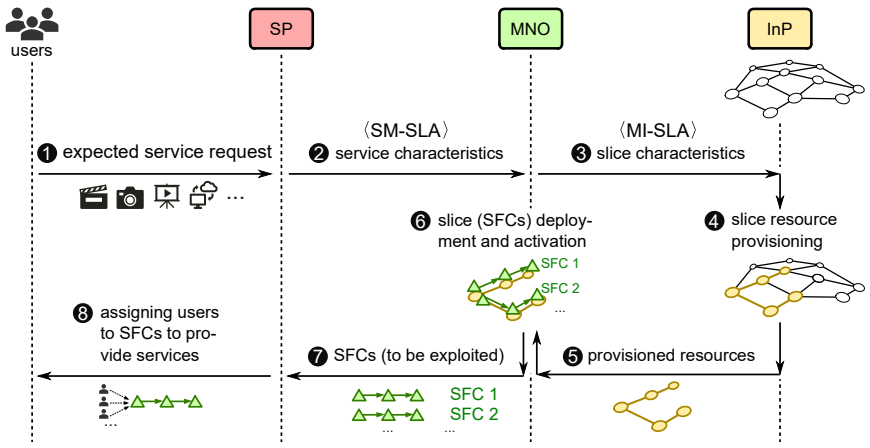


Figure 4.1: Network slicing entities and their SLA-based relationships.

4.2 Slice Provisioning Requests

A discrete-time model is considered, in which time is slotted into slots of constant duration T (typically of few tens of minutes). The value of T results from a compromise between the need to update the provisioning and the level of conservatism in the amount of provisioned resources required to satisfy fast fluctuating user demands;

The slot of index $k \in \mathbb{N}$ lasts over the time interval $[kT, (k+1)\,T[$. One considers that the slice lifetime spans over one or several time slots of duration T. Resources have to be provisioned so as to be compliant with the variations of the number of users and of their demands during the slice lifetime. The service characteristics are assumed stable over each time slot, and may vary from one time slot to the next.

Let t_s be the time instant at which the provisioning request for a slice s is received by the InP. This slice is also characterized by the index k_s^{on} of the time slot at the beginning of which it has to be activated (put into service), and the index k_s^{off} of the time slot at the end of which it has to be deactivated. Thus, the slice s is active over the time interval $\left[k_s^{\text{on}}T, \left(k_s^{\text{off}}+1\right)T\right[$.

Figure 4.2 depicts an example of arrivals of slice provisioning requests, as well as the time slots over which the corresponding services have to be active.

In Chapters 5–7, a *just-in-time* provisioning approach is considered, in which the lifetime of each slice is one single time slot. In this approach, in each time slot, the resource provisioning is performed only for the slices that need to be activated in the next time slot.

In Chapter 8, one considers a longer time scale of numerous time slots. For each time slot, a slice resource provisioning decision has to be made in advance, before the beginning of the time slot of which the slice has to be activated. Different

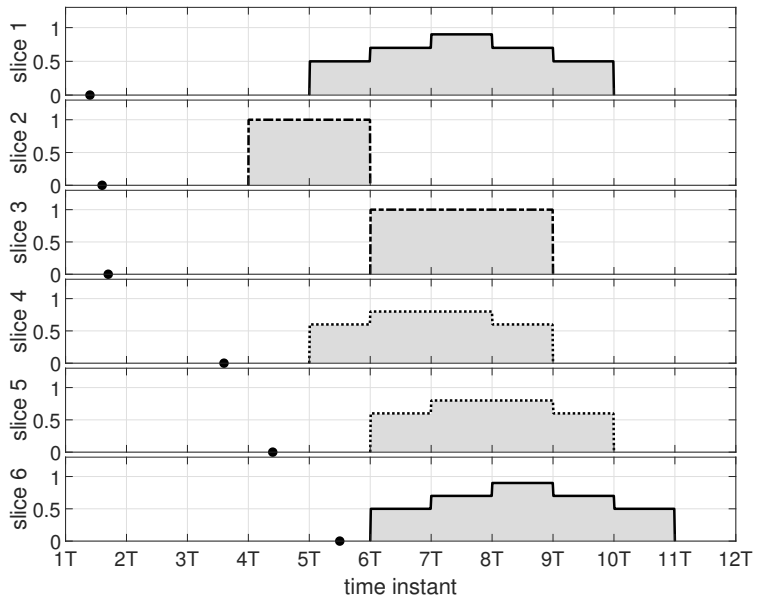


Figure 4.2: Arrivals of slice provisioning requests as a function of time; Black circles represent the arrival times t_s of each request; The types of slices are illustrated by different plot line styles; The slice resource demands evolve with time; Peak demands have been normalized.

slice request processing strategies are proposed. The time needed for the resource provisioning, resource allocation, and slice deployment/activation process are also taken into account.

4.2.1 Infrastructure Network

Consider an infrastructure network managed by a single InP. This network is represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of infrastructure nodes and $\mathcal{E}$ is the set of infrastructure links, which correspond to the wired connections between and within nodes (loop-back links) of the infrastructure network. In cases where the infrastructure network is managed by several InPs, the provisioning approach should be adapted accordingly. We discuss this important consideration in Section 4.2.

In this thesis, one considers a cloud network infrastructure made of a central large-sized datacenter, several small-sized datacenters geographically dispersed at edge nodes (*i.e.*, edge cloud) and a set of base stations. In our model, a node represents a datacenter, central or at edge, or a base station. A link represents an inter-datacenter connection or a backhauling connection between an edge datacenter and a base station.

Each infrastructure node $i \in \mathcal{N}$ is characterized by a given amount $a_n(i)$ of available resource of different types, where n is the type of node resource. An operation cost paid by the InP is attributed to each unit of node resource. The per-unit node resource cost associated to a given node i consists of a fixed part $c_{\mathrm{f}}(i)$ for node disposal (paid for each slice using node i), and variable part $c_n(i)$, which

depend linearly on the amount of resources provided by that node. Furthermore, an additional fixed cost per infrastructure node $c_{\mathrm{d}}(i)$ accounts for downloading a VNF image in a local image registry at the node (*i.e.*, datacenter) level. This fastens the slice deployment time, by avoiding to download the VNF image from a remote repository from a central location in the operator domain. One assumes this cost as independent from the VNF type, *i.e.*, the influence of the size of VNF is negligible and one considers that the cost mainly depends on the transfer distance from the node to the operator domain.

In Chapter 5 of Part II and all chapters of Part III, one considers three types of node resource: computing, memory, and wireless resources. In Chapter 6 of Part II, where coverage constraints are taken into account, the considered types of node resource are: computing, memory, and radio block (RB). It should be pointed out that the wireless resources, considered in Chapters 5, 7, and 8, can be referred to as the useful throughput (goodput), *i.e.*, the amount of data that can be delivered from the considered node, expressed in bits per unit of time. The set of node resources is denoted as Υ. The considered elements of Υ in each chapter of this thesis are listed in Table 4.1.

Table 4.1: Considered set of node resource types (Υ) in each chapter.

Part II		Part III	
Chap. 5	Chap. 6	Chap. 7	Chap. 8
$\Upsilon = \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$	$\Upsilon = \{\mathrm{c}, \mathrm{m}, \mathrm{r}\}$	$\Upsilon = \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$	$\Upsilon = \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$

c: computing, m: memory, w: wireless, r: radio block

Similarly, each infrastructure link $ij \in \mathcal{E}$ connecting node i to j has an available bandwidth $a_{\mathrm{b}}(ij)$, and an associated per-unit bandwidth cost $c_{\mathrm{b}}(ij)$. Several distinct VNFs of the same slice may be deployed on a given infrastructure node. When communication between these VNFs is required, an internal (loop-back) infrastructure link $ii \in \mathcal{E}$ can be used at each node $i \in \mathcal{N}$, as in [Wang et al., 2009], in the case of interconnected virtual machines (VMs) deployed on the same host. The associated per-unit bandwidth cost, in that case, is $c_{\mathrm{b}}(ii)$.

In Chapter 6, where radio resource and coverage constraints are considered, a Cloud Radio Access Network (C-RAN) architecture [China Mobile, 2011] is adopted, as illustrated in Figure 4.3. The C-RAN nodes (*i.e.*, eNB for 4G and gNB for 5G) mainly consists of two parts: (1) the distributed Remote Radio Heads (RRHs) plus antennas deployed at the cellular radio sites; and (2) the centralized Base Band Unit (BBU) pool hosted in an edge cloud data center [Tran et al., 2017].

The BBU pool hosts multiple virtual BBUs and handles higher layer processing functions, whereas all basic radio functions remain at the cellular radio station with the RRH. In 4G, the BBU handles all the L1-L2-L3 functional layers whereas radio frequency functions reside at the RRH. Within 5G, the gNB is split in three parts, namely Central Unit (CU), Distributed Unit (DU), and Radio Unit (RU), and different functional splits are under study where in some options the RU can support

some L2 functions thus reducing the capacity required for the fronthaul link [ITU-T, 2018]. The link (interface) between the BBU and the RRH is known as the fronthaul whereas the backhaul network connects the BBU with the core network functions hosted in the regional or central cloud.

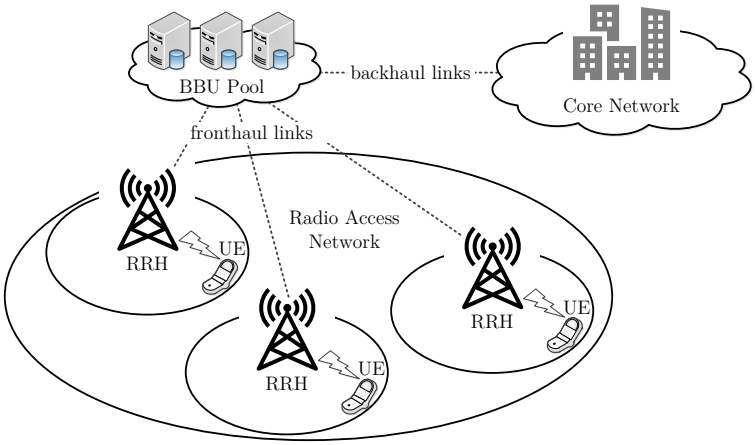


Figure 4.3: General architecture of C-RAN

Table 4.2 summarizes the main notations used in this thesis.

Table 4.2: Main notations used throughout this thesis.

Symbol	*Description*
$\mathcal{G}$	Infrastructure network graph, $\mathcal{G} = (\mathcal{N}, \mathcal{E})$
$\mathcal{N}$	Set of infrastructure nodes
$\mathcal{E}$	Set of infrastructure links
Υ	Set of node resource types, $\Upsilon = \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$
$a_n(i)$	Available resource of type $n \in \Upsilon$ at node i
$a_\mathrm{b}(ij)$	Available bandwidth of link ij
$c_n(i)$	Per-unit cost of resource of type $n \in \Upsilon$ for node i
$c_\mathrm{b}(ij)$	Per-unit cost for link ij
$c_\mathrm{f}(i)$	Fixed cost for using node i
$c_\mathrm{a}(i)$	Provisioning adaptation cost at node i
s	Slice index
$\mathcal{G}_s$	SFC graph, $\mathcal{G}_s = (\mathcal{N}_s, \mathcal{E}_s)$
$\mathcal{N}_s$	Set of virtual network functions
$\mathcal{E}_s$	Set of virtual links
$\boldsymbol{r}_s$	Resource demands of an SFC (SFC-RD)
$\mathbf{U}_s$	Resource demands of a typical user (U-RD)
$\mathbf{R}_s$	Resource demands of slice s (S-RD)
$\mathbf{B}$	Resources consumed by background

4.2.2 Best-Effort Background Services

In Chapters 5 and 6, the infrastructure resources are considered to be consumed only by network slices. In Chapters 7 and 8, we consider that the available resources in the infrastructure network are also partly consumed by other best-effort background services for which no resource provisioning has been performed. The vector gathering all resources consumed by the background services is denoted as

$$\mathbf{B} = \left(B_n(i), B_\mathrm{b}(ij)\right)^\top_{n\in\Upsilon,(i,ij)\in\mathcal{G}} \tag{4.1}$$

The elements $B_n(i)$, $\forall i \in \mathcal{N}$, $\forall n \in \Upsilon$ and $B_{\mathrm{b}}(ij)$, $\forall ij \in \mathcal{E}$ of $\mathbf{B}$ are the random variables representing the aggregate amount of node resources and link bandwidth consumed by these best-effort services.

4.2.3 Resource Demands

A demand of resources is defined on the basis of an SLA between an SP and the MNO, *i.e.*, SM-SLA. Throughout this thesis, we consider that a slice is devoted to a single type of service supplied by a given type of SFC. Several instances of that SFC may have to be deployed so as to satisfy the user demand. The topology of each SFC of slice s is represented by a *virtual*, *unweighted* graph $\mathcal{G}_s = (\mathcal{N}_s, \mathcal{E}_s)$ representing the VNFs and their interconnections. Each virtual node $v \in \mathcal{N}_s$ represents an instance of a VNF, and each virtual link $vw \in \mathcal{E}_s$ represents the connection between virtual nodes v and w.

Based on $\mathcal{G}_s$, one introduces the vectors $\boldsymbol{r}_s$, $\mathbf{U}_s$, and $\mathbf{R}_s$, respectively representing the resource demands of a single SFC (SFC-RD), of a typical user (U-RD), and of all users of slice s (S-RD).

- $\boldsymbol{r}_s = \left(r_{s,n}(v), r_{s,\mathrm{b}}(vw)\right)^{\top}_{n\in\Upsilon,(v,vw)\in\mathcal{G}_s}$ is the SFC-RD vector gathering the computing ($r_{s,\mathrm{c}}(v)$), memory ($r_{s,\mathrm{m}}(v)$), wireless ($r_{s,\mathrm{w}}(v)$), and bandwidth ($r_{s,\mathrm{b}}(vw)$) resource demands of the VNFs $v \in \mathcal{N}_s$ and the virtual links $vw \in \mathcal{E}_s$ of a single SFC;
- $\mathbf{U}_s = \left(U_{s,n}(v), U_{s,\mathrm{b}}(vw)\right)^{\top}_{n\in\Upsilon,(v,vw)\in\mathcal{G}_s}$ is the vector of resource demands a typical user (U-RD) of slice s, in which $U_{s,n}(v)$, $n \in \Upsilon$, and $U_{s,\mathrm{b}}(vw)$ are the amount of resources of VNF instance v and of virtual link vw employed by that user. The vector $\mathbf{U}_s$ is only introduced in Part III of this thesis;
- $\mathbf{R}_s = \left(R_{s,n}(v), R_{s,\mathrm{b}}(vw)\right)^{\top}_{n\in\Upsilon,(v,vw)\in\mathcal{G}_s}$ is the vector of resource demands of slice s (S-RD). $R_{s,n}(v)$, $n \in \Upsilon$, and $R_{s,\mathrm{b}}(vw)$ represent the aggregate amount of resources employed by N_s independent users of slice s.

Throughout this thesis, the elements of $\boldsymbol{r}_s$ are considered to be *deterministic*. The other variables (N_s and the elements of $\mathbf{U}_s$ and of $\mathbf{R}_s$) are considered to be *deterministic* in Part I, and to be *random vectors* in Part II. In Table 4.3, we summarize the characteristic (deterministic or random) of the variables considered in each chapter of this thesis.

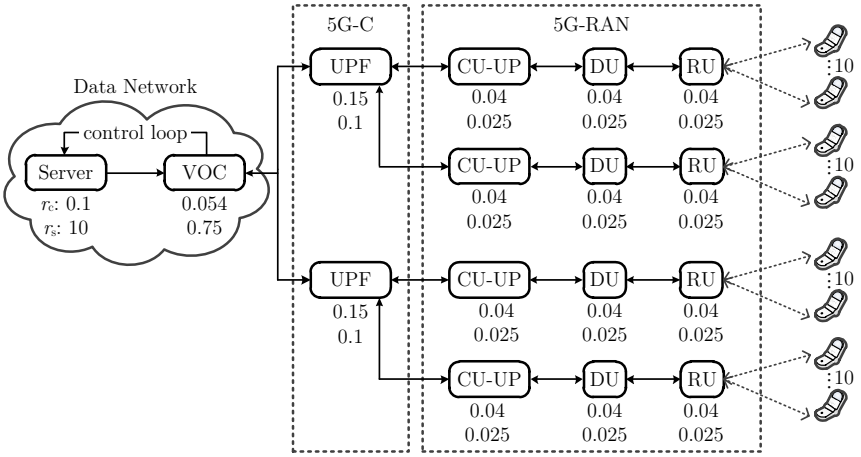
Figure 4.4 illustrates the virtual graph $\mathcal{G}_s$ whose nodes and links are weighted by (*i*) the SFC-RD $\boldsymbol{r}_s$ (Figure 4.4a); and the S-RD $\mathbf{R}_s$ (Figure 4.4b). These resources and network functionalities are devoted to the deployment of an adaptive wireless video streaming service [Savi et al., 2016]. The virtual graph $\mathcal{G}_s$ consists of six VNFs: three for the user-plane of the 5G-RAN (RU, DU, CU-UP), one for the 5G-Core (UPF), and two VNFs placed in the data network: (server and Video Optimization Controller (VOC)). The server archives videos with different qualities. Using the information received from users such as the bandwidth or end-to-end latency, the VOC dynamically adjusts the video bitrate to provide to the users. Notice that, in

Table 4.3: Considered characteristic of variables in each chapter.

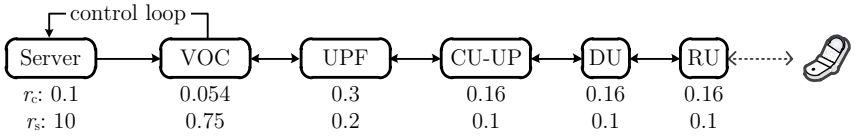
Variable		Part II		Part III	
		Chap. 5	Chap. 6	Chap. 7	Chap. 8
Infrastructure	$a_n(i)$, $a_{\mathrm{b}}(ij)$	dv	dv	⋆	⋆
Background services	$B_n(i)$, $B_{\mathrm{b}}(ij)$	null	null	rv	rv
Resource demands	$U_{s,n}(v)$, $U_{s,\mathrm{b}}(vw)$	dv	dv	rv	rv
	$r_{s,n}(v)$, $r_{s,\mathrm{b}}(vw)$	dv	dv	dv	dv
	$R_{s,n}(v)$, $R_{s,\mathrm{b}}(vw)$	dv	dv	rv	rv

dv: deterministic variables ⋆: partially employed by background services ($\mathbf{B}$)
rv: random variables

this example, the value of N_s and $\mathbf{U}_s$ are deterministic, leading to a deterministic value of $\mathbf{R}_s$.



(a) Graph of SFCs.



(b) Corresponding S-RD graph.

Figure 4.4: Virtual graph and the required computing (in CPUs) and memory (in GBytes) resources for the deployment of SFCs and slice dedicated to an adaptive wireless video streaming service.

When it is possible to provision enough resources, the MNO will be ensured to be able to deploy a collection of SFCs needed and satisfy the SLA. When, for example, the user density over some subarea is larger than stated in the SLA, some users may not be served. Nevertheless, from the perspective of the InP, the SLA is still satisfied. On contrary, when the user density/requirements are less than the maximum specified in the SLA, some provisioned resources may remain unused, but this is the price to pay when provisioning resources.

4.3 Conclusion

This chapter presents the notations, assumptions and hypotheses that are used in the next chapters of this thesis. A typical network slicing system is described, with all the involved entities. The relations and interactions between these entities, *e.g.*, the exchange of the characteristics of user demand, slice demand, and the dedicated service are also detailed.

Part II

Resource Provisioning for Deterministic Demands

Chapter 5

Resource Provisioning for the Core Network

This chapter is based on Q.-T. Luu, M. Kieffer, and A. Mouradian, and S. Kerboeuf, "Aggregated Resource Provisioning for Network Slices," in Proc. IEEE Global Communications Conference, Abu Dhabi, Dec. 2018, pp. 1-6 [Luu et al., 2018].

This chapter aims to provide an answer to Challenge 1 introduced in Chapter 1. We address the problem of infrastructure core network resource provisioning for network slices. The available resource in the infrastructure and the slice resource demands are considered as deterministic.

The rest of the chapter is structured as follows. In Section 5.2, we summarize the originality of the work proposed here, compared to the state of the art. Section 5.3 provides the formulation of the resource provisioning problem, which is an optimization problem that differs significantly from the traditional SFC embedding problem. Evaluation of the proposed algorithm is presented in Section 5.4. Finally, Section 5.5 details some conclusions and perspectives.

5.1 Related Work

As discussed in Chapter 3, the majority of prior works considered the problem of SFC and VNF deployment (or embedding), *e.g.*, [Cohen et al., 2015, Riggio et al., 2016, Riera et al., 2016, Vizarreta et al., 2017]. The problem of SFC embedding is usually formulated as an ILP [Vizarreta et al., 2017, Cohen et al., 2015, Riera et al., 2016], an MILP [Chowdhury et al., 2012, Kang et al., 2017], or an IQP [Tajiki et al., 2018], with objective to minimize the SFC deployment cost due to the use of infrastructure resources. Many heuristics have also been proposed in the literature, with the aim of reducing the complexity of the embedding problem. Using different approaches, *e.g.*, column generation [Huin et al., 2017, Liu et al., 2017], eigendecomposition [Mechtri et al., 2016], or local search for neighbor nodes [Riggio et al., 2016], these heuristics can solve large problems in a reasonable amount of time.

5.2 Contributions

The originality of the work presented in this chapter lies in the proposition of a resource provisioning for network slices. The main idea is to propose an alternative approach to previous best-effort approaches, *e.g.*, [Riera et al., 2016, Riggio et al., 2016, Vizarreta et al., 2017], where the slice SFCs are iteratively deployed on the infrastructure network. In practice, within a slice, SFCs are created, managed, and released in an asynchronous way. Iterative deployment strategies are thus well-suited to such dynamic slice management. Nevertheless, when several concurrent slices are managed in parallel by some MNOs, nothing ensures that enough infrastructure resources will be available to deploy a new SFC. This type of SFC management makes it difficult to satisfy the desired SLA expressed in terms of guaranteed amount of deployed SFCs. With our proposed approach, infrastructure resources are *provisioned* (reserved) in advance. The MNO is thus ensured to be able to deploy the required amount of SFCs within a slice corresponding to the slice resource demands.

As introduced in Chapter 4, the slice resource demand (S-RD) of a given slice s,

$$\mathbf{R}_s = \left(R_{s,n}\left(v\right), R_{s,\mathrm{b}}\left(vw\right)\right)^{\top}_{n\in\Upsilon,(v,vw)\in\mathcal{G}_s}$$

aggregates the resource requirements of N_s independent users of slice s. The S-RD is evaluated by the MNO to satisfy the QoS requirements imposed by the SP. The InP has then to provision enough infrastructure resources to meet the SLA. The resource provisioning for slice s is represented by a mapping between the infrastructure graph $\mathcal{G}$ and the virtual graph $\mathcal{G}_s$. Due to the fact that nodes or links of the graph of $\mathcal{G}_s$ represent aggregate requirements, several infrastructure nodes may have to be gathered and parallel physical links have to be considered to satisfy the various S-RDs. This is the main difference with respect to the deployment approach considered in [Riggio et al., 2016, Vizarreta et al., 2017], where each VNF is deployed on a single node. In [Riggio et al., 2016, Vizarreta et al., 2017], virtual nodes and links are mapped on the infrastructure network to allocate resources to VNFs and virtual links. In this chapter, one will provision a sufficient amount of infrastructure nodes and links, so that the aggregate provisioned resources meet the slice demands $\mathbf{R}_s$.

5.3 Problem Formulation

As introduced in Section 4.2, in this chapter, a *just-in-time* provisioning approach is applied, in which the slice resource provisioning is performed during a given time slot and considering only provisioning requests of slices that have to be activated in the next time slot. The set of those slice requests is denoted as $\mathcal{S}$.

The objective of slice resource provisioning is to minimize the cost of provisioning resources for all slices in $\mathcal{S}$. To that end, we introduce the following set of

variables $\mathbf{\Phi} = \{\mathbf{\Phi}_s\}_{s\in\mathcal{S}}$, with

$$\mathbf{\Phi}_s = \left\{\phi_{s,n}(i,v), \phi_{s,\mathrm{b}}(ij,vw), \widetilde{\phi}_{s,n}(i,v)\right\}_{s\in\mathcal{S},(i,ij)\in\mathcal{G},(v,vw)\in\mathcal{G}_s,n\in\Upsilon},$$

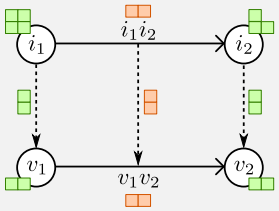
where $\phi_{s,n}(i,v) \in [0,1]$ represents the proportion of resources of type n provisioned by the infrastructure node $i \in \mathcal{N}$ for the virtual node $v \in \mathcal{N}_s$ of slice s;

$\phi_{s,\mathrm{b}}(ij,vw) \in [0,1]$ represents the proportion of bandwidth provisioned by the infrastructure link $ij \in \mathcal{E}$ for the virtual link $vw \in \mathcal{E}_s$ of slice s. When one of the variables $\phi_{s,n}(i,v)$ and $\phi_{s,\mathrm{b}}(ij,vw)$ holds zero, there is no mapping between the infrastructure and the virtual node/link;

$\widetilde{\phi}_{s,n}(i,v) \in \{0,1\}$ is the node mapping indicator, *i.e.*, $\widetilde{\phi}_{s,n}(i,v) = 1$ if $\phi_{s,n}(i,v) > 0$ and $\widetilde{\phi}_{s,n}(i,v) = 0$ otherwise. This mapping indicator is used to determine whether an infrastructure node has provisioned resources for virtual nodes.

Illustration to the elements of $\mathbf{\Phi}_s$ is provided in Example 5.1.

Example 5.1 (Illustration of $\mathbf{\Phi}_s$)**.** The figure below depicts a mapping between the infrastructure nodes i_1, i_2 and the virtual nodes v_1, v_2; and between the infrastructure link i_1i_2 and the virtual link v_1v_2, *i.e.*, $\widetilde{\phi}_{s,n}(i_1,v_1) = \widetilde{\phi}_{s,n}(i_2,v_2) = 1$. The squares at i_1, i_2 and i_1i_2 represent $a_n(i_1)$, $a_n(i_2)$, and $a_{\mathrm{b}}(i_1i_2)$, while the squares at v_1, v_2, and v_1v_2 represent $R_n(v_1)$, $R_n(v_2)$, and $R_{\mathrm{b}}(v_1v_2)$.



In this example, the proportions of resources that i_1, i_2, and i_1i_2 provisioned to v_1, v_2, and v_1v_2 are respectively $\phi_{s,n}(i_1,v_1) = \frac{1}{2}$, $\phi_{s,n}(i_2,v_2) = \frac{2}{3}$, and $\phi_{s,\mathrm{b}}(i_1i_2,v_1v_2) = 1$.

5.3.1 Resource Provisioning for a Single Slice

In this section, we start with the resource provisioning for a single slice $s \in \mathcal{S}$.

The total provisioning cost c_{total} associated to ϕ_s is defined as

$$\begin{aligned} c_{\text{total}}(\mathbf{\Phi}_s) = &\sum_{i,v,n} a_n(i)\,\phi_{s,n}(i,v)\,c_n(i) + \sum_{ij,vw} a_{\mathrm{b}}(ij)\,\phi_{s,\mathrm{b}}(ij,vw)\,c_{\mathrm{b}}(ij) \\ &+ \sum_{i,v} \widetilde{\phi}_{s,n}(i,v)\,c_d(i), \end{aligned} \tag{5.1}$$

where the first and second terms indicate the total cost for leasing resources from infrastructure nodes and links, while the third term represents the cost for deploying VNFs in infrastructure nodes. The minimization of $c_{\text{total}}(\mathbf{\Phi}_s)$ has to be such that several constraints are satisfied.

In what follows, (5.2–5.3) describe the resource requirement constraints for slice nodes and links. (5.4–5.5) guarantee that the provisioned resource cannot exceed the available infrastructure resources. Eq. (5.6) describes the relation between the variables $\widetilde{\phi}_{s,n}(i,v)$ and $\phi_{s,n}(i,v)$ of $\mathbf{\Phi}_s$. Constraints on node mapping are described in (5.7). (5.8) ensures that enough resources are provisioned to deploy an integer number of VNF instances. Finally, (5.9–5.11) describes the resource proportionality and flow reservation constraints.

The resources provisioned by all infrastructure nodes $i \in \mathcal{N}_\mathrm{s}$ to a virtual node v should satisfy the resource demands of v. This leads to

$$\sum_i a_n(i)\,\phi_{s,n}(i,v) \geq R_n(v), \forall v \in \mathcal{N}_s, n \in \Upsilon. \tag{5.2}$$

Similarly, the resources provided by all infrastructure links $ij \in \mathcal{E}$ to a virtual link vw should satisfy the resource demands of vw,

$$\sum_{ij} a_\mathrm{b}(ij)\,\phi_{s,\mathrm{b}}(ij,vw) \geq R_\mathrm{b}(vw), \forall vw \in \mathcal{E}_s. \tag{5.3}$$

Since, the sum of proportions of resources provisioned by a given infrastructure node i cannot exceed one, one has

$$\sum_v \phi_{s,n}(i,v) \leq 1, \forall i \in \mathcal{N}, n \in \Upsilon. \tag{5.4}$$

Similarly, the proportions of resources provisioned by a given infrastructure link ij cannot exceed one. As a consequence,

$$\sum_{vw} \phi_{s,\mathrm{b}}(ij,vw) \leq 1, \forall ij \in \mathcal{E}. \tag{5.5}$$

The elements $\widetilde{\phi}_{s,n}(i,v)$ of $\mathbf{\Phi}_s$ are such that $\widetilde{\phi}_{s,n}(i,v) = 1$ if $\phi_{s,n}(i,v) > 0$. The relation between $\phi_{s,n}(i,v)$ and $\widetilde{\phi}_{s,n}(i,v)$ is non-linear. To address this issue, both quantities may be combined with the following linear constraints

$$\phi_{s,n}(i,v) \leq \widetilde{\phi}_{s,n}(i,v) < \phi(i,v) + 1, \forall i \in \mathcal{N}, v \in \mathcal{N}_s, n \in \Upsilon. \tag{5.6}$$

We impose that any infrastructure node $i \in \mathcal{N}$ cannot provision resources for more than a single virtual node of a given slice. This increases robustness to infrastructure node failures and can be translated into

$$\sum_v \widetilde{\phi}_{s,n}(i,v) \leq 1, \forall i \in \mathcal{N}. \tag{5.7}$$

The amount of resources provided by a given infrastructure node i to a virtual node v has to be equal to an integer multiple of the minimum amount of resources

$r_n(v)$ for a VNF associated to the virtual node v

$$a_n(i)\,\phi_{s,n}(i,v) = \kappa_{s,n}(i,v)\,r_n(v), \forall i \in \mathcal{N}, v \in \mathcal{N}_s, n \in \Upsilon \tag{5.8}$$

where $\kappa_{s,n}(i,v)$ is a positive integer belonging to the set of variables of the optimization problem. This ensures that enough resources are provisioned by an infrastructure node i to be able to deploy an integer number $\kappa_{s,n}(i,v)$ of VNF instances associated to the virtual node v.

Moreover, provisioning of infrastructure node and link resources has to be performed in a balanced way, consistent with the virtual graph. When an infrastructure node provisions resources for a slice, the amount of provisioned computing, memory, and wireless capacity has to be in the same proportion as the corresponding resources of the virtual node. In this chapter, we consider that every virtual node requires computing and memory resources ($R_{s,\mathrm{c}}(v) > 0$ and $R_{s,\mathrm{m}}(v) > 0, \forall v \in \mathcal{N}_s$), while only certain nodes need wireless capacity. The node resource proportionality satisfaction may be formulated as, $\forall i \in \mathcal{N}$, $v \in \mathcal{N}_s$ such that $R_{s,\mathrm{w}}(v) > 0$:

$$\frac{a_{\mathrm{c}}(i)\,\phi_{s,\mathrm{c}}(i,v)}{R_{s,\mathrm{c}}(v)} = \frac{a_{\mathrm{m}}(i)\,\phi_{s,\mathrm{m}}(i,v)}{R_{s,\mathrm{m}}(v)} = \frac{a_{\mathrm{w}}(i)\,\phi_{s,\mathrm{w}}(i,v)}{R_{s,\mathrm{w}}(v)}, \tag{5.9}$$

and $\forall i \in \mathcal{N}$, $v \in \mathcal{N}_s$ such that $R_{s,\mathrm{w}}(v) = 0$:

$$\frac{a_{\mathrm{c}}(i)\,\phi_{s,\mathrm{c}}(i,v)}{R_{s,\mathrm{c}}(v)} = \frac{a_{\mathrm{m}}(i)\,\phi_{s,\mathrm{m}}(i,v)}{R_{s,\mathrm{m}}(v)}. \tag{5.10}$$

These constraints ensure that the infrastructure node i provisions the same proportion of memory, computing, and possibly wireless resources for the virtual node v. Unbalanced resource provisioning is thus avoided.

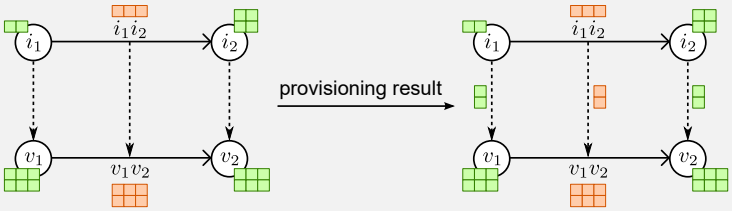
Finally, some flow conservation constraints have to be satisfied when resources are provisioned on the infrastructure link ij for the virtual link vw. That is, for each link $vw \in \mathcal{E}_s$, there must exist a continuous connection between *each* pair of infrastructure nodes that are mapped to the pair (v,w) of virtual nodes.

The proportion of resources on the flow of the virtual link vw entering/leaving an infrastructure node mapped onto v or w should be equal to the proportion of any resource this node provides to v or w. Focusing on the computing resource, this constraint can be formulated as

$$\sum_{j \in \mathcal{N}} \left(\frac{a_{\mathrm{b}}(ij)}{R_{s,\mathrm{b}}(vw)} \phi_{s,\mathrm{b}}(ij,vw) - \frac{a_{\mathrm{b}}(ji)}{R_{s,\mathrm{b}}(vw)} \phi_{s,\mathrm{b}}(ji,vw) \right)$$
$$= \frac{a_{\mathrm{c}}(i)}{R_{s,\mathrm{c}}(v)} \phi_{s,\mathrm{c}}(i,v) - \frac{a_{\mathrm{c}}(i)}{r_{\mathrm{c}}(w)} \phi_{s,\mathrm{c}}(i,w), \forall i \in \mathcal{N}, vw \in \mathcal{E}_s. \tag{5.11}$$

The consistency with the other provisioned resources is ensured by (5.9) or (5.10). An example to illustrate constraint (5.11) is given in Example 5.2.

Example 5.2 (Illustration of the flow conservation constraint). The figure below depicts a mapping between the infrastructure nodes i_1, i_2 and the virtual nodes v_1, v_2; and between the infrastructure link i_1i_2 and the virtual link v_1v_2.



Consider i_1 and v_1v_2, constraint (5.11) leads to $\frac{3}{6}\phi_{s,n}(i_1i_2, v_1v_2) - 0 = \frac{2}{6}\phi_{s,n}(i_1, v_1) - 0$. Since $0 \leq \phi_{s,n}(i_1, v_1), \phi_{s,n}(i_1i_2, v_1v_2) \leq 1$, and to minimize the total provisioning cost depending on $\phi_{s,n}(i_1, v_1)$ and $\phi_{s,n}(i_1i_2, v_1v_2)$, one obtains $\phi_{s,n}(i_1i_2, v_1v_2) = \frac{2}{3}$ and $\phi_{s,n}(i_1, v_1) = 1$. Similarly $\phi_{s,n}(i_2, v_2) = \frac{1}{2}$. The proportions of provisioned node and link resources are then consistent with the proportions of nodes and link resource demands. Constraint (5.11) is satisfied with the obtained $\mathbf{\Phi}_s$.

Finally, the resource provisioning for a single slice can be cast as an MILP in Problem 5.1.

Problem 5.1: MILP Single Slice Resource Provisioning

$$\min_{\mathbf{\Phi}_s} \quad c_{\text{total}}(\mathbf{\Phi}_s),$$

subject to

$$\sum_i a_n(i)\,\phi_{s,n}(i,v) \geq R_n(v), \forall v \in \mathcal{N}_s, n \in \Upsilon,$$

$$\sum_{ij} a_{\text{b}}(ij)\,\phi_{s,\text{b}}(ij,vw) \geq R_{\text{b}}(vw), \forall vw \in \mathcal{E}_s,$$

$$\sum_v \phi_{s,n}(i,v) \leq 1, \forall i \in \mathcal{N}, n \in \Upsilon,$$

$$\sum_{vw} \phi_{s,\text{b}}(ij,vw) \leq 1, \forall ij \in \mathcal{E},$$

$$\sum_v \widetilde{\phi}_{s,n}(i,v) \leq 1, \forall i \in \mathcal{N},$$

$$\forall i \in \mathcal{N}, v \in \mathcal{N}_s, n \in \Upsilon:$$

$$\phi_{s,n}(i,v) \leq \widetilde{\phi}_{s,n}(i,v) < \phi(i,v) + 1,$$

$$a_n(i)\,\phi_{s,n}(i,v) = \kappa_{s,n}(i,v)\,r_n(v),$$

$$\forall i \in \mathcal{N}, v \in \mathcal{N}_s \text{ such that } R_{s,\text{w}}(v) > 0:$$

$$\frac{a_{\text{c}}(i)\,\phi_{s,\text{c}}(i,v)}{R_{s,\text{c}}(v)} = \frac{a_{\text{m}}(i)\,\phi_{s,\text{m}}(i,v)}{R_{s,\text{m}}(v)} = \frac{a_{\text{w}}(i)\,\phi_{s,\text{w}}(i,v)}{R_{s,\text{w}}(v)},$$

$$\forall i \in \mathcal{N}, v \in \mathcal{N}_s \text{ such that } R_{s,\text{w}}(v) = 0:$$

$$\frac{a_{\text{c}}(i)\,\phi_{s,\text{c}}(i,v)}{R_{s,\text{c}}(v)} = \frac{a_{\text{m}}(i)\,\phi_{s,\text{m}}(i,v)}{R_{s,\text{m}}(v)},$$

$$\sum_{j \in \mathcal{N}} \left(\frac{a_{\text{b}}(ij)}{R_{s,\text{b}}(vw)} \phi_{s,\text{b}}(ij,vw) - \frac{a_{\text{b}}(ji)}{R_{s,\text{b}}(vw)} \phi_{s,\text{b}}(ji,vw) \right)$$

$$= \frac{a_{\mathrm{c}}(i)}{R_{s,\mathrm{c}}(v)}\phi_{s,\mathrm{c}}(i,v) - \frac{a_{\mathrm{c}}(i)}{r_{\mathrm{c}}(w)}\phi_{s,\mathrm{c}}(i,w), \forall i \in \mathcal{N}, vw \in \mathcal{E}_s.$$

5.3.2 Resource Provisioning for Multiple Slices

When resource provisioning has to be performed for several slices $s \in \mathcal{S}$, the objective function (5.1) becomes

$$\begin{aligned} c_{\text{total}}(\mathbf{\Phi}) &= \sum_{s} c_{\text{total}}(\mathbf{\Phi}_s) \\ &= \sum_{s,i,v,n} a_n(i)\,\phi_{s,n}(i,v)\,c_n(i) \\ &+ \sum_{s,ij,vw} r_e(ij)\,\phi_{s,\mathrm{b}}(ij,vw)\,c_{\mathrm{b}}(ij) + \sum_{s,i,v} \widetilde{\phi}_{s,n}(i,v)\,c_{\mathrm{d}}(i), \end{aligned} \tag{5.12}$$

The constraints for this problem are similar to those of the resource provisioning for a single slice presented in Section 5.3.1, with the following minor modifications. The sum over all slice $s \in \mathcal{S}$ is added to the left side of constraints (5.5) and (5.6), *i.e.*,

$$\sum_{s,v} \phi_{s,n}(i,v) \leq 1, \forall i \in \mathcal{N}, n \in \Upsilon, \tag{5.13}$$

$$\sum_{s,vw} \phi_{s,b}(ij,vw) \leq 1, \forall ij \in \mathcal{E}. \tag{5.14}$$

Moreover, the remaining constraints (5.2), (5.4), (5.7)–(5.11) hold for all slices $s \in \mathcal{S}$ and for all $n \in \Upsilon$. Finally, the resource provisioning for multiple slices can be cast as an MILP in Problem 5.2.

Problem 5.2: MILP Multiple Slice Resource Provisioning

$$\min_{\mathbf{\Phi}} \quad c_{\text{total}}(\mathbf{\Phi}) = \sum_{s\in\mathcal{S}} c_{\text{total}}(\mathbf{\Phi}_s),$$

subject to

$$\sum_{i} a_n(i)\,\phi_{s,n}(i,v) \geq R_n(v), \forall s \in \mathcal{S}, v \in \mathcal{N}_s, n \in \Upsilon$$

$$\sum_{ij} a_{\mathrm{b}}(ij)\,\phi_{s,\mathrm{b}}(ij,vw) \geq R_{\mathrm{b}}(vw), \forall s \in \mathcal{S}, vw \in \mathcal{E}_s,$$

$$\sum_{s,v} \phi_{s,n}(i,v) \leq 1, \forall i \in \mathcal{N}, n \in \Upsilon,$$

$$\sum_{s,vw} \phi_{s,\mathrm{b}}(ij,vw) \leq 1, \forall ij \in \mathcal{E},$$

$$\sum_{v} \widetilde{\phi}_{s,n}(i,v) \leq 1, \forall s \in \mathcal{S}, i \in \mathcal{N},$$

$$\forall s \in \mathcal{S}, i \in \mathcal{N}, v \in \mathcal{N}_s, n \in \Upsilon:$$

$$\phi_{s,n}(i,v) \leq \widetilde{\phi}_{s,n}(i,v) < \phi(i,v) + 1,$$

$$a_n(i)\,\phi_{s,n}(i,v) = \kappa_{s,n}(i,v)\,r_n(v),$$

$$\forall s \in \mathcal{S}, i \in \mathcal{N}, v \in \mathcal{N}_s \text{ such that } R_{s,\mathrm{w}}(v) > 0:$$
$$\frac{a_{\mathrm{c}}(i)\,\phi_{s,\mathrm{c}}(i,v)}{R_{s,\mathrm{c}}(v)} = \frac{a_{\mathrm{m}}(i)\,\phi_{s,\mathrm{m}}(i,v)}{R_{s,\mathrm{m}}(v)} = \frac{a_{\mathrm{w}}(i)\,\phi_{s,\mathrm{w}}(i,v)}{R_{s,\mathrm{w}}(v)},$$
$$\forall s \in \mathcal{S}, i \in \mathcal{N}, v \in \mathcal{N}_s \text{ such that } R_{s,\mathrm{w}}(v) = 0:$$
$$\frac{a_{\mathrm{c}}(i)\,\phi_{s,\mathrm{c}}(i,v)}{R_{s,\mathrm{c}}(v)} = \frac{a_{\mathrm{m}}(i)\,\phi_{s,\mathrm{m}}(i,v)}{R_{s,\mathrm{m}}(v)},$$
$$\sum_j \Big(\frac{a_{\mathrm{b}}(ij)}{R_{s,\mathrm{b}}(vw)} \phi_{s,\mathrm{b}}(ij,vw) - \frac{a_{\mathrm{b}}(ji)}{R_{s,\mathrm{b}}(vw)} \phi_{s,\mathrm{b}}(ji,vw) \Big)$$
$$= \frac{a_n(i)}{r_n(v)} \phi_{s,n}(i,v) - \frac{a_n(i)}{r_n(w)} \phi_{s,n}(i,w), \forall s \in \mathcal{S}, i \in \mathcal{N}, vw \in \mathcal{E}_s.$$

5.4 Evaluation

In this section, we compare the performance of two different resource provisioning schemes: Sequential (SP) and joint (JP) slice resource provisioning. In the sequential approach, resources are provisioned slice by slice by solving the single slice MILP (Problem 5.1). In the joint approach, provisioning is performed taking into account all slices simultaneously and solving the multiple slices MILP (Problem 5.2). Both schemes are evaluated using the CPLEX MILP solver interfaced with MATLAB.

5.4.1 Infrastructure Network Topology

The infrastructure network is generated from a K-ary fat tree topology, as in [Riggio et al., 2016, Bouten et al., 2017]. A typical fat-tree topology is depicted in Figure 5.1 when $K = 4$. The leaf nodes represent the Remote Radio Heads (RRHs). The other nodes represent the edge, regional, and central data centers. Simulations are done with $K = 2, 4, 6$, resulting in a total of 8, 16, and 54 RRHs. The layers of the fat-tree-based infrastructure are depicted in Figure 5.1. Infrastructure nodes and links provide a given amount of computing, memory, and possibly wireless resources $(a_{\mathrm{c}}, a_{\mathrm{m}}, a_{\mathrm{w}})$ expressed in available number of CPUs, Gbytes of memory, and Gbps of transmission capacity, depending on the layer they are located. The cost of leasing each unit of infrastructure resource is set to 1 for every types $n \in \Upsilon$.

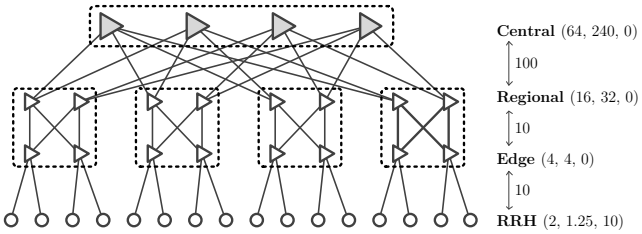


Figure 5.1: Description of the infrastructure network. Nodes provide a given amount of computing, memory, and possibly wireless resources $(a_{\mathrm{c}}, a_{\mathrm{m}}, a_{\mathrm{w}})$ measured in number of used CPUs, Gbytes, and Gbps, respectively. Links are assigned with a given amount of bandwidth (a_{b}) measured in Gbps.

5.4.2 Slice Resource Demands

In this work, we consider two types of slices, each of which has a *deterministic* resource demand.

- Slices of type 1 are devoted to the delivery of UHD video streaming services at 20 Mbps to $N_s = 2000$ users;
- Slices of type 2 are dedicated to provide HD video streaming services at 4 Mbps for $N_s = 4000$ users.

Both slice types address the same type of service, and implement similar SFCs, but with different resource requirements. Each SFC consists of three chained VNFs: a virtual BS, a virtual Gateway/Firewall (GW/FW), and a virtual Video Optimizer Controller (VOC). The minimum resource requirements for running each VNF instance correspond to the aggregate resource demands of 200 users.

Details of each slice type as well as the slice resource demands, $\mathbf{R}_s$, of each type are given in Table 5.1. The values in Table 5.1 have been adapted from [Savi et al., 2016].

Table 5.1: S-RD Parameters of Two Types of Slices.

Type 1: UHD video streaming at 20 Mbps, $N_s = 2000$.

Node	$(R_{s,\mathrm{c}}, r_{s,\mathrm{c}})$	$(R_{s,\mathrm{m}}, r_{s,\mathrm{m}})$	$(R_{s,\mathrm{w}}, r_{s,\mathrm{w}})$	*Link*	$R_{s,\mathrm{b}}$
vVOC	$(54, 5.4)$	$(150, 15)$	—	vVOC→vGW	40
vGW	$(9.0, 0.9)$	$(5.0, 0.5)$	—	vGW→vBBU	40
vBBU	$(8.0, 0.8)$	$(5.0, 0.5)$	$(40, 4.9)$		

Type 2: HD video streaming at 4 Mbps, $N_s = 4000$.

Node	$(R_{s,\mathrm{c}}, r_{s,\mathrm{c}})$	$(R_{s,\mathrm{m}}, r_{s,\mathrm{m}})$	$(R_{s,\mathrm{w}}, r_{s,\mathrm{w}})$	*Link*	$R_{s,\mathrm{b}}$
vVOC	$(21, 1.1)$	$(60, 3.0)$	—	vVOC→vGW	40
vGW	$(3.6, 0.2)$	$(2.0, 0.1)$	—	vGW→vBBU	40
vBBU	$(3.2, 0.2)$	$(2.0, 0.1)$	$(16, 0.8)$		

5.4.3 Results

5.4.3.1 Comparison of Provisioning Algorithms

Figures 5.2a, 5.2b, and 5.2c illustrate respectively the infrastructure node utilization, the infrastructure link utilization, and the computing time for different fat-tree sizes $K \in \{2, 4, 6\}$ considering two slices of different types (UHD and HD). Infrastructure node or link utilization reflects the percentage of infrastructure nodes or links provisioned for the considered slices. Figures 5.2d, 5.2e, and 5.2f consider the same metrics with a fat-tree of size $K = 2$ and different number of slices ($|\mathcal{S}|$) of the same type (HD). Both JP and SP approaches have been compared.

It can be seen that the JP scheme slightly outperforms the SP approach, in terms of both node and link utilization, as the aim of JP is to find the optimal solution

for the whole problem, *i.e.*, provisioning for all the slices, while the SP method only takes care of one slice at a time. The difference in performance of these two methods becomes more significant when the size of the infrastructure network is relatively small, *i.e.*, it is harder to find suitable nodes and links to provision resources for slices. For instance, at $K = 2$ with two slices (see Figure 5.2a), the JP method uses only 67% of the total infrastructure links to provision resources needed to support the slices, while about 73.6% of the links are used by the SP method to provision the required resources.

Nevertheless, as expected, the SP provisioning method performs better than the JP approach in terms of computing time. The reason is that, increasing of number of slices leads to variable set $\mathbf{\Phi}$ of higher cardinality, and therefore increases the computation duration, as shown in Figures 5.2c and 5.2f.

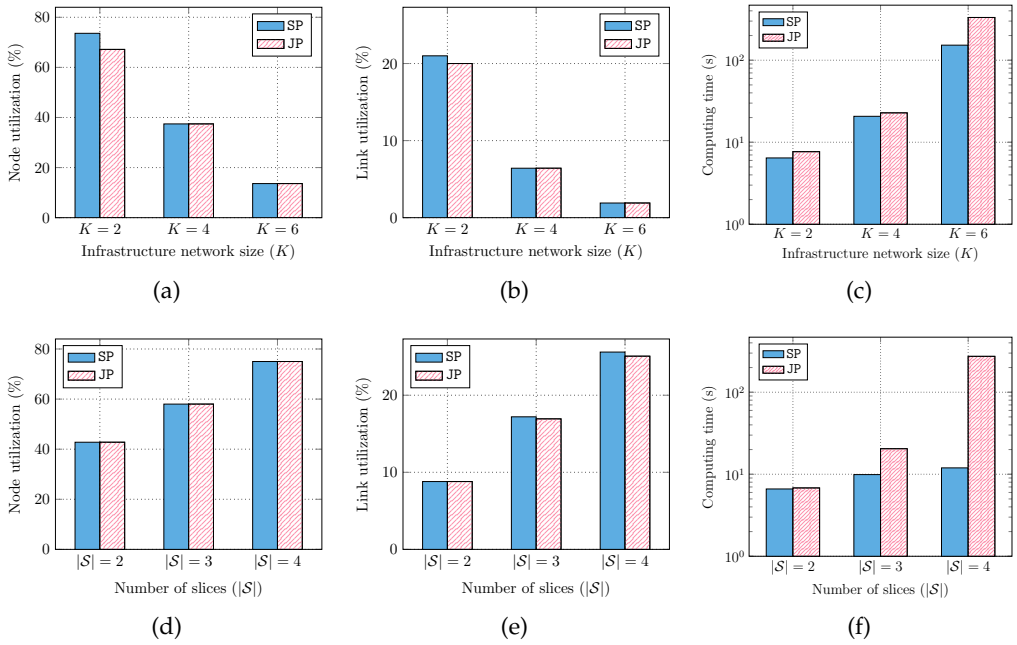


Figure 5.2: Performance of JP and SP as function of the fat-tree size (K) and of the number of slices ($|\mathcal{S}|$), in terms of (a, d) utilization of infrastructure nodes, (b, e) utilization of infrastructure link, and (c, f) computing time.

5.4.3.2 Resource Provisioning vs Direct Embedding

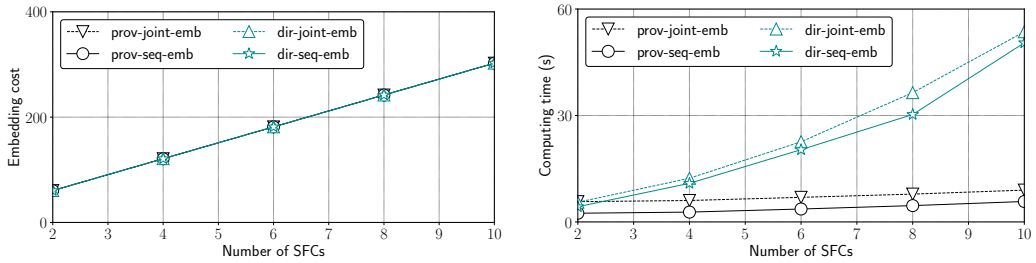
To evaluate the benefits of a provisioning approach prior to SFC embedding, the latter is compared to a direct SFC embedding approach. A single slice of type 1 is considered.

For the SFC deployment, the ILP-based SFC embedding algorithm is adapted from [Riggio et al., 2016]. Specifically, the objective function in [Riggio et al., 2016] is modified to allow the simultaneous embedding of multiple SFCs. Both sequential and joint SFC embedding schemes are performed. The proposed methods, where provisioning is done before a joint and sequential SFC embedding, are denoted respectively as `prov-joint-emb` and `prov-seq-emb`. Direct joint and sequential SFC

embedding are denoted as `dir-joint-emb` and `dir-seq-emb`, where prior provisioning is not considered.

The K-ary fat-tree infrastructure topology considered in Section 5.4.3.1 is used here again. The amount of network infrastructure resource available at each node and link of the infrastructure remains the same.

Figures 5.3a and 5.3b show respectively the cost and the required computing time for different number of SFCs belonging to Slice of type 1 to be embedded (ranging from 2 to 10). The embedding cost reflects the amount of infrastructure node and link resources used for embedding these SFCs. The proposed methods, *i.e.*, `prov-joint-emb` and `prov-seq-emb`, have similar cost performance as that of the direct embedding, *i.e.*, `dir-joint-emb` and `dir-seq-emb`. Nevertheless, as depicted in Figure 5.3b, the proposed approach is faster than a direct embedding, when either performing in a joint or sequential fashion. The difference increases with the number of SFCs to embed. Note that in the proposed approach (*i.e.*, `prov-joint-emb` or `prov-seq-emb`), the computing time for the provisioning step has been taken into account.



(a) (b)

Figure 5.3: (a) Embedding costs and (b) computing time of `prov-joint-emb`, `prov-seq-emb`, `dir-joint-emb`, and `dir-seq-emb` approaches as a function of the number of SFCs to embed.

5.5 Conclusion

This chapter proposes a method of resource provisioning for network slices. Resource demands at SFC and slice level are defined, both are associated with a virtual graph representing the VNFs (virtual nodes) and their interconnections. The SFC-level resource demand (SFC-RD) represent the resource requirements needed so as the VNFs can operate properly, whereas the slice-level demand (S-RD) of a given slice describe the aggregate resource requirements of all users associated to that slice.

We have shown that infrastructure resource provisioning can be formulated as a mixed integer linear programming problem with constraints differing from those considered, *e.g.*, in [Riggio et al., 2016, Bouten et al., 2017], since, for example, the resources of several infrastructure nodes have to be gathered to satisfy the slice resource demands associated to virtual nodes.

Two provisioning approaches have been introduced and evaluated: sequential (SP) and joint (JP). The SP approach provisions resources slice-by-slice. In the JP approach, all slices are considered simultaneously. The complexity increases exponentially with the size of the infrastructure network and with the number of slices. The SP approach is more efficient in terms of computing time. The price to be paid is a somewhat degraded link utilization and a higher embedding cost compared to the joint approach.

Once resources have been provisioned, the approach introduced in [Riggio et al., 2016, Bouten et al., 2017] may be used to deploy SFCs, but considering only a simplified infrastructure network reduced to the nodes and links which have provisioned resources. Numerical results show that provisioning and then deploying is more efficient in terms of computing time than direct SFC embedding.

In Chapter 6, the resource provisioning approach introduced in this chapter is extended to account for some coverage constraints.

Chapter 6

Coverage-Constrained Resource Provisioning

This chapter is based on the following publications

> *Q.-T. Luu, M. Kieffer, A. Mouradian, and S. Kerboeuf, "Coverage-Aware Resource Provisioning Method for Network Slicing," in IEEE/ACM Transactions on Networking, vol. 28, no. 6, pp. 2393-2406, Dec. 2020* [Luu et al., 2020a];
>
> *Q.-T. Luu, S. Kerboeuf, A. Mouradian, and M. Kieffer, "Radio Resource Provisioning for Network Slicing with Coverage Constraints," in Proc. IEEE International Conference on Communications (ICC), Dublin, Ireland, June, 2020, pp. 1-6* [Luu et al., 2020b].

This chapter aims to provide an answer to Challenge 2 introduced in Chapter 1. We extend the study considered in Chapter 5 by considering the problem of provisioning for joint core and radio access network resources, accounting some coverage constraints. To that end, the slice resource provisioning consists in finding (*i*) a set of Base Stations (BS) that sufficiently provides radio resources to mobile users so as to satisfy slice coverage constraints; (*ii*) the placement of the VNFs on the data center nodes; and (*iii*) the routing of data flows between the VNFs, while respecting the structure of SFCs and optimizing a given objective (*e.g.*, minimizing the infrastructure and software fees cost). Updates may be necessary when the service characteristics have changed significantly.

As mentioned in Section 4.2, in this chapter, we consider a time scale of one time slot. The slice duration over which the slice is active is equal to one time slot. The processing of the slice provisioning is done *just-in time*, meaning that it is performed on the time slot just prior the one over which the slice has to be activated. The provisioning processing consider all slice requests that need to be activated in the next time slot.

The rest of the chapter is structured as follows. The main contributions of this chapter are summarized in Section 6.1. In Section 6.2, the system model used in this chapter is presented. The problem of slice resource provisioning is then formulated in Section 6.3 as a mixed integer linear programming problem accounting for cloud network and radio resource constraints for the deployment of multiple slices. An

optimal and four suboptimal variants of a coverage-constrained slice resource provisioning algorithm are provided in Section 6.4. Numerical results are presented in Section 6.5. Finally, Section 6.6 draws some conclusions and perspectives.

6.1 Contributions

A literature review on the problem of slice resource allocation with coverage constraints has been carried out in Section 3.2. In what follows, we present the main contributions of this chapter.

Compared to the related works [Lee et al., 2016, Chatterjee et al., 2018, Teague et al., 2018, D'Oro et al., 2018] reported in Section 3.2, in this chapter, we consider the slice resource demands in terms of coverage and traffic requirements in the radio access part of the network as well as network, memory, and computing requirements from a cloud infrastructure of interconnected data centers for the rest of the network. Extending the study introduced in Chapter 5, the work in this chapter adapts to get a the joint radio and network infrastructure resource provisioning approach. Constraints related to the infrastructure network considered in [Riggio et al., 2016, Vizarreta et al., 2017, Luu et al., 2018, Halabian, 2019] are combined with coverage and radio resource constraints introduced in [Lee et al., 2016, Chatterjee et al., 2018, Teague et al., 2018, D'Oro et al., 2018]. The coverage constraints are very important to satisfy mobile service requirements.

When provisioning resources for slices, some coverage constraints are considered, in which slices are assumed to cover a specific region in some geographical areas, that is part of the SLA between the MNO and the SP (SM-SLA). The amount of radio resources required depends on the location of users. A simple radio propagation model is thus introduced in the provisioning phase. The coverage constraints reduce the flexibility to select the nodes on which SFCs are deployed. In our model, radio resource blocks are allocated and the channel between the RRH nodes and users is taken into account. Compared with [Chatterjee et al., 2018], the selected BS is not necessarily the nearest one. Moreover, both downlink and uplink traffic are considered for the service rate model.

6.2 System Model

As discussed in Section 4.2, in this chapter, a *just-in-time* provisioning approach is applied, in which one focuses on a given time slot and on the requests of slices that have to be activated in the next time slot. The provisioning processing considers only slice requests that need to be activated in the next time slot. The set of those slice requests is denoted as $\mathcal{S}$.

Consider a set of SPs whose aim is to provide different services, indexed by $s = 1, \ldots, |\mathcal{S}|$, to mobile users. The geographical area under study is denoted by $\mathcal{A} \subset \mathbb{R}^2$ and the subarea over which service s has to be made available is denoted

by $\mathcal{A}_s$. For that purpose, each SP forwards his service requirements to the MNO, whose aim is to design a network slice able to satisfy these requirements.

Figure 6.1 illustrates three typical geographical subareas over which three different services have to be deployed.

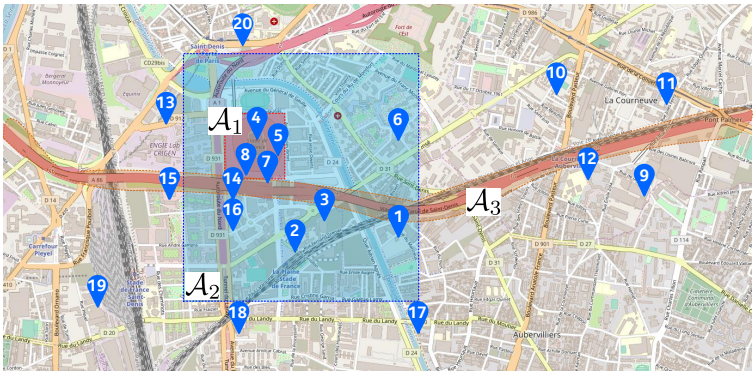


Figure 6.1: The considered metropolitan area including the Stade de France (covered by the red rectangle representing $\mathcal{A}_1$), its surrounding (blue rectangle representing $\mathcal{A}_2$), and part of the A86 highway (orange shape representing $\mathcal{A}_3$); Blue markers show the location of RRH nodes of Orange.

In this chapter, the S-RD sent by the MNO to the InP within the MI-SLA consists of (*i*) virtual graph $\mathcal{G}_s$ accounting for the structure and SLA of the slices $s \in \mathcal{S}$; and (*ii*) the S-RD coverage information related to the area $\mathcal{A}_s$ over which the service will have to be made available.

The InP is then in charge of provisioning enough infrastructure resources to deploy the SFCs whose resource demands have been described by the virtual graph $\mathcal{G}_s$.

The following sections detail the model of the infrastructure provided by the InP and the way a service with wireless coverage constraints can be mapped to a slice with specific virtual graph. Table 6.1 summarizes the newly introduced notations in this chapter.

Table 6.1: Newly introduced notations in Chapter 6.

Symbol	*Description*
Υ	Set of node resource types, $\Upsilon = \{\mathrm{c}, \mathrm{m}, \mathrm{r}\}$
$\mathcal{N}_\mathrm{r}$	Set of RRH nodes, $\mathcal{N}_\mathrm{r} \subset \mathcal{N}$
$\mathcal{A}_s$	Coverage area of slice s
$\mathcal{Q}_s$	Set of all divided subareas in $\mathcal{A}_s$
$\mathcal{A}_{s,q}$	Subarea of index $q \in \mathcal{Q}_s$
$\rho_s(x)$	Maximum user/device density at $x \in \mathcal{A}$
$R_{s,\mathrm{u}}$	Aggregate uplink data rate demands
$R_{s,\mathrm{d}}$	Aggregate downlink data rate demands

6.2.1 Infrastructure model

We consider an infrastructure graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ similar to that in Chapter 6. Instead of wireless resources, here we consider radio resources in terms of radio resource blocks. Radio resources are exclusively provided by a subset $\mathcal{N}_\mathrm{r} \subset \mathcal{N}$ of RRH nodes, whose location in some Cartesian frame attached to $\mathcal{A}$ is denoted by x_i^r. Each RRH node $i \in \mathcal{N}_\mathrm{r}$ has an amount of $a_\mathrm{r}(i)$ RBs. The cost of using an RB is $c_\mathrm{r}(i)$.

6.2.2 S-RD Model

The SLA between the the SP and the MNO (SM-SLA) describes the slice needs within its lifetime. This SLA is also expressed in terms of supported service type and targeted QoS such as a minimum average data rate $U_{s,\mathrm{u}}$ and $U_{s,\mathrm{d}}$[1] for the wireless uplink and downlink traffic required by each user. The geographical distribution function $\rho_s(x)$, with $x \in \mathcal{A}$, describes the *maximum* density of UEs around location x.

The virtual graph $\mathcal{G}_s$ of a given slice s has the following characteristics:

Each virtual node $v \in \mathcal{N}_s$ is characterized by a given amount of *required* computing and memory resources, denoted as $R_{s,\mathrm{c}}(v)$ and $R_{s,\mathrm{m}}(v)$ to sustain the aggregate demand for all instances of a given VNF in the slice. The minimum resources to deploy a single VNF instance are denoted as $r_\mathrm{c}(v)$ and $r_\mathrm{m}(v)$.

Each link $vw \in \mathcal{E}_s$, connecting node v to w in the virtual graph, is characterized by the bandwidth $R_{s,\mathrm{b}}(vw)$ required to sustain the aggregate traffic demand between the VNFs associated to v and w.

In the virtual graph $\mathcal{G}_s$, one assumes that the uplink and downlink radio resource demands are associated to a single node v_r. The aggregate uplink and downlink data rates $R_{s,\mathrm{u}}(v_\mathrm{r})$ and $R_{s,\mathrm{d}}(v_\mathrm{r})$ are associated to the coverage constraint of slice s

$$\begin{aligned} R_{s,\mathrm{u}}(v_\mathrm{r}) &= U_{s,\mathrm{u}} \int_{\mathcal{A}_s} \rho_s(x)\,\mathrm{d}x, \\ R_{s,\mathrm{d}}(v_\mathrm{r}) &= U_{s,\mathrm{d}} \int_{\mathcal{A}_s} \rho_s(x)\,\mathrm{d}x. \end{aligned} \tag{6.1}$$

6.3 Problem Formulation

6.3.1 Accounting for S-RD Coverage Constraints

For the slice s, the InP has to provide a minimum average data rate ($u_{s,\mathrm{u}}$ for uplink and $u_{s,\mathrm{d}}$ for downlink) to each mobile user spread over $\mathcal{A}_s$ with a density $\rho_s(x)$. For that purpose, the InP will have to provision resources from the physical RRH nodes in $\mathcal{N}_\mathrm{r}$. One assumes that every RRH node is able to provide a fixed amount $a_\mathrm{r}(i)$ of resource blocks (RB) per time unit to exchange data (up and downlink) with users.

[1] The notation U is reused in Chapters 7 and 8, but referring to the resource demand of different types of a typical user (*i.e.*, U-RD) of a slice.

The amount of data transmitted using a single RB depends on the characteristics of the RRH, of the User Equipment (UE), and on the transmission channel between the RRH and the user.

During the resource provisioning phase, the locations of users are unknown. Various approaches can be used to address this problem. One approach is to consider different realizations of a point process representing the location of users can be considered, *e.g.*, in [Teague et al., 2018]. In this work, we consider an approach inspired by the subarea partitioning technique introduced in [Shi and Hou, 2007]. $\mathcal{A}_s$ is partitioned into Q_s convex subareas $\mathcal{A}_{s,q}$, $q \in \mathcal{Q}_s = \{1, \ldots, Q_s\}$. Instead of allocating RBs to users, RRH nodes allocate RBs to subareas. The way the partitioning of $\mathcal{A}$ is performed is not detailed here. One may consider, *e.g.*, a partitioning into squares of equal surfaces or a partitioning based on ρ_s that provides an equal average number of users per subarea.

One introduces the set

$$\boldsymbol{\eta}_s = \left\{ \eta_{s,\mathrm{u}}\left(i,q\right), \eta_{s,\mathrm{d}}\left(i,q\right), \widetilde{\eta}_s\left(i\right) \right\}_{i \in \mathcal{N}_\mathrm{r}, q \in \mathcal{Q}_s},$$

where $\eta_{s,\mathrm{u}}\left(i,q\right) \in [0,1]$ and $\eta_{s,\mathrm{d}}\left(i,q\right) \in [0,1]$ represent the proportion of RBs provisioned by RRH i to users in $\mathcal{A}_{s,q}$ for uplink and downlink traffic. These quantities represent average proportions of RBs available during some typical interval of time and provisioned by RRH i. The time interval may be, *e.g.*, of one second[2].

$\widetilde{\eta}_s\left(i\right) \in \{0,1\}$ indicates whether an RRH $i \in \mathcal{N}_\mathrm{r}$ has provisioned some RBs to any subarea for slice s, *i.e.*, $\widetilde{\eta}_s\left(i\right) = 1$ if $\sum_{q \in \mathcal{Q}_s} \left(\eta_{s,\mathrm{u}}\left(i,q\right) + \eta_{s,\mathrm{d}}\left(i,q\right) \right) > 0$, and $\widetilde{\eta}_s\left(i\right) = 0$ otherwise.

To ensure the provisioned RBs do not exceed the RRH capacity, the summed proportions of RBs provided by a given RRH i must be less than one

$$\sum_{s \in \mathcal{S}} \sum_{q \in \mathcal{Q}_s} \left(\eta_{s,\mathrm{u}}\left(i,q\right) + \eta_{s,\mathrm{d}}\left(i,q\right) \right) \leqslant 1, \, \forall i \in \mathcal{N}_\mathrm{r}. \tag{6.2}$$

For each slice s and each subarea $\mathcal{A}_{s,q}$, the total data rate provided by the allocated resource blocks should satisfy the minimum average user demand. Then, $\forall q \in \mathcal{Q}_s, \forall s \in \mathcal{S}$, one should have

$$\sum_{i \in \mathcal{N}_\mathrm{r}} \eta_{s,\mathrm{u}}\left(i,q\right) a_\mathrm{r}\left(i\right) b_\mathrm{u}\left(x_i^\mathrm{r}, \mathcal{A}_{s,q}\right) \geqslant U_{s,\mathrm{u}} \int_{\mathcal{A}_{s,q}} \rho_s\left(x\right) \mathrm{d}x, \tag{6.3}$$

$$\sum_{i \in \mathcal{N}_\mathrm{r}} \eta_{s,\mathrm{d}}\left(i,q\right) a_\mathrm{r}\left(i\right) b_\mathrm{d}\left(x_i^\mathrm{r}, \mathcal{A}_{s,q}\right) \geqslant U_{s,\mathrm{d}} \int_{\mathcal{A}_{s,q}} \rho_s\left(x\right) \mathrm{d}x, \tag{6.4}$$

which correspond to the satisfaction of the geographical coverage constraints for uplink and downlink traffic. Here, $b_\mathrm{u}\left(x_i^\mathrm{r}, \mathcal{A}_{s,q}\right)$ and $b_\mathrm{d}\left(x_i^\mathrm{r}, \mathcal{A}_{s,q}\right)$ denote the amount of data (bits) carried by a RB for a user located in $\mathcal{A}_{s,q}$ for up and downlink. Depending on the level of conservatism, $b_\mathrm{u}\left(x_i^\mathrm{r}, \mathcal{A}_{s,q}\right)$ and $b_\mathrm{d}\left(x_i^\mathrm{r}, \mathcal{A}_{s,q}\right)$ may represent

[2]Since $\eta_{s,\mathrm{u}}\left(i,q\right)$ and $\eta_{s,\mathrm{d}}\left(i,q\right)$ are averages, they may be accurately represented by real numbers in the range $[0,1]$, even if in reality both quantities should be rational numbers.

the minimum or the average amount of data evaluated over the possible locations of users in $\mathcal{A}_{s,q}$. The terms $b_{\mathrm{u}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right)$, $b_{\mathrm{d}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right)$, and $\int_{\mathcal{A}_{s,q}} \rho_s\left(x\right) \mathrm{d}x$ are fixed quantities that only depend on the RRH location x_i^{r}, on the user density ρ_s, and on the way the partitioning of $\mathcal{A}_s$ has been performed. These terms may thus be evaluated in advance, see Section 6.5.1.3.

Summing (6.3) over all $q \in \mathcal{Q}_s$ and using (6.1), one gets

$$\sum_{q \in \mathcal{Q}_s} \sum_{i \in \mathcal{N}_{\mathrm{r}}} \eta_{s,\mathrm{u}}\left(i,q\right) a_{\mathrm{r}}\left(i\right) b_{\mathrm{u}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right) \geqslant R_{s,\mathrm{u}}\left(v_{\mathrm{r}}\right), \tag{6.5}$$

$$\sum_{q \in \mathcal{Q}_s} \sum_{i \in \mathcal{N}_{\mathrm{r}}} \eta_{s,\mathrm{d}}\left(i,q\right) a_{\mathrm{r}}\left(i\right) b_{\mathrm{d}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right) \geqslant R_{s,\mathrm{d}}\left(v_{\mathrm{r}}\right), \tag{6.6}$$

which ensure, for slice s, the satisfaction of the part of the S-RD related to the uplink and downlink radio resource demands.

For each RRH i, the amount of provisioned uplink and downlink resources should be proportional to the demand expressed in the S-RD through $R_{s,\mathrm{u}}\left(v_{\mathrm{r}}\right)$ and $R_{s,\mathrm{d}}\left(v_{\mathrm{r}}\right)$. This avoids provisioning RRH resources taking care only of the uplink or only of the downlink traffic. This has to be ensured for all subareas $q \in \mathcal{Q}_s$

$$\frac{\eta_{s,\mathrm{u}}\left(i,q\right) a_{\mathrm{r}}\left(i\right) b_{\mathrm{u}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right)}{R_{s,\mathrm{u}}\left(v_{\mathrm{r}}\right)} = \frac{\eta_{s,\mathrm{d}}\left(i,q\right) a_{\mathrm{r}}\left(i\right) b_{\mathrm{d}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right)}{R_{s,\mathrm{d}}\left(v_{\mathrm{r}}\right)}. \tag{6.7}$$

The relation between $\eta_s\left(i,q\right)$ and $\widetilde{\eta}_s\left(i\right)$ is nonlinear. Nevertheless, both quantities can be linked with the following linear constraints, $\forall s \in \mathcal{S}$, $i \in \mathcal{N}_{\mathrm{r}}$,

$$0 \leq \widetilde{\eta}_s\left(i\right) - \sum_{q \in \mathcal{Q}_s} \eta_s\left(i,q\right) < 1, \tag{6.8}$$

where $\eta_s\left(i,q\right) = \eta_{s,\mathrm{u}}\left(i,q\right) + \eta_{s,\mathrm{d}}\left(i,q\right)$.

The leasing cost related to the radio resource provisioning for a given slice s gathers the fixed costs $c_{\mathrm{f}}\left(i\right) \widetilde{\eta}_s\left(i\right)$ related to the use of a RRH by the slice and the variable costs $c_{\mathrm{r}}\left(i\right) a_{\mathrm{r}}\left(i\right) \eta_s\left(i,q\right)$ related to the amount of RBs provided by each RRH to the slice. A bias towards RB allocation by RRHs providing a high spectral efficiency is obtained by the introduction of a rate-related discount $\lambda b\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right) a_{\mathrm{r}}\left(i\right) \eta_s\left(i,q\right)$, where λ is a positive discount factor. The resulting cost function for the *radio* resources is

$$c_{\mathrm{rr}}\left(\boldsymbol{\eta}\right) = \sum_{s \in \mathcal{S}} c_{s,\mathrm{rr}}\left(\boldsymbol{\eta}_s\right), \tag{6.9}$$

where $c_{s,\mathrm{rr}}\left(\boldsymbol{\eta}_s\right)$ is the radio provisioning cost of a single slice s, given by

$$\begin{aligned} c_{s,\mathrm{rr}}\left(\boldsymbol{\eta}_s\right) = & \sum_{i \in \mathcal{N}_{\mathrm{r}}} c_{\mathrm{f}}\left(i\right) \widetilde{\eta}_s\left(i\right) \\ & + \sum_{i \in \mathcal{N}_{\mathrm{r}}} \sum_{q \in \mathcal{Q}_s} \left[c_{\mathrm{r}}\left(i\right) - \lambda b_{\mathrm{u}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right)\right] a_{\mathrm{r}}\left(i\right) \eta_{s,\mathrm{u}}\left(i,q\right) \\ & + \sum_{i \in \mathcal{N}_{\mathrm{r}}} \sum_{q \in \mathcal{Q}_s} \left[c_{\mathrm{r}}\left(i\right) - \lambda b_{\mathrm{d}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right)\right] a_{\mathrm{r}}\left(i\right) \eta_{s,\mathrm{d}}\left(i,q\right). \end{aligned} \tag{6.10}$$

Finally, The *Radio resource Provisioning* problem, denoted by RP, is summarized in Problem 6.1.

Problem 6.1: Radio Resource Provisioning (RP)

$$
\begin{aligned}
&\min_{\boldsymbol{\eta}} \quad c_{\mathrm{rr}}\left(\boldsymbol{\eta}\right), \\
&\text{subject to} \\
&\quad \sum_{s\in\mathcal{S}}\sum_{q\in\mathcal{Q}_s}\left(\eta_{s,\mathrm{u}}\left(i,q\right)+\eta_{s,\mathrm{d}}\left(i,q\right)\right)\leqslant 1,\ \forall i\in\mathcal{N}_{\mathrm{r}}, \\
&\quad \sum_{i\in\mathcal{N}_{\mathrm{r}}}\eta_{s,\mathrm{u}}\left(i,q\right)a_{\mathrm{r}}\left(i\right)b_{\mathrm{u}}\left(x_i^{\mathrm{r}},\mathcal{A}_{s,q}\right)\geqslant U_{s,\mathrm{u}}\int_{\mathcal{A}_{s,q}}\rho_s\left(x\right)\mathrm{d}x,\forall s\in\mathcal{S},q\in\mathcal{Q}_s, \\
&\quad \sum_{i\in\mathcal{N}_{\mathrm{r}}}\eta_{s,\mathrm{d}}\left(i,q\right)a_{\mathrm{r}}\left(i\right)b_{\mathrm{d}}\left(x_i^{\mathrm{r}},\mathcal{A}_{s,q}\right)\geqslant U_{s,\mathrm{d}}\int_{\mathcal{A}_{s,q}}\rho_s\left(x\right)\mathrm{d}x,\forall s\in\mathcal{S},q\in\mathcal{Q}_s, \\
&\quad \sum_{q\in\mathcal{Q}_s}\sum_{i\in\mathcal{N}_{\mathrm{r}}}\eta_{s,\mathrm{u}}\left(i,q\right)a_{\mathrm{r}}\left(i\right)b_{\mathrm{u}}\left(x_i^{\mathrm{r}},\mathcal{A}_{s,q}\right)\geqslant R_{s,\mathrm{u}}\left(v_{\mathrm{r}}\right),\forall s\in\mathcal{S}, \\
&\quad \sum_{q\in\mathcal{Q}_s}\sum_{i\in\mathcal{N}_{\mathrm{r}}}\eta_{s,\mathrm{d}}\left(i,q\right)a_{\mathrm{r}}\left(i\right)b_{\mathrm{d}}\left(x_i^{\mathrm{r}},\mathcal{A}_{s,q}\right)\geqslant R_{s,\mathrm{d}}\left(v_{\mathrm{r}}\right),\forall s\in\mathcal{S}, \\
&\quad \frac{\eta_{s,\mathrm{u}}\left(i,q\right)a_{\mathrm{r}}\left(i\right)b_{\mathrm{u}}\left(x_i^{\mathrm{r}},\mathcal{A}_{s,q}\right)}{R_{s,\mathrm{u}}\left(v_{\mathrm{r}}\right)}=\frac{\eta_{s,\mathrm{d}}\left(i,q\right)a_{\mathrm{r}}\left(i\right)b_{\mathrm{d}}\left(x_i^{\mathrm{r}},\mathcal{A}_{s,q}\right)}{R_{s,\mathrm{d}}\left(v_{\mathrm{r}}\right)},\forall q\in\mathcal{Q}_s, \\
&\quad 0\leq\widetilde{\eta}_s\left(i\right)-\sum_{q\in\mathcal{Q}_s}\eta_s\left(i,q\right)<1,\forall s\in\mathcal{S},i\in\mathcal{N}_{\mathrm{r}}.
\end{aligned}
$$

6.3.2 Accounting for other S-RD Constraints

This section introduces a set of constraints which have to be satisfied to address the other resource demands for each $s\in\mathcal{S}$, while being consistent with the coverage constraints. For that purpose, the set of variables introduced in Chapter 5 is reused:

$$
\boldsymbol{\Phi}_s=\left\{\phi_{s,n}\left(i,v\right),\phi_{s,\mathrm{b}}\left(ij,vw\right),\widetilde{\phi}_s\left(i\right)\right\}_{(i,ij)\in\mathcal{G},(v,vw)\in\mathcal{G}_s,n\in\{\mathrm{c},\mathrm{m}\}},
$$

where $\phi_{s,n}\left(i,v\right)\in[0,1]$ represents the proportion of resources of type n provisioned on the physical node $i\in\mathcal{N}$ for the virtual node $v\in\mathcal{N}_s$ of the slice s;

$\phi_{s,\mathrm{b}}\left(ij,vw\right)\in[0,1]$ represents the proportion of bandwidth of the physical link $ij\in\mathcal{E}$ provisioned for the virtual link $vw\in\mathcal{E}_s$ of the slice s. When one of the variables $\phi_{s,n}\left(i,v\right)$ and $\phi_{s,\mathrm{b}}\left(ij,vw\right)$ holds zero, there is no mapping between the infrastructure and the virtual node/link;

$\widetilde{\phi}_s\left(i\right)\in\{0,1\}$ indicates whether an infrastructure node i have provisioned resources for some virtual node of slice s. $\widetilde{\phi}_s\left(i\right)=1$ if at least one of the elements of $\{\phi_{s,\mathrm{c}}\left(i,v\right),\phi_{s,\mathrm{m}}\left(i,v\right)\}_{v\in\mathcal{N}_s}$ is strictly positive, and $\widetilde{\phi}_s\left(i\right)=0$ otherwise.

In what follows, the constraints accounted for other S-RD constraints are described. Some constraints have already been presented in Chapter 5.

The provisioned infrastructure resources should satisfy the resource demands

of each slice node and link. This leads to

$$\sum_{i\in\mathcal{N}} a_n(i)\,\phi_{s,n}(i,v) \geq R_n(v), \forall s\in\mathcal{S}, v\in\mathcal{N}_s, n\in\{\mathrm{c},\mathrm{m}\}, \tag{6.11}$$

$$\sum_{ij\in\mathcal{E}} a_{\mathrm{b}}(ij)\,\phi_{s,\mathrm{b}}(ij,vw) \geq R_{\mathrm{b}}(vw), \forall s\in\mathcal{S}, vw\in\mathcal{E}_s. \tag{6.12}$$

The resources provisioned by a given infrastructure node/link should not exceed the available resource at that node/link. One has

$$\sum_{s\in\mathcal{S}}\sum_{v\in\mathcal{N}_s} \phi_{s,n}(i,v) \leq 1, \forall n\in\{\mathrm{c},\mathrm{m}\}, i\in\mathcal{N}, \tag{6.13}$$

$$\sum_{s\in\mathcal{S}}\sum_{vw\in\mathcal{E}_s} \phi_{s,\mathrm{b}}(ij,vw) \leq 1, \forall ij\in\mathcal{E}. \tag{6.14}$$

The amount of resources provisioned by a given infrastructure node i for a virtual node v has to be equal to an integer multiple of the minimum amount of resources $r_n(v)$ for a VNF associated to the virtual node v

$$a_n(i)\,\phi_{s,n}(i,v) = \kappa_{s,n}(i,v)\,r_n(v), \forall i\in\mathcal{N}, v\in\mathcal{N}_s, n\in\{\mathrm{c},\mathrm{m}\}, \tag{6.15}$$

where $\kappa_{s,n}(i,v)$ is a positive integer belonging to the set of variables of the optimization problem. This constraint has been discussed in Chapter 5, Eq. (5.8).

As also discussed in Chapter 5, provisioning of infrastructure node and link resources has to be performed in a balanced way, consistent with the virtual graph. This translates into the following resource provisioning proportionality constraints, for each $s\in\mathcal{S}$ and $v\in\mathcal{N}_s$,

$$\frac{a_{\mathrm{c}}(i)\,\phi_{s,\mathrm{c}}(i,v)}{R_{s,\mathrm{c}}(v)} = \frac{a_{\mathrm{s}}(i)\,\phi_{s,\mathrm{m}}(i,v)}{R_{s,\mathrm{m}}(v)}, \ \forall i\in\mathcal{N}, \tag{6.16}$$

$$\overset{\text{also}}{=} \frac{a_{\mathrm{r}}(i)}{R_{s,\mathrm{r}}(v_{\mathrm{r}})}\sum_{q\in\mathcal{Q}_s}\left[\begin{array}{l}\eta_{s,\mathrm{u}}(i,q)\,b_{\mathrm{u}}(x_i^{\mathrm{r}},\mathcal{A}_{s,q})\\ +\eta_{s,\mathrm{d}}(i,q)\,b_{\mathrm{d}}(x_i^{\mathrm{r}},\mathcal{A}_{s,q})\end{array}\right], \forall i\in\mathcal{N}_{\mathrm{r}}. \tag{6.17}$$

The constraints (6.16) and (6.17) ensure a balanced resource provisioning by infrastructure nodes. In (6.17), $R_{s,\mathrm{r}}(v_{\mathrm{r}})$ is the total radio resource demand of v_{r} of slice s in both up and downlink, *i.e.*, $R_{s,\mathrm{r}}(v_{\mathrm{r}}) = R_{s,\mathrm{u}}(v_{\mathrm{r}}) + R_{s,\mathrm{d}}(v_{\mathrm{r}})$.

Moreover, link resources should be consistently provisioned with the radio resource of the RRH for both uplink and downlink. Thus, for downlink traffic (links with RRH as egress), one should have for each $s\in\mathcal{S}$, $j\in\mathcal{N}_{\mathrm{r}}$, and $vv_{\mathrm{r}}\in\mathcal{E}_s$,

$$\sum_{i\in\mathcal{N}\setminus\mathcal{N}_{\mathrm{r}}} \frac{a_{\mathrm{b}}(ij)\,\phi_{s,\mathrm{b}}(ij,vv_{\mathrm{r}})}{R_{s,\mathrm{b}}(vv_{\mathrm{r}})} = \Big(\frac{R_{s,\mathrm{b}}(vv_{\mathrm{r}})}{\sum_{uv_{\mathrm{r}}\in\mathcal{E}_s} R_{s,\mathrm{b}}(uv_{\mathrm{r}})}\Big)\frac{a_{\mathrm{r}}(j)}{R_{s,\mathrm{d}}(v_{\mathrm{r}})}\sum_{q\in\mathcal{Q}_s}\eta_{s,\mathrm{d}}(j,q)\,b_{\mathrm{d}}\left(x_j^{\mathrm{r}},\mathcal{A}_{s,q}\right). \tag{6.18}$$

In (6.18), the term $\frac{a_{\mathrm{r}}(j)\sum_{q\in\mathcal{Q}_s}\eta_{s,\mathrm{d}}(j,q)b_{\mathrm{d}}\left(x_j^{\mathrm{r}},\mathcal{A}_{s,q}\right)}{R_{s,\mathrm{d}}(v_{\mathrm{r}})}$ represents the proportion of downlink radio resources provided by RRH j to satisfy the downlink demand of v_{r}. When

several virtual links feed v_r, the term $\frac{R_{s,\mathrm{b}}(vv_\mathrm{r})}{\sum_{uv_\mathrm{r}\in\mathcal{E}_s} R_{s,\mathrm{b}}(uv_\mathrm{r})}$ represents the proportion of (downlink) traffic demand associated to the virtual link vv_r. The right-hand side of (6.18) represents thus the proportion of the data traffic that *has to be provisioned* for the virtual link vv_r to satisfy the part of the downlink radio resource provided by RRH j to satisfy the part of the downlink demand of v_r. The left-hand side of (6.18), represents the proportion of the data traffic that *is provided* by all infrastructure links ij, $i \in \mathcal{N}\backslash\mathcal{N}_\mathrm{r}$ for the virtual link vv_r. Both terms have thus to be equal.

For uplink traffic (links with RRH as ingress), one has, for each $s \in \mathcal{S}$, $i \in \mathcal{N}_\mathrm{r}$, and $v_\mathrm{r}v \in \mathcal{E}_s$,

$$\sum_{j\in\mathcal{N}\backslash\mathcal{N}_\mathrm{r}} \frac{a_\mathrm{b}(ij)\,\phi_{s,\mathrm{b}}(ij,v_\mathrm{r}v)}{R_{s,\mathrm{b}}(v_\mathrm{r}v)} = \Big(\frac{R_{s,\mathrm{b}}(v_\mathrm{r}v)}{\sum_{v_\mathrm{r}u\in\mathcal{E}_s} R_{s,\mathrm{b}}(v_\mathrm{r}u)}\Big)\frac{a_\mathrm{r}(i)}{R_{s,\mathrm{u}}(v_\mathrm{r})}\sum_{q\in\mathcal{Q}_s}\eta_{s,\mathrm{u}}(i,q)\,b_\mathrm{u}(x_i^\mathrm{r},\mathcal{A}_{s,q})\,. \tag{6.19}$$

In (6.19), the term $\frac{a_\mathrm{r}(i)\sum_{q\in\mathcal{Q}_s}\eta_{s,\mathrm{u}}(i,q)b_\mathrm{u}(x_i^\mathrm{r},\mathcal{A}_{s,q})}{R_{s,\mathrm{u}}(v_\mathrm{r})}$ represents now the proportion of uplink radio resources provided by RRH i to satisfy the uplink demand of v_r. When several virtual links depart from v_r, the term $\frac{R_{s,\mathrm{b}}(v_\mathrm{r}v)}{\sum_{v_\mathrm{r}u\in\mathcal{E}_s} R_{s,\mathrm{b}}(v_\mathrm{r}u)}$ represents the proportion of (uplink) traffic demand associated to the virtual link $v_\mathrm{r}v$. The right-hand side of (6.19) represents thus the proportion of the data traffic that *has to be provisioned* for the virtual link $v_\mathrm{r}v$ to convey the part of the uplink radio resource provided by RRH i to satisfy the part of the uplink demand of v_r. The left-hand side of (6.19), represents the proportion of the data traffic that *is provided* by all infrastructure links ij, $j \in \mathcal{N}\backslash\mathcal{N}_\mathrm{r}$ for the virtual link $v_\mathrm{r}v$. Both terms have again to be equal. Combined with (6.6), the constraints (6.18) and (6.19) impose that the total radio resources provisioned by the RRHs are above the required resources $R_{s,\mathrm{d}}(v_\mathrm{r})$ and $R_{s,\mathrm{u}}(v_\mathrm{r})$.

Finally, Eq. (6.20) describes the flow conservation constraints, which have to be satisfied when resources are provisioned on the infrastructure link ij for the virtual link vw.

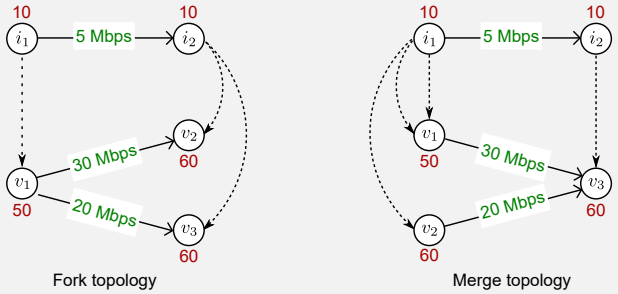
$$\begin{aligned}&\sum_{j\in\mathcal{N}}\left[\frac{a_\mathrm{b}(ij)\,\phi_{s,\mathrm{b}}(ij,vw)}{R_{s,\mathrm{b}}(vw)} - \frac{a_\mathrm{b}(ji)\,\phi_{s,\mathrm{b}}(ji,vw)}{R_{s,\mathrm{b}}(vw)}\right]\\ &= \Big(\frac{R_{s,\mathrm{b}}(vw)}{\sum_{vu\in\mathcal{E}_s} R_{s,\mathrm{b}}(vu)}\Big)\frac{a_\mathrm{c}(i)}{R_{s,\mathrm{c}}(v)}\phi_{s,\mathrm{c}}(i,v) - \Big(\frac{R_{s,\mathrm{b}}(vw)}{\sum_{uw\in\mathcal{E}_s} R_{s,\mathrm{b}}(uw)}\Big)\frac{a_\mathrm{c}(i)}{R_{s,\mathrm{c}}(w)}\phi_{s,\mathrm{c}}(i,w)\,.\end{aligned} \tag{6.20}$$

Notice that 6.20 is an extension to the flow conservation constraints introduced in Chapter 5, allowing the mapping of an infrastructure node to multiple virtual nodes of a slice with branched topology. In (6.20), when several virtual links depart from v, the term $\frac{R_{s,\mathrm{b}}(vw)}{\sum_{vu\in\mathcal{E}_s} R_{s,\mathrm{b}}(vu)}$ represents the proportion of traffic demand that departs from v associated to the virtual link vw. Example 6.1 provides an illustration to the flow conservation constraint (6.20), considering the virtual graph with branched topologies.

In (6.20), the consistency with the other provisioned resources is ensured by (6.16).

Note that the flow conservation constraints (6.20) imposes a relation between the different $\phi_{s,n}(i,v)$ for different i and v. Since $\phi_{s,n}(i,v)$ and $\kappa_{s,n}(i,v)$ are proportional (see (6.15)), the relations between $\kappa_{s,n}(i,v)$ for different i and v are also imposed without specifying any additional constraint.

Example 6.1 (Illustration of constraint (6.20))**.** The figures below illustrate the constraint (6.20) considering an S-RD weighted virtual graph with a fork (left figure) and merge (right figure) topology. The nodes i_1, i_2 and the link i_1i_2 belong to the infrastructure graph; The nodes v_1, v_2, v_3 and the links connecting them belong to the virtual graph.



In the fork topology example, the node i_1 is mapped onto v_1, i_2 is mapped onto the pair (v_2, v_3); and the link i_1i_2 is mapped onto the pair (v_1v_2, v_1v_3). Considering the infrastructure node i_1 and the virtual link v_1v_2, the constraint (6.20) leads to

$$\frac{5}{30}\phi_{s,\mathrm{b}}(i_1i_2, v_1v_2) - 0 = \frac{30}{50}\frac{10}{50}\phi_{s,n}(i_1, v_1) - 0,$$

yielding $\phi_{s,\mathrm{b}}(i_1i_2, v_1v_2) = \frac{18}{25}\phi_{s,n}(i_1, v_1)$. Similarly, considering i_1 and v_1v_3, one gets $\phi_{s,\mathrm{b}}(i_1i_2, v_1v_3) = \frac{8}{25}\phi_{s,n}(i_1, v_1)$. Thus

$$\phi_{s,\mathrm{b}}(i_1i_2, v_1v_2) + \phi_{s,\mathrm{b}}(i_1i_2, v_1v_3) = \frac{26}{25}\phi_{s,n}(i_1, v_1).$$

Here $\phi_{s,\mathrm{b}}(i_1i_2, v_1v_2) + \phi_{s,\mathrm{b}}(i_1i_2, v_1v_3)$ is the total resources provisioned by the link i_1i_2 to the virtual links v_1v_2 and v_1v_3, and has to be less than or equal to 1. Therefore, the largest value that $\phi_{s,n}(i_1, v_1)$ can take is $\frac{25}{26}$, which leads to $\phi_{s,\mathrm{b}}(i_1i_2, v_1v_2) = \frac{8}{26}$ and $\phi_{s,\mathrm{b}}(i_1i_2, v_1v_3) = \frac{18}{26}$.

In the merge topology example, through similar calculations, one gets

$$\begin{cases} \phi_{s,n}(i_1, v_1) = \frac{1}{2}; & \phi_{s,\mathrm{b}}(i_1i_2, v_1v_3) = \frac{9}{15}; \\ \phi_{s,n}(i_1, v_2) = \frac{2}{5}; & \phi_{s,\mathrm{b}}(i_1i_2, v_2v_3) = \frac{4}{15}; \\ \phi_{s,n}(i_2, v_3) = 1. \end{cases}$$

The proportions of provisioned infrastructure node and link resources are then consistent with the proportions of node and link resource demands. The proportionality of provisioned resources for links entering or leaving the same vertices is also ensured.

The relation between $\phi_{s,n}(i,v)$ and $\widetilde{\phi}_s(i)$ is again nonlinear. As in (6.8), both quantities may be linearly related as follows, for each $s \in \mathcal{S}$ and $i \in \mathcal{N}$,

$$\sum_{v \in \mathcal{N}_s} \sum_{n \in \{\mathrm{c},\mathrm{m}\}} \frac{\phi_{s,n}(i,v)}{2|\mathcal{N}_s|} \leq \widetilde{\phi}_s(i) < \sum_{v \in \mathcal{N}_s} \sum_{n \in \{\mathrm{c},\mathrm{m}\}} \frac{\phi_{s,n}(i,v)}{2|\mathcal{N}_s|} + 1. \tag{6.21}$$

The leasing cost related to the provisioning of computing, memory, and bandwidth resources in the *wired* part of the infrastructure network for all slices in $\mathcal{S}$ can be expressed as

$$c_{\mathrm{wr}}(\mathbf{\Phi}) = \sum_{s \in \mathcal{S}} c_{s,\mathrm{wr}}(\mathbf{\Phi}_s), \tag{6.22}$$

where $c_{s,\mathrm{wr}}(\mathbf{\Phi}_s)$ is the wired resource provisioning cost of a single slice s, given by

$$\begin{aligned} c_{s,\mathrm{wr}}(\mathbf{\Phi}_s) = & \sum_{i \in \mathcal{N} \setminus \mathcal{N}_{\mathrm{Ir}}} \widetilde{\phi}_s(i)\, c_{\mathrm{f}}(i) \\ & + \sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{N}_s} \sum_{n \in \{\mathrm{c},\mathrm{m}\}} a_n(i)\, \phi_{s,n}(i,v)\, c_n(i) \\ & + \sum_{ij \in \mathcal{E}} \sum_{vw \in \mathcal{E}_s} a_{\mathrm{b}}(ij)\, \phi_{s,\mathrm{b}}(ij,vw)\, c_{\mathrm{b}}(ij), \end{aligned} \tag{6.23}$$

where the first term represents the cost for deploying VNFs in infrastructure nodes, while the second and the third term indicate the total cost for leasing resources from infrastructure nodes and links. In the first term, the fixed infrastructure node disposal cost related to RRH nodes is not considered, since it has already been taken into account in (6.9).

Finally, the *Network resource Provisioning*, denoted by NP, is summarized in Problem 6.2.

Problem 6.2: Network Resource Provisioning (NP)

$$\begin{aligned} & \underset{\mathbf{\Phi}}{\text{minimize}} \quad c_{\mathrm{wr}}(\mathbf{\Phi}), \\ & \text{subject to} \\ & \sum_{i \in \mathcal{N}} a_n(i)\, \phi_{s,n}(i,v) \geq R_n(v), \forall v \in \mathcal{N}_s, n \in \{\mathrm{c},\mathrm{m}\}, \\ & \sum_{ij \in \mathcal{E}} a_{\mathrm{b}}(ij)\, \phi_{s,\mathrm{b}}(ij,vw) \geq R_{\mathrm{b}}(vw), \forall vw \in \mathcal{E}_s, \\ & \sum_{s \in \mathcal{S}} \sum_{v \in \mathcal{N}_s} \phi_{s,n}(i,v) \leq 1, \forall n \in \{\mathrm{c},\mathrm{m}\}, i \in \mathcal{N}, \\ & \sum_{s \in \mathcal{S}} \sum_{vw \in \mathcal{E}_s} \phi_{s,\mathrm{b}}(ij,vw) \leq 1, \forall ij \in \mathcal{E}, \\ & a_n(i)\, \phi_{s,n}(i,v) = \kappa_{s,n}(i,v)\, r_n(v), \forall i \in \mathcal{N}, v \in \mathcal{N}_s, n \in \{\mathrm{c},\mathrm{m}\}, \\ & \frac{a_{\mathrm{c}}(i)\, \phi_{s,\mathrm{c}}(i,v)}{R_{s,\mathrm{c}}(v)} = \frac{a_{\mathrm{s}}(i)\, \phi_{s,\mathrm{m}}(i,v)}{R_{s,\mathrm{m}}(v)}, \ \forall i \in \mathcal{N}, v \in \mathcal{N}_s, \\ & \forall s \in \mathcal{S}, i \in \mathcal{N}_{\mathrm{r}}: \end{aligned}$$

$$\frac{a_{\mathrm{c}}(i)\,\phi_{s,\mathrm{c}}(i,v_{\mathrm{r}})}{R_{s,\mathrm{c}}(v_{\mathrm{r}})}=\frac{a_{\mathrm{s}}(i)\,\phi_{s,\mathrm{m}}(i,v_{\mathrm{r}})}{R_{s,\mathrm{m}}(v_{\mathrm{r}})}=\frac{a_{\mathrm{r}}(i)}{r_{\mathrm{r}}(v_{\mathrm{r}})}\sum_{q\in\mathcal{Q}_s}\left[\begin{array}{l}\eta_{s,\mathrm{u}}(i,q)\,b_{\mathrm{u}}(x_i^{\mathrm{r}},\mathcal{A}_{s,q})\\+\eta_{s,\mathrm{d}}(i,q)\,b_{\mathrm{d}}(x_i^{\mathrm{r}},\mathcal{A}_{s,q})\end{array}\right],$$

$$\forall s\in\mathcal{S},j\in\mathcal{N}_{\mathrm{r}},vv_{\mathrm{r}}\in\mathcal{E}_s:$$

$$\sum_{i\in\mathcal{N}\setminus\mathcal{N}_{\mathrm{r}}}\frac{a_{\mathrm{b}}(ij)\,\phi_{s,\mathrm{b}}(ij,vv_{\mathrm{r}})}{R_{s,\mathrm{b}}(vv_{\mathrm{r}})}=\Big(\frac{R_{s,\mathrm{b}}(vv_{\mathrm{r}})}{\sum_{uv_{\mathrm{r}}\in\mathcal{E}_s}R_{s,\mathrm{b}}(uv_{\mathrm{r}})}\Big)\frac{a_{\mathrm{r}}(j)}{R_{s,\mathrm{d}}(v_{\mathrm{r}})}\times\sum_{q\in\mathcal{Q}_s}\eta_{s,\mathrm{d}}(j,q)\,b_{\mathrm{d}}\left(x_j^{\mathrm{r}},\mathcal{A}_{s,q}\right),$$

$$\forall s\in\mathcal{S},i\in\mathcal{N}_{\mathrm{r}},v_{\mathrm{r}}v\in\mathcal{E}_s:$$

$$\sum_{j\in\mathcal{N}\setminus\mathcal{N}_{\mathrm{r}}}\frac{a_{\mathrm{b}}(ij)\,\phi_{s,\mathrm{b}}(ij,v_{\mathrm{r}}v)}{R_{s,\mathrm{b}}(v_{\mathrm{r}}v)}=\Big(\frac{R_{s,\mathrm{b}}(v_{\mathrm{r}}v)}{\sum_{v_{\mathrm{r}}u\in\mathcal{E}_s}R_{s,\mathrm{b}}(v_{\mathrm{r}}u)}\Big)\frac{a_{\mathrm{r}}(i)}{R_{s,\mathrm{u}}(v_{\mathrm{r}})}\times\sum_{q\in\mathcal{Q}_s}\eta_{s,\mathrm{u}}(i,q)\,b_{\mathrm{u}}(x_i^{\mathrm{r}},\mathcal{A}_{s,q}),$$

$$\forall s\in\mathcal{S},i\in\mathcal{N},vw\in\mathcal{E}_s:$$

$$\sum_{j\in\mathcal{N}}\left[\frac{a_{\mathrm{b}}(ij)\,\phi_{s,\mathrm{b}}(ij,vw)}{R_{s,\mathrm{b}}(vw)}-\frac{a_{\mathrm{b}}(ji)\,\phi_{s,\mathrm{b}}(ji,vw)}{R_{s,\mathrm{b}}(vw)}\right]=\Big(\frac{R_{s,\mathrm{b}}(vw)}{\sum_{vu\in\mathcal{E}_s}R_{s,\mathrm{b}}(vu)}\Big)\frac{a_{\mathrm{c}}(i)}{R_{s,\mathrm{c}}(v)}\phi_{s,\mathrm{c}}(i,v)-\Big(\frac{R_{s,\mathrm{b}}(vw)}{\sum_{uw\in\mathcal{E}_s}R_{s,\mathrm{b}}(uw)}\Big)\frac{a_{\mathrm{c}}(i)}{R_{s,\mathrm{c}}(w)}\phi_{s,\mathrm{c}}(i,w),$$

$$\forall s\in\mathcal{S},i\in\mathcal{N}:$$

$$\sum_{v\in\mathcal{N}_s}\sum_{n\in\{\mathrm{c},\mathrm{m}\}}\frac{\phi_{s,n}(i,v)}{2\left|\mathcal{N}_s\right|}\leq\widetilde{\phi}_s(i)<\sum_{v\in\mathcal{N}_s}\sum_{n\in\{\mathrm{c},\mathrm{m}\}}\frac{\phi_{s,n}(i,v)}{2\left|\mathcal{N}_s\right|}+1.$$

6.4 Single-Step vs Two-Step Slice Resource Provisioning

6.4.1 Single-Step Provisioning

The global provisioning problem has to account for memory and computing constraints, as well as coverage constraints. It leads to the minimization of the sum of the costs (6.9) and (6.22)

$$c_{\mathrm{tot}}(\boldsymbol{\eta},\boldsymbol{\Phi})=c_{\mathrm{rr}}(\boldsymbol{\eta})+c_{\mathrm{wr}}(\boldsymbol{\Phi}) \tag{6.24}$$

with the constraints introduced in Sections 6.3.1 and 6.3.2. The provisioning approach minimizing (6.24) and considering all slices jointly is denoted as `JRN` (Joint Radio and Network) provisioning and is summarized in Problem 6.3.

Problem 6.3: Global Slice Resource Provisioning (`JRN`)

$$\begin{array}{ll}\underset{\boldsymbol{\eta},\boldsymbol{\Phi}}{\text{minimize}} & c_{\mathrm{tot}}(\boldsymbol{\eta},\boldsymbol{\Phi})=c_{\mathrm{rr}}(\boldsymbol{\eta})+c_{\mathrm{wr}}(\boldsymbol{\Phi}),\\ \text{subject to} & \text{(6.2)–(6.8)}, \quad \text{(constraints of RP)}\\ & \text{(6.11)–(6.21)}. \quad \text{(constraints of NP)}\end{array}$$

6.4.2 Two-Step Provisioning

When the number of variables in $\mathbf{\Phi}$ and $\boldsymbol{\eta}$ increases, the problem may become intractable. Therefore, a two-step provisioning algorithm, denoted as CARP (Coverage-Aware Resource Provisioning), see Algorithm 6.1, is introduced where both terms of (6.24) are minimized separately. Problem 6.1 is solved first, followed by the solving of Problem 6.2.

When solving Problem 6.1 and Problem 6.2 for several slices, each of the RP and NP problems can be addressed either sequentially for each slice, or jointly for all slices. Let SR and JR denote the sequential and joint RP, and similarly SN and JN denote the sequential and joint NP. As a result, we have four different variants of CARP, namely JR-JN, JR-SN, SR-JN, and SR-SN, as illustrated in Figure 6.2.

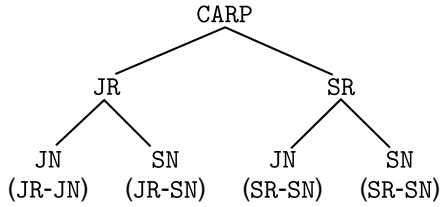


Figure 6.2: Four variants of CARP.

During initialization of CARP, the slice coverage information $\mathcal{A}_s$ is obtained from the S-RD, and $\mathcal{A}_s$ is partitioned into Q_s convex subareas $\mathcal{A}_{s,q}$, $q \in \mathcal{Q}_s = \{1, \ldots, Q_s\}$.

In Step 1 (Lines 3–4 (for JR) or Lines 5–10 (for SR) of Algorithm 6.1), the value of $\boldsymbol{\eta}$ minimizing $c_{\text{rr}}(\boldsymbol{\eta})$ while satisfying all constraints related to radio provisioning (6.2)–6.8 are evaluated;

In Step 2 (Line 13–14 (for JN) or Lines 15–21 (for SN) of Algorithm 6.1), the value of $\mathbf{\Phi}$ minimizing $c_{\text{wr}}(\mathbf{\Phi})$, subject to the constraints (6.11)–(6.21) are evaluated. The constraints (6.17), (6.18), (6.19) are evaluated with the help of $\boldsymbol{\eta}$ obtained at Step 1.

Table 6.2 summarizes the number of optimization problems and the corresponding number of variables per problem to be handled by each provisioning variant. The complexity of the single-step JRN algorithm, performing a simultaneous joint radio and network provisioning for all slices is provided as a reference. In Table 6.2, the variables $\kappa_{s,n}(i,v)$ introduced in (6.15) are not taken into account, since they are directly related to $\phi_{s,n}(i,v)$.

The sequential variants (SR and SN) require to solve $|\mathcal{S}|$ optimization problems, but with $|\mathcal{S}|$ less variables compared to the joint variants (JR and JN). Since each problem is NP-hard, the sequential variants may obviously be solved faster than the joint variants. Table 6.2 summarizes the number of problems and of variables per problem of each provisioning variant. In Section 6.5.1, these variants are evaluated via simulations.

When the amount of available infrastructure resources is not sufficient to accommodate all slices, the proposed joint approaches return no solution. In the sequential approach, the provisioning is performed slice-by-slice. The first processed requests are likely to be satisfied. Next requests may only be satisfied when

resources are released. This solution works on a first-arrived-first-served strategy, and has thus some fairness. The main drawback is the suboptimality of the sequential approach, which will be discussed in the next section.

Algorithme 6.1 : Coverage-Aware Resource Provisioning (CARP)

Input : $\mathcal{G}, \mathcal{S}, \{\mathcal{G}_s, s \in \mathcal{S}\}, \{\mathcal{A}_s, s \in \mathcal{S}\}$

Output : $\widehat{\boldsymbol{\eta}}$ and $\widehat{\boldsymbol{\Phi}}$

```
# Solve RP problem
switch RP_variant do
    case JR (joint radio resource provisioning) do
        Solve Problem 6.1 for all slices in S to obtain η̂;
    case SR (sequential radio resource provisioning) do
        for s ∈ S do
            Solve Problem 6.1 for slice s to obtain η̂_s;
            # Update available infrastructure radio resource
            for i ∈ N_r do
                a_r(i) = a_r(i) − Σ_{q∈Q_s} η_s(i,q) a_r(i);
# Solve NP problem
switch NP_variant do
    case JN (joint network resource provisioning) do
        Solve Problem 6.2 for all slices in S to obtain Φ̂;
    case SN (sequential network resource provisioning) do
        for s ∈ S do
            Solve Problem 6.2 for slice s to obtain Φ̂_s;
            # Update available infrastructure network resources
            for (i, ij) ∈ G do
                a_n(i) = a_n(i) − Σ_{v∈N_s} φ_s(i,v) a_n(i);
                a_b(ij) = a_b(ij) − Σ_{vw∈E_s} φ_s(ij,vw) a_b(ij);
```

Alternatively, in the joint approach, one may renegotiate the SLAs of all slices to provide some fairness by deploying a part of the services. This may be done by provisioning resources so as to satisfy only a fixed proportion $\delta \in]0, 1]$ of demands of each slice. The search for δ may be done by dichotomy.

6.5 Evaluation

In this section, one evaluates via simulations the performance of the proposed provisioning algorithms. The simulation set-up is described in Section 6.5.1. The vari-

Table 6.2: Number of MILP problems, variables, and constraints involved in each variant.

Variant	*#problems*	*#variables/problem*
JRN	1	$\lvert\mathcal{S}\rvert(\lvert\mathcal{N}_\mathrm{r}\rvert\,(1+\lvert\mathcal{Q}_s\rvert)+2\lvert\mathcal{N}\rvert\lvert\mathcal{N}_s\rvert+\lvert\mathcal{N}\rvert+\lvert\mathcal{E}\rvert\lvert\mathcal{E}_\mathrm{V}^\sigma\rvert)$
SR-SN	$\lvert\mathcal{S}\rvert$ RP	$\lvert\mathcal{N}_\mathrm{r}\rvert\,(1+\lvert\mathcal{Q}_s\rvert)$
	$\lvert\mathcal{S}\rvert$ NP	$2\lvert\mathcal{N}\rvert\lvert\mathcal{N}_s\rvert+\lvert\mathcal{N}\rvert+\lvert\mathcal{E}\rvert\lvert\mathcal{E}_\mathrm{V}^\sigma\rvert$
SR-JN	$\lvert\mathcal{S}\rvert$ RP	$\lvert\mathcal{N}_\mathrm{r}\rvert\,(1+\lvert\mathcal{Q}_s\rvert)$
	1 NP	$\lvert\mathcal{S}\rvert\left(2\lvert\mathcal{N}\rvert\lvert\mathcal{N}_s\rvert+\lvert\mathcal{N}\rvert+\lvert\mathcal{E}\rvert\lvert\mathcal{E}_\mathrm{V}^\sigma\rvert\right)$
JR-SN	1 RP	$\lvert\mathcal{S}\rvert\lvert\mathcal{N}_\mathrm{r}\rvert\,(1+\lvert\mathcal{Q}_s\rvert)$
	$\lvert\mathcal{S}\rvert$ NP	$2\lvert\mathcal{N}\rvert\lvert\mathcal{N}_s\rvert+\lvert\mathcal{N}\rvert+\lvert\mathcal{E}\rvert\lvert\mathcal{E}_\mathrm{V}^\sigma\rvert$
JR-JN	1 RP	$\lvert\mathcal{S}\rvert\lvert\mathcal{N}_\mathrm{r}\rvert\,(1+\lvert\mathcal{Q}_s\rvert)$
	1 NP	$\lvert\mathcal{S}\rvert\left(2\lvert\mathcal{N}\rvert\lvert\mathcal{N}_s\rvert+\lvert\mathcal{N}\rvert+\lvert\mathcal{E}\rvert\lvert\mathcal{E}_\mathrm{V}^\sigma\rvert\right)$

ants of the provisioning algorithm introduced in Section 6.4 are first compared in Section 6.5.2. All simulations are performed with the CPLEX MILP solver interfaced with MATLAB.

6.5.1 Simulation Conditions

6.5.1.1 Infrastructure Topology

Consider the $1.43\,\mathrm{km}\times4.95\,\mathrm{km}$ area around the Stade de France in Seine-Saint-Denis (suburban area of the city of Paris) shown in Figure 6.1. The map includes real coordinates of RRH nodes (indicated by blue markers) taken from the open database provided by the French National Agency of Frequencies[3]. In Figure 6.1, only the RRH nodes are represented. The locations of the remaining parts of the infrastructure network (central, regional, and edge nodes) are not displayed.

For the wired part of the infrastructure network, the fat-tree topology introduced in Chapter 5 is reused. The leasing costs of each resource of the infrastructure network is detailed in Table 6.3.

Table 6.3: Infrastructure cost.

Node	$c_\mathrm{f}(i)$	$c_\mathrm{r}(i)$	$c_\mathrm{c}(i)$	$c_\mathrm{s}(i)$
$i\in\mathcal{N}_\mathrm{I}\backslash\mathcal{N}_\mathrm{Ir}$	20	–	1	1
$i\in\mathcal{N}_\mathrm{Ir}$	25	0.05	1	1

6.5.1.2 Slice Resource Demand (S-RD)

Three types of slices are considered.

- Slices of type 1 cover the *Stade de France* and aim to provide an HD video streaming service at 4 Mbps for at most 200 VIP users within the stadium (downlink traffic);

[3]L'Agence nationale des fréquences (ANFR): https://data.anfr.fr/

- Slices of type 2 are dedicated to provide an SD video streaming service at 0.5 Mbps, and cover the blue-highlighted area in Figure 6.1 (downlink traffic);
- Slices of type 3 aim to provide a video surveillance and traffic monitoring service at 1 Mbps for each of 50 cameras installed on the A86 highway (uplink traffic).

The first two slice types address a video streaming service, and have a similar functional structure to those considered in Chapter 5 (see Table 5.1), with the resource demands adapted for respectively 4 Mbps and 0.5 Mbps.

The third slice type video surveillance service and a consists of five virtual functions: a vBBU, a vGW, a virtual Traffic Monitor (vTM), a vVOC, and a virtual Intrusion Detection Prevention System (vIDPS). The resource demands of slice of Type 3 are summarized in Table 6.4.

In the following, different scenarios are considered with an increasing number of slices whose distribution among each type is given in Table 6.5. This represents, *e.g.*, situations where slices of the same type are provided by different SPs.

The coverage area $\mathcal{A}_s$ associated to each slice type is partitioned into rectangular subareas $\mathcal{A}_{s,q}$ of $90\,\mathrm{m}\times 103\,\mathrm{m}$.

Table 6.4: S-RD of slices of Type 3.

Node	$(R_{s,\mathrm{c}}, r_{s,\mathrm{c}})$	$(R_{s,\mathrm{m}}, r_{s,\mathrm{m}})$	*Link*	$R_{s,\mathrm{b}}$
vBBU	$(0.20, 0.02)$	$(0.01, 0.001)$	vBBU→vGW	0.05
vGW	$(0.05, 0.01)$	$(0.01, 0.001)$	vGW→vTM	0.05
vTM	$(0.67, 0.07)$	$(0.01, 0.001)$	vTM→vVOC	0.05
vVOC	$(0.27, 0.03)$	$(0.19, 0.02)$	vVOC→vIDPS	0.05
vIDPS	$(0.54, 0.05)$	$(0.01, 0.001)$		

Table 6.5: Number of slices of each type as a function of $|\mathcal{S}|$.

Case	*#Type 1*	*#Type 2*	*#Type 3*
$\|\mathcal{S}\| = 4$	2	1	1
$\|\mathcal{S}\| = 6$	2	2	2
$\|\mathcal{S}\| = 8$	4	1	2

6.5.1.3 Rate Function

The models of $b_{\mathrm{d}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right)$ and $b_{\mathrm{u}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right)$, which are introduced in Section 6.3.1 for the amount of data carried by an RB for a user located in $\mathcal{A}_{s,q}$ and served by an RRH located in x_i^{r}, are now detailed.

Let $d\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right)$ be the distance between x_i^{r} and the center of each rectangle $\mathcal{A}_{s,q}$. Focusing on downlink traffic, according to [Tse and Pramod, 2004], one assumes that

$$b_{\mathrm{d}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right) = W_i \log_2\left(1 + \frac{P_{\mathrm{rx,d}}\left(d\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right)\right)}{P_{\mathrm{n}}}\right), \tag{6.25}$$

where W_i is the bandwidth (in Hz) of an RB provided by RRH i, P_{n} is the noise power given by $P_{\text{n}} = W_i N_0$, where N_0 is the noise power spectral density. $P_{\text{rx}}(d)$ is the obtained signal power at the receiver evaluated as

$$P_{\text{rx,d}}(d) = P_{\text{tx,d}} + G_{\text{tx,d}} + G_{\text{rx,d}} - PL\,(d)\,, \tag{6.26}$$

where P_{tx} is the transmission power of the transmitter, G_{tx} and G_{rx} are the antenna gains of the transmitter and of the receiver, and $PL(d)$ is the path loss given by the adapted $\alpha\beta\gamma$-model introduced in [Sun et al., 2016] for 5G mobile network

$$PL(d) = 10\alpha \log_{10}(d) + \beta + 10\gamma \log_{10}(f_i)\,, \tag{6.27}$$

where α and γ are respectively coefficients accounting for the dependency of the path loss with distance and frequency f_i, β is an optimized offset value for path loss (dB). PL, d, and f_i are expressed in dB, meters, and GHz, respectively. An expression similar to (6.25) may be derived for $b_{\text{u}}\,(x_i^{\text{r}}, \mathcal{A}_{s,q})$.

All RRH $i \in \mathcal{N}_{\text{r}}$ and all UEs are assumed to be identical. The parameters for the models $b_{\text{d}}\,(x_i^{\text{r}}, \mathcal{A}_{s,q})$ and $b_{\text{u}}\,(x_i^{\text{r}}, \mathcal{A}_{s,q})$ are summarized in Table 6.6 and have been partly taken from [ETSI, 2016].

Table 6.6: Parameters of RRH and $\alpha\beta\gamma$-model.

Parameter	*Definition*	*Value*
$a_{\text{r}}(i)$	Number of RBs available at RRH i	100
f_i	Carrier frequency of RRH i	2.6 GHz
W_i	Bandwidth of a RB of RRH i	0.2 MHz
$P_{\text{tx,d}}$	Antenna transmit power of each RRH	43 dBm
$G_{\text{tx,d}}$	Antenna gain of each RRH	15 dBi
$P_{\text{tx,u}}$	Antenna transmit power of each UE	23 dBm
$G_{\text{tx,u}}$	Antenna gain of each UE	3 dBi
N_0	Noise power spectral density	-174 dBm/Hz
(α, β, γ)	$\alpha\beta\gamma$-model parameters	$(3.6, 7.6, 2)$

6.5.2 Comparison of Provisioning Algorithms

This section illustrates the performance of the `JRN` joint approach and of the four variants of the `CARP` two-step provisioning algorithm described in Table 6.2 when four, six, and eight slices of different types have to be deployed, see Table 6.5.

Figure 6.3a illustrates the radio provisioning costs obtained with the various approaches. One observes that the joint `RP` schemes (`JRN`, `JR-SN`, and `JR-JN`) yield a smaller cost whatever the `NP` allocation method. Note that the `JRN` scheme provides a wireless provisioning cost slightly larger than that of the `JR-SN` or `JR-JN` approaches.

Figure 6.3b illustrates the cost related to the wired part of the infrastructure network. The `JRN` scheme provides the best results and is always able to compen-

sate for the somewhat larger radio provisioning cost, as illustrated in Figure 6.3c, which shows the total provisioning costs. Considering the suboptimal approaches, Figures 6.3b and 6.3c show that the JR-JN scheme performs better than the other approaches and SR-SN provides always the largest costs, as expected.

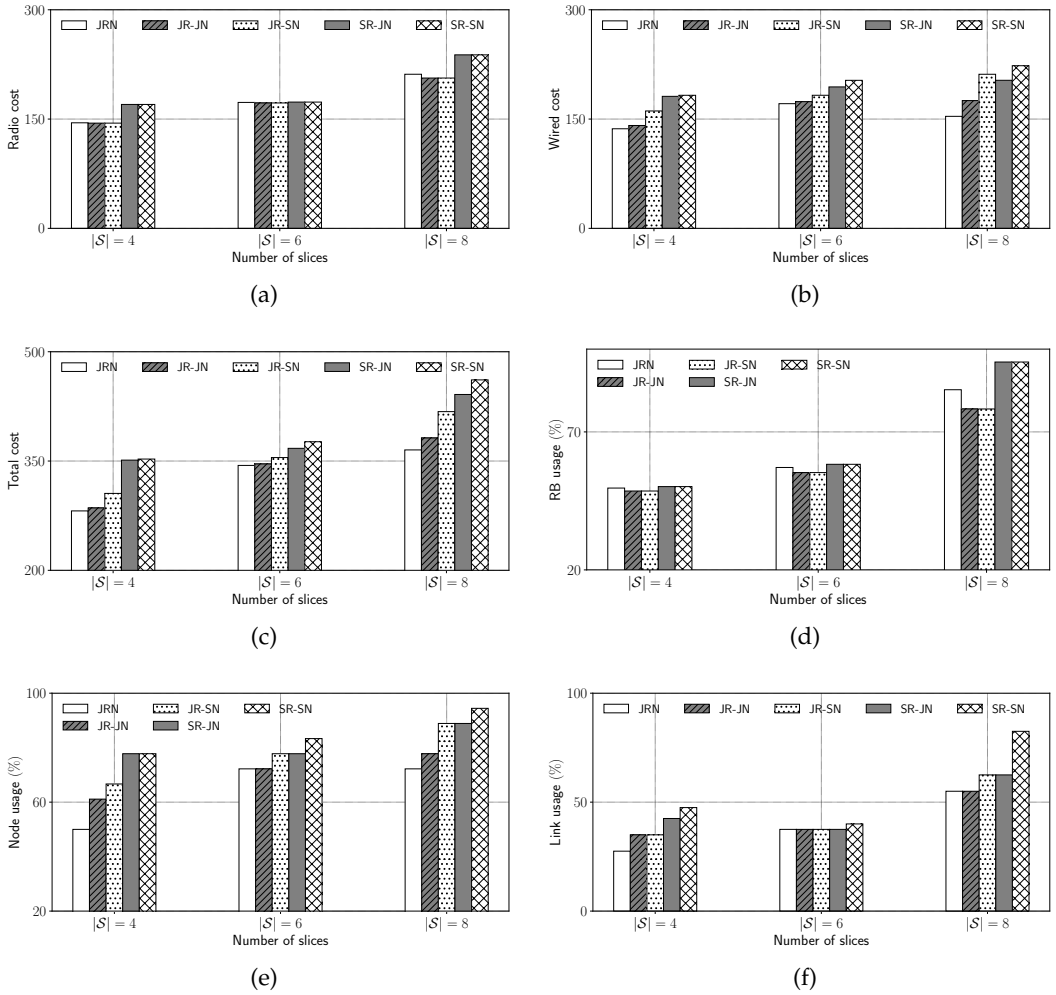


Figure 6.3: Performance comparison of 4 variants in terms of (a) radio cost, (b) wired cost, (c) total provisioning cost, utilization of (d) RBs, (e) infrastructure nodes, and (f) infrastructure links.

To explain these results, one may consider first the use of radio resource blocks detailed in Figure 6.3d. The results are consistent with those in Figure 6.3a: the joint RP approaches (JRN, JR-SN, and JR-JN) outperform the sequential approaches (SR-JN and SR-SN), since the joint RP aims at finding the optimal wireless provisioning for all the slices, while the sequential method only accounts for the constraints of each slice sequentially. The JRN approach does not select the best RRHs for the radio resource provisioning, as compared to the JR-JN or the JR-SN approach, but rather selects the RRHs so as to facilitate the wired network resource provisioning. This leads to a slightly higher utilization of RBs and radio cost (see Figures 6.3d and

6.3a), but lower utilization of infrastructure nodes and links (see Figures 6.3e and 6.3f), and finally allows the JRN approach to obtain the lowest total cost.

For the suboptimal approaches, the joint RP approach also leads to an efficient utilization of infrastructure nodes and links when solving the NP problem, as shown in Figures 6.3e and 6.3f.

The difference in performance of these two sets of methods (JR-SN and JR-JN versus SR-JN and SR-SN) becomes more significant when the number of slices increases. For instance, with six slices, a difference of 11.11% in terms of link utilization is observed in favor of the JR-JN approach, see Figure 6.3f, whereas with eight slices, the difference is 16.67%. Overall, the JR-JN approach provides the best performance in terms of provisioning costs among the four suboptimal methods.

As expected, the methods involving sequential provisioning (SR and SN) perform better than the joint approaches (JR, JN, and JRN) in terms of computing time. Increasing the number of slices leads to an increase of the cardinality of the sets of variables $\boldsymbol{\eta}$ and $\boldsymbol{\Phi}$ and therefore increases the computing time. In sequential provisioning, slices are considered successively. Therefore, among the four suboptimal methods, the SR-SN approach and the JR-JN approach are respectively the least and most time-consuming, as shown in Figure 6.4. The computing time of the optimal JRN is up to 4 times larger than that of the SR-SN approach. Moreover, it increases faster than the other approaches when the number of slices increases.

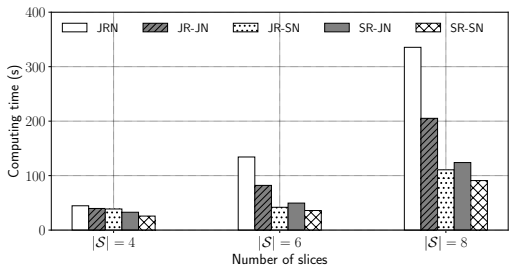


Figure 6.4: Computing time of the 4 proposed provisioning variants

Figure 6.6 illustrates the way RBs are provisioned by the various RRHs for each slice, when $|\mathcal{N}_r| = 8$ and $|\mathcal{S}| = 8$. The use of RBs by each slice for different provisioning variants are further illustrated in Figures 6.7, 6.8, and 6.9.

Thanks to the rate-related discount introduced in the objective function, RRHs that are close to the coverage area of each slice are chosen in priority. For instance, with the JRN and JR approach, Slice 1, which covers the stadium, has its radio resource demand provisioned by RRH 5 and RRH 7. With the SR approach, radio resource demand of Slice 1 is provisioned by RRH 4 and RRH 7. These three RRHs are both close to the stadium.

The advantage of the JRN and JR over the SR approach can be observed: with the SR approach, all RRHs are required to provision resources, whereas with the JRN or JR approach, only seven RRHs are needed.

Finally, Figure 6.5 focuses on the RP problem and shows the maximum supported data rate in the case of sequential and joint radio resource provisioning (*i.e.*, SR and JR) as a function of the aggregate data rate demand from users, *i.e.*, $\sum_{s\in\mathcal{S}} N_s U_s$, where N_s is the number of users in s, when $|\mathcal{N}_\mathrm{r}| = 8$ and 3 slices of type 1, 2, and 3 have to be deployed. U_s remains constant for each slice s. The total number of users N_s associated to each slice varies, but their relative proportions among slices remain constant. With the JR approach, a larger aggregate data rate is supported: provisioning of slices with more users is then possible.

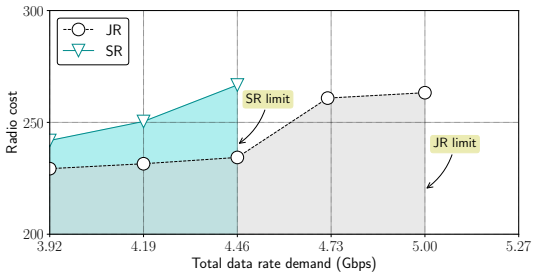


Figure 6.5: Maximum supported data rate associated to the SR and JR provisioning approaches when 3 slices of type 1, 2, and 3 have to be deployed.

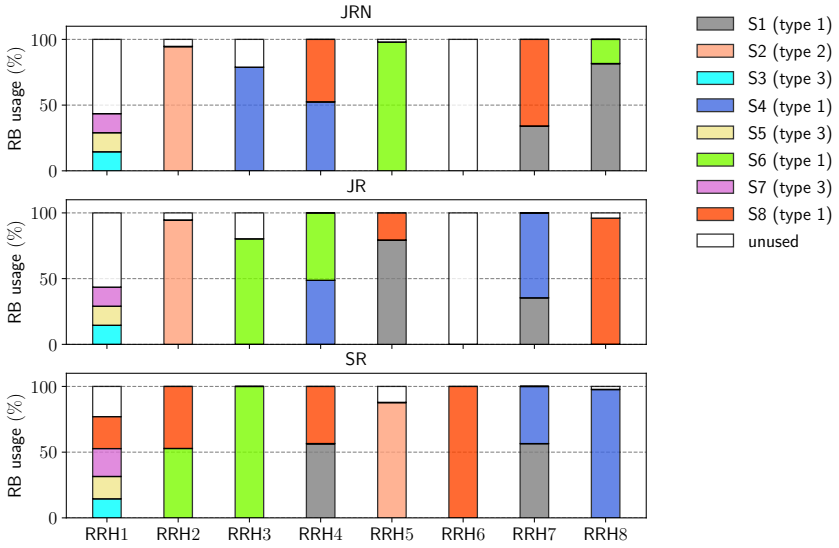
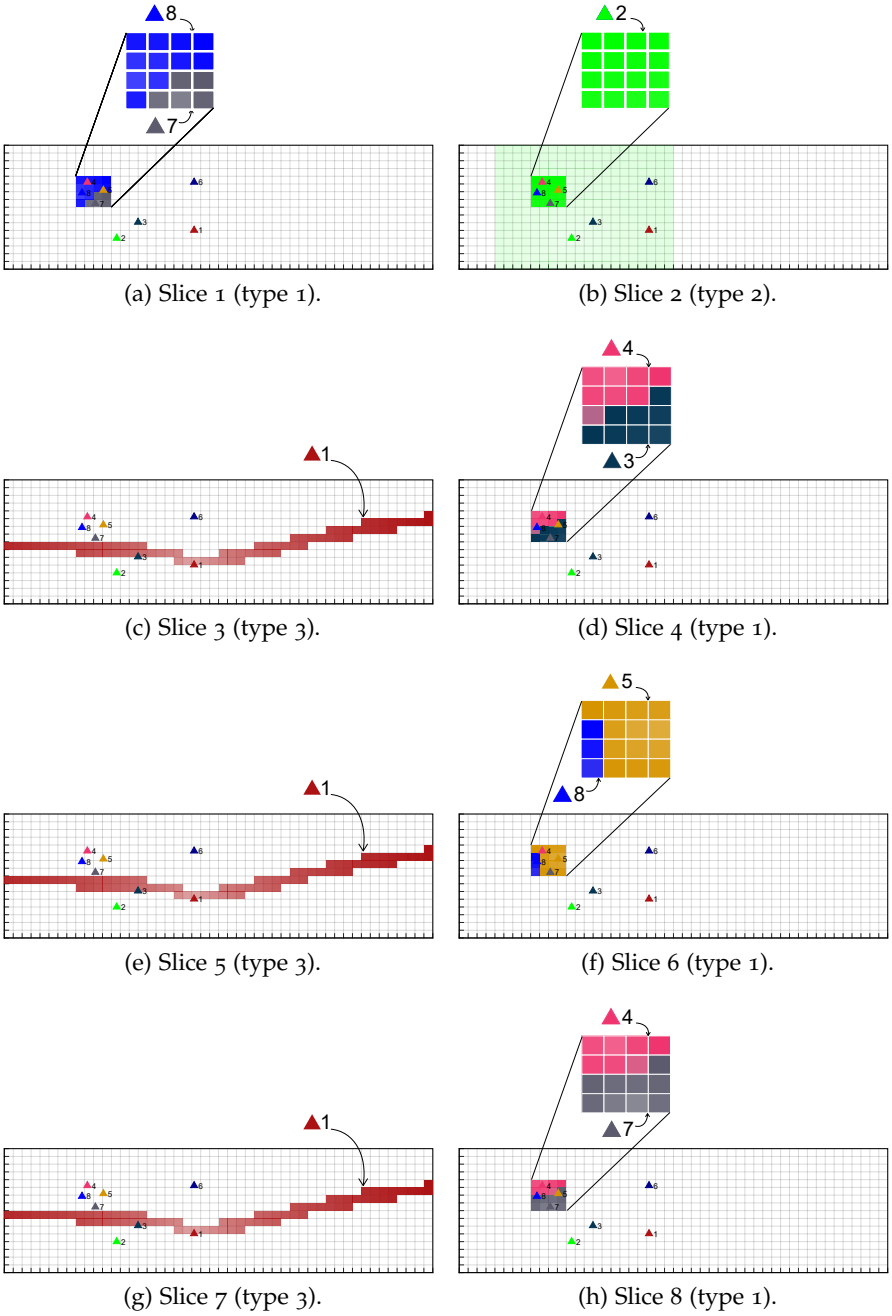
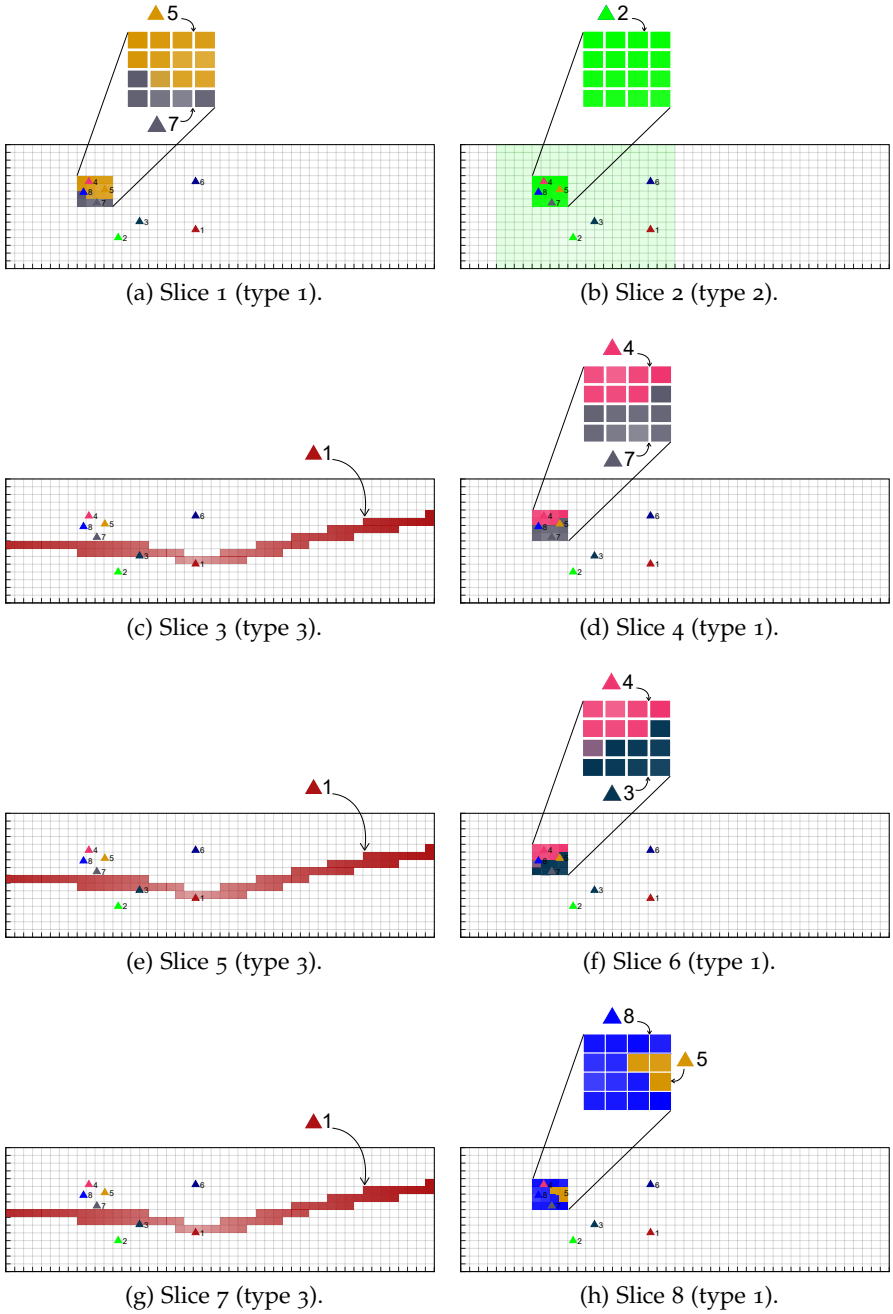


Figure 6.6: Provisioned RBs by RRHs for each slice considering the JRN (top), the JR (middle), and the SR (bottom) approaches.



(a) Slice 1 (type 1).
(b) Slice 2 (type 2).
(c) Slice 3 (type 3).
(d) Slice 4 (type 1).
(e) Slice 5 (type 3).
(f) Slice 6 (type 1).
(g) Slice 7 (type 3).
(h) Slice 8 (type 1).

Figure 6.7: Usage of RBs by each slice using JRN method. RRHs are represented by triangles with different colors. Subareas (small squares) of each provisioned slice are filled by the same color of the RRH that provisions RBs for that slice. The color density reflects the amount of provisioned RBs (the darker, the more RBs).



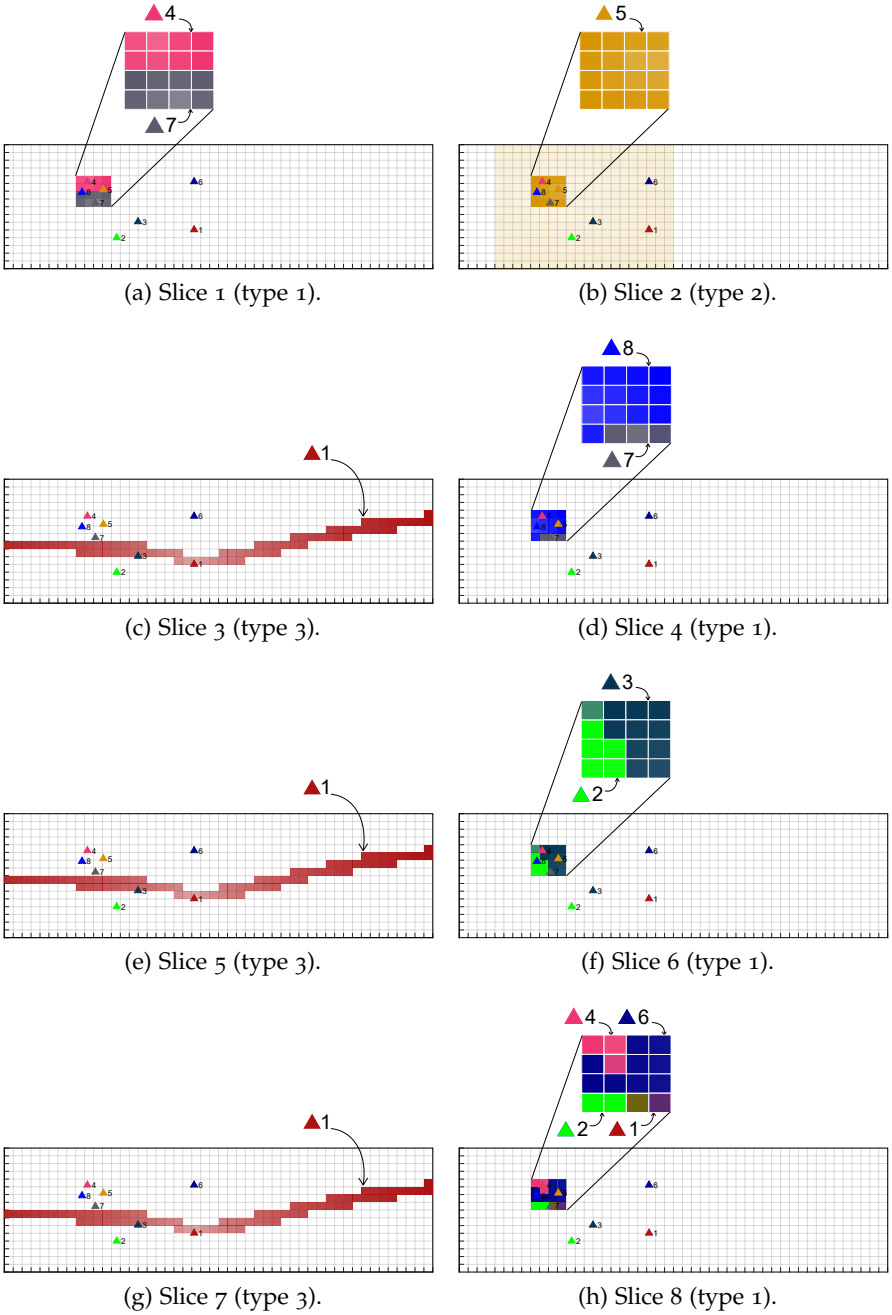
(a) Slice 1 (type 1). (b) Slice 2 (type 2).

(c) Slice 3 (type 3). (d) Slice 4 (type 1).

(e) Slice 5 (type 3). (f) Slice 6 (type 1).

(g) Slice 7 (type 3). (h) Slice 8 (type 1).

Figure 6.8: Usage of RBs by each slice using JR-SN method. RRHs are represented by triangles with different colors. Subareas (small squares) of each provisioned slice are filled by the same color of the RRH that provisions RBs for that slice. The color density reflects the amount of provisioned RBs.



(a) Slice 1 (type 1). (b) Slice 2 (type 2).

(c) Slice 3 (type 3). (d) Slice 4 (type 1).

(e) Slice 5 (type 3). (f) Slice 6 (type 1).

(g) Slice 7 (type 3). (h) Slice 8 (type 1).

Figure 6.9: Usage of RBs by each slice using SR-SN method. RRHs are represented by triangles with different colors. Subareas (small squares) of each provisioned slice are filled by the same color of the RRH that provisions RBs for that slice. The color density reflects the amount of provisioned RBs.

6.6 Conclusion

This chapter extends the study considered in Chapter 5 by considering the problem of provisioning for joint core and radio access network resources, accounting some coverage constraints. The problem is cast in the framework of MILP problem.

A two-step approach is proposed to address the complexity of this problem. Radio resources on RRH are provisioned first to ensure the satisfaction of the cov-

erage constraints. Other constraints as defined by the S-RD and the associated virtual graph are then considered. When resources have to be provisioned for several concurrent slices, two variants have again been considered. At each step, constraints related to each slice may be considered either sequentially, or jointly. Due to the exponential worst-case complexity in the number of variables of the MILP, as expected, sequential methods are shown, through simulations, to better scale to network topologies of realistic size. The price to be paid is a somewhat degraded link utilization and a higher provisioning cost compared to the joint approach. When both coverage and infrastructure network constraints have to be taken into account simultaneously, *i.e.*, with the `JRN` approach, a minimum provisioning cost could be achieved, but this approach requires a much larger time complexity than the four variants of the suboptimal `CARP`.

Once resources have been provisioned, the approach introduced in [Riggio et al., 2016, Bouten et al., 2017] may be used to deploy SFCs, but considering only a simplified infrastructure network reduced to the nodes and links which have provisioned resources. Simulations show that provisioning and then deploying is more efficient in terms of computing time than direct SFC embedding.

Only static provisioning is considered in this chapter. Resource provisioning was done for a given time interval specified in the SLA over which the service characteristics and constraints are assumed constant and compliant with the variations of user demands within a slice. A level of conservatism in the amount of provisioned resources is then required to satisfy fast fluctuating user demands.

One could imagine adaptive SLAs to meet more closely the actual demands. The SLA may consider several time intervals over each of which the service characteristics and constraints are assumed constant, but may vary from one interval to the next one. On the other hand, the traffic dynamics in individual slices, *e.g.*, arrivals and departures of slice requests, may also lead to the variations of total resource demands of slices. In Part III, the uncertainties of slice resource demands as well as the traffic dynamics will be carefully addressed.

Part III

Dynamic Resource Provisioning with Uncertainties

Chapter 7

Uncertainty-Aware Resource Provisioning

This chapter is based on Q.-T. Luu S. Kerboeuf, and M. Kieffer, "Uncertainty-Aware Resource Provisioning for Network Slicing," in IEEE Transactions on Network and Service Management, vol. 18, no. 1, Feb. 2021 [Luu et al., 2021c].

The traffic dynamics in individual slices, such as flow arrival/departure, as well as the dynamics of resource availability on the network infrastructure, may lead to slice QoS below the level expected by the Service Provider (SP) managing the slice. The traditional approach, in which allocated/provisioned resources are tailored to peak demands, as studied in Part I and in the majority of literature, *e.g.*, in [Huin et al., 2017, Su et al., 2019, Barakabitze et al., 2020], does not provide optimal results. Therefore, to fully unleash the power of network slicing in dynamic environments, uncertainties related to the resource demands need to be carefully addressed. This chapter aims to provide an answer to Challenge 3 introduced in Chapter 1. We investigate a method to provision infrastructure resources for network slices, while being robust to a partly unknown number of users with a random usage of the slice resources. Moreover, since some parts of the infrastructure network on which slices should be deployed are often already employed by low-priority background services, the provisioning approach will be performed so as to limit its impact on these services.

The rest of the chapter is structured as follows. Section 7.1 highlights our main contributions. The model of the infrastructure network and of the slice resource demands are presented in Section 7.2. The robust slice resource provisioning problem with uncertainties in the number of users as well as in the resource demands and accounting for the best-effort background services is then formulated in Section 7.3. This problem is transformed into a mixed integer linear programming (MILP) problem in Section 7.4. Numerical results are presented in Section 7.5. Finally, Section 7.6 draws some conclusions and perspectives.

7.1 Contributions

In the related works reported in Section 3.3, the effect of the best effort background services combined with a approach robust to uncertainties in the demands and in the infrastructure resources has not yet been considered for the slice provisioning problem. In this chapter, a slice resource provisioning method robust to randomness of resource demands is proposed. The randomness is due to a partly unknown number of users with a random usage of the slice resources. The robustness is achieved by providing a probabilistic guarantee that the amount of provisioned network resources for a slice will meet the slice requirements. Moreover, in the related literature, *e.g.*, [Fendt et al., 2019, Wen et al., 2019], uncertainties in the available network resource are usually considered. Here, we consider best-effort background services running in parallel with the network slices on the infrastructure network. The proposed method tries to maintain the impact of resource provisioning on those background services at a prescribed level. Previous results on slice resource provisioning have been presented in [Luu et al., 2020a]. Nevertheless, uncertainties in the number of users of a slice and in the way they consume resources, as well as concurrent best-effort services sharing the infrastructure network have not been taken into account.

7.2 Notations and Hypotheses

In what follows, the characteristics of the SLAs[1] between different entities involved in the network slicing system are described.

In Chapters 5 and 6, the SLA between the SP and the MNO (SM-SLA) contains a description of the characteristics of the service and of the way it is employed by a typical user/device[2]. In this chapter, the SM-SLA has the following additional characteristics

(1) a probability mass function (pmf) describing the target number of users/devices to be supported by the slice;

(2) a target probability of service satisfaction.

As discussed in Section 4.2, in this chapter, a *just-in-time* provisioning approach is applied, in which one focuses on a given time slot and on the requests of slices that have to be activated in the next time slot. The provisioning processing considers only slice requests that need to be activated in the next time slot. The set of those slice requests is denoted as $\mathcal{S}$.

To characterize the variability over time and among users of these demands, we assume that the MNO considers a probabilistic description of the consumption of

[1] See Section 4.1. Throughout this thesis, one considers two SLAs: the SM-SLA between the SP and the MNO and the MI-SLA between the MNO and the InP.

[2] In Chapter 6, the SM-SLA also contains some slice coverage constraints.

slice resources by a typical user. The MNO then forwards to the InP these characteristics as part of an SLA between them (MI-SLA). Each InP then provisions infrastructure resources needed for the SFCs. Under the MI-SLA, this provisioning has to meet the target probability of service satisfaction. This translates the fact that enough resources of various types have been provisioned to satisfy the resource demands of the users of the service. This probability is evaluated considering the pmf describing the number of users of the service and the probabilistic description of the slice resource consumption by a typical user. When performing the provisioning, each InP has to limit the impact on other best-effort service running on its infrastructure network.

One considers an infrastructure owned by a single InP. To perform the provisioning, the InP has to identify the infrastructure nodes which will provide resources for future deployment of VNFs and the links able to transmit data between these nodes, while respecting the structure of SFCs and optimizing a given objective (*e.g.*, minimizing the infrastructure and software fee costs).

Table 7.1 summarizes additional notations involved in this chapter.

Table 7.1: Additional notations used in Chapter 7.

Symbol	*Description*
$\boldsymbol{r}_s$	Vector of resource demands of an SFC
$\mathbf{U}_s$	Vector of resource demands of a typical user
$\mathbf{R}_s$	Vector of aggregate resource demands
$\mathbf{B}$	Vector of resources consumed by background
$\overline{U}$, $\overline{R}$, $\overline{B}$	Mean of U, R, B
$\widetilde{U}$, $\widetilde{R}$, $\widetilde{B}$	Standard deviation of U, R, B

7.2.1 Infrastructure Network

As presented in Chapter 4, the considered types of resources at node level are $\Upsilon = \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$, denoting respectively computing, memory, and wireless resources. The model of the infrastructure network is similar to that introduced in Chapter 5.

7.2.2 Model of the Slice Resource Demand

The vectors of resource demands for SFC-RD ($\boldsymbol{r}_s$), U-RD ($\mathbf{U}_s$), and S-RD ($\mathbf{R}_s$) introduced in Chapter 4 are used to model the slice resource demands. As pointed out in Section 4.2.3, $\boldsymbol{r}_s$ is a deterministic vector, while $\mathbf{U}_s$ and of $\mathbf{R}_s$ are random vectors.

Considering the analysis of co-allocated online services of large scale data centers reported in [Jiang et al., 2019], the utilization of CPU and memory of virtual machines (VMs) have a positive correlation in the majority of cases. Moreover, this correlation is particularly strong at the VMs that execute the same jobs, showing correlation coefficients larger than 0.85. Based on this observation, for a typical user,

the resource demands of different types for a given node $v \in \mathcal{N}_s$ are considered to be correlated. The demands for resources of the same type among virtual nodes are also correlated. Finally, the resulting traffic demands between nodes is usually also correlated with the resource demands for a given virtual node. To represent this correlation, consider the vector of joint resource demands for a typical user of an SFC of slice s

$$\mathbf{U}_s = \left(U_{s,n}\left(v\right), U_{s,\mathrm{b}}\left(vw\right)\right)^{\top}_{n\in\Upsilon,(v,vw)\in\mathcal{G}_s}.$$

Assuming that $U_{s,\mathrm{c}}\left(v\right)$, $U_{s,\mathrm{m}}\left(v\right)$, $U_{s.\mathrm{w}}\left(v\right)$, and $U_{s,\mathrm{b}}\left(vw\right)$ are normally distributed, $\mathbf{U}_s$ follows a multivariate normal distribution with probability density

$$f\left(\mathbf{x};\boldsymbol{\mu}_s,\boldsymbol{\Gamma}_s\right) = \frac{1}{\sqrt{\left(2\pi\right)^{\mathrm{card}(\mathbf{U}_s)}\left|\boldsymbol{\Gamma}_s\right|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_s)^{\top}(\boldsymbol{\Gamma}_s)^{-1}(\mathbf{x}-\boldsymbol{\mu}_s)}, \tag{7.1}$$

with mean

$$\boldsymbol{\mu}_s = \left(\overline{U}_{s,n}\left(v\right), \overline{U}_{s,\mathrm{b}}\left(vw\right)\right)^{\top}_{n\in\Upsilon,(v,vw)\in\mathcal{G}_s},$$

and covariance matrix $\boldsymbol{\Gamma}_s$ such that

$$\mathrm{diag}\left(\boldsymbol{\Gamma}_s\right) = \left(\widetilde{U}^2_{s,n}\left(v\right), \widetilde{U}^2_{s,\mathrm{b}}\left(vw\right)\right)^{\top}_{n\in\Upsilon,(v,vw)\in\mathcal{G}_s},$$

the off-diagonal elements of $\boldsymbol{\Gamma}_s$ representing the correlation between different types of resource demands. In (7.1), $\mathrm{card}\left(\mathbf{U}_s\right)$ is the number of elements of $\mathbf{U}_s$. One has thus

$$\begin{aligned} U_{s,n}(v) \sim& \mathcal{N}\left(\overline{U}_{s,n}\left(v\right), \widetilde{U}^2_{s,n}\left(v\right)\right), \forall n \in \Upsilon, \text{ and} \\ U_{s,\mathrm{b}}\left(vw\right) \sim& \mathcal{N}\left(\overline{U}_{s,\mathrm{b}}\left(vw\right), \widetilde{U}^2_{s,\mathrm{b}}\left(vw\right)\right). \end{aligned}$$

Assume that the number of users N_s to be supported by slice s is described by the pmf

$$p_k = \Pr\left(N_s = k\right). \tag{7.2}$$

Since the amount of resources of the VNF v and of the virtual link vw consumed by different users is represented by independently and identically distributed copies of $\mathbf{U}_s$, the joint distribution of the aggregate amount $\mathbf{U}_{s,k}$ of resources consumed by k independent users is $f\left(\mathbf{x}, k\boldsymbol{\mu}_s, k^2\boldsymbol{\Gamma}_s\right)$. The total amount of resources employed by a random number N_s of independent users, $\mathbf{R}_s \triangleq \mathbf{U}_{s,N_s}$, is distributed according to

$$g\left(\mathbf{x},\boldsymbol{\mu}_s,\boldsymbol{\Gamma}_s\right) = \sum_{k=0}^{\infty} p_k f\left(\mathbf{x}, k\boldsymbol{\mu}_s, k^2\boldsymbol{\Gamma}_s\right). \tag{7.3}$$

The typical joint distribution of two components of $\mathbf{U}_s$ and $\mathbf{R}_s$ is illustrated in Example 7.1.

Example 7.1 (Distribution example of $\mathbf{U}_s$ and $\mathbf{R}_s$)**.** Considering a virtual node v of a given slice s, Figure 7.1 represents the joint distribution of $U_{s,\mathrm{c}}\left(v\right)$ and $U_{s,\mathrm{m}}\left(v\right)$ and the resulting joint distribution of $R_{s,\mathrm{c}}\left(v\right)$ and $R_{s,\mathrm{m}}\left(v\right)$. Here N_s

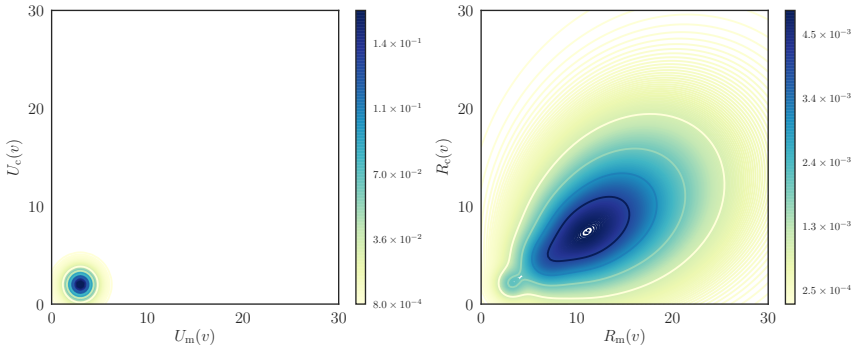
follows the binomial distribution $N_s \sim \mathcal{B}(10, 0.5)$, $\boldsymbol{\mu}_s = [2, 3]^\top$. In Figure 7.1a,

$$\boldsymbol{\Gamma}_s = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

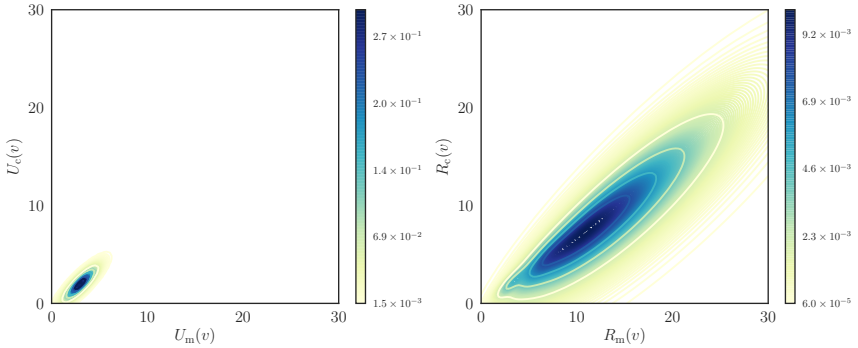
is diagonal. Even if the level sets of $f(\mathbf{x}; \boldsymbol{\mu}_s, \boldsymbol{\Gamma}_s)$ are circles, the level sets of the resulting $g(\mathbf{x}, \boldsymbol{\mu}_s, \boldsymbol{\Gamma}_s)$ illustrate the correlation between $R_{s,\mathrm{c}}(v)$ and $R_{s,\mathrm{m}}(v)$. In Figure 7.1b,

$$\boldsymbol{\Gamma}_s = \begin{bmatrix} 1 & 0.85 \\ 0.85 & 1 \end{bmatrix}$$

is non-diagonal, *i.e.*, $U_{s,\mathrm{c}}(v)$ and $U_{s,\mathrm{m}}(v)$ are correlated, the correlation between $R_{s,\mathrm{c}}(v)$ and $R_{s,\mathrm{m}}(v)$ increases significantly.



(a) Uncorrelated demands.



(b) Correlated demands.

Figure 7.1: Joint distribution of $U_{s,\mathrm{c}}(v)$ and $U_{s,\mathrm{m}}(v)$ (top left and bottom left); and of $R_{s,\mathrm{c}}(v)$ and $R_{s,\mathrm{m}}(v)$ (top right and bottom right), when $U_{s,\mathrm{c}}(v)$ and $U_{s,\mathrm{m}}(v)$ are (a) uncorrelated, and (b) correlated. The number of users N_s follows the binomial distribution $N_s \sim \mathcal{B}(10, 0.5)$.

7.2.3 Resource Consumption of Best-Effort Background Services

As assumed in Section 4.2.2, in the considered time slot, a given part of the available resources is consumed by other best-effort background services for which no resource provisioning has been performed. Since $\Upsilon = \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$, one has

$$\mathbf{B} = \left(B_{\mathrm{c}}(i), B_{\mathrm{m}}(i), B_{\mathrm{w}}(i), B_{\mathrm{b}}(ij)\right)^\top_{(i,ij)\in\mathcal{G}}. \tag{7.4}$$

In this chapter, one assumes the elements $B_{\mathrm{c}}(i)$, $B_{\mathrm{m}}(i)$, $B_{\mathrm{w}}(i)$, $\forall i \in \mathcal{N}$, and $B_{\mathrm{b}}(ij)$, $\forall ij \in \mathcal{E}$ of $\mathbf{B}$ are uncorrelated and Gaussian distributed, *i.e.*,

$$B_n(i) \sim \mathcal{N}\left(\overline{B}_n(i), \widetilde{B}^2_n(i)\right), \forall i \in \mathcal{N}, \forall n \in \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}, \text{and}$$
$$B_n(i) \sim \mathcal{N}\left(\overline{B}_{\mathrm{b}}(ij), \widetilde{B}^2_{\mathrm{b}}(ij)\right), \forall ij \in \mathcal{E}.$$

Finally, assuming that the elements of $\mathbf{B}$ are uncorrelated. Hence, $\mathbf{B}$ is distributed according to $f(\mathbf{x}; \boldsymbol{\mu}_{\mathrm{B}}, \boldsymbol{\Gamma}_{\mathrm{B}})$ with

$$\boldsymbol{\mu}_{\mathrm{B}} = \left(\overline{B}_{\mathrm{c}}(i), \overline{B}_{\mathrm{m}}(i), \overline{B}_{\mathrm{w}}(i), \overline{B}_{\mathrm{b}}(ij)\right)^{\top}_{(i,ij)\in\mathcal{G}}, \text{ and}$$
$$\boldsymbol{\Gamma}_{\mathrm{B}} = \mathrm{diag}\left(\widetilde{B}^2_{\mathrm{c}}(i), \widetilde{B}^2_{\mathrm{m}}(i), \widetilde{B}^2_{\mathrm{w}}(i), \widetilde{B}^2_{\mathrm{b}}(ij)\right)^{\top}_{(i,ij)\in\mathcal{G}}.$$

7.3 Optimal Slice Resource Provisioning

To provision infrastructure resource for a given slice $s \in \mathcal{S}$, the InP has to determine the amount of resources that its infrastructure nodes and links has to reserve to satisfy the slice resource demands with a given probability. Moreover, the InP has to preserve enough resource for background services. This will be done by evaluating and bounding the probability that the provisioning impacts (reduces) the resources and traffic involved in the best-effort services.

As specified in Section 4.2, the slice resource provisioning is represented by a mapping between the infrastructure graph $\mathcal{G}$ and the S-RD weighted graph $\mathcal{G}_s$. In Part I of this thesis, the set of variables $\boldsymbol{\Phi} = \{\boldsymbol{\Phi}_s\}$, where

$$\boldsymbol{\Phi}_s = \left\{\phi_{s,n}(i,v), \phi_{s,\mathrm{b}}(ij,vw), \widetilde{\phi}_{s,n}(i,v)\right\}_{(i,ij)\in\mathcal{G},(v,vw)\in\mathcal{G}_s,n\in\Upsilon} \tag{7.5}$$

have been used to solve the slice resource provisioning problem for the core network. Nevertheless, the constraints imposed by (5.8) in Chapter 5 and by (6.15) in Chapter 6 require introducing additional variables $\kappa_{s,n}(i,v)$. To simplify the formulation, in this chapter, we propose a novel formulation based on the $\kappa_{s,n}(i,v)$ and is entirely independent of $\boldsymbol{\Phi}_s$. The novel set of variables $\boldsymbol{\kappa} = \{\boldsymbol{\kappa}_s\}_{s\in\mathcal{S}}$ is defined as

$$\boldsymbol{\kappa}_s = \left\{\kappa_s(i,v), \kappa_s(ij,vw), \widetilde{\kappa}_s(i,v), \widetilde{\kappa}(i)\right\}_{(i,ij)\in\mathcal{G},(v,vw)\in\mathcal{G}_s},$$

where

- $\kappa_s(i,v) \in \mathbb{N}$ represents the number of VNF instances of type $v \in \mathcal{N}_s$ that node i will be able to host. Consequently the amount of resource of type $n \in \Upsilon$ provisioned by node i for a VNF instance of type v is $\kappa_s(i,v)\, r_{s,n}(v)$. Similarly, $\kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw)$ represents the bandwidth provisioned by link ij to support the traffic between virtual nodes of type v and w, with $\kappa_s(ij,vw) \in \mathbb{N}$;
- $\widetilde{\kappa}_s(i,v) \in \{0,1\}$ is a node mapping indicator, *i.e.*, $\widetilde{\kappa}_s(i,v) = 1$ if $\kappa_s(i,v) > 0$ and

Table 7.2: Involved variables and the corresponding size in the $\boldsymbol{\Phi}$-based and the $\boldsymbol{\kappa}$-based formulation.

$\boldsymbol{\Phi}$-based formulation (Chapters 5 and 6)		$\boldsymbol{\kappa}$-based formulation (Chapters 7 and 8)	
Variable	*Size*	*Variable*	*Size*
$\phi_{s,n}(i,v)$	$3\,\lvert\mathcal{N}\rvert\,\lvert\mathcal{N}_s\rvert$	$\kappa_s(i,v)$	$\lvert\mathcal{N}\rvert\,\lvert\mathcal{N}_s\rvert$
$\phi_{s,\mathrm{b}}(ij,vw)$	$\lvert\mathcal{E}\rvert\,\lvert\mathcal{E}_s\rvert$	$\kappa_s(ij,vw)$	$\lvert\mathcal{N}\rvert\lvert\mathcal{N}_s\rvert$
$\widetilde{\phi}_{s,n}(i,v)$	$\lvert\mathcal{N}\rvert\,\lvert\mathcal{N}_s\rvert$	$\widetilde{\kappa}_s(i,v)$	$\lvert\mathcal{N}\rvert\,\lvert\mathcal{N}_s\rvert$
$\kappa_{s,n}(i,v)$	$3\,\lvert\mathcal{N}\rvert\,\lvert\mathcal{N}_s\rvert$	$\widetilde{\kappa}(i)$	$\lvert\mathcal{N}\rvert$
Total: $7\,\lvert\mathcal{N}\rvert\,\lvert\mathcal{N}_s\rvert+\lvert\mathcal{E}\rvert\,\lvert\mathcal{E}_s\rvert$		Total: $\lvert\mathcal{N}\rvert+3\,\lvert\mathcal{N}\rvert\,\lvert\mathcal{N}_s\rvert+\lvert\mathcal{E}\rvert\,\lvert\mathcal{E}_s\rvert$	

$\widetilde{\kappa}_s(i,v)=0$ otherwise;

$\widetilde{\kappa}(i)\in\{0,1\}$ indicates whether the infrastructure node i has been used for the slice resource provisioning.

Table 7.2 compares the $\boldsymbol{\Phi}$-based and the $\boldsymbol{\kappa}$-based formulation and the resulting number of variables required to formulate the resource provisioning problem. Compared to the $\boldsymbol{\Phi}$-based formulations used in Chapters 5 and 6, the novel $\boldsymbol{\kappa}$-based formulation has a lower number of variables. Moreover, since the novel formulation is based on $\kappa_s(i,v)$, the constraints (5.8) in Chapter 5 and (6.15) in Chapter 6 are no longer required. With these reasons, one obtains thus a simplified optimization problems with reduced number of variables and constraints, which may make the convergence to the optimal solution faster.

A solution of the provisioning problem for slice s is thus defined by a given assignment of the variables $\boldsymbol{\kappa}_s$. This assignment has to satisfy some constraints to ensure a satisfying behavior of the SFC and the satisfaction of the MI-SLA for slice s defined in terms of probability of satisfaction of the aggregate user demands $\underline{p}_s$, see Section 7.3.1. In addition, from the perspective of the InP, this assignment has also to have a limited impact on the operation of background best-effort services. Section 7.3.2 presents the model of cost, income, and earning. The first attempt to formulate the slice resource provisioning problem is introduced in Section 7.3.3.

7.3.1 Constraints

Consider slice $s\in\mathcal{S}$ and a given assignment of the variables $\boldsymbol{\kappa}_s$. For a given node $v\in\mathcal{N}_s$, the probability that enough resources are provisioned in the infrastructure network to satisfy the resource demand $R_{s,n}(v)$ of type $n\in\Upsilon$ is

$$p_{s,n}(v)=\Pr\Big\{\sum_{i\in\mathcal{N}}\kappa_s(i,v)\,r_{s,n}(v)\geqslant R_{s,n}(v)\Big\}. \tag{7.6}$$

Similarly, for a given virtual link $vw \in \mathcal{E}_s$, the probability that enough bandwidth is provisioned in the infrastructure network to satisfy the demand $R_{s,\mathrm{b}}(vw)$ is

$$p_{s,\mathrm{b}}(vw) = \Pr\Big\{ \sum_{ij \in \mathcal{E}} \kappa_s(ij, vw)\, r_{s,\mathrm{b}}(vw) \geqslant R_{s,\mathrm{b}}(vw) \Big\}. \tag{7.7}$$

In both cases, the assignment has to be such that, for each infrastructure node $i \in \mathcal{N}$ and link $ij \in \mathcal{E}$, the total amount of provisioned resources for all slices $s \in \mathcal{S}$ is less or equal than the amount of available resources

$$\sum_{s \in \mathcal{S}} \sum_{v \in \mathcal{N}_s} \kappa_s(i, v)\, r_{s,n}(v) \leqslant a_n(i), \forall i \in \mathcal{N}, n \in \Upsilon, \tag{7.8}$$

$$\sum_{s \in \mathcal{S}} \sum_{v \in \mathcal{E}_s} \kappa_s(ij, vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij), \forall ij \in \mathcal{E}. \tag{7.9}$$

The constraints (7.8)–(7.9) may leave no resources for the background best-effort services, when $\sum_{s,v} \kappa_s(i, v)\, r_{s,n}(v)$ and $\sum_{s,vw} \kappa_s(ij, vw)\, r_{s,\mathrm{b}}(vw)$ are close to $a_n(i)$, $\forall n \in \Upsilon$, and $a_{\mathrm{b}}(ij)$. The probability that the background best-effort services are impacted at a node i or on the link ij by the provisioning for each slice $s \in \mathcal{S}$ and each $n \in \Upsilon$ are

$$p_n^{\mathrm{im}}(i) = \Pr\Big\{ \sum_{s \in \mathcal{S}} \sum_{v \in \mathcal{N}_s} \kappa_s(i, v)\, r_{s,n}(v) \geqslant a_n(i) - B_n(i) \Big\}, \tag{7.10}$$

and

$$p_{\mathrm{b}}^{\mathrm{im}}(ij) = \Pr\Big\{ \sum_{s \in \mathcal{S}} \sum_{v \in \mathcal{E}_s} \kappa_s(ij, vw)\, r_{s,\mathrm{b}}(vw) \geqslant a_{\mathrm{b}}(ij) - B_{\mathrm{b}}(ij) \Big\}. \tag{7.11}$$

The impact probabilities (ImPs) of the provisioning for all slice $s \in \mathcal{S}$ on the node and link resources employed by best-effort services have to be such that

$$p_n^{\mathrm{im}}(i) \leqslant \overline{p}^{\mathrm{im}}, \forall i \in \mathcal{N}, n \in \Upsilon, \tag{7.12}$$

$$p_{\mathrm{b}}^{\mathrm{im}}(ij) \leqslant \overline{p}^{\mathrm{im}}, \forall ij \in \mathcal{E}, \tag{7.13}$$

where $\overline{p}^{\mathrm{im}}$ is the maximum tolerated impact probability. The value of $\overline{p}^{\mathrm{im}}$ is chosen by the InP to provide sufficient resources for the background services at every infrastructure nodes and links. A small value of $\overline{p}^{\mathrm{im}}$ leads to a small impact of slice resource provisioning on background services, but makes the provisioning problem more difficult to solve compared to a situation where $\overline{p}^{\mathrm{im}}$ is close to one.

To ensure the data is correctly carried between VNFs, we also have the flow conservation constraints, similar to those introduced in Chapters 5 and 6. With the $\boldsymbol{\kappa}$-based formulation, the flow conservation constraints become, for each $s \in \mathcal{S}$, $i \in \mathcal{N}$, and $vw \in \mathcal{E}_s$,

$$\sum_{j \in \mathcal{N}} \left[\kappa_s(ij, vw) - \kappa_s(ji, vw) \right] = \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{vu \in \mathcal{E}_s} r_{s,\mathrm{b}}(vu)} \right) \kappa_s(i, v) - \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{uw \in \mathcal{E}_s} r_{s,\mathrm{b}}(uw)} \right) \kappa_s(i, w). \tag{7.14}$$

Finally, considering an assignment $\boldsymbol{\kappa} = \{\boldsymbol{\kappa}_s\}_{s\in\mathcal{S}}$ satisfying (7.8)–(7.14), the probability that this assignment is compliant with the constraints imposed for slice s and by the infrastructure, *i.e.*, the Service Satisfaction Probability (SSP) for slice s is

$$p_s(\boldsymbol{\kappa}_s) = \Pr\Big\{ \sum_{i\in\mathcal{N}} \kappa_s(i,v)\, r_{s,n}(v) \geqslant R_{s,n}(v), \forall v, n, \\ \sum_{ij\in\mathcal{E}} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant R_{s,\mathrm{b}}(vw), \forall vw \Big\}, \tag{7.15}$$

and, as stated in the MI-SLA, the InP has to ensure a minimum SSP of $\underline{p}_s$ for each slice $s \in \mathcal{S}$, *i.e.*,

$$p_s(\boldsymbol{\kappa}_s) \geqslant \underline{p}_s, \forall s \in \mathcal{S}. \tag{7.16}$$

7.3.2 Costs, Incomes, and Earnings

Considering the perspective of the InP, this section presents the cost, income, and earning model for the slice resource provisioning problem.

Consider a given slice $s \in \mathcal{S}$ and its related assignment of the variables $\boldsymbol{\kappa}_s$. Let

$$x_s(\boldsymbol{\kappa}_s) = \begin{cases} 1 & \text{if } p_s(\boldsymbol{\kappa}_s) \geqslant \underline{p}_s \\ 0 & \text{else} \end{cases} \tag{7.17}$$

indicate whether the MI-SLA for slice s is satisfied.

Define I_s as the income obtained for a slice s whose MI-SLA is satisfied. The income awarded to the InP from the MNO is then $I_s x_s(\boldsymbol{\kappa}_s)$.

The total provisioning cost $C_s(\boldsymbol{\kappa}_s)$ of a given slice s for the InP is

$$C_s(\boldsymbol{\kappa}_s) = \sum_{i\in\mathcal{N}} \widetilde{\kappa}_s(i)\, c_{\mathrm{f}}(i) + \sum_{i\in\mathcal{N}} \sum_{v\in\mathcal{N}_s} \sum_{n\in\Upsilon} \kappa_s(i,v)\, r_{s,n}(v)\, c_n(i) \\ + \sum_{ij\in\mathcal{E}} \sum_{vw\in\mathcal{E}_s} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw)\, c_{\mathrm{b}}(ij), \tag{7.18}$$

where

$$\widetilde{\kappa}_s(i) = \begin{cases} 1 & \text{if } \sum_{v\in\mathcal{N}_s} \kappa_s(i,v) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{7.19}$$

The first term of $C_s(\boldsymbol{\kappa}_s)$ represents the fixed costs associated to the use of infrastructure nodes by slice s, whereas the second and the third terms indicate the cost of reserved resources from infrastructure nodes and links. The variable $\widetilde{\kappa}_s(i)$ indicates whether the infrastructure node i is used by slice s.

Finally, the total earnings $E_s(\boldsymbol{\kappa}_s)$ obtained by the InP for the successful provisioning of slice s is

$$E_s(\boldsymbol{\kappa}_s) = I_s x_s(\boldsymbol{\kappa}_s) - C_s(\boldsymbol{\kappa}_s). \tag{7.20}$$

7.3.3 Nonlinear Constrained Optimization Problem

Consider a set of slices $\mathcal{S}$. The resource provisioning problem for all slices $s \in \mathcal{S}$, which accounts for uncertain slice user demands and tries to limit the impact on background services, can be formulated as

Problem 7.1: Nonlinear Constrained Optimization

$$\underset{\boldsymbol{\kappa}=\{\boldsymbol{\kappa}_s\}_{s\in\mathcal{S}}}{\text{maximize}} \quad \sum_{s\in\mathcal{S}} E_s\left(\boldsymbol{\kappa}_s\right) = \sum_{s\in\mathcal{S}} \left(I_s x_s\left(\boldsymbol{\kappa}_s\right) - C_s\left(\boldsymbol{\kappa}_s\right)\right),$$

subject to

$$\sum_{s\in S}\sum_{v\in\mathcal{N}_s} \kappa_s\left(i,v\right) r_{s,n}\left(v\right) \leqslant a_n\left(i\right), \forall i \in \mathcal{N}, n \in \Upsilon,$$

$$\sum_{s\in S}\sum_{vw\in\mathcal{E}_s} \kappa_s\left(ij,vw\right) r_{s,\mathrm{b}}\left(vw\right) \leqslant a_{\mathrm{b}}\left(ij\right), \forall ij \in \mathcal{E},$$

$$p_n^{\mathrm{im}}\left(i\right) \leqslant \overline{p}^{\mathrm{im}}, \forall i \in \mathcal{N}, n \in \Upsilon,$$

$$p_{\mathrm{b}}^{\mathrm{im}}\left(ij\right) \leqslant \overline{p}^{\mathrm{im}}, \forall ij \in \mathcal{E},$$

$$p_s\left(\boldsymbol{\kappa}_s\right) \geqslant \underline{p}_s, \forall s \in \mathcal{S},$$

and for each $s \in \mathcal{S}$, $i \in \mathcal{N}$, and $vw \in \mathcal{E}_s$:

$$\sum_{j\in\mathcal{N}} \left[\kappa_s\left(ij,vw\right) - \kappa_s\left(ji,vw\right)\right] = \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum\limits_{vu} r_{s,\mathrm{b}}(vu)}\right)\kappa_s\left(i,v\right) - \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum\limits_{uw} r_{s,\mathrm{b}}(uw)}\right)\kappa_s\left(i,w\right).$$

Solving Problem 7.1 is complex due to the need to evaluate $p_s\left(\boldsymbol{\kappa}_s\right)$ defined by (7.15) in the verification of the constraint $p_s\left(\boldsymbol{\kappa}_s\right) \geqslant \underline{p}_s$, $\forall s \in \mathcal{S}$ in (7.16). Section 7.4 introduces a simpler method to solve Problem 7.1.

7.4 Reduced-Complexity Slice Resource Provisioning

In this section, a parameterized ILP formulation of Problem 7.1 is introduced. The main idea is to replace the constraints (7.12), (7.13), and (7.16) involving probabilities related to random variables describing the aggregate user demands and best-effort services by linear deterministic constraints.

7.4.1 Linear Inequality Constraints for the SSP

For a given slice $s \in \mathcal{S}$ and for each $v \in \mathcal{N}_s$, $vw \in \mathcal{E}_s$, and $n \in \Upsilon$, let

$$\widehat{R}_{s,n}\left(v,\gamma_s\right) = \overline{R}_{s,n}\left(v\right) + \gamma_s \widetilde{R}_{s,n}\left(v\right), \tag{7.21}$$

$$\widehat{R}_{s,\mathrm{b}}\left(vw,\gamma_s\right) = \overline{R}_{s,\mathrm{b}}\left(vw\right) + \gamma_s \widetilde{R}_{s,\mathrm{b}}\left(vw\right), \tag{7.22}$$

be the target aggregate user demand, depending on some parameter $\gamma_s > 0$. $\overline{R}_{s,n}\left(v\right)$ and $\widetilde{R}_{s,n}\left(v\right)$ are the mean and deviation of $R_{s,n}\left(v\right)$, while $\overline{R}_{s,\mathrm{b}}\left(vw\right)$ and $\widetilde{R}_{s,\mathrm{b}}\left(vw\right)$ are the mean and deviation of $R_{s,\mathrm{b}}\left(vw\right)$. Calculation for those quantities are based on the mean and standard deviation of N_s and $\mathbf{U}_s$ as follows.

Assuming the number of users of slice s (N_s) and the resource demands of each

user of this slice ($U_{s,n}(v)$ and $U_{s,\mathrm{b}}(vw)$) are independently distributed. Consider first $R_{s,n}(v)$. Denoting $\mathbb{E}[N_s] = \overline{N}_s$, $\mathrm{Var}(N_s) = \widetilde{N}_s^2$, $\mathbb{E}[U_{s,n}(v)] = \overline{U}_{s,n}(v)$ and $\mathrm{Var}(U_{s,n}(v)) = \widetilde{U}_{s,n}^2(v)$. The mean and variance of $R_{s,n}(v)$, for all $n \in \Upsilon$ and $v \in \mathcal{N}_s$, can be evaluated as

$$\overline{R}_{s,n}(v) = \mathbb{E}[N_s U_{s,n}(v)] = \mathbb{E}[N_s]\,\mathbb{E}[U_{s,n}(v)] = mp\overline{U}_{s,n}(v),$$

$$\begin{aligned}\overline{R}_{s,n}(v) =& \mathbb{E}(N_s U_{s,n}(v)) = \overline{N}_s \overline{U}_{s,n}(v),\\ \widetilde{R}_{s,n}^2(v) =& \overline{N}_s^2 \widetilde{U}_{s,n}^2(v) + \overline{U}_{s,n}^2(v)\,\widetilde{N}_s^2 + \widetilde{N}_s^2 \widetilde{U}_{s,n}^2(v),\end{aligned}$$

see [Goodman, 1960]. Similarly, for $R_{s,\mathrm{b}}(vw)$, $\forall vw \in \mathcal{E}_s$, one obtains

$$\begin{aligned}\overline{R}_{s,\mathrm{b}}(vw) &= \overline{N}_s \overline{U}_{s,\mathrm{b}}(vw),\\ \widetilde{R}_{s,\mathrm{b}}^2(vw) &= \overline{N}_s^2 \widetilde{U}_{s,\mathrm{b}}^2(vw) + \overline{U}_{s,\mathrm{b}}^2(vw)\,\widetilde{N}_s^2 + \widetilde{N}_s^2 \widetilde{U}_{s,\mathrm{b}}^2(vw).\end{aligned}$$

Now consider

$$p_s(\gamma_s) = \Pr\left\{\begin{array}{r}\widehat{R}_{s,n}(v,\gamma_s) \geqslant R_{s,n}(v), \forall v, n,\\ \widehat{R}_{s,\mathrm{b}}(vw,\gamma_s) \geqslant R_{s,\mathrm{b}}(vw), \forall vw\end{array}\right\}, \tag{7.23}$$

one has to determine the smallest value of γ_s such that $p_s(\gamma_s) \geqslant \underline{p}_s$(see (7.16)). Then if, for a given assignment $\boldsymbol{\kappa}_s$,

$$\sum_i \kappa_s(i,v)\, r_{s,n}(v) \geqslant \widehat{R}_{s,n}(v,\gamma_s), \forall n, v, \tag{7.24}$$

$$\sum_{ij} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant \widehat{R}_{s,\mathrm{b}}(vw,\gamma_s), \forall vw, \tag{7.25}$$

are satisfied, then combining (7.24)–(7.25) with the definition of $p_s(\boldsymbol{\kappa}_s)$ given in (7.15), the SSP constraint (7.16) is satisfied.

The main difficulty is now to determine the smallest value of γ_s such that $p_s(\gamma_s) \geqslant \underline{p}_s$, since the larger γ_s, the more difficult the satisfaction of (7.24) and (7.25).

Using (7.3), one has

$$p_s(\gamma_s) = \sum_{\eta=1}^{m} p_\eta \int_{\widehat{\mathcal{R}}(\gamma_s)} f\left(\mathbf{x}, \eta\boldsymbol{\mu}, \eta^2\boldsymbol{\Gamma}\right) \mathrm{d}\mathbf{x}, \tag{7.26}$$

where $\widehat{\mathcal{R}}(\gamma_{s,k}) = \left\{\mathbf{x} \in \mathbb{R}^{n_\mathrm{R}} \,|\, \mathbf{x} \leqslant \widehat{\mathbf{R}}(\gamma_{s,k})\right\}$ and

$$\widehat{\mathbf{R}}(\gamma_s) = \left[\widehat{R}_{s,\mathrm{c}}(v_1,\gamma_s), \widehat{R}_{s,\mathrm{m}}(v_1,\gamma_s), \cdots, \widehat{R}_{s,\mathrm{b}}(v_1v_2,\gamma_s), \cdots\right] \tag{7.27}$$

is a vector of size $n_\mathrm{R} = 3|\mathcal{N}_s| + |\mathcal{E}_s|$.

Since the pmf of the number of users p_η, $\eta = 1, \ldots, m$ is assumed to be known,

the value of γ_s such that $p_s(\gamma_s) = \underline{p}_s$ may be obtained, *e.g.*, by the bisection methods [Burden and Douglas Faires, 2011]. The multidimensional integral in (7.26) can be evaluated using a quasi-Monte Carlo integration algorithm presented in [Genz, 2004]. An example of the evolution of $p_s(\gamma_s)$ as function of γ_s for a given slice s of Type 1 is depicted in Figure 7.2, using the simulation setting described in Section 7.5.1.

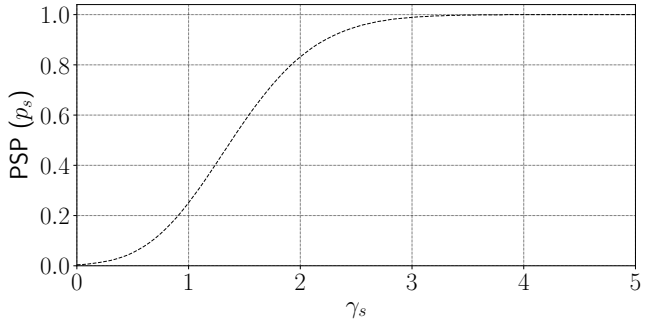


Figure 7.2: Evolution of p_s as function of γ_s.

7.4.2 Linear Inequality Constraints for the ImP

For each $i \in \mathcal{N}$, $ij \in \mathcal{E}$, and $n \in \Upsilon$, consider the following target level of background service demands

$$\widehat{B}_n(i, \gamma_{\mathrm{B}}) = \overline{B}_n(i) + \gamma_{\mathrm{B}} \widetilde{B}_n(i), \tag{7.28}$$

$$\widehat{B}_{\mathrm{b}}(ij, \gamma_{\mathrm{B}}) = \overline{B}_{\mathrm{b}}(ij) + \gamma_{\mathrm{B}} \widetilde{B}_{\mathrm{b}}(ij), \tag{7.29}$$

where $\gamma_{\mathrm{B}} > 0$ is some tuning parameter. For an assignment $\boldsymbol{\kappa} = \{\boldsymbol{\kappa}_s\}_{s \in \mathcal{S}}$ that satisfies

$$\sum_{s,v} \kappa_s(i, v)\, r_{s,n}(v) \leqslant a_n(i) - \widehat{B}_n(i, \gamma_{\mathrm{B}}), \forall n \in \Upsilon, i \in \mathcal{N}, \tag{7.30}$$

$$\sum_{s,vw} \kappa_s(ij, vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij) - \widehat{B}_{\mathrm{b}}(ij, \gamma_{\mathrm{B}}), \forall ij \in \mathcal{E}, \tag{7.31}$$

and (7.8, 7.9, 7.14), the ImP defined in (7.10) can be evaluated as follows

$$\begin{aligned} p_n^{\mathrm{im}}(i) = \Pr\left\{ B_n(i) \geqslant \widehat{B}_n(i, \gamma_{\mathrm{B}}) \right\} &= \int_{\widehat{B}_n(i,\gamma_{\mathrm{B}})}^{+\infty} f\left(x; \overline{B}_n(i), \widetilde{B}_n^2(i)\right) \mathrm{d}x \\ &= 1 - \int_{-\infty}^{\widehat{B}_n(i,\gamma_{\mathrm{B}})} f\left(x; \overline{B}_n(i), \widetilde{B}_n^2(i)\right) \mathrm{d}x \\ &= 1 - \Phi(\gamma_{\mathrm{B}}), \end{aligned} \tag{7.32}$$

where Φ is the cumulative distribution function (CDF) of the zero-mean, unit-variance normal distribution. Similarly, the ImP defined in (7.11) can also be evalu-

ated as

$$p_{s,\mathrm{b}}^{\mathrm{im}}(ij) = \Pr\left\{B_{\mathrm{b}}(ij) \geqslant \widehat{B}_{\mathrm{b}}(ij, \gamma_{\mathrm{B}})\right\} = 1 - \Phi(\gamma_{\mathrm{B}}). \tag{7.33}$$

It is observed that, from (7.32) and (7.33), $p_n^{\mathrm{im}}(i)$ and $p_{s,\mathrm{b}}^{\mathrm{im}}(ij)$ are independent of $\boldsymbol{\kappa}_s$, for all $s \in \mathcal{S}$. To satisfy the impact constraints imposed by (7.10, 7.11), γ_{B} has to be chosen such that

$$1 - \Phi(\gamma_{\mathrm{B}}) \leqslant \overline{p}^{\mathrm{im}} \Leftrightarrow \gamma_{\mathrm{B}} \geqslant \Phi^{-1}\left(1 - \overline{p}^{\mathrm{im}}\right). \tag{7.34}$$

Since the larger γ_{B}, the more difficult the satisfaction of (7.30) and (7.31), the optimal γ_{B} would be $\gamma_{\mathrm{B}} = \Phi^{-1}\left(1 - \overline{p}^{\mathrm{im}}\right)$.

7.4.3 ILP Formulation for Multiple Slice Provisioning

With the relaxed SSP and ImP constraints introduced in Sections 7.4.1 and 7.4.2, we replace the nonlinear-constrained optimization in Problem 7.1, by a relaxed parameterized optimization in Problem 7.2.

Problem 7.2: ILP for Multiple Slice Resource Provisioning

$$\underset{\{\boldsymbol{d},\boldsymbol{\kappa}\}=\{d_s,\boldsymbol{\kappa}_s\}_{s\in\mathcal{S}}}{\text{maximize}} \quad \sum_{s\in\mathcal{S}} \left(I_s d_s - C_s(\boldsymbol{\kappa}_s)\right),$$

subject to

$$\sum_{i\in\mathcal{N}} \kappa_s(i,v)\, r_{s,n}(v) \geqslant \widehat{R}_{s,n}(v,\gamma_s)\, d_s, \forall s \in \mathcal{S}, v \in \mathcal{N}_s, n \in \Upsilon$$

$$\sum_{ij\in\mathcal{E}} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant \widehat{R}_{s,\mathrm{b}}(vw,\gamma_s)\, d_s, \forall s \in \mathcal{S}, vw \in \mathcal{E}_s,$$

$$\sum_{s\in\mathcal{S}} \sum_{v\in\mathcal{N}_s} \kappa_s(i,v)\, r_{s,n}(v) \leqslant a_n(i) - \widehat{B}_n(i,\gamma_{\mathrm{B}}), \forall i \in \mathcal{N}, n \in \Upsilon,$$

$$\sum_{s\in\mathcal{S}} \sum_{vw\in\mathcal{E}_s} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij) - \widehat{B}_{\mathrm{b}}(ij,\gamma_{\mathrm{B}}), \forall ij \in \mathcal{E}.$$

and for each $s \in \mathcal{S}$, $i \in \mathcal{N}$, and $vw \in \mathcal{E}_s$:

$$\sum_{j\in\mathcal{N}} \left[\kappa_s(ij,vw) - \kappa_s(ji,vw)\right] = \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{vu} r_{s,\mathrm{b}}(vu)}\right)\kappa_s(i,v) - \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{uw} r_{s,\mathrm{b}}(uw)}\right)\kappa_s(i,w).$$

Problem 7.2 is now an ILP. The binary variable d_s, $s \in \mathcal{S}$, indicates whether resources are actually provisioned for slice s. When $d_s = 0$, the minimization of the provisioning cost $C_s(\boldsymbol{\kappa}_s)$ imposed by the objective function of Problem 7.2 will enforce $\kappa_s = 0$ in the first two constraints of Problem 7.2. Remind that γ_s and γ_{B} are evaluated by dichotomy search, as discussed in Sections 7.4.1 and 7.4.2, before solving Problem 7.2.

7.4.4 ILP Formulation for Slice-by-Slice Provisioning

The number of variables involved in Problem 7.2 may be relatively large when several slices have to be considered jointly. This section introduces a reduced-complexity formulation where provisioning is performed slice-by-slice.

Consider the set of n_s slices $\mathcal{S} = \{s_1, \ldots, s_{n_\mathrm{s}}\}$ for which resources have to be provisioned. Assume that the slice-by-slice resource provisioning has been performed up to slice $s_{\ell-1}$, $1 \leqslant \ell - 1 < n_\mathrm{s}$. A successful provisioning is indicated by $d_s = 1$, whereas $d_s = 0$ indicates that resources cannot be provisioned for slice s, due, *e.g.*, to the lack of infrastructure resources leading to the non-satisfaction of the SSP or ImP constraints. The corresponding assignments are represented by $\boldsymbol{\kappa}_s$, $s \in \{s_1, \ldots, s_{\ell-1}\}$.

Slice s_ℓ is now considered. In the provisioning for slice s_ℓ, one has simply to account for the amount of infrastructure resources left after the provisioning of all slices $s \in \{s_1, \ldots, s_{\ell-1}\}$. Consequently, only the third and fourth constraint of Problem 7.2 have to be updated to get the following new ILP formulation (Problem 7.3) for slice-by-slice resource provisioning.

Problem 7.3: ILP for Slice-by-Slice Resource Provisioning

$$\underset{d_{s_\ell}, \boldsymbol{\kappa}_{s_\ell}}{\text{maximize}} \; I_{s_\ell} d_{s_\ell} - C_{s_\ell}\left(\boldsymbol{\kappa}_{s_\ell}\right),$$

subject to

$$(1) \sum_{i \in \mathcal{N}} \kappa_{s_\ell}(i,v)\, r_{s,n}(v) \geqslant \widehat{R}_{s_\ell,n}(v, \gamma_{s_\ell})\, d_{s_\ell}, \forall v \in \mathcal{N}_s, n \in \Upsilon,$$

$$(2) \sum_{ij \in \mathcal{E}} \kappa_{s_\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant \widehat{R}_{s_\ell,\mathrm{b}}(vw, \gamma_{s_\ell})\, d_{s_\ell}, \forall vw \in \mathcal{E}_s,$$

and for each $i \in \mathcal{N}$ and $n \in \Upsilon$:

$$(3) \sum_{v \in \mathcal{N}_s} \kappa_{s_\ell}(i,v)\, r_{s,n}(v) \leqslant a_n(i) - \widehat{B}_n(i, \gamma_\mathrm{B}) - \sum_{s \in \{s_1,\ldots,s_{\ell-1}\}} \kappa_s(i,v)\, r_{s,n}(v)\, d_s,$$

and for each $ij \in \mathcal{E}$:

$$(4) \sum_{vw \in \mathcal{E}_s} \kappa_{s_\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_\mathrm{b}(ij) - \widehat{B}_\mathrm{b}(ij, \gamma_\mathrm{B}) - \sum_{s \in \{s_1,\ldots,s_{\ell-1}\}} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw)\, d_s,$$

and for each $i \in \mathcal{N}$, and $vw \in \mathcal{E}_s$:

$$(5) \sum_{j \in \mathcal{N}} \left[\kappa_s(ij,vw) - \kappa_s(ji,vw)\right] = \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{vu} r_{s,\mathrm{b}}(vu)}\right) \kappa_s(i,v) - \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{uw} r_{s,\mathrm{b}}(uw)}\right) \kappa_s(i,w).$$

The order in which the provisioning is performed is important. One may choose to provision the slices by decreasing income I_s. An other possibility is to perform a greedy search, starting with the slice $s^1 \in \mathcal{S}$ for which $I_s d_s - C_s(\boldsymbol{\kappa}_s)$ is maximized, when deployed alone. Then, assuming that resources have been provisioned for s^1, one may search $s^2 \in \mathcal{S} \setminus \{s^1\}$ maximizing $I_s d_s - C_s(\boldsymbol{\kappa}_s)$ with the remaining infrastructure resources, and so forth.

7.4.5 Slice Resource Provisioning Algorithms

From the suboptimal methods introduced in Sections 7.4.3 and 7.4.4, we propose four uncertainty-aware slice resource provisioning variants, JP-B and JP considering the joint provisioning approach introduced in Problem 7.2; SP-B and SP considering the sequential provisioning approach introduced in Problem 7.3.

The JP-B and SP-B approaches account for the impact of provisioning on background services, whereas the JP and SP approaches do not take those services into account. This is obtained by setting $\overline{B}_n(i, \gamma_{\mathrm{B}}) = 0, \forall n, i$ and $\overline{B}_{\mathrm{b}}(ij, \gamma_{\mathrm{B}}) = 0, \forall ij$ in Problems 2 and 3, see also Chapter 5, Sections 5.3.1 and 5.3.2, where slice resource demands are considered to be deterministic. Compared to the approaches introduced in Chapter 5, the SP and JP approaches in this chapter account additionally for the uncertainties of slice resource demands. Moreover, while the main decision variables in the original SP and JP approaches in [Luu et al., 2018] are the proportion of available resources in the infrastructure, here, the main decision variables are the number of SFC instances for which resources have to be provisioned for a future deployment.

Algorithme 7.1 : Joint Approaches (JP-B and JP)

Input : $\mathcal{G} = (\mathcal{N}, \mathcal{E}), \mathcal{S}, \{\mathcal{G}_s, s \in \mathcal{S}\}$
Output : $\widehat{\boldsymbol{\kappa}} = \{\widehat{\boldsymbol{\kappa}}_s\}_{s \in \mathcal{S}}$

```
switch provisioning_variant do
    case JP-B (background traffics taken into account) do
        Solve Problem 7.2 to obtain κ̂;
    case JP (background traffic ignored) do
        Solve Problem 7.2 with B = ∅ to obtain κ̂;
```

Algorithme 7.2 : Sequential Approaches (SP-B and SP)

Input : $\mathcal{G} = (\mathcal{N}, \mathcal{E}), \mathcal{S}, \{\mathcal{G}_s, s \in \mathcal{S}\}$
Output : $\widehat{\boldsymbol{\kappa}} = \{\widehat{\boldsymbol{\kappa}}_s\}_{s \in \mathcal{S}}$

```
switch provisioning_variant do
    case SP-B (background traffic taken into account) do
        for ℓ = 1, ..., |S| do
            Solve Problem 7.3 to obtain κ̂_{s_ℓ};
    case SP (background traffics ignored) do
        for ℓ = 1, ..., |S| do
            Solve Problem 7.3 with B = ∅ to obtain κ̂_{s_ℓ};
```

These four provisioning variants are summarized in Algorithms 7.1 and 7.2. Each variant requires the solution of one or several MILPs, whose complexity is exponential in the number of variables in the worst case. The number of MILPs, variables, and of constraints involved in each variant are summarized in Table 7.3.

The JP-B and JP approaches (Algorithm 7.1) require the solution of one single MILP, while the SP-B and SP approaches (Algorithm 7.2) split the work into $|\mathcal{S}|$ subproblems, each of which implies $|\mathcal{S}|$ times less variables than the joint variants (JP-B and JP). Due to the exponential complexity of each problem, solutions for the sequential variants may be obtained faster than with the joint variants. Section 7.5 presents a more detailed performance comparison of these variants.

Table 7.3: Number of MILPs, variables, and of constraints involved in each variant.

Variant	*#probs*	*#variables/problem*	*#constraints/problem*
JP-B and JP	1	$\|\mathcal{N}_s\| + \|\mathcal{S}\|\,(1 + \|\mathcal{N}\|\,\|\mathcal{N}_s\|) + \|\mathcal{S}\|\,\|\mathcal{E}\|\,\|\mathcal{E}_s\|$	$\|\mathcal{S}\|\,(\|\mathcal{N}\|\,\|\mathcal{E}_s\| + 3\,\|\mathcal{N}_s\| + \|\mathcal{E}_s\|) + 3\,\|\mathcal{N}\| + \|\mathcal{E}\|$
SP-B and SP	$\|\mathcal{S}\|$	$\|\mathcal{N}_s\| + \|\mathcal{N}\|\,\|\mathcal{N}_s\| + \|\mathcal{E}\|\,\|\mathcal{E}_s\| + 1$	$\|\mathcal{S}\|\,\|\mathcal{N}\|\,\|\mathcal{E}_s\| + 3\,\|\mathcal{N}_s\| + \|\mathcal{E}_s\| + 3\,\|\mathcal{N}\| + \|\mathcal{E}\|$

7.5 Evaluation

In this section, one evaluates via simulations the performance of the four variants (JP-B, SP-B, JP, and SP) of the provisioning algorithms described in Section 7.4. The simulation setup is described in Section 7.5.1. All numerical results presented in Section 7.5.2 have been performed with the CPLEX MILP solver interfaced with MATLAB.

7.5.1 Simulation Conditions

7.5.1.1 Infrastructure Topology

The fat-tree topology introduced in Chapter 5 is reused, with $K = 2$. The cost of using each resource of the infrastructure network is $c_n(i) = 1$, $\forall n \in \Upsilon$, $c_{\mathrm{f}}(i) = 65$, 60, 55, 50 for respectively central, regional, edge, RRH nodes, and $c_{\mathrm{b}}(ij) = 1$, $\forall ij \in \mathcal{E}$.

7.5.1.2 Background Services

One considers a simplified model in which the resources consumed by best-effort background services at each infrastructure node link follow the same distribution. Precisely, at each infrastructure node $i \in \mathcal{N}$ and link $ij \in \mathcal{E}$, the resources consumed by background services follow a normal distribution with mean and standard deviation equal to respectively $20\,\%$ and $5\,\%$ of the available resource at that node and link, *i.e.*, $\mu_{\mathrm{B},n}(i) = 0.2a_n(i)$, $\sigma_{\mathrm{B},n}(i) = 0.05a_n(i)$, $\forall i \in \mathcal{N}$, $\forall n \in \Upsilon$, and $\mu_{\mathrm{B,b}}(ij) = 0.2a_{\mathrm{b}}(ij)$, $\sigma_{\mathrm{B,b}}(ij) = 0.05a_{\mathrm{b}}(ij)$, $\forall ij \in \mathcal{E}$.

7.5.1.3 Slice Resource Demand (S-RD)

The three types of slices considered in Chapter 6 are reused. The characteristics of each slice of each type within the considered time slot are as follows

- Slices of type 1 aim to provide an HD video streaming service at an average rate of 4 Mbps for VIP users, *e.g.*, in a stadium. The number of users follows a binomial distribution $\mathcal{B}(300, 0.9)$;
- Slices of type 2 are dedicated to provide an SD video streaming service at an average rate of 2 Mbps. The number of users follows a binomial distribution $\mathcal{B}(1000, 0.8)$;
- Slices of type 3 aim to provide a video surveillance and traffic monitoring service at an average rate of 1 Mbps for 100 cameras, *e.g.*, installed along a highway.

The functional architecture of each type is given in Chapter 6, Section 6.5.1.2. The values of U-RD, SFC-RD, and S-RD are given in Table 7.4. Numerical values in Table 7.4 have been adapted from [Savi et al., 2016].

Table 7.4: Parameters of U-RD, SFC-RD, and S-RD.

Type 1: HD video streaming at 4 Mbps. $N_s \sim \mathcal{B}(300, 0.9)$, $I_s = 900$, $\underline{p}_s = 0.99$

Node	$(\overline{U}_{s,\mathrm{c}}, \widetilde{U}_{s,\mathrm{c}})$	$(\overline{U}_{s,\mathrm{m}}, \widetilde{U}_{s,\mathrm{m}})$	$(\overline{U}_{s,\mathrm{w}}, \widetilde{U}_{s,\mathrm{w}})$	$(r_{\mathrm{c}}, r_{\mathrm{m}}, r_{\mathrm{w}})$	*Link*	$(\overline{U}_{s,\mathrm{b}}, \widetilde{U}_{s,\mathrm{b}})$	$r_{s,\mathrm{b}}$
vVOC	(5.4, 0.54) e-3	(1.5, 0.15) e-2	—	(0.29, 0.81, 0)	vVOC→vGW	(4, 0.4) e-3	0.22
vGW	(9.0, 0.90) e-4	(5.0, 0.50) e-4	—	(0.05, 0.03, 0)	vGW→vBBU	(4, 0.4) e-3	0.22
vBBU	(8.0, 0.80) e-4	(5.0, 0.50) e-4	(4, 0.4) e-3	(0.04, 0.03, 0.2)			

Type 2: SD video streaming at 2 Mbps. $N_s \sim \mathcal{B}(1000, 0.8)$, $I_s = 1000$, $\underline{p}_s = 0.95$

Node	$(\overline{U}_{s,\mathrm{c}}, \widetilde{U}_{s,\mathrm{c}})$	$(\overline{U}_{s,\mathrm{m}}, \widetilde{U}_{s,\mathrm{m}})$	$(\overline{U}_{s,\mathrm{w}}, \widetilde{U}_{s,\mathrm{w}})$	$(r_{\mathrm{c}}, r_{\mathrm{m}}, r_{\mathrm{w}})$	*Link*	$(\overline{U}_{s,\mathrm{b}}, \widetilde{U}_{s,\mathrm{b}})$	$r_{s,\mathrm{b}}$
vVOC	(1.1, 0.11) e-3	(7.5, 0.75) e-3	—	(0.17, 1.20, 0)	vVOC→vGW	(2, 0.2) e-3	0.32
vGW	(1.8, 0.18) e-4	(2.5, 0.25) e-4	—	(0.03, 0.04, 0)	vGW→vBBU	(2, 0.2) e-3	0.32
vBBU	(0.8, 0.08) e-4	(2.5, 0.25) e-4	(2, 0.2) e-3	(0.01, 0.04, 0.3)			

Type 3: Video surveillance and traffic monitoring at 1 Mbps. $N_s = 50$, $I_s = 800$, $\underline{p}_s = 0.9$

Node	$(\overline{U}_{s,\mathrm{c}}, \widetilde{U}_{s,\mathrm{c}})$	$(\overline{U}_{s,\mathrm{m}}, \widetilde{U}_{s,\mathrm{m}})$	$(\overline{U}_{s,\mathrm{w}}, \widetilde{U}_{s,\mathrm{w}})$	$(r_{\mathrm{c}}, r_{\mathrm{m}}, r_{\mathrm{w}})$	*Link*	$(\overline{U}_{s,\mathrm{b}}, \widetilde{U}_{s,\mathrm{b}})$	$r_{s,\mathrm{b}}$
vBBU	(2.0, 0.20) e-4	(1.3, 0.13) e-4	(1, 0.1) e-3	(0.4, 0.25, 2) e-2	vBBU→vGW	(1, 0.1) e-3	0.02
vGW	(9.0, 0.90) e-4	(1.3, 0.13) e-4	—	(0.018, 0.003, 0)	vGW→vTM	(1, 0.1) e-3	0.02
vTM	(1.1, 0.11) e-3	(1.3, 0.13) e-4	—	(0.266, 0.003, 0)	vTM→vVOC	(1, 0.1) e-3	0.02
vVOC	(5.4, 0.54) e-3	(3.8, 0.38) e-3	—	(0.108, 0.080, 0)	vVOC→vIDPS	(1, 0.1) e-3	0.02
vIDPS	(1.1, 0.11) e-2	(1.3, 0.13) e-4	—	(0.214, 0.003, 0)			

7.5.2 Results

This section illustrates the performance of the various resource provisioning variants, in terms of: utilization of infrastructure nodes and links, maximal probability of impact p^{im} on the background services at every infrastructure node and link, provisioning cost, total earnings of the InP, and number of impacted nodes and links, *i.e.*, the number of nodes $i \in \mathcal{N}$ such that, for each $n \in \Upsilon$, $p_n^{\mathrm{im}}(i) > \overline{p}^{\mathrm{im}}$; and links $ij \in \mathcal{E}$ such that $p_{\mathrm{b}}^{\mathrm{im}}(ij) > \overline{p}^{\mathrm{im}}$.

We first evaluate the influence of background services on the slice resource provisioning approaches. This is done by comparing the two variants SP-B and SP in Section 7.5.2.1 and 7.5.2.2, considering (*i*) a single and (*ii*) a multiple slice provisioning problem. In Section 7.5.2.3, the performance of the four proposed resource provisioning variants (JP, SP, JP-B, and SP-B) are compared. Finally, the benefits of the uncertainty-aware slice resource provisioning approach in terms of improved probability of successful provisioning are illustrated in Section 7.5.2.4.

7.5.2.1 Provisioning of a Single Slice

Table 7.5 shows the performance of two variants SP-B and SP for the provisioning of a single slice of Type 1, where $\underline{p}_s = 0.99$ and $\overline{p}^{\text{im}} = 0.1$. These two variants differ from each other in whether the impact on background service is considered or not.

The SP variant, which does not account for the impact on background services, has a lower link usage and provisioning cost, and yields a higher earning for the InP than that of the SP-B variant. Nevertheless, as expected, the SP variant has a higher impact on background services, with a maximal impact probability of 0.58 exceeding the maximum tolerated impact probability $\overline{p}^{\text{im}}$ at one infrastructure node, as summarized in Table 7.5.

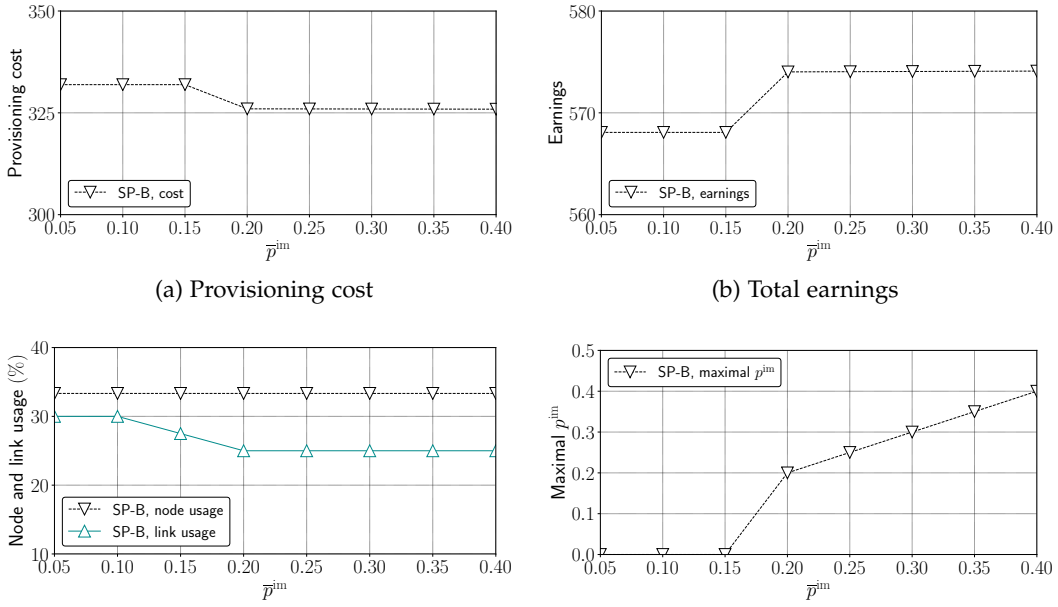
Table 7.5: Performance of SP-B and SP on resource provisioning for a single slice.

Criteria	SP-B	SP
Node usage	33%	33%
Link usage	28%	25%
Maximal p^{im}	1.26e-4	0.58
Provisioning cost	332	326
Total earnings	568	574
#impacted nodes	0	1
#impacted links	0	0

The way $\overline{p}^{\text{im}}$ affects the performance of the SP-B approach is shown in Figures 7.3a–7.3d, where $\underline{p}_s = 0.99$ and $\overline{p}^{\text{im}}$ ranges from 0.05 to 0.4. One observes that, the higher $\overline{p}^{\text{im}}$, the lower the provisioning cost and the higher the earnings for the InP. This is due to the fact that, with higher $\overline{p}^{\text{im}}$, it is easier to provision slices with a reduced amount of resources. This can be observed in the decrease of the link usage in Figure 7.3c. On the other hand, the impact probability p^{im} is always kept under the threshold $\overline{p}^{\text{im}}$ imposed by the InP, as shown in Figure 7.3d.

7.5.2.2 Provisioning Several Slices of the Same Type

Now, considering 10 slices of type 1, the SP-B and SP variants are compared in terms of acceptance rate, *i.e.*, percentage of slices that have been successfully provisioned (given by $\sum_{s\in\mathcal{S}} \frac{d_s}{|\mathcal{S}|}$) and number of impacted nodes and links (for which the impact probability is larger than $\overline{p}^{\text{im}}$), for different value of $\underline{p}_s$, see Figure 7.4a. The tolerated impact probability $\overline{p}^{\text{im}}$ is set to 0.1. As expected, when $\underline{p}_s$ increases, the acceptance



(a) Provisioning cost

(b) Total earnings

(c) Node and link usage

(d) Evolution of p^{im} as a function of $\overline{p}^{\text{im}}$

Figure 7.3: Performance of the SP-B approach on single slice provisioning problem with different values of $\overline{p}^{\text{im}}$, in terms of (a) provisioning costs, (b) total earnings, (c) node and link utilization, and (d) maximal impact probability p^{im}.

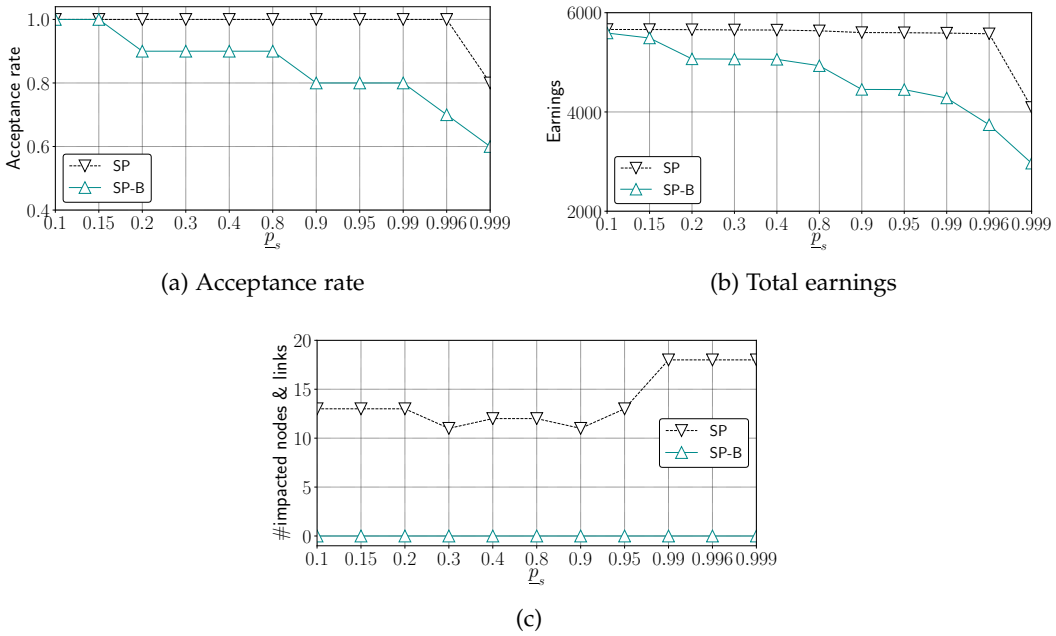
rate decreases for both approaches. The SP approach, which does not account for background services, always has a higher acceptance rate and earnings compared to the SP-B approach, but its impact on the background services is significantly larger (see also Figures 7.4b and 7.4c). Using the SP approach, provisioned resources are concentrated on a fewer amount of nodes and links. Consequently, the background services running on such nodes and links may then be affected.

7.5.2.3 Provisioning of Several Slices of Different Types

The performance of the four variants is illustrated in this section, when resources of 2 to 8 slices of three different types have to be provisioned. The number of slices of each type and their associated $\underline{p}_s$ are detailed in Tables 7.4 and 7.6. The impact probability threshold $\overline{p}^{\text{im}}$ is set to 0.1 in all scenarios.

Table 7.6: Number of slices of each type as a function of $|\mathcal{S}|$

Case	*#Type 1*	*#Type 2*	*#Type 3*
$\lvert\mathcal{S}\rvert = 2$	1	1	0
$\lvert\mathcal{S}\rvert = 4$	2	1	1
$\lvert\mathcal{S}\rvert = 6$	2	2	2
$\lvert\mathcal{S}\rvert = 8$	3	2	3



(a) Acceptance rate

(b) Total earnings

(c)

Figure 7.4: Performance of the SP-B and SP approaches the provisioning of multiple slices of one type, with different required minimum SSP, in terms of (a) acceptance rate and (b) total earnings.

The use of infrastructure nodes and links is shown in Figures 7.5a and 7.5b. The joint provisioning approaches (JP and JP-B) require a reduced amount of nodes and links compared to the sequential schemes (SP and SP-B). Moreover, considering the impact on background services requires, again, provisioning resources on more nodes and links.

Figure 7.5c shows the provisioning costs obtained with the various approaches. One observes that the JP variant yields the smallest cost among all variants, as it aims at finding an optimal solution for all slices, without considering the impact probability, contrary to the JP-B variant. This leads to the highest earnings for the InP, as shown in Figure 7.5d.

The total number of impacted nodes and links is shown in Figure 7.5e. The JP-B and SP-B variants have no impacted nodes or links, whereas the provisioning performed by the JP and SP approaches significantly impact the background services. The SP variant has a higher impact on the background services, due to the higher utilization of infrastructure nodes and links, as shown in Figures 7.5a and 7.5b.

From the InP perspective, the use of the JP and SP maximizes the earnings of the InP but violates background services at a significant number of infrastructure nodes and links. This may necessitate to reconfigure those background services. On contrary, by using the JP-B and SP-B variants, the InP can provision slices and preserve a tolerable impact on the background services. The price to be paid is somewhat degraded efficiency of node and link utilization and a higher provisioning cost compared to the impact-aware variants, leading to lower earnings for the InP. For instance, when provisioning for 4 slices, the JP-B variant uses around 72%

of the total infrastructure nodes to aggregate resources needed to support the slices, while only 66.7% of the nodes are employed by the JP method, leading to a reduction of 3.5% of total earnings, as depicted in Figures 7.5a and 7.5d.

As expected, the sequential provisioning methods (SP-B and SP) perform better in terms of computing time than the joint approaches (JP-B and JP), as shown in Figure 7.5f. Increasing the number of slices leads to an increase of the cardinality of the sets of variables $\boldsymbol{d}$ and $\boldsymbol{\kappa}$, and therefore increases the computing time. In sequential provisioning, slices are considered successively. There is only a very small difference (usually less than 5%) in computing time between the SP-B and SP approaches and between the JP-B and JP approaches.

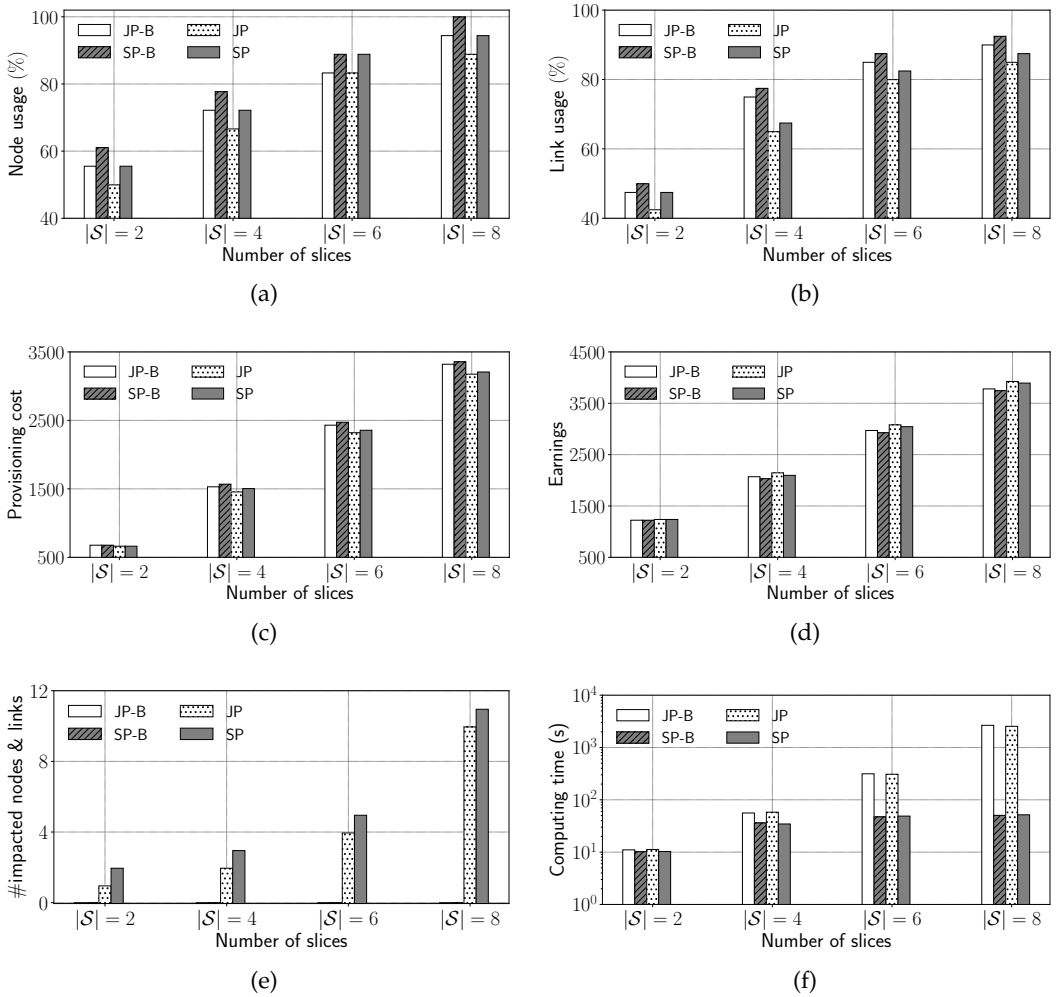


Figure 7.5: Performance comparison of 4 variants in terms of (a) utilization of infrastructure nodes, (b) utilization of infrastructure links, (c) provisioning costs, (d) total earnings, (e) number of impacted nodes and links, and (f) computing time.

7.5.2.4 Benefits of the Uncertainty-Aware Slice Resource Provisioning

In this section, we show the benefits of the proposed uncertainty-aware slice resource provisioning method, in terms of deployment efficiency, when considering the SFC embedding. Slice resource provisioning is performed for a *single slice*. JP-B and SP-B behave thus similarly. This is also the case for JP and SP.

The JP-B approach is compared to the JP approach, which does not account for the uncertainty of slice resource demands. Problem 7.2 is solved in the latter case with a slice resource demand corresponding to its mean value. This is done by choosing $\gamma_s = 0$ in the first and second constraint of Problem 7.2. Once provisioning is performed, the SFC embedding step is realized and a randomly generated number of users following the same distribution as that used in the provisioning process is considered. One gets an uncertainty-aware provisioning and embedding solution (UPE) and a deterministic provisioning and embedding solution (DPE). These solutions are compared in terms of satisfaction of the user demands.

A single slice of type 1 is considered. The U-RD, SFC-RD, and infrastructure parameters used in the previous parts of Section 7.5.2 are used again. For the S-RD, the number of users associated to the slice follows a binomial distribution $\mathcal{B}(m, p)$, where m is fixed to 300, and p varies. One thousand independent drawings of the number of users are performed. The number of SFCs that have to be actually deployed can be then deduced from the resulting number of users. SFCs can only be deployed when enough resources have been provisioned for the slice. Finally, the SFC acceptance rate, *i.e.*, the number of provisioned SFCs divided by the number of required SFCs, of the UPE and DPE solution is compared.

Figure 7.6 shows the average, minimum, and maximum SFC acceptance rates, when the probability p of the binomial distribution ranges from 0.4 to 0.9. The UPE solution provides a successful deployment of all SFCs. The DPE solution, which does not take into account the uncertainties of slice resource demands, cannot ensure the deployment for all SFCs, when not enough resources have been provisioned. In addition, as expected, when p is higher, *i.e.*, the slice resource demands become less uncertain, DPE yields a higher acceptance rate, with a smaller gap between the minimum, and maximum SFC acceptance rates, as shown in Figure 7.6.

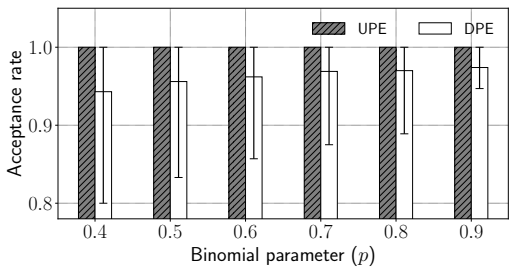


Figure 7.6: Performance comparison of the UPE and DPE solutions in terms of SFC acceptance rate.

7.6 Conclusion

This chapter investigates a resource provisioning method for network slicing robust to a partly unknown number of users whose resource demands are uncertain. Adopting the point of view of the InP, one tries to maximize its earnings, while providing a probabilistic guarantee that the slice resource demands are fulfilled. In addition to that, the proposed resource provisioning method is performed to keep the impact on the background services under a threshold imposed by the InP.

The uncertainty-aware slice resource provisioning is formulated as a nonlinear constrained optimization problem. A parameterized MILP formulation is then proposed. With the MILP formulation, four variants (JP, SP, JP-B, and SP-B) are introduced, for the solution of the provisioning problem for multiple slices jointly or sequentially, without or with consideration of the impact on background services.

With the JP-B and SP-B variants, resource provisioning is performed with a controlled impact on the background services. The JP and SP variants, on the other hand, do not account for the impact of resource provisioning on those services. Consequently, all resources of several infrastructure nodes and links may be consumed when using the JP and SP variants. This may impose a reconfiguration of background services. The price to be paid for the InP when performing the JP-B and SP-B variants is a reduction of its earnings.

Moreover, due to the exponential worst-case complexity in the number of variables of the MILP formulation, as expected, sequential approaches are shown to better scale to a larger number of slices. Sequential approaches have a somewhat degraded node and link utilization, a higher provisioning cost, which results in lower earnings, compared to the joint approaches.

In this chapter, uncertainties related to the fluctuation of user demands and the background services have been taken into account for the slice resource provisioning. A prospective extension to this work is to consider the dynamic behavior of slice provisioning requests, that is, each slice request has an arrival, activation (execution), and deactivation time. This approach will be considered in the next chapter.

Chapter 8

Admission Control and Resource Provisioning for Prioritized Slice Requests with Uncertainties

This chapter is based on Q.-T. Luu, S. Kerboeuf, and M. Kieffer, "Admission Control and Resource Provisioning for Prioritized Slice Requests with Uncertainties," submitted to IEEE Transactions on Network and Service Management, 2021. [Luu et al., 2021a].

This chapter aims to provide an answer to Challenge 4 introduced in Chapter 1. We propose a prioritized admission control mechanism for concurrent slices based on an infrastructure resource provisioning approach. It is an extension to the study presented in Chapter 7, by providing a more general provisioning mechanism that can be robust against both (*i*) the uncertainties due to the randomness of amount of resources employed by each user; and (*ii*) the dynamic behavior of the slice provisioning requests, each of which has an arrival, activation (execution), and deactivation time.

The remainder of this chapter is organized as follows. We start by highlighting the main contributions of this chapter in Section 8.1. In Section 8.2, we describe some important notations and hypotheses. The infrastructure and the slice resource demands are described in detail. Following that, we introduce in Section 8.3 some approaches to address an efficient adaptive slice resource provisioning, accounting for the dynamic nature of slice requests and robust to the uncertainties related to infrastructure and slice parameters. Numerical results are then provided in Section 8.5 to compare the performance of the dynamic provisioning approach, in comparison with a quasi-static approach, which we use as a baseline. Finally, Section 8.6 draws some conclusions and perspectives.

8.1 Contributions

Related works to this chapter can be found in Section 3.4 of Chapter 3. It can be observed that only few of previous papers cover both (*i*) the uncertain characteristic of resource demands and available infrastructure resources; and (*ii*) the dynamic characteristic of user requests, especially in a network slicing context. One may find,

for instance, in [Ghaznavi et al., 2015], a preliminary study where the VNF instance placement is optimized in response to the uncertain and dynamic characteristic of slice requests. Nevertheless, this paper only studied the problem of placing individual VNFs of a single type, thus cannot be applied to an SFC or slicing context, which usually involves a cooperation of VNFs of various types.

In this chapter, we adapt the slice resource provisioning approach studied in previous chapters to the problem of dynamic slice resource provisioning, with taking into account both above-mentioned dynamic aspects. This is the main difference with respect to the traditional SFC embedding/resource allocation approach considered in [Ghaznavi et al., 2015] and [Liu et al., 2017], and other related works on the problem of slice admission control, *e.g.*, in [Bega et al., 2017, Noroozi et al., 2019, Bega et al., 2020, Ebrahimi et al., 2020, Han et al., 2020].

8.2 Notations and Hypotheses

In this chapter, the characteristics of the SM-SLAs are similar to those in Chapter 7. In addition, the slice characteristics within an SM-SLA also include the priority class of the slice. When performing slice resource provisioning, the priority level of slices is taken into account.

Table 8.1 summarizes the main notations introduced in this chapter.

Table 8.1: Newly introduced notations in Chapter 8.

Symbol	*Description*
k	Time slot index
$\mathcal{P}_k$	Processing time interval in time slot k
T	Duration of a time slot
εT	Processing duration (of $\mathcal{P}_k$)
$\mathcal{S}_k$	Slices requests received before $(k+1)\,T - \varepsilon T$
$\mathcal{R}_k$	Slices requests processed during $\mathcal{P}_k$
P_s^{c}	Priority class
$P_{s,k}$	Priority level at time k
$\mathcal{K}_s$	Slice active interval, $\mathcal{K}_s = \left[k_s^{\mathrm{on}}, k_s^{\mathrm{off}}\right]$
$\boldsymbol{r}_s$	Vector of resource demands of an SFC
$\boldsymbol{U}_{s,k}$	Vector of resource demands of a typical user in time slot k
$\boldsymbol{R}_{s,k}$	Vector of aggregate resource demands in time slot k
$\boldsymbol{B}_k$	Vector of resources consumed by background services in time slot k

8.2.1 Network Model

As presented in Section 4.2.1, the considered types of resources at node level are $\Upsilon = \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$, denoting respectively computing, memory, and wireless resources. The model of the infrastructure network is similar to that introduced in Section 4.2.1.

8.2.2 Slice Provisioning Requests and Deployment Costs

8.2.2.1 Request Arrivals

As presented in Section 4.2, in this chapter, one considers that the lifetime of each slice s spans over several time slots denoted as $\left[k_s^{\text{on}}T, \left(k_s^{\text{off}}+1\right)T\right[$.

Resources have to be provisioned so as to be compliant with the variations of resource demands during the slice lifetime. The slice characteristics are assumed stable over each time slot, and may vary from one time slot to the next.

8.2.2.2 Slice Resource Demand

Within the SM-SLA, each slice s is associated with a priority class P_s^{c} indicating its priority level.

We add an additional subscript k to the notations of the U-RD and S-RD introduced in Section 4.2.3, *i.e.*, $\mathbf{U}_{s,k}$ and $\mathbf{R}_{s,k}$, to indicate the resource demands corresponding to each time slot k within the slice lifetime. Similarly, one denotes by $N_{s,k}$ the random number of independent users of slice s during time slot k. With the same calculation in Section 8.2.2.2 of Chapter 7, the total amount of resources employed by a random number $N_{s,k}$ of independent users, $\mathbf{R}_{s,k}$, is distributed according to

$$g\left(\mathbf{x}, \boldsymbol{\mu}_{s,k}, \boldsymbol{\Gamma}_{s,k}\right) = \sum_{\eta=0}^{\infty} p_{s,k,\eta} f\left(\mathbf{x}, \eta\boldsymbol{\mu}_{s,k}, \eta^2\boldsymbol{\Gamma}_{s,k}\right), \tag{8.1}$$

where $p_{s,k,\eta}$ is the probability that the number of users to be supported by slice s in the k-th time slot ($N_{s,k}$) is equal to η, *i.e.*, $p_{s,k,\eta} = \Pr\left(N_{s,k} = \eta\right)$.

8.2.2.3 Provisioning Adaptation Costs

During the lifetime of a slice, the amount of required slice resources may evolve. An increase of the required resources may impact the provisioning scheme by requiring more infrastructure resources to be provisioned. Compared to a situation where the resource provisioning is static for the whole lifespan of a slice, this induces more operations to be performed on the network infrastructure (assignment or re-assignment of resources, launching virtual machines on which VNFs will be operated) and results in additional costs to the InP. A cost $c_{\text{a}}\left(i\right)$ for each unit *increase* of the amount of VNF instances between two time slots is assumed to be charged by the InP to the MNO. Resource release costs are assumed to be incorporated within $c_{\text{a}}\left(i\right)$.

As will be seen in Section 8.3.5, this cost reduces SFC migrations within a given slice between consecutive time slots.

8.2.3 Resource Consumption of Background Services

We also add an additional subscript k to the notations of the background service, *i.e.*, $\mathbf{B}_k$. Performing the same calculations as in Section 7.2.3, during each time slot

k, $\mathbf{B}_k$ is distributed according to $f\left(\mathbf{x}; \boldsymbol{\mu}_{\mathrm{B},k}, \boldsymbol{\Gamma}_{\mathrm{B},k}\right)$, with

$$\boldsymbol{\mu}_{\mathrm{B},k} = \left[\overline{B}_{n,k}\left(i\right), \overline{B}_{\mathrm{b},k}\left(ij\right)\right]^{\top}_{(i,ij)\in\mathcal{G},n\in\Upsilon}, \tag{8.2}$$

$$\boldsymbol{\Gamma}_{\mathrm{B},k} = \mathrm{diag}\left[\widetilde{B}^{2}_{\mathrm{c},k}\left(i\right), \widetilde{B}^{2}_{\mathrm{b},k}\left(vw\right)\right]_{(i,ij)\in\mathcal{G},n\in\Upsilon}. \tag{8.3}$$

The evolution of resources consumed by background services over time slots may be predicted from past observations [Tan et al., 2011]. The smaller variations within each time slot is taken into account in the probability distribution.

8.3 Slice Resource Provisioning Approaches

Resource provisioning for slice s, which is represented by the mapping between $\mathcal{G}$ and $\mathcal{G}_s$, may evolve between successive time intervals due to the evolution of the characteristics of the MI-SLA for slice s, to the arrival of new slice provisioning requests, and to resources released by terminated slices.

8.3.1 Prioritized Processing Provisioning Requests

Several provisioning strategies may be considered to account for the dynamicity of requests. A first approach consists in processing provisioning requests as soon as they arrive. The advantage is to be able to immediately indicate whether enough infrastructure resources are available to satisfy the request. A second approach is to wait some time and process several requests simultaneously. This second approach, considered in this chapter, helps in organizing the resource provisioning, since the InP has a better view of concurrent requests. Nevertheless, the slice request processing delay has to be adjusted depending on the priority class of the slice.

When processing a new request, the already provisioned resources for slices in service or to be activated in the future may be adjusted. This update possibility gives more degrees of freedom to the InP to satisfy new requests, but comes at the price of a higher computational complexity. Moreover, updates should be done so as to still satisfy previous requests which have been indicated to the MNOs as granted. In this chapter, we have chosen not to change any assignment of previously successfully processed slice requests.

Independently of the chosen provisioning strategy, the InP has to account for the time required for infrastructure resource provisioning as well as slice deployment and activation (lasting few minutes, as indicated in [Boubendir et al., 2018]). Consequently, provisioning requests for slices to be activated at $(k+1)\,T$ should reach the InP before $(k+1)\,T - \varepsilon T$, where $\varepsilon T < T$ accounts for the provisioning operations as well as the slice activation and update delays.

Let $\mathcal{S}_k$ be the set of slices whose provisioning requests have been received before $(k+1)\,T - \varepsilon T$. A flag $f_s \in \{0,1\}$ is associated to each slice $s \in \mathcal{S}_k$, indicating whether the request has been processed ($f_s = 1$) (granted or denied) or is still to be

processed ($f_s = 0$).

In what follows, the proposed prioritized slice provisioning approach, considers two classes of slices, namely Premium and Standard. Each slice request, when received for the first time in the interval $\mathcal{T}_k = [kT - \varepsilon T, (k+1)T - \varepsilon T[$, gets $f_s = 0$, and is assigned a priority level $P_{s,k} \in \mathbb{R}$ depending on its class

$$P_{s,k} = \begin{cases} P_{\max}, & \text{for Premium slices,} \\ 0 & \text{for Standard slice, if } k_s^{\text{on}} > k+1, \\ P_{\max} - 1 & \text{for Standard slice, if } k_s^{\text{on}} = k+1. \end{cases} \tag{8.4}$$

Standard slices requests, which have to be activated in the next time slot, get thus a higher priority level.

Then, only slices whose priority level is above a certain threshold

$$P_{\text{thres}} = \alpha \left(P_{\max} - 1\right), \text{ with } \alpha \in [0, 1], \tag{8.5}$$

are processed in the time interval $\mathcal{P}_k = [(k+1)T - \varepsilon T, (k+1)T[$ of duration εT. The set of slices to be processed during the time interval $\mathcal{P}_k$ is then

$$\mathcal{R}_k \triangleq \{s \in \mathcal{S}_k : f_s = 0, P_{s,k} \geqslant P_{\text{thres}}\}. \tag{8.6}$$

Once the request of a slice in $\mathcal{R}_k$ is processed, its flag is set to $f_s = 1$. All standard slice requests with $P_{s,k} < P_{\text{thres}}$ (pending requests) are delayed and may be processed in the next time interval $\mathcal{P}_{k+1}$. Their priority is updated as

$$P_{s,k+1} = \begin{cases} \min\{P_{s,k} + \Delta P, P_{\max} - 1\} & \text{if } k_s^{\text{on}} > k+2, \\ P_{\max} - 1 & \text{if } k_s^{\text{on}} = k+2, \end{cases} \tag{8.7}$$

where $\Delta P \geqslant 0$ is some priority increment. When several slices of equal priority have to be processed in a given time slot, those who have to be activated first are processed first, then those who have been submitted first. Premium slices are always processed first. The processing delay of Standard slice requests depends thus on α and ΔP. Deferring more the processing of Standard slice requests gives more chance to satisfy Premium slice requests.

When $\alpha = 0$, whatever the value of ΔP, all slices received in the time interval $\mathcal{T}_k$ are processed, starting from the Premium slices, with the risk of having no resources available for Premium slice requests received in the few next time slots;

When $\alpha = 1$ and $\Delta P = 0$, the processing of Standard slice requests is delayed until the time slot preceding their activation, leaving a maximum amount of resources available for Premium slice requests.

Figure 8.1 illustrates a scenario taking place during the processing time interval $\mathcal{P}_k$ when the processing of Standard slice requests is maximally delayed ($\alpha = 1$ and $\Delta P = 0$). The three slice requests s_1, s_2, and s_3 in $\mathcal{S}_k$ are assumed still to

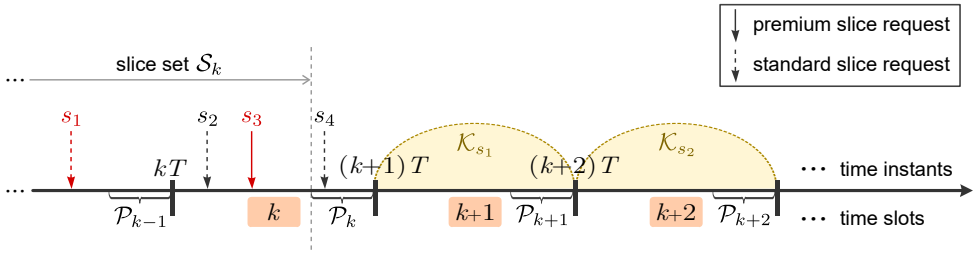


Figure 8.1: Time slots, arrival times of the slice provisioning requests, and time intervals during which the slice provisioning decisions are made.

be processed. The slice request s_4 arrives within $\mathcal{P}_k$ and will thus be considered in $\mathcal{P}_{k+1}$. Among the slices s_1, s_2, and s_3, only s_3 is Premium (request time instant indicated by a solid arrow), and is therefore processed in $\mathcal{P}_k$. The slice requests s_1 and s_2 are Standard (request time instants indicated by dashed arrows). They have to be active in the time slots $\mathcal{K}_{s_1} = \left[k^{\text{on}}_{s_1}, k^{\text{off}}_{s_1}\right]$ and $\mathcal{K}_{s_2} = \left[k^{\text{on}}_{s_2}, k^{\text{off}}_{s_2}\right]$. Since $k^{\text{on}}_{s_1} = k+1$ and $k^{\text{on}}_{s_2} = k+2$, only s_1 is processed in $\mathcal{P}_k$. Finally, the set of slice requests to be processed in $\mathcal{P}_k$ is $\mathcal{R}_k = \{s_1, s_3\}$ (highlighted by red arrows).

8.3.2 Decision Variables

Provisioning resources for some slice $s \in \mathcal{R}_k$ amounts to defining a mapping $\boldsymbol{\kappa}_{s,\ell}$ between the graphs $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and $\mathcal{G}_s = (\mathcal{N}_s, \mathcal{E}_s)$ for each time slot $\ell \in \mathcal{K}_s \triangleq \left[k^{\text{on}}_s, k^{\text{off}}_s\right]$ during which slice s is active. This mapping describes

(1) the number $\kappa_{s,\ell}(i,v) \in \mathbb{N}$ of VNF instances of type $v \in \mathcal{N}_s$ for which node $i \in \mathcal{N}$ will provision resources;

(2) the number $\kappa_{s,\ell}(ij,vw) \in \mathbb{N}$ of links $vw \in \mathcal{E}_s$ between VNF instances for which the InP will provision resources on the infrastructure link $ij \in \mathcal{E}$, both in time slot ℓ.

The amount of resource of type $n \in \Upsilon$ provisioned by node i for a VNF instance of type v is $\kappa_{s,\ell}(i,v)\, r_{s,n}(v)$ and $\kappa_{s,\ell}(ij,vw)\, r_{s,\text{b}}(vw)$ represents the bandwidth provisioned on link ij to support the traffic between two virtual nodes of type v and w.

The mapping $\boldsymbol{\kappa}_{s,\ell}$ is thus defined as

$$\boldsymbol{\kappa}_{s,\ell} = \left\{\kappa_{s,\ell}(i,v), \kappa_{s,\ell}(ij,vw)\right\}_{(i,ij)\in\mathcal{G},(v,vw)\in\mathcal{G}_s}$$

for each $\ell \in \mathcal{K}_s$. By convention, $\boldsymbol{\kappa}_{s,\ell} = \mathbf{0}$ when $\ell \notin \mathcal{K}_s$. Moreover, one introduces

$$\boldsymbol{\kappa}_s = \{\boldsymbol{\kappa}_{s,\ell} : \ell \in \mathcal{K}_s\}$$

to indicate the assignment that has to be performed for a given slice.

In some situations, not enough infrastructure resources may be available for a given slice $s \in \mathcal{R}_k$. The binary decision variable d_s indicates whether all conditions are met to satisfy the provisioning request for slice s and consequently whether resources are actually provisioned for slice s ($d_s = 1$) or not ($d_s = 0$). These conditions are detailed in the following sections.

Consequently, the set of variables which have to be assigned by the InP in the processing time interval $\mathcal{P}_k$ are

$$\mathbf{d}_{\mathcal{R}_k} = \{d_s : s \in \mathcal{R}_k\}, \text{ and}$$
$$\boldsymbol{\kappa}_{\mathcal{R}_k} = \{\boldsymbol{\kappa}_{s,\ell} : s \in \mathcal{R}_k, \ell \in \mathcal{K}_s\}.$$

The vector $\mathbf{d}_{\mathcal{R}_k}$ indicates which slice requests in $\mathcal{R}_k$ have been granted, and $\boldsymbol{\kappa}_k$ describes the associated provisioning schemes proposed by the InP.

8.3.3 Provisioning Constraints

During the processing time interval $\mathcal{P}_k$ of time slot k, the InP has to account for all provisioning requests of slices $s \in \mathcal{S}_{k\text{-}1}$ which have previously been processed, *i.e.*, with $f_s = 1$. The set of these slices is denoted as

$$\mathcal{S}^{\mathrm{p}}_{k\text{-}1} = \{s \in \mathcal{S}_{k\text{-}1} : f_s = 1\}.$$

Moreover, the mappings $\boldsymbol{\kappa}_{s,\ell}$ for all slices $s \in \mathcal{R}_k$ have to satisfy some constraints to ensure that (*i*) enough resources are provisioned to properly deploy the SFCs; and (*ii*) the probability of satisfying provisioning $\underline{p}^{\mathrm{sp}}_s$ is reached. These constraints have to be satisfied for all time slots during which the slice is active. The InP has also to keep the impact probability on background services below $\overline{p}^{\mathrm{im}}$. These constraints are described in what follows.

The total amount of resources provisioned by each infrastructure node $i \in \mathcal{N}$ and each infrastructure link $ij \in \mathcal{E}$ for all slices $s \in \mathcal{R}_k$ has to be less than their available resources, see Section 8.2.1. Consequently, the following constraints have to be satisfied, for each $\ell = \min_{s\in\mathcal{R}_k}\{k^{\mathrm{on}}_s\} \geqslant k, \ldots, \max_{s\in\mathcal{R}_k}\{k^{\mathrm{off}}_s\}$,

$$\sum_{s\in\mathcal{R}_k}\sum_{v\in\mathcal{N}_s} \kappa_{s,\ell}(i,v)\, r_{s,n}(v) \leqslant a_n(i) - \sum_{s\in\mathcal{S}^{\mathrm{p}}_{k\text{-}1}}\sum_{v\in\mathcal{N}_s} \kappa_{s,\ell}(i,v)\, r_{s,n}(v), \forall i, n, \tag{8.8}$$

$$\sum_{s\in\mathcal{R}_k}\sum_{vw\in\mathcal{E}_s} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij) - \sum_{s\in\mathcal{S}^{\mathrm{p}}_{k\text{-}1}}\sum_{vw\in\mathcal{E}_s} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw), \forall ij. \tag{8.9}$$

In (8.8) and (8.9), the right-hand sides of the inequalities represent the remaining part of the resources once previous provisioning requests have been processed. When updates of the provisioning scheme of granted slice requests are allowed, $\boldsymbol{\kappa}_{s,\ell}$, $s \in \mathcal{S}^{\mathrm{p}}_{k\text{-}1}$ are considered as variables, but not d_s, $s \in \mathcal{S}^{\mathrm{p}}_{k\text{-}1}$, since the status of successfully processed slice requests should not be changed. In what follows, one considers that such updates are not allowed.

The inequalities (8.8) and (8.9) may be more compactly written for $\ell > k$ as follows

$$\sum_{s\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{p}}_{k\text{-}1}}\sum_{v\in\mathcal{N}_s}\kappa_{s,\ell}\left(i,v\right)r_{s,n}\left(v\right)\leqslant a_n\left(i\right), \tag{8.10}$$

$$\sum_{s\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{p}}_{k\text{-}1}}\sum_{vw\in\mathcal{E}_s}\kappa_{s,\ell}\left(ij,vw\right)r_{s,\mathrm{b}}\left(vw\right)\leqslant a_{\mathrm{b}}\left(ij\right). \tag{8.11}$$

The conditions (8.8) and (8.10) are equivalent, as $\boldsymbol{\kappa}_{s,\ell}=\mathbf{0}$ when $\ell\notin\mathcal{K}_s$. The same is also true for conditions (8.9) and (8.11).

Finally, the flow conservation constraint similar to that introduced in Chapter 7 has also to be satisfied, for each $\ell\in\mathcal{K}_s$, $s\in\mathcal{R}_k$, $i\in\mathcal{N}$, and $vw\in\mathcal{E}_s$,

$$\begin{aligned}&\sum_{j\in\mathcal{N}}\left[\kappa_{s,\ell}\left(ij,vw\right)-\kappa_{s,\ell}\left(ji,vw\right)\right]=\\&\left(\frac{r_{s,\mathrm{b}}(vw)}{\sum\limits_{vu\in\mathcal{E}_s}r_{s,\mathrm{b}}(vu)}\right)\kappa_{s,\ell}\left(i,v\right)-\left(\frac{r_{s,\mathrm{b}}(vw)}{\sum\limits_{uw\in\mathcal{E}_s}r_{s,\mathrm{b}}(uw)}\right)\kappa_{s,\ell}\left(i,w\right).\end{aligned} \tag{8.12}$$

8.3.4 Demand Satisfaction and Impact Probabilities

For each time slot $\ell\in\mathcal{K}_s$ during the lifetime of slice s, one considers the constraints on SSP and ImP similar to those presented in Section 7.4 as follows

$$p_{s,\ell}\left(\boldsymbol{\kappa}_{s,\ell},d_s\right)\geqslant\underline{p}^{\mathrm{sp}}_s,\ell\in\mathcal{K}_s,s\in\mathcal{R}_k, \tag{8.13}$$

$$p^{\mathrm{im}}_{n,\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k},i\right)\leqslant\overline{p}^{\mathrm{im}},\forall n\in\Upsilon,\forall i\in\mathcal{N}, \tag{8.14}$$

$$p^{\mathrm{im}}_{\mathrm{b},\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k},ij\right)\leqslant\overline{p}^{\mathrm{im}},\forall ij\in\mathcal{E}, \tag{8.15}$$

where

$$\begin{aligned}p_{s,\ell}\left(\boldsymbol{\kappa}_{s,\ell},d_s\right)=\Pr\Big\{&\sum_{i\in\mathcal{N}}\kappa_{s,\ell}\left(i,v\right)r_{s,n}\left(v\right)\geqslant d_sR_{s,n,\ell}\left(v\right),\forall v,n,\\&\sum_{ij\in\mathcal{E}}\kappa_{s,\ell}\left(ij,vw\right)r_{s,\mathrm{b}}\left(vw\right)\geqslant d_sR_{s,\mathrm{b},\ell}\left(vw\right),\forall vw\Big\},\end{aligned} \tag{8.16}$$

and

$$p^{\mathrm{im}}_{n,\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k},i\right)=\Pr\Big\{\sum_{s\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{p}}_{k\text{-}1}}\sum_{v\in\mathcal{N}_s}\kappa_{s,\ell}\left(i,v\right)r_{s,n}\left(v\right)\geqslant a_n\left(i\right)-B_{n,\ell}\left(i\right)\Big\}, \tag{8.17}$$

$$p^{\mathrm{im}}_{\mathrm{b},\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k},ij\right)=\Pr\Big\{\sum_{s\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{p}}_{k\text{-}1}}\sum_{vw\in\mathcal{E}_s}\kappa_{s,\ell}\left(ij,vw\right)r_{s,\mathrm{b}}\left(vw\right)\geqslant a_{\mathrm{b}}\left(ij\right)-B_{\mathrm{b},\ell}\left(ij\right)\Big\}. \tag{8.18}$$

The evaluations of (8.17) and (8.18) for all $\ell>k$ involve the assignments $\boldsymbol{\kappa}_{\mathcal{S}^{\mathrm{p}}_{k\text{-}1}}$ which have already been evaluated in previous processing time intervals and are considered as constants in the current processing time interval, *i.e.*, $\mathcal{P}_k$. The dependency in $\boldsymbol{\kappa}_{\mathcal{S}^{\mathrm{p}}_{k\text{-}1}}$ of $p^{\mathrm{im}}_{n,\ell}$ and $p^{\mathrm{im}}_{\mathrm{b},\ell}$ is omitted to lighten notations.

8.3.5 Costs and Incomes

Consider the processing time interval $\mathcal{P}_k$ during which a provisioning scheme for all slices $s \in \mathcal{R}_k$ has to be evaluated. This amounts at evaluating $\mathbf{d}_{\mathcal{R}_k}$ and the assignments $\boldsymbol{\kappa}_{\mathcal{R}_k}$.

The costs charged by the InP to an MNO for a provisioning scheme for slice $s \in \mathcal{R}_k$ described by $\boldsymbol{\kappa}_{s,\ell}$ in time slot ℓ are spread between node and bandwidth *resource* provisioning costs

$$\begin{aligned} C_{\mathrm{r}}\left(\boldsymbol{\kappa}_{s,\ell}\right) = & \sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{N}_s} \sum_{n \in \Upsilon} \kappa_{s,\ell}\left(i,v\right) r_n\left(v\right) c_n\left(i\right) \\ & + \sum_{ij \in \mathcal{E}} \sum_{vw \in \mathcal{E}_s} \kappa_{s,\ell}\left(ij,vw\right) r_{\mathrm{b}}\left(vw\right) c_{\mathrm{b}}\left(ij\right) \end{aligned} \tag{8.19}$$

as well as *fixed* node disposal costs

$$C_{\mathrm{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right) = \sum_{i \in \mathcal{N}} \widetilde{\kappa}_{s,\ell}\left(i\right) c_{\mathrm{f}}\left(i\right) \tag{8.20}$$

for the infrastructure nodes used, where

$$\widetilde{\kappa}_{s,\ell}\left(i\right) = \begin{cases} 1 & \text{if } \sum_{v \in \mathcal{N}_s} \kappa_{s,\ell}\left(i,v\right) > 0, \\ 0 & \text{otherwise} \end{cases} \tag{8.21}$$

indicates whether node i is used by slice s in time slot ℓ.

Additionally, when the amount of provisioned resources for slice s increases during two consecutive time slots, provisioning adaptation costs are also charged by the InP to the MNO

$$C_{\mathrm{a}}\left(\boldsymbol{\kappa}_{s,\ell}, \boldsymbol{\kappa}_{s,\ell-1}\right) = \sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{N}_s} \max\left\{\kappa_{s,\ell}\left(i,v\right) - \kappa_{s,\ell\text{-}1}\left(i,v\right), 0\right\} c_{\mathrm{a}}\left(i\right), \tag{8.22}$$

see Section 8.2.2.3.

Once a provisioning request for a slice $s \in \mathcal{R}_k$ has been granted by the InP, the MNO will be able to deploy the slice (see the *commissioning* and *operation* blocks of Figure 2.7) and receives from the SP some income I_s depending on the complexity and of the load of the slice.

8.3.6 Optimization Problem

For a given assignment $\boldsymbol{\kappa}_{\mathcal{R}_k}$, the earnings of the InP are the costs charged to the MNOs

$$\begin{aligned} E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}\right) = & \sum_{s \in \mathcal{R}_k} \sum_{\ell \in \mathcal{K}_s} \left(C_{\mathrm{r}}\left(\boldsymbol{\kappa}_{s,\ell}\right) + C_{\mathrm{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right)\right. . \\ & \left. + C_{\mathrm{a}}\left(\boldsymbol{\kappa}_{s,\ell}, \boldsymbol{\kappa}_{s,\ell-1}\right)\right. . \end{aligned} \tag{8.23}$$

The InP may thus be interested in an assignment $\boldsymbol{\kappa}_{\mathcal{R}_k}$ that maximizes $E_k^{\text{InP}}(\boldsymbol{\kappa}_{\mathcal{R}_k})$. Nevertheless, such assignment will not lead to the maximum of the earnings of the MNOs expressed as

$$E_k^{\text{MNO}}(\boldsymbol{\kappa}_{\mathcal{R}_k}) = \sum_{s \in \mathcal{R}_k} d_s I_s - E_k^{\text{InP}}(\boldsymbol{\kappa}_{\mathcal{R}_k}). \tag{8.24}$$

MNOs may not be interested by InPs applying an optimization strategy maximizing $E_k^{\text{InP}}(\boldsymbol{\kappa}_{\mathcal{R}_k})$.

Alternatively, the InP may try to find an assignment which maximizes $E_k^{\text{MNO}}(\boldsymbol{\kappa}_{\mathcal{R}_k})$. This approach reduces the per-slice income for the InP, but allows more slice provisioning requests to be granted. Nevertheless, InPs are usually unaware of the income I_s obtained by the MNOs from the SP, therefore $E_k^{\text{MNO}}(\boldsymbol{\kappa}_{\mathcal{R}_k})$ cannot be evaluated by the InP. Consequently, we will consider a scenario where the InP tries to find the provisioning scheme which maximizes the amount of slices for which the provisioning is successful, while minimizing the provisioning costs charged to the MNOs. This leads to a max-min optimization problem, which provides the maximum earnings to MNOs whose slice requests have been granted, while potentially saving infrastructure resources to satisfy future provisioning requests.

Consequently, the problem of provisioning resources *jointly* for all slices $s \in \mathcal{R}_k$ during the processing time interval $\mathcal{P}_k$ can be formulated as in Problem 8.1.

Problem 8.1: Max-Min Joint Slice Resource Provisioning

$$\max_{\mathbf{d}_{\mathcal{R}_k}} \min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\text{InP}}(\boldsymbol{\kappa}_{\mathcal{R}_k})$$
subject to, $\forall \ell > k$,
$$\sum_{s \in \mathcal{R}_k \cup \mathcal{S}_{k\text{-}1}^{\text{p}}} \sum_{v \in \mathcal{N}_s} \kappa_{s,\ell}(i,v)\, r_{s,n}(v) \leqslant a_n(i), \forall n \in \Upsilon, \forall i \in \mathcal{N},$$
$$\sum_{s \in \mathcal{R}_k \cup \mathcal{S}_{k\text{-}1}^{\text{p}}} \sum_{vw \in \mathcal{E}_s} \kappa_{s,\ell}(ij,vw)\, r_{s,\text{b}}(vw) \leqslant a_{\text{b}}(ij), \forall ij \in \mathcal{E},$$
(8.12), $\forall s \in \mathcal{R}_k, i \in \mathcal{N}, vw \in \mathcal{E}_s$,
$p_{s,\ell}(\boldsymbol{\kappa}_{s,\ell}, d_s) \geqslant \underline{p}_s^{\text{sp}}, \forall s \in \mathcal{R}_k$,
$p_{n,\ell}^{\text{im}}(\boldsymbol{\kappa}_{\mathcal{R}_k}, i) \leqslant \overline{p}^{\text{im}}, \forall i \in \mathcal{N}, \forall n \in \Upsilon$,
$p_{\text{b},\ell}^{\text{im}}(\boldsymbol{\kappa}_{\mathcal{R}_k}, ij) \leqslant \overline{p}^{\text{im}}, \forall ij \in \mathcal{E}$.

Solving Problem 8.1 is a complex max-min optimization problem involving probabilistic constraints. Section 8.4 presents some heuristics allowing one to obtain suboptimal solutions to Problem 8.1.

8.4 Slice Resource Provisioning Algorithms

In this section, the probabilistic constraints (8.13), (8.14), and (8.15) are relaxed and the objective function of Problem 8.1 is linearized. Finally, two heuristics are introduced to provide approximate solutions to Problem 8.1, performing either se-

quential or joint slice resource provisioning.

8.4.1 Relaxation of Probabilistic Constraints

Using the same approach in Section 7.4, the probabilistic SSP can be replaced by the following linear deterministic constraint, for all $s \in \mathcal{R}_k$ and $\ell \in \mathcal{K}_s$,

$$\sum_{i \in \mathcal{N}} \kappa_{s,\ell}(i,v)\, r_{s,n}(v) \geqslant d_s \widehat{R}_{s,n,\ell}(v, \gamma_{s,\ell}), \forall v, n, \tag{8.25}$$

$$\sum_{ij \in \mathcal{E}} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant d_s \widehat{R}_{s,\mathrm{b},\ell}(vw, \gamma_{s,\ell}), \forall vw, \tag{8.26}$$

where

$$\widehat{R}_{s,n,\ell}(v, \gamma_{s,\ell}) = \overline{R}_{s,n,\ell}(v) + \gamma_{s,\ell} \widetilde{R}_{s,n,\ell}(v), \tag{8.27}$$

$$\widehat{R}_{s,\mathrm{b},\ell}(vw, \gamma_{s,\ell}) = \overline{R}_{s,\mathrm{b},\ell}(vw) + \gamma_{s,\ell} \widetilde{R}_{s,\mathrm{b},\ell}(vw), \tag{8.28}$$

are the target aggregate user demands, depending on some parameter $\gamma_{s,\ell} > 0$. $\overline{R}_{s,n,\ell}(v)$ and $\widetilde{R}_{s,n,\ell}(v)$ are the mean and standard deviation of $R_{s,n,\ell}(v)$, while $\overline{R}_{s,\mathrm{b},\ell}(vw)$ and $\widetilde{R}_{s,\mathrm{b},\ell}(vw)$ are the mean and standard deviation of $R_{s,\mathrm{b},\ell}(vw)$.

Similarly, the ImP constraints (8.14, 8.15) can be replaced, for all $(i,ij) \in \mathcal{G}$ and $n \in \Upsilon$, by

$$\sum_{s \in \mathcal{R}_k \cup \mathcal{S}^{\mathrm{p}}_{k-1}} \sum_{v \in \mathcal{N}_s} \kappa_{s,\ell}(i,v)\, r_{s,n}(v) \leqslant a_n(i) - \widehat{B}_{n,\ell}(i, \gamma_{\mathrm{B},\ell}), \tag{8.29}$$

$$\sum_{s \in \mathcal{R}_k \cup \mathcal{S}^{\mathrm{p}}_{k-1}} \sum_{vw \in \mathcal{E}_s} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij) - \widehat{B}_{\mathrm{b},\ell}(ij, \gamma_{\mathrm{B},\ell}) \tag{8.30}$$

for $\ell > k$, where

$$\widehat{B}_{n,\ell}(i, \gamma_{\mathrm{B},\ell}) = \overline{B}_{n,\ell}(i) + \gamma_{\mathrm{B},\ell} \widetilde{B}_{n,\ell}(i), \tag{8.31}$$

$$\widehat{B}_{\mathrm{b},\ell}(ij, \gamma_{\mathrm{B},\ell}) = \overline{B}_{\mathrm{b},\ell}(ij) + \gamma_{\mathrm{B},\ell} \widetilde{B}_{\mathrm{b},\ell}(ij) \tag{8.32}$$

are the considered target level of background service demands.

In (8.25, 8.26, 8.29, 8.30), the selection of values for $\gamma_{s,\ell}$ and $\gamma_{\mathrm{B},\ell}$ is similar to that introduced in Chapter 7.

8.4.2 Linearization of the Cost Function

In Problem 8.1, the term $C_{\mathrm{a}}(\boldsymbol{\kappa}_{s,\ell}, \boldsymbol{\kappa}_{s,\ell-1})$ introduced in (8.22) makes the objective function nonlinear. To address this issue, one may introduce the set of variables

$$\mathbf{y}_s = \{y_{s,\ell}(i,v) : \ell \in \mathcal{K}_s, i \in \mathcal{N}, v \in \mathcal{N}_s\}$$

for each $s \in \mathcal{R}_k$ and

$$\mathbf{y}_{\mathcal{R}_k} = \{\mathbf{y}_s : s \in \mathcal{R}_k\}$$

and reformulate the objective function (8.23) as

$$E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \mathbf{y}_{\mathcal{R}_k}\right) = \sum_{s\in\mathcal{R}_k}\sum_{\ell\in\mathcal{K}_s}\Big(C_{\mathrm{r}}\left(\boldsymbol{\kappa}_{s,\ell}\right) + C_{\mathrm{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right) + \sum_{i\in\mathcal{N}}\sum_{v\in\mathcal{N}_s} y_{s,\ell}\left(i,v\right) c_{\mathrm{a}}\left(i\right)\Big), \quad (8.33)$$

with the additional constraints, to be satisfied for all $s \in \mathcal{R}_k$, $\ell \in \mathcal{K}_s$, $i \in \mathcal{N}$, and $v \in \mathcal{N}_s$

$$y_{s,\ell}\left(i,v\right) \geqslant \kappa_{s,\ell}\left(i,v\right) - \kappa_{s,\ell\text{-}1}\left(i,v\right), \quad (8.34)$$

$$y_{s,\ell}\left(i,v\right) \geqslant 0. \quad (8.35)$$

For a given value of $\mathbf{d}_{\mathcal{R}_k}$, the objective function has now to be minimized with respect to $\boldsymbol{\kappa}_{s,\ell}$, $s \in \mathcal{R}_k$, $\ell \in \mathcal{K}_s$, and $\mathbf{y}_{\mathcal{R}_k}$.

Moreover, the evaluation of $C_{\mathrm{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right)$ involves $\widetilde{\kappa}_{s,\ell}\left(i\right)$ defined in (8.21). The variable $\widetilde{\kappa}_{s,\ell}\left(i\right)$ can be related to $\sum_v \kappa_{s,\ell}\left(i,v\right)$ using the following linear inequality constraints

$$\begin{aligned}\sum_v \kappa_{s,\ell}\left(i,v\right) &\geqslant 0,\\ \widetilde{\kappa}_{s,\ell}\left(i\right)\overline{N} &\geqslant \sum_v \kappa_{s,\ell}\left(i,v\right),\\ \widetilde{\kappa}_{s,\ell}\left(i\right) &\in \{0,1\},\end{aligned}$$

where $\overline{N}$ is an upper bound on the number of VNF instances of all types for which resources may be provisioned by a given infrastructure node.

8.4.3 Relaxed Joint Max-Min Optimization Problem

Even with the results of Sections 8.4.1 and 8.4.2, the solution of a relaxed version of Problem 8.1 requires the solution of a constrained max-min optimization problem, which is still quite complex. To address this issue, for a fixed value of $\mathbf{d}_{\mathcal{R}_k}$, we introduce the following optimization problem.

Problem 8.2: Joint Slice Resource Provisioning Given $\mathbf{d}_{\mathcal{R}_k}$

$$\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}, \mathbf{y}_{\mathcal{R}_k}} E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \mathbf{y}_{\mathcal{R}_k}\right)$$

s.t. $\forall s \in \mathcal{R}_k, \ell > k$:

(8.12), $\forall i \in \mathcal{N}, \forall vw \in \mathcal{E}_s$,

$$\sum_{i\in\mathcal{N}} \kappa_{s,\ell}\left(i,v\right) r_{s,n}\left(v\right) \geqslant d_s \widehat{R}_{s,n,\ell}\left(v,\gamma_{s,\ell}\right), \forall v \in \mathcal{N}_s, \forall n \in \Upsilon,$$

$$\sum_{ij\in\mathcal{E}} \kappa_{s,\ell}\left(ij,vw\right) r_{s,\mathrm{b}}\left(vw\right) \geqslant d_s \widehat{R}_{s,\mathrm{b},\ell}\left(vw,\gamma_{s,\ell}\right), \forall vw \in \mathcal{E}_s,$$

$$y_{s,\ell}\left(i,v\right) \geqslant \kappa_{s,\ell}\left(i,v\right) - \kappa_{s,\ell\text{-}1}\left(i,v\right), \forall i \in \mathcal{N}, \forall v \in \mathcal{N}_s,$$

$$y_{s,\ell}\left(i,v\right) \geqslant 0, \forall i \in \mathcal{N}, \forall v \in \mathcal{N}_s,$$

and s.t. $\forall \ell > k, i \in \mathcal{N}, n \in \Upsilon$:

$$\sum_{s\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{P}}_{k\text{-}1}}\sum_{v\in\mathcal{N}_s}\kappa_{s,\ell}\left(i,v\right)r_{s,n}\left(v\right)\leqslant a_n\left(i\right)-\widehat{B}_{n,\ell}\left(i,\gamma_{\mathrm{B},\ell}\right),$$
and s.t. $\forall\ell>k, ij\in\mathcal{E}$:
$$\sum_{s\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{P}}_{k\text{-}1}}\sum_{vw\in\mathcal{E}_s}\kappa_{s,\ell}\left(ij,vw\right)r_{s,\mathrm{b}}\left(vw\right)\leqslant a_{\mathrm{b}}\left(ij\right)-\widehat{B}_{\mathrm{b},\ell}\left(ij,\gamma_{\mathrm{B},\ell}\right).$$

In Problem 8.2 a greedy solution approach to Problem 8.1 is considered, where the slices to be processed in $\mathcal{R}_k=\{s_1,\ldots,s_{R_k}\}$ are assumed to be ordered, with $R_k=|\mathcal{R}_k|$, see Section 8.3.1. When several MNOs have submitted slices in the same time slot, the InP may also prioritize MNOs.

8.4.4 Relaxed Single Slice Max-Min Optimization Problem

The number of variables involved in Problem 8.2 introduced in Section 8.4.3 may become relatively large when provisioning has to be performed for several slices. For this reason, we introduce a reduced-complexity version of Problem 8.2, where the resource provisioning is performed slice by slice.

One focuses on a slice $s\in\mathcal{R}_k$ which provisioning request has to be processed. Some provisioning requests for slices $s'\in\mathcal{R}_k$, $s'\neq s$ may have been previously processed, in which case, when the request is granted, $d_{s'}=1$ and $\boldsymbol{\kappa}_{s'}\neq\mathbf{0}$ and when it is not granted, $d_{s'}=0$ and $\boldsymbol{\kappa}_{s'}=\mathbf{0}$. For not yet processed requests of slices $s'\in\mathcal{R}_k$, $s'\neq s$, one considers that $d_{s'}=0$ and $\boldsymbol{\kappa}_{s'}=\mathbf{0}$. With these assumptions, provisioning resources for slice $s\in\mathcal{R}_k$ requires the solution of Problem 8.3.

Problem 8.3: Single Slice Resource Provisioning

$$\min_{\boldsymbol{\kappa}_s,\mathbf{y}_s}E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_s,\mathbf{y}_s\right)$$
s.t. $\forall\ell>k$:
(8.12), $\forall i\in\mathcal{N},\forall vw\in\mathcal{E}_s$,
$$\sum_{i\in\mathcal{N}}\kappa_{s,\ell}\left(i,v\right)r_{s,n}\left(v\right)\geqslant d_s\widehat{R}_{s,n,\ell}\left(v,\gamma_{s,\ell}\right),\forall v\in\mathcal{N}_s,\forall n\in\Upsilon,$$
$$\sum_{ij\in\mathcal{E}}\kappa_{s,\ell}\left(ij,vw\right)r_{s,\mathrm{b}}\left(vw\right)\geqslant d_s\widehat{R}_{s,\mathrm{b},\ell}\left(vw,\gamma_{s,\ell}\right),\forall vw\in\mathcal{E}_s,$$
$$y_{s,\ell}\left(i,v\right)\geqslant\kappa_{s,\ell}\left(i,v\right)-\kappa_{s,\ell\text{-}1}\left(i,v\right),\forall i\in\mathcal{N},\forall v\in\mathcal{N}_s,$$
$$y_{s,\ell}\left(i,v\right)\geqslant 0,\forall i\in\mathcal{N},\forall v\in\mathcal{N}_s,$$
and s.t. $\forall\ell>k, i\in\mathcal{N}, n\in\Upsilon$:
$$\sum_{s'\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{P}}_{k\text{-}1}}\sum_{v\in\mathcal{N}_s}\kappa_{s',\ell}\left(i,v\right)r_{s',n}\left(v\right)\leqslant a_n\left(i\right)-\widehat{B}_{n,\ell}\left(i,\gamma_{\mathrm{B},\ell}\right),$$
and s.t. $\forall\ell>k, ij\in\mathcal{E}$:
$$\sum_{s'\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{P}}_{k\text{-}1}}\sum_{vw\in\mathcal{E}_s}\kappa_{s',\ell}\left(ij,vw\right)r_{s',\mathrm{b}}\left(vw\right)\leqslant a_{\mathrm{b}}\left(ij\right)-\widehat{B}_{\mathrm{b},\ell}\left(ij,\gamma_{\mathrm{B},\ell}\right).$$

Here d_s is initially set to 1 in the second and third constraint of Problem 8.3. Once Problem 8.3 is solved, if a feasible solution cannot be found, d_s will be reset

to 0, *i.e.*, the resource provisioning request of slice s is refused.

Assuming again that the slice provisioning requests are ordered, see Section 8.3.1, one may get a second greedy provisioning algorithm where slice provisioning requests are processed slice by slice solving Problem 8.3 for each slice. The highest priority slice is processed first. The lower priority slices are then processed, whatever the provisioning result of a higher priority slice. Even if high-priority slices may have their provisioning request rejected, lower-priority slice requests may be granted for slices with smaller resource requirements.

8.4.5 Slice Resource Provisioning Approaches

For the suboptimal algorithms introduced in Sections 8.4.3 and 8.4.4, two Prioritized slice resource Provisioning (PP) variants are then considered, depending on whether slices provisioning requests are processed jointly (J-PP) or sequentially (S-PP).

8.4.5.1 Joint Approach

In the J-PP approach, all slices in $\mathcal{R}_k$ are processed jointly. This is done by solving Problem 8.2, considering $\mathbf{d}_{\mathcal{R}_k} = (1, \ldots, 1)$. If the provisioning is successful, the algorithm stops. If no solution is returned, the provisioning request of the slice with lowest priority is not granted, *i.e.*, $d_{R_k} = 0$. Problem 8.2 is solved again considering $\mathbf{d}_{\mathcal{R}_k} = (1, \ldots, 1, 0)$. If there is still no solution, the provisioning request for the slice with second lowest priority is not granted, and so forth. If more than two slice requests have the same lowest priority, the last arrived one is not granted.

The first part of Algorithm 8.1 (Lines 4–16) summarizes the J-PP approach, which tries to jointly provision resources for a decreasing number of slices within each processing time interval.

8.4.5.2 Sequential Approach

In the S-PP approach, slices in $\mathcal{R}_k$ are sequentially provisioned. This is done by solving Problem 8.3. The second part of Algorithm 8.1 (Lines 17–19) summarizes the S-PP approach. Note that, S-PP when $\alpha = 0$ is implements a first-arrived first-served processing policy.

8.4.5.3 Complexity Analysis

Each variant in Algorithm 8.1 requires the solution of one or several ILPs, whose complexity increases exponentially with the number of variables in the worst case. The number of ILPs, variables, and of constraints involved in each variant are summarized in Table 8.2.

The J-PP variant considers a single ILP, while the S-PP variant splits the task into $|\mathcal{R}_k|$ subproblems, each of which implies $|\mathcal{R}_k|$ times less variables than the joint variant. Consequently, due to the exponential complexity of the NP-hard ILP,

the sequential approach may provide a solution faster than the joint variant. In Section 8.5, the two proposed variants are compared via numerical simulations.

Algorithme 8.1 : Prioritized Slice Resource Provisioning

Input : $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, `n_target` (target nb. of processing requests)
Output : $\widehat{\mathbf{d}}_{\mathcal{R}_k}$,$\widehat{\boldsymbol{\kappa}}_{\mathcal{R}_k}$

```
# Initialization
k = 1;
n_proc = 0;                                   ▷ number of processed slice requests
while n_proc < n_target do
    foreach processing time interval P_k do
        Determine R_k = {s ∈ S_k : f_s = 0, P_{s,k} ⩾ P_thres};
        Assign priority P_{s,k} to each s ∈ R_k w.r.t (8.4);
        Determine the prioritized slice set P_k;
        switch provisioning_variant do
            case J-PP (joint prioritized provisioning) do
                Initialize d_{R_k} = (1, ..., 1);
                i = |R_k|;
                while i > 0 do
                    Solve Problem 8.2;
                    if κ̂_{R_k} = ∅ then
                        d_i ← 0;
                        i = i − 1;
                    else
                        break;
            case S-PP (sequential prioritized provisioning) do
                foreach s ∈ R_k do
                    Solve Problem 8.3 for slice s to get d̂_s and κ̂_s;
        # Update slice priority
        foreach s ∈ S_k with f_s = 0 do
            P_{s,k+1} = min{P_{s,k} + ΔP, P_max − 1};
        # Update flag and number of processed slice requests
        Set f_s = 1, ∀s ∈ R_k;
        n_proc = n_proc + |R_k|;
```

8.5 Evaluation

This section presents simulations to evaluate the performance of the two provisioning algorithms, J-PP and S-PP, described in Section 8.4. The simulation setup is described in Section 8.5.1. All simulation results described in Section 8.5.2 are performed with the CPLEX MILP solver interfaced with MATLAB.

Table 8.2: Number of problems, variables, and constraints involved in each variant.

Variant	*#pbs*	*#variables/problem*	*#constraints/problem*
J-PP	1	$\sum_{s,\ell}(\lvert\mathcal{N}\rvert + 2\lvert\mathcal{N}\rvert\lvert\mathcal{N}_s\rvert) + \sum_{s,\ell}\lvert\mathcal{E}\rvert\lvert\mathcal{E}_s\rvert$	$\sum_{s,\ell}(\lvert\mathcal{N}\rvert\lvert\mathcal{E}_s\rvert + \lvert\mathcal{N}_s\rvert\lvert\Upsilon\rvert + \lvert\mathcal{E}_s\rvert) + \sum_{s,\ell}2\lvert\mathcal{N}\rvert\lvert\mathcal{N}_s\rvert + \sum_{\ell}(\lvert\mathcal{N}\rvert + \lvert\mathcal{E}\rvert)$
S-PP	$\lvert\mathcal{R}_k\rvert$	$\sum_{\ell}(\lvert\mathcal{N}\rvert + 2\lvert\mathcal{N}\rvert\lvert\mathcal{N}_s\rvert) + \sum_{\ell}\lvert\mathcal{E}\rvert\lvert\mathcal{E}_s\rvert$	$\sum_{\ell}(\lvert\mathcal{N}\rvert\lvert\mathcal{E}_s\rvert + \lvert\mathcal{N}_s\rvert\lvert\Upsilon\rvert + \lvert\mathcal{E}_s\rvert) + \sum_{\ell}(2\lvert\mathcal{N}\rvert\lvert\mathcal{N}_s\rvert + \lvert\mathcal{N}\rvert + \lvert\mathcal{E}\rvert)$

8.5.1 Simulation Conditions

8.5.1.1 Infrastructure Topology

The fat-tree topology introduced in Chapter 5 is used again here, with $K = 2$. The cost of using each resource of the infrastructure network ($c_n(i)$ and $c_\mathrm{b}(ij)$, $\forall n \in \Upsilon$, $\forall (i,ij) \in \mathcal{G}$) is set to 1. The fixed cost $c_\mathrm{f}(i)$ and the adaptation cost $c_\mathrm{a}(i)$ are respectively set to 10 and 20, $\forall i \in \mathcal{N}$

8.5.1.2 Slice Resource Demand (S-RD)

The number of users of a slice s is assumed to follow a binomial distribution of parameter $p_{s,k}$. One considers two patterns to represent the evolution with time of $p_{s,k}$, which impact the evolution of the slice resource demands. The first, illustrated in Figure 8.2a corresponds to a constant demand $p_{s,k} = 1$ during the whole lifetime of the slice. The second, shown in Figure 8.2b, describes a slice whose demand evolves from one time slot to the next.

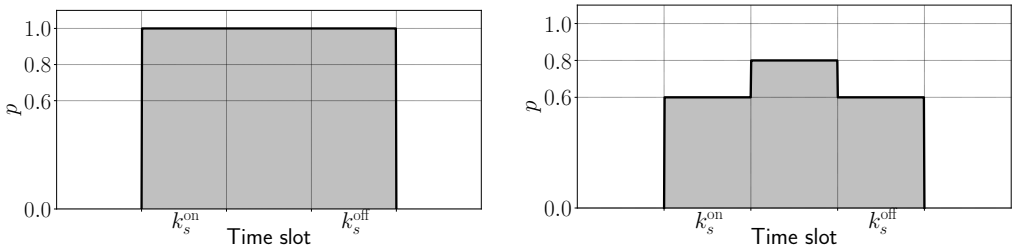


Figure 8.2: Probability pattern of service usage: (a) constant over a time interval and (b) piece-wise constant.

We reuse the three types of slices introduced in Section 7.5.1.3

- Slices of type 1 aim to provide an HD video streaming service at an average rate of 6 Mbps for VIP users, *e.g.*, in a stadium. The number of users of a slice s of type 1 follows a binomial distribution $\mathcal{B}(500, p_{s,k})$. The required SSP for type 1-slices is $\underline{p}_s^{\mathrm{sp}} = 0.99$;

- Slices of type 2 are dedicated to provide an SD video streaming service at an average rate of 4 Mbps. The number of users of a slice s of type 2 follows a binomial distribution $\mathcal{B}(2000, p_{s,k})$. The required SSP for type 2-slices is $\underline{p}_s^{\text{sp}} = 0.95$;
- Slices of type 3 aim to provide a video surveillance and traffic monitoring service at an average rate of 2 Mbps for 200 cameras. The required SSP for type 3-slices is $\underline{p}_s^{\text{sp}} = 0.9$.

The functional architecture of each type is given in Chapter 6, Section 6.5.1.2. The values of U-RD and SFC-RD are given in Table 7.4.

The slice type is chosen uniformly at random. For slices of type 1 and 2, the demand pattern is also chosen uniformly at random.

A normalized unit duration time slot is considered, *i.e.*, $T = 1$. The processing duration has value of $\varepsilon T = 0.1T$. The number of provisioning request arrivals in each time slot obeys a Poisson distribution Pois (μ) of parameter $\mu = 2$. The arrival time of each slice request is uniformly distributed within each time interval $\mathcal{T}_k$. The activation delay (*i.e.*, $k_s^{\text{on}} - k_s$) follows the uniform distribution $\mathcal{U}(1, 6)$ and the lifetime follows the uniform distribution $\mathcal{U}(1, 3)$.

8.5.1.3 Background Services

At each infrastructure node $i \in \mathcal{N}$ and link $ij \in \mathcal{E}$ and for all time slots k, we assume that the resources consumed by best-effort background services follow a normal distribution with mean and standard deviation equal to respectively 20% and 5% of the available resources at a node and at a link, *i.e.*,

$$\left\{\overline{B}_{n,k}(i), \widetilde{B}_{n,k}(i)\right\} = \{0.2a_n(i), 0.05a_n(i)\} \,\forall i \in \mathcal{N}, \forall n \in \Upsilon,$$
$$\left\{\overline{B}_{\text{b},k}(ij), \widetilde{B}_{\text{b},k}(ij)\right\}, = \{0.2a_{\text{b}}(ij), 0.05a_{\text{b}}(ij)\}, \forall ij \in \mathcal{E}.$$

The provisioning impact probability threshold $\overline{p}^{\text{im}}$ is set to 0.1.

8.5.2 Results

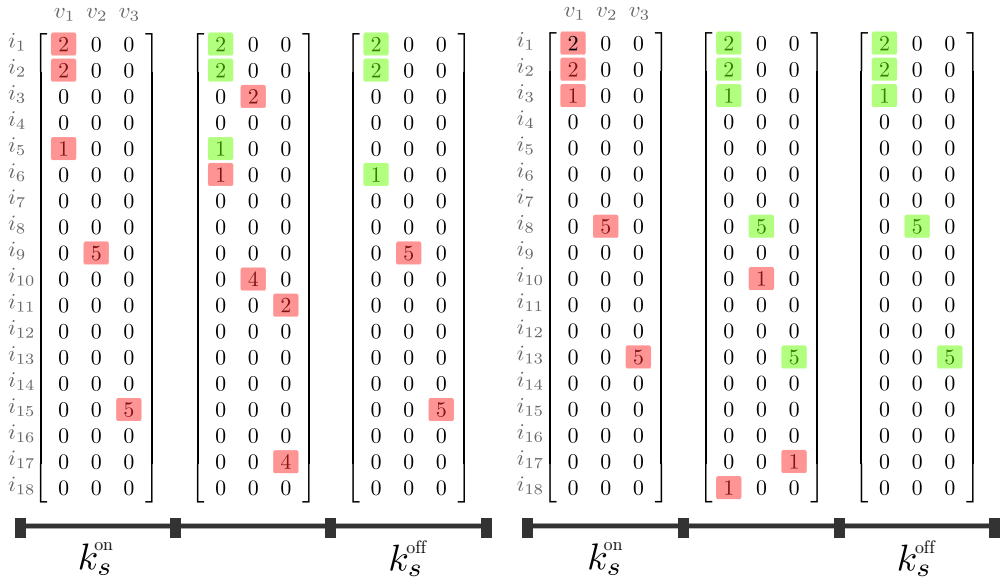
The performance of the provisioning variants (J-PP and S-PP) is evaluated considering the following metrics: slice request acceptance rate, per-slice provisioning cost, average response delay (*i.e.*, time between the time instant the request arrives and the time instant at which it is processed), average number of adjusted VNF instances per slice, and average computing time for each slice request.

8.5.2.1 Resource Provisioning for a Single Slice

A first simulation aims at illustrating the impact of the adaptation costs described in Section 8.2.2.3, on the adjustments of the provisioned resources between consecutive time slots. A single slice of type 1 with the demand pattern of Figure 8.2b is

considered. Consequently, the J-PP and S-PP provisioning variants yield the same provisioning assignment $\boldsymbol{\kappa}_{s,\ell}$.

Figure 8.3b illustrates the evolution with the time index ℓ of $\boldsymbol{\kappa}_{s,\ell}$ for a slice s of type 2, characterized by an activation duration of three time slots and a demand pattern of type 2 (increasing for the second time slot and decreasing for the third one). In Figure 8.3b, the entries for which $y_{s,\ell}(i,v) = \max\{\kappa_{s,\ell}(i,v) - \kappa_{s,\ell\text{-}1}(i,v), 0\} > 0$ are highlighted in red, indicating an increase of the provisioned resources for slice s during consecutive time slots. Comparing Figure 8.3a, where $c_{\mathrm{a}} = 0$ and Figure 8.3b, where $c_{\mathrm{a}} > 0$, one observes that the number of adjustments of node assignment $\kappa_{s,\ell}(i,v)$ is reduced when $c_{\mathrm{a}} > 0$, as expected.



(a) $\kappa_{s,\ell}(i,v)$ when $c_{\mathrm{a}} = 0$. (b) $\kappa_{s,\ell}(i,v)$ when $c_{\mathrm{a}} > 0$.

Figure 8.3: Evolution of the provisioning assignment $\kappa_{s,\ell}(i,v)$ for a single slice (for each matrix, rows correspond to i, columns to v) when (a) $c_{\mathrm{a}} = 0$ and (b) $c_{\mathrm{a}} > 0$; the matrix entries with $\kappa_{s,\ell}(i,v) - \kappa_{s,\ell\text{-}1}(i,v) > 0$ are highlighted in red, whereas entries with $\kappa_{s,\ell}(i,v) - \kappa_{s,\ell\text{-}1}(i,v) \leqslant 0$ and $\kappa_{s,\ell}(i,v) > 0$ are in green.

8.5.2.2 Resource Provisioning for Multiple Slices

In this simulation, 1000 slice requests are generated among which 250 are tagged as Premium uniformly at random. Four choices are considered for the parameters α and ΔP, all with $P_{\max} = 3$, see Section 8.3.1. These choices impact the processing strategy of Premium and Standard slice requests. When $(\alpha, \Delta P) = (0.5, 0)$, Premium requests are processed immediately and Standard requests are processed in the time slot preceding their activation time slot. When $\alpha = 0$, whatever the value of ΔP, Premium and Standard requests are processed immediately, starting with the Premium requests. With $(\alpha, \Delta P) = (0.5, 0.5)$ and $(\alpha, \Delta P) = (0.5, 1)$, intermediate processing delays are obtained for Standard slices.

Figure 8.4 compares the performance of the J-PP and S-PP provisioning variants considering the four slice requests processing strategies induced by the choices of

$(\alpha, \Delta P)$.

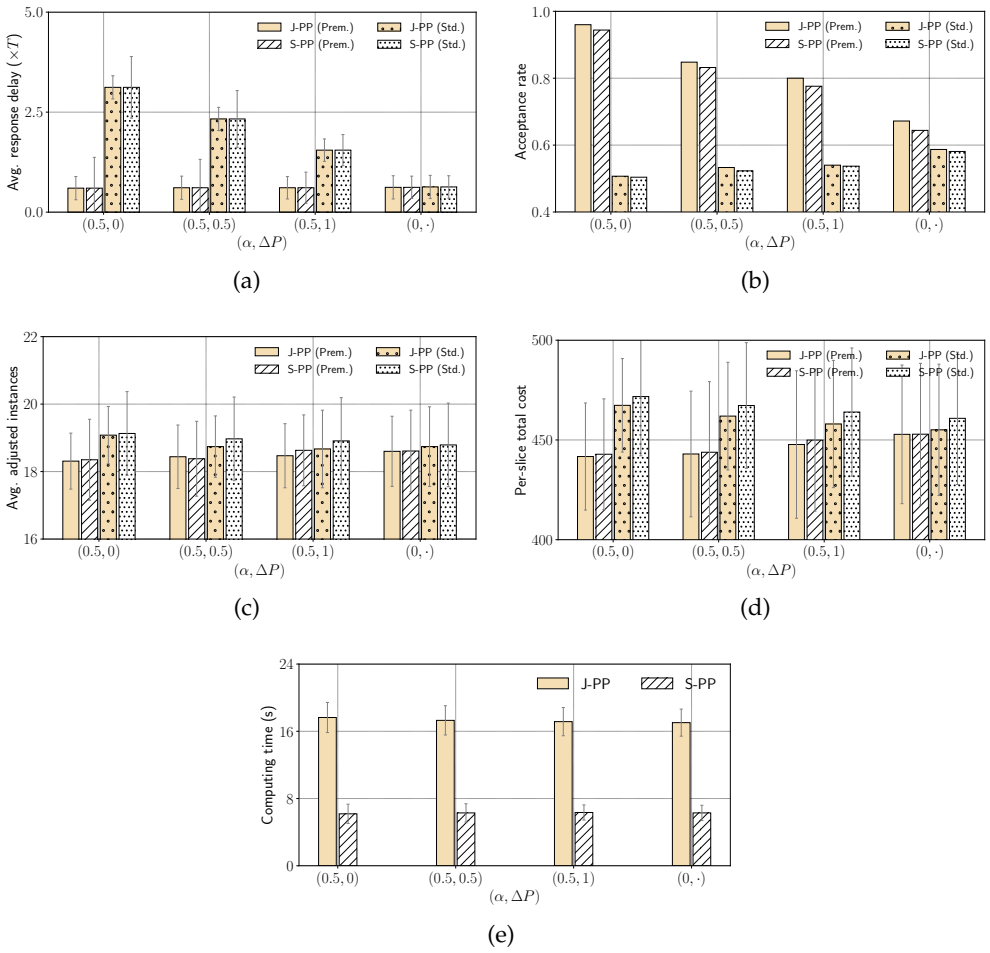


Figure 8.4: Performance comparison of the different processing strategies $(\alpha, \Delta P)$ with the J-PP and S-PP variants, in terms of (a) average response delay, (b) acceptance rate of slice requests, (b) average adjusted instances, (d) average cost per slice, and (e) computing time.

The average response delay for each slice request is shown in Figure 8.4a. The J-PP and S-PP variants share the same prioritized processing policy, therefore, both variants provide the same result in terms of response delay. When $\alpha = 0$, all requests are processed immediately, independently of their priority. The observed delay is only due to the processing which takes place at the end of each time slot during the processing time interval of duration εT. When $\alpha = 0.5$, the processing delay remains constant for Premium slices and increases when ΔP decreases for Standard slices.

Figure 8.4b illustrates the acceptance rate for the various processing strategies. Processing the slices jointly yields a slightly higher acceptance rate compared to a sequential approach. The acceptance rate of Premium slice requests is higher than that of Standard ones. The difference decreases when the average processing delay of Standard slice requests decreases. The difference is minimum when Stan-

dard slices are processed just after Premium slices in the same processing slot, *i.e.*, when $(\alpha, \Delta P) = (0, \cdot)$. Selecting the processing strategy allows one to adjust the acceptance rate difference between Premium slices and Standard slices.

Figure 8.4c illustrates the average number of adjustments of node assignments $y_{s,\ell}(i, v)$ per slice and per time slot. A joint approach is again more efficient than a sequential approach. Moreover, when the processing delay of Standard slices decreases, the number of adjustments for Standard slices decreases too, while the average number of adjustments of node assignments increases for Premium slices. This is explained by the fact that delaying more the processing of Standard slices facilitates finding assignments with fewer adjustments during the lifetime of Premium slices. The price to be paid is more adjustments for Standard slices.

Figure 8.4d shows the average per-slice provisioning cost charged by the InP to the MNO. Provisioning resources jointly leads to lower costs compared to a sequential provisioning. The provisioning costs increase for Premium slices when the processing of Standard slices is less delayed. For Standard and Premium slices, the provisioning costs are consistent with the evolution of the average number of adjustments of node assignments observed in Figure 8.4c.

Figure 8.4e shows that the computing times are independent of the processing strategy of Premium and Standard slice requests. As expected the `S-PP` variant is less time-consuming than the `J-PP` variant, due to the reduced number of variables involved.

8.6 Conclusions

This chapter considers a network slicing scenario with slice requests characterized by variable delays between their submission and activation and by different priority levels (*e.g.*, Premium and Standard). Considering these hypotheses, we introduce a prioritized slice admission control and resource provisioning mechanism. Admission decisions are provided and resources required for admitted slices are provisioned with a response delay depending on the slice priority and on the time left before its activation.

Adopting the perspective of the InPs, slice admission control and resource provisioning is formulated as a max-min optimization problem. The aim for the InP is to maximize the amount of admitted slices, *i.e.*, slices for which enough resources can be provisioned, while minimizing the cost charged to the MNOs. Uncertainties in the slice resource demands, as well as the presence of background service sharing the infrastructure are taken into account. Two reduced-complexity provisioning variants, namely `J-PP` and `S-PP`, are proposed to get solutions to the max-min problem.

Numerical results show that the proportion of admitted slices can be efficiently adjusted depending on the difference in the processing delay between Premium and Standard slices. When the delay difference increases, Premium slice requests are granted significantly more frequently, with less adjustments with time in the

provisioning scheme. This directly impacts the provisioning costs, which are reduced for Premium slices compared to Standard slices when the delay difference is large.

Chapter 9

Conclusions and Perspectives

In this chapter, we summarize the main contributions of this work and discuss possible future research directions.

9.1 Contributions

In what follows, we return to the research challenges posed in Chapter 1 and briefly review the answers provided in this thesis.

> **Challenge 1.** Enough infrastructure resources should be provisioned to accommodate slice resource demands, so that the desired service requirements are satisfied. The amount of resources provisioned to a slice depends on the characteristics of the service it provides, its QoS requirements expressed, *e.g.*, in terms of bandwidth, computing, and storage requirements.

This thesis proposes solutions that provision resources for slices to accommodate slice resource demands. The proposed approach goes beyond previous best-effort approaches, where the SFCs of a slice are deployed sequentially in the infrastructure network. With the approach proposed in this thesis, once resources are provisioned for a given slice, the SFCs of that slice are ensured to get enough resources to operate properly. This facilitates the satisfaction of the contracted service requirements with desired quality. In addition, numerical results show that the proposed provisioning solutions yield a reduction of the computational resources needed to deploy the SFCs.

In our provisioning approach, resource demands of a given slice are the aggregate resource demands of users associated to that slice. The aggregate resource demands of a slice are stated in the SLA between the MNO and the InP (MI-SLA). The MI-SLA may also include other required constraints, *e.g.*, the successful provisioning probability when accounting the uncertainties of slice resource demands. When performing the slice resource provisioning, the MI-SLA has to be satisfied, thus guaranteeing enough infrastructure resources are provisioned for the targeted slices.

Challenge 2. In a wireless slicing context, *e.g.*, RAN slicing, some constraints related to the coverage area of the slice as well as the user location also have to be taken into account.

This challenge has been addressed in Chapter 6. This chapter considers the problem of provisioning for joint core and radio access network resources, accounting some coverage constraints. To address the problem of user location (unknown during the resource provisioning phase), we have adopted a subarea partitioning approach. The coverage areas of slices are partitioned in subareas and, instead of provisioning radio blocks to users, one tries to provision radio blocks to each subarea. Several additional constraints have been presented to satisfy the coverage requirements. The main coverage constraints include: a constraint to ensure the provisioned radio resources (RBs) do not exceed the capacity of RRHs; a constraint to satisfy the minimum average user demand and the total slice radio resource demand for both uplink and downlink traffic; and finally a constraint to ensure the proportionality between provisioned radio resources for uplink and downlink. These additional constraints lead to a complicated optimization problem when considering the problem of joint radio and network resource provisioning. The joint provisioning problem (called *one-step provisioning*) becomes intractable when the number of slices increases. To cope with this issue, we have introduced an alternative approach (called *two-step provisioning*), in which the radio resource provisioning and network resource provisioning are performed sequentially. This approach is shown to have a lower time complexity than that of the joint approach, while still yielding good performance in terms of provisioning cost and efficiency of infrastructure resource utilization.

Challenge 3. An efficient slice resource provisioning mechanism should be robust against the uncertainties related to slice resource demands. Moreover, the proposed provisioning approach has to be performed so as to limit its impact on low-priority background services, which may co-exist with slices in the infrastructure network.

As pointed out in Chapter 7, the dynamics of traffics in individual slices (flow arrivals/departures), as well as of resource availability on the network infrastructure, may lead to slice QoS below the level expected by the Service Provider managing the slice. The traditional approach, in which allocated/provisioned resources are tailored to peak demands, may lead to over-provisioning of resources, thus decreases the efficiency of infrastructure resource utilization.

In Chapter 7, a slice resource provisioning method robust to randomness of resource demands has been proposed. The randomness is due to a partly unknown number of users with a random usage of the slice resources. The robustness is achieved by providing a probabilistic guarantee that the amount of provisioned network resources for a slice will meet the slice requirements. The proposed method

also tries to maintain the impact of resource provisioning on those background services (which are also time-varying) at a prescribed level.

> **Challenge 4.** Slice provisioning requests should be processed in an anticipated way, largely before their activation time, to guarantee the availability of infrastructure resources at the deployment time and during the life-time of the slices. The resulting slice admission control mechanism should take into account the dynamic nature of slice provisioning requests.

This challenge has been addressed in Chapter 8. Slice requests are characterized by variable delays between their submission and activation time; and by different priority levels (*e.g.*, Premium and Standard). We designed a prioritized slice admission control and resource provisioning mechanism. Admission decisions are provided and resources required for admitted slices are provisioned with a response delay depending on the slice priority and on the time left before its activation. In addition, different processing strategies have been proposed, each of which has a different impact on the processing of slice requests of different priority levels.

Numerical results show that the proportion of admitted slices can be efficiently adjusted depending on the difference in the processing delay between Premium and Standard slices. When the delay difference increases, Premium slice requests are granted significantly more frequently, with less adjustments with time in the provisioning scheme. This directly impacts the provisioning costs, which are reduced for Premium slices compared to Standard slices when the delay difference is large.

9.2 Perspectives

9.2.1 Accounting Additional Constraints

As discussed in Chapter 2, 5G aims to guarantee services with higher capacity, higher speed, and lower latency. To support diversified services with different requirements, some additional constraints, *e.g.*, latency or end-to-end error rate probability, should be added to the optimization formulation. For instance, for URLLC services, which require stringent constraints in terms of latency (order of milliseconds), some latency constraints should be taken into account. In general, latency comes from several sources such as transmission delay, propagation delay, queuing, processing delays, *etc*. The combination of those delay sources produces a complex and variable network latency profile. Some latency constraints have been considered in the literature for the SFC embedding problem, *e.g.*, [Alleg et al., 2017, Qu et al., 2019]. In an SFC embedding problem, only one infrastructure path is used to map to one SFC. It is thus easier to formulate the latency constraints than when considering a slice resource provisioning problem, where a slice may stretch across multiple paths in the infrastructure. A way to address this issue is to perform a worst-case analysis.

9.2.2 Development of Heuristics

To cope with a large number of variables and constraints involved in large optimization instances (ILPs or MILPs), various heuristics should be developed. In what follows, we present two prospective candidates for the design of heuristics, the *column generation* (CG) approach and the *eigendecompostion* approach, and see how these techniques could be used in the context of slice resource provisioning.

Column Generation CG is an efficient method for solving large linear programs [Nemhauser, 2012]. The principal idea of CG is that it is not necessary to consider all variables of a problem explicitly since, most of the variables are practically *non-basic*, *i.e.*, take a null value in the optimal solution of the linear program (LP). In that case, when solving an LP instance, theoretically only a subset of variables of the LP need to be considered. Considering an LP having minimization form, CG leverages this idea to identify, or *generate*, only the variables which potentially *reduce* the objective function[1], *i.e.*, to find variables with negative *reduced cost*.

CG decomposes the original LP (called *master problem*) into two problems: the *restricted master problem* (RMP) and the *pricing problem* (PP). The RMP takes exactly the form of the original LP with only a small subset of considered variables (called *generated columns*). The other unconsidered variables fall in the second group of variables (called *non-generated columns*). In the pricing problem, one checks whether any non-generated column that has been left out has *negative* reduced cost—if so, that column is added to the RMP and the RMP is solved again. This iterative CG algorithm is repeated until no negative reduced cost variables are identified. When the sub-problem return a solution with *non-negative* reduced cost, one can conclude that the current solution to the master problem is optimal. This decomposition of problems into master and sub-problems has enabled great reduction of time complexity compared to the original LP. The principles of the column generation method is summarized in Example 9.1.

Example 9.1 (Principles of column generation method)**.** Consider the following master problem (MP) and its dual problem (MP-Dual)

$$\begin{aligned}
\text{(MP)}\quad \min_{\boldsymbol{x}} \quad & \sum_{j\in\mathcal{J}} c_j x_j && \rightarrow \text{(MP-Dual)}\quad \max_{\boldsymbol{y}} \quad && \sum_{i\in\mathcal{I}} c_j y_j \\
\text{s.t.} \quad & \sum_{j\in\mathcal{J}} a_{ij} x_j \leqslant b_i, \forall i \in \mathcal{I}, && \text{s.t.} \quad && \sum_{i\in\mathcal{I}} a_{ij} y_i \geq c_j, \forall j \in \mathcal{J}, \\
& x_j \geq 0, \forall j \in \mathcal{J}. && && y_i \geq 0, \forall i \in \mathcal{I}.
\end{aligned}$$

Here $\boldsymbol{y}$ is the dual variables of $\boldsymbol{x}$. In general, given $y_i \geqslant 0, \forall i \in \mathcal{I}$, we want to find the index j of variable x_j yielding the minimum reduced cost $\bar{c}_j^*$, that is,

[1] When the LP is a maximization problem, CG will generate only the variables that potentially *increase* the objective function.

$$j^\star = \arg\min_{j \in \mathcal{J}} \ \bar{c}_j,$$

where $\bar{c}_j = c_j - \sum_{i \in \mathcal{I}} a_{ij} y_i$. In the simplex method, this is accomplished by calculating all possible reduced costs $\bar{c}_j$, $j \in \mathcal{J}$ and then selecting the most negative one. However, such explicit search of $j^\star$ in $\mathcal{J}$ is very time consuming, especially when dealing with $\mathcal{J}$ of large size. In the column generation method, we instead consider the MP with a reasonably small subset $\overline{\mathcal{J}} \subseteq \mathcal{J}$. This transforms the original MP to a RMP as follows

$$\begin{aligned} \text{(RMP)} \quad \min_{\boldsymbol{x}} \quad & \sum_{j \in \overline{\mathcal{J}}} c_j x_j \\ \text{s.t.} \quad & \sum_{j \in \overline{\mathcal{J}}} a_{ij} x_j \leqslant b_i, \forall i \in \mathcal{I}, \\ & x_j \geq 0, \forall j \in \overline{\mathcal{J}}. \end{aligned}$$

After solving the restricted master problem, we get the dual variables y_i , $\forall i \in \mathcal{I}$. In order to check the optimality of the current solution, we solve the following pricing problem

$$\text{(PP)} \quad \bar{c}_j^* = \min_j \left(c_j - \sum_{i \in \mathcal{I}} a_{ij} y_i \right)$$

If the objective value $\bar{c}_j^*$ of the PP is negative, the corresponding variable with index j (*i.e.*, x_j) is added to the RMP, and we repeat the optimization by solving again the RMP. Otherwise, $\bar{c}_j^* \geqslant 0$ indicates that no reduced cost $\bar{c}_j$ is negative, the solution cannot be more improved and the CG algorithm stops.

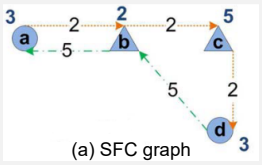
Some prior researches, *e.g.*, [Hu et al., 2013, Jarray and Karmouch, 2015, Liu et al., 2017], have used CG to solve the SFC embedding problem. For instance, [Liu et al., 2017] applies CG to the problem of dynamic SFC embedding. The master problem, which addresses the embedding of newly arrived SFCs and readjustment of in-service SFCs, is formulated as an ILP. In the proposed CG formulation, the pricing problem, in each iteration, tries to minimize the reduced cost and returns a feasible embedding solution for a *single* SFC. The RMP inherits the ILP formulation from the master problem but only considers the set of generated embedding solutions (columns) obtained from solving the PP.

When applying CG to the slice resource provisioning problem studied throughout this thesis, the most difficult step is to model the pricing problem. In [Liu et al., 2017], each column (embedding solution) contains the node and link mapping results for a single SFC. Each mapping solution uses only one infrastructure path. When considering the provisioning solution for a given slice, several nodes and paths may be used to provision resources for the slice. Modeling a column and the corresponding pricing problem thus becomes more challenging.

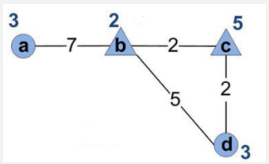
Eigendecomposition This method was first introduced in [Umeyama, 1988] to solve weighted graph matching problems (WGMPs). As discussed in Section 3.1, the work in [Mechtri et al., 2016] is one attempt to formulate the SFC embedding as a WGMP and then solve it using the eigendecompostion approach. In [Mechtri et al., 2016], the SFC graph and the infrastructure graph are modeled as weighted graphs, on which each node and each link have their own weight corresponding to their required resource (for the SFC graph), or their available resource (for the infrastructure graph). The eigendecompostion then tries to find the optimum matching between the SFC graph and the infrastructure graph.

Example 9.2 shows the weighted graphs of SFC and infrastructure network and how the matrix M is constructed to find the potential matching between these graphs.

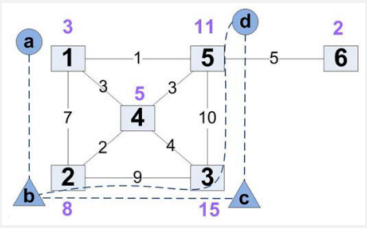
Example 9.2 (SFC embedding using eigendecomposition)**.** The figures below illustrate the (a) SFC graph, (b) Network Connectivity Topology (NCT) graph, and (c) the infrastructure graph (example taken from [Mechtri et al., 2016]). The weights in the infrastructure graph represent the available resources whereas those in the SFC and the NCT graphs refer to the resource demands.



(a) SFC graph



(b) Network Connectivity Topology (NCT) graph



(c) Infrastructure graph

The NCT graph takes exactly the same topology of the SFC graph, with the same weights on nodes. The weights in links of the NCT graph represent the aggregate bandwidth demand of the SFC links.

The eigendecomposition algorithm proposed in [Mechtri et al., 2016] first computes the adjacency matrices A_{I} and A_{NCT} as follows

$$A_{\mathrm{I}} = \begin{bmatrix} 3 & 7 & 0 & 3 & 1 & 0 \\ 7 & 8 & 9 & 2 & 0 & 0 \\ 0 & 9 & 15 & 4 & 10 & 0 \\ 3 & 2 & 4 & 5 & 3 & 0 \\ 1 & 0 & 10 & 3 & 11 & 5 \\ 0 & 0 & 0 & 0 & 5 & 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 3 & 7 & 7 & 3 & 1 & 5 \\ 7 & 8 & 9 & 2 & 9 & 5 \\ 7 & 9 & 15 & 4 & 10 & 5 \\ 3 & 2 & 4 & 5 & 3 & 4 \\ 1 & 9 & 10 & 3 & 11 & 5 \\ 5 & 5 & 5 & 4 & 5 & 2 \end{bmatrix}, \tag{9.1}$$

$$A_{\text{NCT}} = \begin{bmatrix} 3 & 7 & 0 & 0 \\ 7 & 2 & 2 & 5 \\ 0 & 2 & 5 & 2 \\ 0 & 5 & 2 & 3 \end{bmatrix} \overset{\text{add padding } (0)}{\Rightarrow} \begin{bmatrix} 3 & 7 & 0 & 0 & 0 & 0 \\ 7 & 2 & 2 & 5 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 & 0 \\ 0 & 5 & 2 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{9.2}$$

A_{I} and A_{NCT} represent respectively the infrastructure and the NCT graphs. In each matrix, the diagonal elements (*i.e.*, $A_{\text{I}}(i,i)$ and $A_{\text{NCT}}(i,i)$) refer to the node weights and the off-diagonal elements (*i.e.*, $A_{\text{I}}(i,j)$ and $A_{\text{NCT}}(i,j)$, $i \neq j$) refer to the link weights between nodes i and j.

In A_{I}, some elements corresponding to indirect links in the infrastructure graph are firstly set to zero (*e.g.*, $A_{\text{I}}(1,3) = A_{\text{I}}(1,3) = 0$, highlighted in red in A_{I}). Those elements are then recalculated based on the maximal bandwidth between the corresponding nodes. For instance, the nodes 1 and 3 have no direct connection and have a maximal bandwidth of 7 (via the path $1 \to 2 \to 3$), hence $A_{\text{I}}(1,3)$ and $A_{\text{I}}(3,1)$ are reset to 7, and so forth, as shown in (9.1).

In A_{NCT}, since the NCT graph is usually smaller than the infrastructure graph, A_{NCT} is transformed into a matrix of the same size as A_{I} by adding zero paddings, as shown in (9.2).

Denote U_{I} and U_{NCT} as the *eigenvector* matrices of A_{I} and A_{NCT}. The eigen-decomposition algorithm finally computes a matrix M based on U_{I} and U_{NCT} as

$$M = \overline{U}_{\text{NCT}} \times \overline{U}_{\text{I}}^{\top}, \tag{9.3}$$

where each element of $\overline{U}_{\text{I}}$ and $\overline{U}_{\text{NCT}}$ takes the absolute value of respectively U_{I} and U_{NCT}. One obtains

$$M = \begin{array}{c} \text{Infrastructure nodes} \\ \begin{array}{cccccc} (1) & (2) & (3) & (4) & (5) & (6) \end{array} \\ \left[\begin{array}{cccccc} 0.05 & 0.03 & 0.04 & 0.01 & 0.22 & 0.11 \\ 0.03 & 0.23 & 0.02 & 0.09 & 0.00 & 0.36 \\ 0.04 & 0.02 & 0.49 & 0.07 & 0.00 & 0.15 \\ 0.01 & 0.09 & 0.07 & 0.56 & 0.16 & 0.05 \\ \hdashline 0.22 & 0.00 & 0.00 & 0.16 & 0.45 & 0.12 \\ 0.11 & 0.36 & 0.15 & 0.05 & 0.12 & 0.08 \end{array}\right] \begin{array}{l} (a) \\ (b) \\ (c) \\ (d) \end{array} \text{SFC nodes} \end{array}$$

The columns of M correspond to the infrastructure nodes, whereas the first four rows of M correspond to the SFC nodes. For each row i of M, the algorithm selects the element $M(i,j)$ with the highest value as a potential candidate for the mapping of the corresponding SFC node i to the infrastructure node j. The mapping is double-checked by verifying whether the resource demands are satisfied. If the node candidate with the highest value does not satisfy the required constraints, the node with the second highest value in the same column is selected, an so forth, until every constraints are satisfied. In

this example, the SFC nodes a and c are mapped to the infrastructure nodes 1 and 3, which have the highest values in M (highlighted in green), whereas the SFC nodes b and d are mapped to infrastructure nodes 2 and 5, which have the second highest values in the corresponding rows in M (the highest values in these rows are highlighted in orange).

Applying the eigendecompostion method to the problem of slice resource provisioning is more difficult, since each slice node may require several infrastructure nodes on which resources are provisioned. In other words, the slice resource provisioning cannot be represented by a one-to-one mapping. One possible approach is, in each time, to provision resources for each SFC of the slice, until the aggregate resource demands of the slice is reached. With this approach, the construction of the matrix M is still useful to have an initial guess for the provisioning solution for each SFC.

9.2.3 Multi-Domain Network Slicing

One of the challenging problems in network slicing is to deploy end-to-end network slices, which refer to slices that span across multiple domains, not just a single domain. For example, network slices may stretch across a huge geographic area at worldwide level, or encompass areas where slice coverage can only be guaranteed by using resources from different MNOs or InPs.

Similarly, some specific services may need computing and storage resources offered by a particular cloud providers [Taleb et al., 2019]. In such situation, the deployment of network slices require an efficient combination of resources provided by different InPs. The problem of slice resource provisioning in a multi-domain context thus becomes very challenging. Each InP may find its own resource provisioning solution for a part of the slice resource demands, and afterwards, there requires an efficient coordination between the InPs to eventually have a feasible provisioning solution that allows the slices to operate across the networks provided by those InPs.

To address this problem, one may introduce two algorithms, one is used by the InPs to solve the provisioning problem for the part of slice resource demands that each InP receives, and one another is used by a central entity that coordinates the InPs. This algorithm should return several possible solutions. And then, the InP coordinator runs the second algorithm with the results given by the InPs to find a feasible final solution that accounts for the total resource demands. Such approach has been considered in the literature, *e.g.*, in [Boutigny et al., 2018, Fossati et al., 2020], but for the problem of SFC embedding. Further investigations are needed when considering this approach for the problem of slice resource provisioning.

9.2.4 Slot-by-Slot Provisioning

In Chapter 8, resources are provisioned for the whole life time of each slice. Considering all active time slots of a given slice simultaneously helps reducing the

required adjustments of node resources, but leads to large optimization problems, especially when the life time of slices spans over many time slots. An alternative approach to address this problem is to perform the slice resource provisioning in a slot-by-slot manner, *i.e.*, instead of considering all active time slots simultaneously, one may try to provision resources for the slice sequentially for each time slot in which it is active.

We now consider the resource provisioning for a single slice s and for a single time slot $\ell \in \mathcal{K}_s$ in which it is active. The objective of Problem 8.3 presented in Chapter 8 can be adapted to the slot-by-slot provisioning approach as follows

$$\max_{d_{s,\ell}} \min_{\boldsymbol{\kappa}_{s,\ell}} \; C_{\text{sbs}}\left(\boldsymbol{\kappa}_{s,\ell}\right) = C_{\text{r}}\left(\boldsymbol{\kappa}_{s,\ell}\right) + C_{\text{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right). \tag{9.4}$$

Compared to the objective function of Problem 8.3, in the objective function C_{sbs}, the adaptation cost $C_{\text{a}} = \sum_{i\in\mathcal{N}} \sum_{v\in\mathcal{N}_s} y_{s,\ell}\left(i,v\right) c_{\text{a}}\left(i\right)$ is omitted since only one time slot is considered. Similarly, the constraints in Problem 8.3 can be adapted for one single time slot ℓ. The final provisioning decision for the considered slice is the aggregate of all provisioning results for each time slot. This approach is expected to yield a reduction in time complexity when solving the slice resource provisioning problem.

Nevertheless, when considering a slot-by-slot provisioning approach using the objective function C_{sbs}, the provisioning result of a given time slot ℓ does not take into account the prior provisioning result obtained in the previous time slot $\ell - 1$, *i.e.*, $\boldsymbol{\kappa}_{s,\ell\text{-}1}$. This possibly yields several changes in $\boldsymbol{\kappa}_{s,\ell}$, thus leading to many redeployment (*e.g.*, migration) of VNF instances. To address this issue, for each time slot $\ell \in \mathcal{K}_s$, one may take the results $\boldsymbol{\kappa}_{s,\ell\text{-}1}$ of time slot $\ell - 1$ as input for the optimization problem run in time slot ℓ. For this approach, called *myopic* provisioning, the objective function C_{sbs} in (9.4) can be updated as

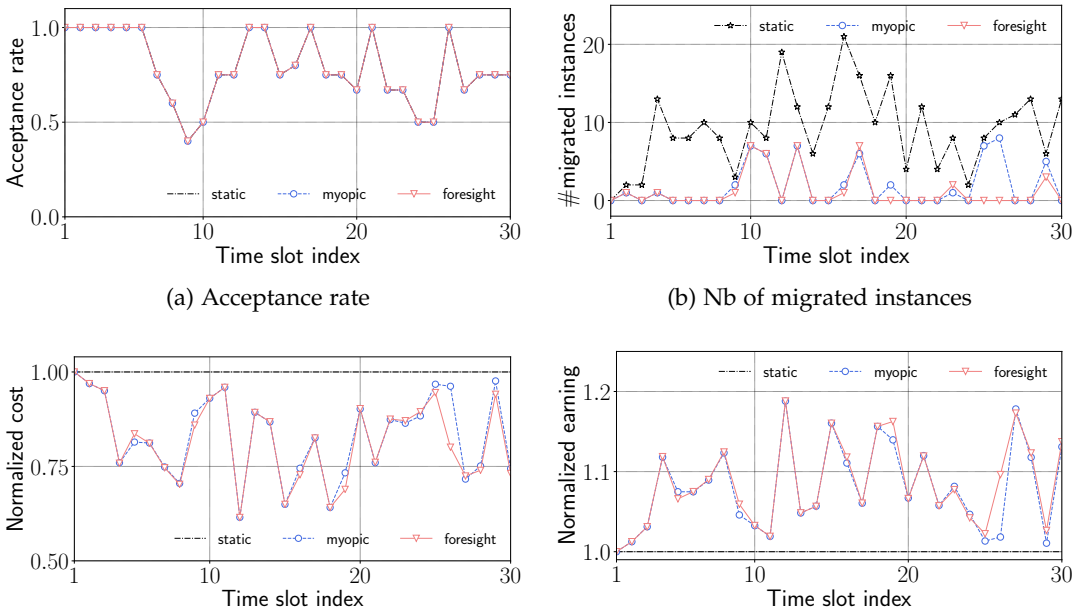
$$C^{\text{my}}_{\text{sbs}}\left(\boldsymbol{\kappa}_{s,\ell}|\boldsymbol{\kappa}_{s,\ell\text{-}1}\right) = C_{\text{r}}\left(\boldsymbol{\kappa}_{s,\ell}\right) + C_{\text{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right) + \sum_{i,v}\left(1 - \widetilde{\kappa}_{s,\ell\text{-}1}\left(i,v\right)\right)\widetilde{\kappa}_{s,\ell}\left(i,v\right) c_{\text{a}}\left(i\right), \tag{9.5}$$

where the additional variable $\widetilde{\kappa}_{s,\ell}\left(i,v\right) \in \{0,1\}$ is the node mapping indicator, *i.e.*,

$$\begin{cases} \widetilde{\kappa}_{s,\ell}\left(i,v\right) = 1 & \text{if } \kappa_{s,\ell}\left(i,v\right) > 0, \\ \widetilde{\kappa}_{s,\ell}\left(i,v\right) = 0 & \text{otherwise.} \end{cases}$$

The last term of (9.5), $\sum_{i,v}\left(1 - \widetilde{\kappa}_{s,\ell\text{-}1}\left(i,v\right)\right)\widetilde{\kappa}_{s,\ell}\left(i,v\right) c_{\text{a}}\left(i\right)$ accounts for the cost of using a new infrastructure node i to provision resources for any virtual node v, *i.e.*, this cost is not counted if a VNF instance reuses the infrastructure node that was previously used.

We can go even further by taking into account both prior information (provisioning solution for the previous time slot, $\boldsymbol{\kappa}_{k\text{-}1}$) and posterior information (predicted variations of slice resource demands in the next consecutive time slots, *i.e.*, $\mathbf{R}_{s,\ell+1}$, $\mathbf{R}_{s,\ell+2}$, ...). This approach is called *foresighted* provisioning.



(a) Acceptance rate

(b) Nb of migrated instances

(c) Normalized cost

(d) Normalized earning

Figure 9.1: Performance comparison of three variants in saturated scenario, in terms of (a) acceptance rate, (b) number of redeployed VNF instances, (c) normalized provisioning cost, and (d) normalized earning.

Considering first the *one-step ahead* foresighted provisioning, *i.e.*, the posterior information only contains $\mathbf{R}_{s,\ell+1}$. The objective function for this approach can be formulated as a sum of two weighted myopic costs $C^{\text{my}}_{\text{sbs}}$, one accounting for the provisioning of slices for the first active time slot ℓ, and an other accounts for the consecutive time slot $\ell+1$, associated with a discount factor λ. The objective function of the one-step ahead foresighted provisioning approach, denoted as $C^{\text{fo},1}_{\text{sbs}}$, can be defined as

$$C^{\text{fo},1}_{\text{sbs}}\left(\boldsymbol{\kappa}_{s,\ell}, \boldsymbol{\kappa}_{s,\ell+1} | \boldsymbol{\kappa}_{s,\ell\text{-}1}\right) = C^{\text{my}}_{\text{sbs}}\left(\boldsymbol{\kappa}_{s,\ell} | \boldsymbol{\kappa}_{s,\ell\text{-}1}\right) + \lambda C^{\text{my}}_{\text{sbs}}\left(\boldsymbol{\kappa}_{s,\ell+1} | \boldsymbol{\kappa}_{s,\ell}\right). \tag{9.6}$$

Similarly, the foresighted cost function for N-step ahead, $C^{\text{fo},N}_{\text{sbs}}$, is given by

$$C^{\text{fo},1}_{\text{sbs}}\left(\boldsymbol{\kappa}_{s,\ell}, \boldsymbol{\kappa}_{s,\ell+1} | \boldsymbol{\kappa}_{s,\ell\text{-}1}\right) = C^{\text{my}}_{\text{sbs}}\left(\boldsymbol{\kappa}_{s,\ell} | \boldsymbol{\kappa}_{s,\ell\text{-}1}\right) + \sum_{i=1}^{N} \lambda_i C^{\text{my}}_{\text{sbs}}\left(\boldsymbol{\kappa}_{s,\ell+i} | \boldsymbol{\kappa}_{s,\ell+(i\text{-}1)}\right), \tag{9.7}$$

where λ_i is the discount factor associated with $C^{\text{my}}_{\text{sbs}}\left(\boldsymbol{\kappa}_{s,\ell+i} | \boldsymbol{\kappa}_{s,\ell+(i\text{-}1)}\right)$.

Figure 9.1 shows some preliminary results of different slot-by-slot provisioning variants: `static`, `myopic`, and `foresight`, when considering respectively C_{sbs}, $C^{\text{my}}_{\text{sbs}}$, and $C^{\text{fo},1}_{\text{sbs}}$ as objective functions. It can be seen that all variants yield a similar acceptance rate of slice requests. Nevertheless, `static` variant requires a much higher number of VNF instances to be redeployed, see Figure 9.1b), thus yielding a higher provisioning cost compared to the `myopic` and `foresight` schemes (see Figure 9.1c), leading to lower earnings for the InP, as shown in Figure 9.1d. In

general, the `foresight` scheme performs the best among the three variants.

In future work, the slot-by-slot slice resource provisioning approach considering the objective functions C_{sbs}, $C_{\text{sbs}}^{\text{my}}$, and $C_{\text{sbs}}^{\text{fo},N}$ will be carefully studied and compared with the methods introduced in Chapter 8.

Bibliography

[3GPP, 2016a] 3GPP (2016a). Feasibility Study on New Services and Markets Technology Enablers - Network Operation. *3GPP Release 15 - Technical Report TR 22.864*, pages 1–15. (cited in page 12.)

[3GPP, 2016b] 3GPP (2016b). Study on Architecture for Next Generation System. *3GPP Release 14- Technical Report TR 23.799*. (cited in page 14.)

[3GPP, 2019] 3GPP (2019). Network Slicing - 3GPP Use Case. *Internet-Draft*, pages 1–12. (cited in page 10.)

[3GPP, 2020] 3GPP (2020). Management and Orchestration; Concepts, Use Cases and Requirements. *3GPP TS 28.530 V17.0.0*. (cited in pages xiv, xvi, 3, 17, 18, 19, and 140.)

[5G Americas, 2016] 5G Americas (2016). Network Slicing for 5G Networks & Services. *White Paper*. (cited in pages 1 and 138.)

[5GPPP, 2017] 5GPPP (2017). View on 5G Architecture. *White Paper*. (cited in page 17.)

[Afolabi et al., 2018] Afolabi, I., Taleb, T., Samdanis, K., Ksentini, A., and Flinck, H. (2018). Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions. *IEEE Commun. Surveys Tuts.*, 20(3):2429–2453. (cited in pages 14 and 17.)

[Alleg et al., 2017] Alleg, A., Ahmed, T., Mosbah, M., Riggio, R., and Boutaba, R. (2017). Delay-Aware VNF Placement and Chaining based on a Flexible Resource Allocation Approach. In *Proc. CNSM*, pages 1–7, Tokyo, Japan. (cited in page 119.)

[Barakabitze et al., 2020] Barakabitze, A. A., Ahmad, A., Mijumbi, R., and Hines, A. (2020). 5G Network Slicing Using SDN and NFV: A Survey of Taxonomy, Architectures and Future Challenges. *Computer Networks*, 167. (cited in pages 1, 4, 13, 16, 17, 73, 138, 141, and 142.)

[Basta et al., 2014] Basta, A., Kellerer, W., Hoffmann, M., Morper, H. J., and Hoffmann, K. (2014). Applying NFV and SDN to LTE Mobile Core Gateways; The Functions Placement Problem. In *ACM AllThingsCellular*, pages 33–38. (cited in pages 1 and 138.)

[Baumgartner et al., 2018] Baumgartner, A., Bauschert, T., D'Andreagiovanni, F., and Reddy, V. S. (2018). Towards Robust Network Slice Design under Correlated Demand Uncertainties. In *Proc. ICC*, pages 1–7A. (cited in page 24.)

[Bauschert and Reddy, 2019] Bauschert, T. and Reddy, V. S. (2019). Genetic Algorithms for the Network Slice Design Problem Under Uncertainty. In *Proc. GECCO Companion*, pages 360–361. (cited in page 24.)

[Bega et al., 2020] Bega, D., Gramaglia, M., Banchs, A., Sciancalepore, V., and Costa-Perez, X. (2020). A Machine Learning Approach to 5G Infrastructure Market Optimization. *IEEE Trans. Mobile Comput.*, 19(3):498–512. (cited in pages 25 and 97.)

[Bega et al., 2017] Bega, D., Gramaglia, M., Banchs, A., Sciancalepore, V., and Samdanis, K. (2017). Optimising 5G Infrastructure Markets: The Business of Network Slicing. In *IEEE INFOCOM*. (cited in pages 25 and 97.)

[Boubendir et al., 2018] Boubendir, A., Guillemin, F., Le Toquin, C., Alberi-Morel, M. L., Faucheux, F., Kerboeuf, S., Lafragette, J. L., and Orlandi, B. (2018). Federation of Cross-Domain Edge Resources: A Brokering Architecture for Network Slicing. In *Proc. IEEE NetSoft*, pages 494–499. IEEE. (cited in page 99.)

[Bouten et al., 2017] Bouten, N., Mijumbi, R., Serrat, J., Famaey, J., Latre, S., and De Turck, F. (2017). Semantically Enhanced Mapping Algorithm for Affinity-Constrained Service Function Chain Requests. *IEEE Trans. Netw. Service Manag.*, 14(2):317–331. (cited in pages 43, 46, 47, and 71.)

[Boutigny et al., 2018] Boutigny, F., Betgé-Brezetz, S., Debar, H., Blanc, G., Lavignotte, A., and Popescu, I. (2018). Multi-Provider Secure Virtual Network Embedding. In *Proc. IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. (cited in page 124.)

[Burden and Douglas Faires, 2011] Burden, R. L. and Douglas Faires, J. (2011). *Numerical Analysis*. Brooks/Cole, Cengage Learning, 9th edition. (cited in page 84.)

[Chatterjee et al., 2018] Chatterjee, S., Abdel-rahman, M. J., and Mackenzie, A. B. (2018). Virtualization Framework for Cellular Networks with Downlink Rate Coverage Probability Constraints. In *Proc. IEEE GLOBECOM*. (cited in pages 4, 22, 49, and 141.)

[China Mobile, 2011] China Mobile (2011). C-RAN: The Road Towards Green RAN. *China Mobile White Paper, ver 2.5*, 5:15–16. (cited in page 30.)

[Chowdhury et al., 2012] Chowdhury, M., Rahman, M. R., and Boutaba, R. (2012). ViNEYard: Virtual Network Embedding Algorithms. *IEEE/ACM Trans. Netw.*, 20(1):206–219. (cited in pages 20 and 36.)

[Chun et al., 2003] Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., and Bowman, M. (2003). PlanetLab: An Overlay Testbed for Broad-Coverage Services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12. (cited in pages 13 and 14.)

[Cohen et al., 2015] Cohen, R., Lewin-Eytan, L., Naor, J. S., and Raz, D. (2015). Near Optimal Placement of Virtual Network Functions. In *Proc. IEEE INFOCOM*, pages 1346–1354. (cited in pages 20 and 36.)

[Coniglio et al., 2015] Coniglio, S., Koster, A. M., and Tieves, M. (2015). Virtual Network Embedding Under Uncertainty: Exact And Heuristic Approaches. In *Proc. DRCN*, pages 1–8. IEEE. (cited in page 23.)

[D'Oro et al., 2018] D'Oro, S., Restuccia, F., Melodia, T., Member, S., Palazzo, S., and Member, S. (2018). Low-Complexity Distributed Radio Access Network Slicing: Algorithms and Experimental Results. *IEEE/ACM Trans. Netw.*, 26(6):2815–2828. (cited in pages 22 and 49.)

[Ebrahimi et al., 2020] Ebrahimi, S., Zakeri, A., Akbari, B., and Mokari, N. (2020). Joint Resource and Admission Management for Slice-enabled Networks. In *Proc. IEEE NOMS*, pages 1–7. (cited in pages 25 and 97.)

[Elliott, 2008] Elliott, C. (2008). GENI - Global Environment for Network Innovations. In *33rd IEEE Conference on Local Computer Networks (LCN)*, pages 8–8, Montreal, QC, Canada. (cited in pages 13 and 14.)

[ETSI, 2012] ETSI (2012). Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges & Call for Action. *White Paper*. (cited in pages 13, 14, and 15.)

[ETSI, 2014] ETSI (2014). Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV. *ETSI Group Specification GS NFV 003 v1.2.1*. (cited in pages xiv, 15, and 16.)

[ETSI, 2016] ETSI (2016). Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) Radio Transmission and Reception. *Tecnical Specification - ETSI TS 136 101 V10.21.0 (2016-04)*. (cited in page 64.)

[ETSI, 2020] ETSI (2020). European Telecommunications Standards Institute, Industry Specification Groups (ISG) - NFV. (cited in page 16.)

[Fendt et al., 2019] Fendt, A., Mannweiler, C., Schmelz, L. C., and Bauer, B. (2019). An Efficient Model for Mobile Network Slice Embedding under Resource Uncertainty. In *Proc. ISWCS*, pages 602–606. (cited in pages 23, 24, and 74.)

[Fischer et al., 2013] Fischer, A., Botero, J. F., Till Beck, M., De Meer, H., and Hesselbach, X. (2013). Virtual Network Embedding: A Survey. *IEEE Commun. Surveys Tuts.*, 15(4):1888–1906. (cited in page 20.)

[Fossati et al., 2020] Fossati, F., Moretti, S., Rovedakis, S., and Secci, S. (2020). Decentralization of 5G Slice Resource Allocation. In *Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS)*, pages 1–9. (cited in page 124.)

[Galis and Makhijani, 2018] Galis, A. and Makhijani, K. (2018). Turorial: Network Slicing Landscape: A Holistic Architectural Approach, Orchestration and Management with Applicability in Mobile and Fixed Networks and Clouds. In *IEEE Network Softwarization (NetSoft)*. (cited in page 12.)

[Genz, 2004] Genz, A. (2004). Numerical Computation of Rectangular Bivariate and Trivariate Normal and t Probabilities. *Statistics and Computing*, 14(3):251–260. (cited in page 84.)

[Ghaznavi et al., 2015] Ghaznavi, M., Khan, A., Shahriar, N., Alsubhi, K., Ahmed, R., and Boutaba, R. (2015). Elastic Virtual Network Function Placement (EVNFP). In *4th IEEE Conference on Cloud Networking (CloudNet) 2015*, number October, pages 255–260. (cited in page 97.)

[Goodman, 1960] Goodman, L. A. (1960). On the Exact Variance of Products. *Journal of the American Statistical Association*, 55(292):708–713. (cited in page 83.)

[GSM Alliance, 2017] GSM Alliance (2017). An Introduction to Network Slicing. *White Paper*. (cited in pages 1, 12, and 138.)

[GSMA, 2018] GSMA (2018). Network Slicing: Use Case Requirements. *GSMA Reference Document*. (cited in pages xiv, 12, and 13.)

[Halabian, 2019] Halabian, H. (2019). Distributed Resource Allocation Optimization in 5G Virtualized Networks. *IEEE J. Sel. Areas Commun.*, 37(3):627–642. (cited in pages 21 and 49.)

[Han et al., 2020] Han, B., Sciancalepore, V., Costa-Perez, X., Feng, D., and Schotten, H. D. (2020). Multiservice-Based Network Slicing Orchestration with Impatient Tenants. *IEEE Trans. Wireless Commun.*, 19(7):5010–5024. (cited in pages 25 and 97.)

[Hu et al., 2013] Hu, Q., Wang, Y., and Cao, X. (2013). Resolve The Virtual Network Embedding Problem: A Column Generation Approach. In *Proc. IEEE INFOCOM*, pages 410–414. (cited in page 121.)

[Huin et al., 2017] Huin, N., Jaumard, B., and Giroire, F. (2017). Optimization of Network Service Chain Provisioning. In *Proc. IEEE ICC*. (cited in pages 4, 21, 23, 36, 73, and 142.)

[Huynh et al., 2019] Huynh, N. V., Hoang, D. T., Nguyen, D. N., and Dutkiewicz, E. (2019). Real-Time Network Slicing with Uncertain Demand : A Deep Learning Approach. In *Proc. IEEE ICC*. (cited in page 24.)

[IETF, 2017] IETF (2017). Network Slicing Architecture. *Internet-Draft*, pages 1–8. (cited in pages 1 and 138.)

[IMT, 2015] IMT (2015). IMT Vision - Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond. *ITU-R Recommendation M.2083*. (cited in pages xiv, 10, 11, and 12.)

[ITU-T, 2018] ITU-T (2018). GSTR-TN5G: Transport Network Support of IMT-2020/5G. *ITU Technical Report*. (cited in page 31.)

[Jarray and Karmouch, 2015] Jarray, A. and Karmouch, A. (2015). Decomposition Approaches for Virtual Network Embedding With One-Shot Node and Link Mapping. *IEEE/ACM Transactions on Networking*, 23(3):1012–1025. (cited in page 121.)

[Jiang et al., 2019] Jiang, C., Han, G., Lin, J., Jia, G., Shi, W., and Wan, J. (2019). Characteristics of Co-Allocated Online Services and Batch Jobs in Internet Data Centers: A Case Study from Alibaba Cloud. *IEEE Access*, 7:22495–22508. (cited in page 75.)

[Kaloxylos, 2018] Kaloxylos, A. (2018). A Survey and an Analysis of Network Slicing in 5G Networks. *IEEE Commun. Std. Mag.*, 2(1):60–65. (cited in pages 1, 3, 138, and 141.)

[Kang et al., 2017] Kang, J., Kang, J., and Simeone, O. (2017). On the Trade-Off between Computational Load and Reliability for Network Function Virtualization. *IEEE Commun. Lett.*, 21(8):1767–1770. (cited in pages 20 and 36.)

[Kazmi et al., 2019] Kazmi, S. M. A., Khan, L. U., Tran, N. H., Hong, C. S., Kazmi, S. M. A., Khan, L. U., Tran, N. H., and Hong, C. S. (2019). *Network Slicing for 5G and Beyond Networks*. Springer. (cited in pages xiv, 11, and 12.)

[Kerboeuf et al., 2018] Kerboeuf, S., Luu, Q.-T., Kieffer, M., and Mouradian, A. (2018). Method and Apparatus for Mapping Network Slices Onto Network Infrastructures With SLA Guarantee. In *WO2020114608A1*. WIPO. (cited in pages 8 and 147.)

[Lee et al., 2016] Lee, Y. L., Loo, J., and Chuah, T. C. (2016). A New Network Slicing Framework for Multi-Tenant Heterogeneous Cloud Radio Access Networks. In *Proc. ICAEES*, pages 414–420. (cited in pages 22 and 49.)

[Li et al., 2017] Li, X., Samaka, M., Chan, A. H., Bhamare, D., Gupta, L., Guo, C., and Jain, R. (2017). Network Slicing for 5G: Challenges and Opportunities. *IEEE Internet Comput.*, 21(5):20–27. (cited in pages 1, 3, 138, and 141.)

[Li et al., 2018] Li, X., Samaka, M., Chan, H. A., Bhamare, D., Gupta, L., Guo, C., and Jain, R. (2018). Network Slicing for 5G: Challenges and Opportunities. *IEEE Internet Computing*. (cited in pages 4 and 142.)

[Liang and Yu, 2014] Liang, C. and Yu, F. R. (2014). Wireless Network Virtualization: A Survey, Some Research Issues and Challenges. *IEEE Commun. Surveys Tuts.*, pages 1–24. (cited in pages 1, 27, and 138.)

[Lindquist et al., 1966] Lindquist, A. B., Seeber, R. R., and Cdmeau, L. W. (1966). A Time-Sharing System Using an Associative Memory. *Proceedings of the IEEE*, 54(12):1774–1779. (cited in pages 12 and 14.)

[Liu et al., 2017] Liu, J., Lu, W., Zhou, F., Lu, P., and Zhu, Z. (2017). On Dynamic Service Function Chain Deployment and Readjustment. *IEEE Trans. Netw. Service Manag.*, 14(3):543–553. (cited in pages 21, 24, 36, 97, and 121.)

[Luu et al., 2021a] Luu, Q.-t., Kerboeuf, S., and Kieffer, M. (2021a). Admission Control and Resource Provisioning for Prioritized Slice Requests with Uncertainties. *submitted to IIEEE Trans. Netw. Service Manag.*, pages 1–16. (cited in pages 7, 96, and 146.)

[Luu et al., 2021b] Luu, Q. T., Kerboeuf, S., and Kieffer, M. (2021b). Foresighted Resource Provisioning for Network Slicing. In *Proc. IEEE HPSR*, pages 1–8. IEEE. (cited in pages 7 and 147.)

[Luu et al., 2021c] Luu, Q.-T., Kerboeuf, S., and Kieffer, M. (2021c). Uncertainty-Aware Resource Provisioning for Network Slicing. *IEEE Trans. Netw. Service Manag.*, 18(1):79–93. (cited in pages 7, 73, and 146.)

[Luu et al., 2020a] Luu, Q.-T., Kerboeuf, S., Mouradian, A., and Kieffer, M. (2020a). A Coverage-Aware Resource Provisioning Method for Network Slicing. *IEEE/ACM Trans. Netw.*, 28(6):2393–2406. (cited in pages 7, 48, 74, and 147.)

[Luu et al., 2020b] Luu, Q.-T., Kerboeuf, S., Mouradian, A., and Kieffer, M. (2020b). Radio Resource Provisioning for Network Slicing with Coverage Constraints. In *Proc. IEEE ICC*. (cited in pages 7, 48, and 147.)

[Luu et al., 2020c] Luu, Q.-T., Kerboeuf, S., Mouradian, A., and Kieffer, M. (2020c). Resource Provisioning for Network Slices with Coverage Constraints. In *ANR MAESTRO-5G Workshop on Orchestration of 5G Networks and Beyond*, CentraleSupélec, Gif-sur-Yvette, France. (cited in pages 8 and 147.)

[Luu et al., 2018] Luu, Q.-T., Kieffer, M., Mouradian, A., and Kerboeuf, S. (2018). Aggregated Resource Provisioning for Network Slices. In *Proc. IEEE GLOBECOM*, pages 1–6, Abu Dhabi, UAE. (cited in pages 8, 36, 49, 87, and 147.)

[McKeown et al., 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review*, 38(2):68–74. (cited in pages 13 and 14.)

[Mechtri et al., 2016] Mechtri, M., Ghribi, C., and Zeghlache, D. (2016). A Scalable Algorithm for the Placement of Service Function Chains. *IEEE Trans. Netw. Service Manag.*, 13(3):533–546. (cited in pages 21, 36, and 122.)

[Mireslami et al., 2019] Mireslami, S., Rakai, L., Wang, M., and Far, B. H. (2019). Dynamic Cloud Resource Allocation Considering Demand Uncertainty. *IEEE Trans. on Cloud Comput.*, 7161(c):1–1. (cited in page 23.)

[Nakao et al., 2017] Nakao, A., Du, P., Kiriha, Y., Granelli, F., Gebremariam, A. A., Taleb, T., and Bagaa, M. (2017). End-to-end Network Slicing for 5G Mobile Networks. *J. Inf. Process.*, 25:153–163. (cited in page 14.)

[Nemhauser, 2012] Nemhauser, G. (2012). Column Generation for Linear and Integer Programming. *Optimization Stories*, I:65–73. (cited in page 120.)

[NGMN Alliance, 2015] NGMN Alliance (2015). 5G White Paper. *NGMN 5G Initiative*. (cited in page 14.)

[NGMN Alliance, 2016] NGMN Alliance (2016). Description of Network Slicing Concept. *Deliverable P1 WS1*, pages 1–7. (cited in pages xiv, 17, and 18.)

[Nokia, 2016] Nokia (2016). Dynamic End-to-End Network Slicing for 5G. *White Paper*, pages 1–10. (cited in page 17.)

[Noroozi et al., 2019] Noroozi, K., Karimzadeh-Farshbafan, M., and Shah-Mansouri, V. (2019). Service Admission Control for 5G Mobile Networks with RAN and Core Slicing. In *Proc. IEEE GLOBECOM*, pages 6–11. IEEE. (cited in pages 25 and 97.)

[ONF, 2014] ONF (2014). SDN Architecture. *ONF Technical Recommendation TR-502*, (1). (cited in pages xiv, 14, and 15.)

[ONF, 2016] ONF (2016). Applying SDN Architecture to 5G Slicing. *ONF Technical Recommendation TR-526*, (1):1–19. (cited in page 14.)

[Orlandi et al., 2018] Orlandi, B., Kerboeuf, S., Faucheux, F., Lafragette, J.-L., Boubendir, A., and Luu, Q.-T. (2018). Network Slicing Made Easy! From Graph-based Design to Automated Deployment of Network Slices in 5G. In *Nokia 5G Smart Campus Event*, Nozay, France. (cited in pages 8 and 147.)

[Perrot et al., 2020] Perrot, N., Kerboeuf, S., and Luu, Q.-T. (2020). Virtual Network Orchestration Framework and Algorithms. *MAESTRO-5G Deliverable D3.1*. (cited in pages 8 and 147.)

[Qu et al., 2019] Qu, K., Zhuang, W., Ye, Q., Shen, X. S., Li, X., and Rao, J. (2019). Delay-Aware Flow Migration for Embedded Services in 5G Core Networks. In *Proc. ICC*, pages 1–6. (cited in page 119.)

[Richart et al., 2016] Richart, M., Baliosian, J., Serrat, J., and Gorricho, J. L. (2016). Resource Slicing in Virtual Wireless Networks: A Survey. *IEEE Trans. Netw. Service Manag.*, 13(3):462–476. (cited in pages 4 and 141.)

[Riera et al., 2016] Riera, J. F., Batalle, J., Bonnet, J., Dias, M., McGrath, M., Petralia, G., Liberati, F., Giuseppi, A., Pietrabissa, A., Ceselli, A., Petrini, A., Trubian, M., Papadimitrou, P., Dietrich, D., Ramos, A., Melian, J., Xilouris, G., Kourtis, A., Kourtis, T., and Markakis, E. K. (2016). TeNOR: Steps towards an orchestration platform for multi-PoP NFV deployment. In *IEEE NetSoft*, pages 243–250. (cited in pages 20, 36, and 37.)

[Riggio et al., 2016] Riggio, R., Bradai, A., Harutyunyan, D., Rasheed, T., and Ahmed, T. (2016). Scheduling Wireless Virtual Networks Functions. *IEEE Trans. Netw. Service Manag.*, 13(2):240–252. (cited in pages 20, 36, 37, 43, 45, 46, 47, 49, and 71.)

[Rost et al., 2017] Rost, P., Mannweiler, C., Michalopoulos, D. S., Sartori, C., Sciancalepore, V., Sastry, N., Holland, O., Tayade, S., Han, B., Bega, D., Aziz, D., and Bakker, H. (2017). Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks. In *IEEE Commun. Mag.*, volume 55, pages 72–79. (cited in pages 1 and 138.)

[Samdanis et al., 2016] Samdanis, K., Costa-perez, X., and Sciancalepore, V. (2016). From Network Sharing to Multi-Tenancy: The 5G Network Slice Broker. *IEEE Communications Magazine*, 54(7):32–39. (cited in page 27.)

[Savi et al., 2016] Savi, M., Tornatore, M., and Verticale, G. (2016). Impact of Processing-Resource Sharing on the Placement of Chained Virtual Network Functions. In *Proc. IEEE NFV-SDN*, pages 191–197. (cited in pages 32, 44, and 89.)

[Shi and Hou, 2007] Shi, Y. and Hou, Y. T. (2007). Approximation Algorithm for Base Station Placement in Wireless Sensor Networks. In *Proc. IEEE SeCON*, pages 512–519. (cited in page 52.)

[Srinivasan et al., 1989] Srinivasan, T. V., Vinrelette, C. J., and Dasgupta, D. (1989). Overlay Network Applications for Network Modernization and Positioning for the Future. In *Fourth IEEE Region 10 International Conference (TENCON)*, pages 306–309. (cited in pages 13 and 14.)

[Su et al., 2019] Su, R., Zhang, D., Venkatesan, R., Gong, Z., Li, C., Ding, F., Jiang, F., and Zhu, Z. (2019). Resource Allocation for Network Slicing in 5G Telecommunication Networks: A Survey of Principles and Models. *IEEE Netw.*, 33(6):172–179. (cited in pages 4, 73, and 142.)

[Sun et al., 2019] Sun, G., Xiong, K., Boateng, G. O., Ayepah-Mensah, D., Liu, G., and Jiang, W. (2019). Autonomous Resource Provisioning and Resource Customization for Mixed Traffics in Virtualized Radio Access Network. *IEEE Systems Journal*, 13(3):2454–2465. (cited in pages 24 and 26.)

[Sun et al., 2016] Sun, S., Rappaport, T. S., Rangan, S., Thomas, T. A., Ghosh, A., Kovacs, I. Z., Rodriguez, I., Koymen, O., Partyka, A., and Jarvelainen, J. (2016). Propagation Path Loss Models for 5G Urban Micro- and Macro-Cellular Scenarios. In *Proc. IEEE VTC*, pages 1–6. (cited in page 64.)

[Tajiki et al., 2018] Tajiki, M. M., Salsano, S., Chiaraviglio, L., Shojafar, M., and Akbari, B. (2018). Joint Energy Efficient and QoS-aware Path Allocation and VNF Placement for Service Function Chaining. *IEEE Trans. Netw. Service Manag.*, (July):1–20. (cited in pages 20 and 36.)

[Taleb et al., 2019] Taleb, T., Afolabi, I., Samdanis, K., and Yousaf, F. Z. (2019). On Multi-Domain Network Slicing Orchestration Architecture and Federated Resource Control. *IEEE Network*, 33(5):242–252. (cited in page 124.)

[Tan et al., 2011] Tan, J., Dube, P., Meng, X., and Zhang, L. (2011). Exploiting Resource Usage Patterns for Better Utilization Prediction. In *Proc. International Conference on Distributed Computing Systems*, pages 14–19. IEEE. (cited in page 99.)

[Teague et al., 2019] Teague, K., Abdel-Rahman, M. J., and Mackenzie, A. B. (2019). Joint Base Station Selection and Adaptive Slicing in Virtualized Wireless Networks: A Stochastic Optimization Framework. In *Proc. International Conference on Computing, Networking and Communications, (ICNC)*, pages 859–863. IEEE. (cited in page 22.)

[Teague et al., 2018] Teague, K. A., Mackenzie, A. B., Buehrer, R. M., and Dasilva, L. A. (2018). *Approaches to Joint Base Station Selection and Adaptive Slicing in Virtualized Wireless Networks Approaches to Joint Base Station Selection and Adaptive Slicing in*. PhD thesis, Virginia Polytechnic Institute and State University. (cited in pages 49 and 52.)

[Tran et al., 2017] Tran, T. X., Younis, A., and Pompili, D. (2017). Understanding the Computational Requirements of Virtualized Baseband Units Using a Programmable Cloud Radio Access Network Testbed. In *Proc. IEEE ICAC*, pages 221–226. (cited in page 30.)

[Trinh et al., 2011] Trinh, T., Esaki, H., and Aswakul, C. (2011). Quality of Service Using Careful Overbooking for Optimal Virtual Network Resource Allocation. In *Proc. ECTI*, pages 296–299. (cited in page 23.)

[Tse and Pramod, 2004] Tse, D. and Pramod, V. (2004). *Fundamentals of Wireless Communication*. (cited in page 63.)

[Umeyama, 1988] Umeyama, S. (1988). An Eigendecomposition Approach to Weighted Graph Matching Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703. (cited in page 122.)

[Vizarreta et al., 2017] Vizarreta, P., Condoluci, M., Machuca, C. M., Mahmoodi, T., and Kellerer, W. (2017). QoS-driven Function Placement Reducing Expenditures in NFV Deployments. In *Proc. IEEE ICC*. (cited in pages 20, 21, 36, 37, and 49.)

[Wang et al., 2019] Wang, G., Feng, G., Quek, T. Q. S., Qin, S., Wen, R., and Tan, W. (2019). Reconfiguration in Network Slicing-Optimizing the Profit and Performance. *IEEE Trans. Netw. Service Manag.*, 16(2):591–605. (cited in page 24.)

[Wang et al., 2017] Wang, G., Feng, G., Tan, W., Qin, S., Ruihan, W., and Sun, S. (2017). Resource Allocation for Network Slices in 5G with Network Resource Pricing. In *Proc. IEEE GLOBECOM*, pages 1–6. (cited in pages 4, 23, and 142.)

[Wang et al., 2009] Wang, J., Wright, K. L., and Gopalan, K. (2009). XenLoop: A Transparent High Performance Inter-VM Network Loopback. *Cluster Comput.*, 12(2 SPEC. ISS.):141–152. (cited in page 30.)

[Weldon, 2015] Weldon, M. K. (2015). *The Future X Network: A Bell Labs Perspective*. CRC Press. (cited in pages 1 and 138.)

[Wen et al., 2019] Wen, R., Feng, G., Tang, J., Quek, T. Q., Wang, G., Tan, W., and Qin, S. (2019). On Robustness of Network Slicing for Next-Generation Mobile Networks. *IEEE Trans. Commun.*, 67(1):430–444. (cited in pages 24 and 74.)

[Xiong et al., 2019] Xiong, K., Samuel Rene Adolphe, S., Boateng, G. O., Liu, G., and Sun, G. (2019). Dynamic Resource Provisioning and Resource Customization for Mixed Traffics in Virtualized Radio Access Network. *IEEE Access*, 7:115440–115453. (cited in page 26.)

[Yang et al., 2018] Yang, Y., Xu, J., Shi, G., and Wang, C.-X. (2018). *5G Wireless Systems*. (cited in pages xvii and 10.)

Appendix A

Synthèse

A.1 Contexte

La cinquième génération de réseau mobile (5G) offre aux opérateurs des opportunités uniques d'aborder de nouveaux modèles économiques pour les entreprises et le grand public. Les réseaux 5G ciblent différents secteurs industriels avec pour objectifs de faciliter la gestion, l'automatisation, la surveillance de processus, *etc*. Les services dédiés aux marchés verticaux, par exemple, l'énergie, l'e-santé, la ville intelligente, les voitures connectées, *etc*., seront plus facilement déployés [Li et al., 2017]. L'architecture 5G apporte la flexibilité requise pour prendre en charge de nombreux services avec différentes niveaux d'exigences en termes de latence, de débit et de disponibilité [Kaloxylos, 2018].

Pour augmenter cette flexibilité, les réseaux mobiles évoluent vers des systèmes constitués de ressources virtuelles qui peuvent être instanciées et libérées à la demande pour répondre aux demandes des clients. Des technologies telles que les réseaux définis par logiciels (RDL) et la virtualisation des fonctions réseau (VFR) jouent un rôle d'une importance croissante pour fournir une telle flexibilité aux réseaux mobiles [Basta et al., 2014].

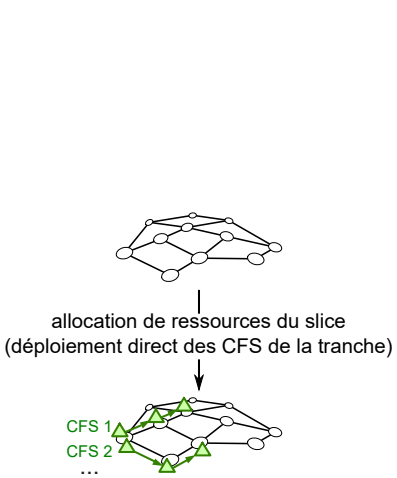
Tirant parti de technologies de RDL et de VFR, le découpage en tranches de réseau (*network slicing* en anglais) est apparu comme une technologie clé [5G Americas, 2016, IETF, 2017, Barakabitze et al., 2020]. Le découpage du réseau réduit les coûts globaux d'équipement et de gestion du réseau [Liang and Yu, 2014] tout en augmentant sa flexibilité d'exploitation [Rost et al., 2017]. Plusieurs réseaux virtuels ou tranches dédiés peuvent être gérés en parallèle sur une même infrastructure réseau. Avec le découpage en tranches du réseau, des marchés verticaux peuvent être abordés : les clients peuvent gérer leurs propres applications en exploitant des tranches adaptées à leurs besoins [GSM Alliance, 2017]. Comme l'indique [Weldon, 2015], l'industrie des réseaux a entamé une transformation massive vers la virtualisation des réseaux, comme en témoigne le nombre croissant de résultats publiés, de brevets déposés, les démonstrations, les preuves de concept, les essais sur le terrain et les accords commerciaux.

A.2 Réservation de Ressources pour des Tranches de Réseau

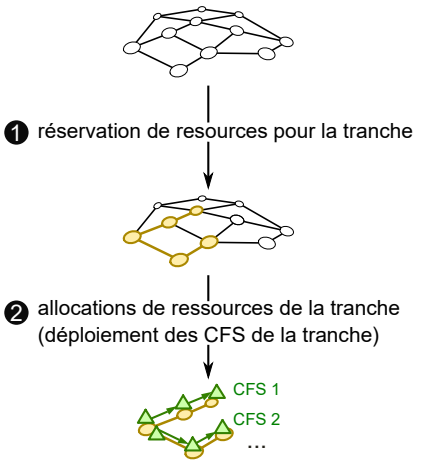
Dans les approches classiques de déploiement de tranches de réseau, les diverses chaînes de fonctions de service (CFS) d'une tranche de réseau sont déployées séquentiellement dans le réseau d'infrastructure. Dans cette thèse, nous proposons des solutions de réservation de ressources pour les tranches de réseau afin de satisfaire les demandes. Dans l'approche proposée, les ressources sont réservées à l'avance dans l'infrastructure pour le déploiement futur des CFS.

Une fois la réservation effectuée pour une tranche de réseau donnée, les CFS de cette tranche ont l'assurance de disposer des ressources réservées. Cela facilite la satisfaction des exigences en termes de qualité de service de la tranche de réseau. Les chapitres 5–8 de cette thèse montrent que les solutions de réservation proposées permettent de réduire les besoins de calcul nécessaires au déploiement des CFS.

La figure A.1 illustre l'approche d'intégration de CFS considérée dans l'état de l'art (figure A.1a), par rapport à l'approche proposée où les ressources nécessaires pour les tranches de réseau sont préalablement réservées (figure A.1b). La figure A.1b montre que le processus d'intégration (ou de déploiement) de CFS est divisé en deux phases : premièrement, la réservation des ressources est effectuée pour une tranche de réseau donnée et deuxièmement, les CFS de cette tranche sont déployées en exploitant les ressources réservées dans la première phase.



(a) Le déploiement direct des CFS.



(b) Déploiement des CFS après réservation de ressources.

Figure A.1: Illustration de (a) le déploiement direct des CFS, et (b) l'approche en deux phases proposée, où la réservation des ressources de tranches de réseau est effectué avant le déploiement des CFS.

Le point de vue d'un fournisseur d'infrastructure réseau (FIR) est adopté dans cette thèse. Plusieurs méthodes de réservation sont étudiées pour tenir compte

de divers cas d'usage. Nous proposons d'abord une méthode de réservation de ressources de tranches de réseau, répondant aux demandes de plusieurs tranches en termes de capacité de calcul, de mémoire et de ressouces radio. Nous étendons ensuite la méthode de réservation en considérant la situation où les tranches doivent être déployées sur différentes zones géographiques. Dans une telle situation, les contraintes associées à la couverture radio telles que des contraintes de débit minimum par utilisateur doivent être prises en compte. Enfin, la réservation de ressources de tranches et le contrôle d'admission sont combinés pour faire face (*i*) aux incertitudes liées aux demandes de ressources de tranches (par exemple, la fluctuation des demandes de ressources des utilisateurs, la variation temporelle du nombre d'utilisateurs d'une tranche, *etc.*) ; et (*ii*) la nature dynamique des demandes de tranches de réseau, c'est-à-dire les demandes d'activation et de desactivation des tranches.

L'approche proposée s'intègre dans la vision 3GPP dela gestion des tranches de réseau [3GPP, 2020]. Les méthodes de réservation des ressources de tranches proposées peuvent être réalisées dans la tâche de *préparation de l'environnement réseau* de la phase de préparation, voir la figure A.2. Au cours de cette phase, la conception et la planification de la capacité d'une tranche, l'intégration et l'évaluation des fonctions réseau requises et la réservation des ressources de l'infrastructure doivent être effectuées avant la création et l'activation des instances de nécessaires pour cette tranche. Les aspects de gestion du découpage du réseau sont considérés au chapitre 2.

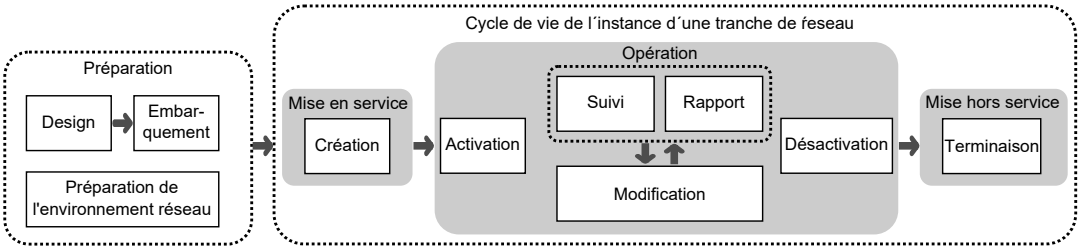


Figure A.2: La vision du 3GPP de la gestion des tranches de réseau [3GPP, 2020].

A.3 Défis de Recherche

L'un des principaux problèmes à résoudre dans ce contexte est de fournir à chaque tranche de réseau la bonne quantité de ressources physiques (calcul, mémoire et radio) afin de répondre aux demandes de ressources formulées afin de satisfaire aux exigences de service définies par l'opérateur de réseaux virtuels. La quantité de ressources réservées pour une tranche dépend des services qui y sont attachés, de leur niveau d'exigence de qualité de service (QdS) exprimé en termes de latence, de bande passante, de calcul et de stockage. Ces exigences dépendent de la demande de consommation de services dans la tranche. Différents types de tranches

de réseau peuvent coexister (par exemple, vidéo ultra-HD, e-santé, réseau de capteurs, systèmes de transport intelligents, jeux vidéos, internet tactile, *etc.*). Une meilleure identification des besoins des tranches de réseau facilite la réservation des ressources. Ceci correspond au défi 1 suivant.

> **Défi 1.** Une quantité suffisante de ressources doit réservée sur l'infrastructure réseau pour répondre aux besoins des tranches, afin de répondre aux exigences de service souhaitées. La quantité de ressources allouées à une tranche dépend des caractéristiques du service qu'il fournit, de ses exigences de QdS exprimées, par exemple, en termes de bande passante, de capacité de calcul, et de mémoire.

De nombreux défis restent à relever lorsque le découpage du réseau intègre la partie sans fil des réseaux 5G [Li et al., 2017, Kaloxylos, 2018], où l'accès radio doit être pris en compte. Par exemple, dans [Chatterjee et al., 2018], les caractéristiques de service requises par un fournisseur service (FS) sont : le débit minimum, la probabilité de couverture avec un débit minimum, la densité des équipements utilisateurs (EU) et la zone géographique à couvrir par la tranche. Dans cette thèse, nous abordons également le problème de réservation de ressources avec des contraintes de couverture. Le défi 2 résume ces problèmes.

> **Défi 2.** Lors du découpage en tranche du réseau d'accès radio, les contraintes liées à la couverture radio ainsi qu'à la localisation des utilisateurs doivent également être prises en compte.

Dans l'article [Barakabitze et al., 2020] sur le découpage en tranches du réseau 5G, les auteurs fournissent une taxonomie du découpage du réseau, des architectures et des défis futurs. L'une des questions ouvertes est de savoir comment répondre aux exigences de tranches de réseau des différents secteurs verticaux, où plusieurs segments de réseau, y compris l'accès radio, le transport et les réseaux cœurs, doivent être pris en compte. Le réseau d'infrastructure sur lequel les tranches sont exploitées doit supporter des services de haute qualité avec une consommation de ressources croissante (diffusion vidéo, téléprésence, réalité augmentée, exploitation de véhicules à distance, jeux, *etc.*). De plus, le nombre d'utilisateurs de chaque tranche, leur emplacement (généralement difficile à prévoir [Richart et al., 2016]) et les demandes de ressources peuvent fluctuer avec le temps. Ces incertitudes peuvent avoir un impact significatif sur les ressources consommées par chaque tranche de réseau et rendre le problème de réservation de ressources plus difficile. Les ressources d'infrastructure suffisantes doivent être dédiées à une tranche donnée pour garantir une qualité de service appropriée malgré les incertitudes concernant le nombre d'utilisateurs de la tranche et la demande de chauqe utilisateur. La surréservation doit également être évitée, afin de limiter les coûts de location des infrastructures et de laisser des ressources à des tranches concurrentes. Cette problématique est résumée dans le défi 3.

Défi 3. Un mécanisme efficace de réservation de ressources de tranches doit être robuste aux incertitudes liées aux demandes de utilisateurs du service fourni par cette tranche. De plus, la technique de réservation proposée doit être mise en œuvre de manière à limiter son impact sur les services d'arrière-plan de faible priorité, qui peuvent coexister avec les tranches de réseau sur le réseau d'infrastructure.

En plus du problème d'incertitude, il est également nécessaire de prendre en compte la nature dynamique des requêtes de tranche de réseau. Ces requêtes pour la création de tranches arrivent à des instants différents, avec des délais d'activation, des durées de vie et des demandes de ressources variables dans le temps. Ces paramètres ont un impact significatif sur les demandes de ressources globales des tranches de réseau. La variété des services proposés par les tranches de réseau induit des exigences de QdS très différentes [Li et al., 2018]. Dans les approches traditionnelles d'allocation de ressources de tranches [Huin et al., 2017, Wang et al., 2017, Su et al., 2019, Barakabitze et al., 2020], les ressources sont allouées juste avant leur activation. Avec une telle gestion *juste-à-temps*, il est difficile de garantir la disponibilité de ressources d'infrastructure au moment du déploiement et pendant la durée de vie d'une tranche. Dans ce cas, les demandes de tranches peuvent être rejetées. Par conséquent, une nouvelle approche de réservation des ressources de tranche doit être introduite, fournissant un contrôle d'admission *anticipé*. Les tranches sont admises, parfois largement avant leur d'activation, lorsque suffisamment de ressources d'infrastructure sont disponibles pour répondre à leurs exigences de qualité de service. Cette problématique est décrite dans le défi 4.

Défi 4. Les requêtes de réservation de resources pour les tranches de réseau doivent être traitées de manière anticipée, largement avant leur activation. Ceci permet de garantir la disponibilité des ressources de l'infrastructure au moment du déploiement et pendant la durée de vie des tranches de réseau. Le mécanisme de contrôle d'admission de tranche qui en résulte doit prendre en compte la nature dynamique des requêtes et le niveau de priorité des tranches de réseau.

A.4 Description de la Thèse

Cette thèse propose des techniques de réservation de ressources pour des tranches de réseau pour les systèmes de communication de 5ème génération et au-delà.

A.4.1 Plan de Thèse

La partie I présente des généralités sur le découpage d'un réseau en tranches et donne un aperçu des techniques de l'état de l'art. Les hypothèses considérées tout au long de la thèse sont également décrites. Les principales contributions de cette thèse sont présentées dans les parties II et III.

Partie I (*Background and Assumptions*) présente des généralités sur le découpage d'un réseau en tranches et met en présente les directions de recherche liées au découpage de réseau tout en donnant des éléments de l'état de l'art.

Chapitre 1 (*Introduction*) introduit le contexte général de cette thèse ainsi que ses contributions sur l'approche proposée de la réservation de ressources pour des tranches de réseau. Ce chapitre présent aussi les défis de recherche liés à l'approche proposée ;

Chapitre 2 (*Network Slicing in 5G*) présente un bref historique du découpage de réseau en tranches, met en évidence les principales technologies permettant une mise en oeuvre de cette technique comme les réseaux définis par logiciels et la virtualisation des fonctions réseau. Ce chapitre décrit également une architecture conceptuelle d'un système de découpage en tranche de réseau et aborde différents aspects tels que la gestion du cycle de vie des tranches de réseaux.

Chapitre 3 (*Related Works*) résume certaines études liées à divers aspects de la virtualisation et du découpage en tranches de réseau. Ce chapitre présent des éléments de l'état de l'art sur (*i*) l'intégration de CFS et l'allocation des ressources (*ii*) l'allocation des ressources avec des contraintes de couverture, (*iii*) l'allocation des ressources tenant compte d'incertitudes et (*iv*) l'allocation dynamique de ressources.

Chapitre 4 (*Hypotheses and Assumptions*) présente les notations et les hypothèses qui sont utilisées tout au long de la thèse. Un système typique de découpage de réseau en tranches est décrit, avec toutes les entités impliquées. La relation et les interactions entre ces entités, par exemple l'échange des caractéristiques de la demande de l'utilisateur, la demande de tranche de réseau et le service dédié, sont également détaillées.

Partie II (*Resource Provisioning for Deterministic Demands*) propose de nouvelles méthodes de réservation de ressources pour des tranches de réseau lorsque les demandes sont déterministes.

Chapitre 5 (*Resource Provisioning for the Core Network*) aborde le défi 1, dans lequel le problème de réservation de ressources de tranches dans le réseau cœur est pris en compte. Dans ce chapitre, les ressources disponibles dans l'infrastructure et les demandes de ressources pour des tranches de réseau sont considérées comme déterministes. La formation d'une demande de ressources et de la manière dont le problème de réservation de ces ressources sont ensuite décrites.

Chapitre 6 (*Coverage-Constrained Resource Provisioning*) aborde le défi 2. Il prolonge l'étude du chapitre 5 en considérant le problème de réservation conjoint des ressources du réseau cœur et RAN. Pour cela, la réservation de ressources consiste à trouver (*i*) un ensemble de Stations de Base (SB)

qui fournit des ressources radio suffisantes aux utilisateurs mobiles pour satisfaire les contraintes de couverture ; (*ii*) le placement des FRV sur les nœuds du centre de données ; et (*iii*) le routage des flux de données entre les FRV, tout en respectant la structure des CFS et en optimisant un objectif donné (par exemple, minimiser les coûts de l'infrastructure et de logiciel).

Partie III (*Resource Provisioning for Slice Requests with Uncertainties*) présente quelques méthodes de réservation de ressources pour des tranches de réseau avec des incertitudes et des demandes dynamiques.

Chapitre 7 (*Uncertainty-Aware Resource Provisioning*) étudie une méthode de réservation des ressources sur l'infrastructure pour les tranches de réseau, tout en étant robuste à un nombre partiellement inconnu d'utilisateurs de ces tranches, ce qui entraîne une incertitude sur l'utilisation des ressources de ces tranches. De plus, étant donné que certaines parties du réseau d'infrastructure sur lesquelles des tranches doivent être déployées sont souvent déjà utilisées par des services en arrière-plan, l'approche de réservation sera effectuée de manière à limiter son impact sur ces services. L'approche proposée dans ce chapitre est une réponse au défi 3 ;

Chapitre 8 (*Admission Control and Resource Provisioning for Prioritized Slice Requests with Uncertainties*) aborde le défi 4. Il prolonge l'étude présentée au chapitre 7 en considérant la réservation des ressources des tranches concurrentes. Il rend compte de la dynamique des requêtes de tranches, qui fait référence au fait que (*i*) la demande de ressources de ces tranches peut évoluer au cours de leur durée de vie, (*ii*) les demandes sont soumises à l'avance, et (*iii*) différentes requêtes de tranches de réseau peuvent être associées à différents niveau de priorité. Plusieurs stratégies de réservation à complexité réduite sont envisagées pour résoudre le problème de réservation de ressources pour des tranches de réseau, en tenant compte du cycle de vie des requêtes de tranches (c'est-à-dire le temps d'arrivée, d'activation et de départ), tout en étant robustes vis-à-vis des incertitudes sur les demandes de ressources. De plus, les méthodes proposées tiennent compte du niveau de priorité des tranches et fournissent un taux d'acceptation dépendant de la priorité des requêtes.

Chapitre 9 (*Conclusion and Perspective*) présente quelques conclusions et perspectives. Ce chapitre aborde certains aspects qui méritent plus de développements et quelques pistes de recherche potentielles.

A.4.2 Résumé des Contributions

Dans ce qui suit, nous passons brièvement en revue les réponses apportées dans cette thèse aux défis de recherche posés au paragraphe A.3.

Défi 1 Pour répondre au défi 1, cette thèse propose des solutions qui fournissent des ressources pour les tranches de réseau afin de répondre à leurs demandes. L'approche proposée va au-delà des approches précédentes de type *best-effort*, où les CFS d'une tranche sont déployés séquentiellement dans le réseau d'infrastructure. Avec l'approche proposée dans cette thèse, une fois que les ressources sont réservées pour une tranche donnée, les CFS de cette tranche sont assurées d'obtenir suffisamment de ressources pour fonctionner correctement. Cela facilite la satisfaction des exigences de service contractées avec la qualité souhaitée. De plus, les résultats numériques montrent que les solutions de réservation proposées permettent de réduire les besoins de calcul nécessaires pour déployer les CFS.

Dans notre approche de réservation, les demandes de ressources d'une tranche donnée agrègent les demandes de ressources des utilisateurs du service proposé par cette tranche. Les demandes de ressources globales d'une tranche sont indiquées dans le contrat (C) entre l'opérateur de réseau virtuel (ORV) et le FIR (C-ORV-FIR). Le C-ORV-FIR peut également inclure d'autres contraintes requises, par exemple, la probabilité réussite de la réservation lors de la prise en compte des incertitudes des demandes de ressources. Lors de l'exécution de la réservation des ressources pour la tranche de réseau, le C-ORV-FIR doit être satisfait, garantissant ainsi que suffisamment de ressources d'infrastructure sont réservées.

Défi 2 Ce défi est abordé au chapitre 6. Ce chapitre examine le problème conjoint de réservation des ressources du réseau cœur et du réseau d'accès radio, en tenant compte de contraintes de couverture. Pour résoudre le problème de la localisation de l'utilisateur (inconnu lors de la phase de réservation des ressources), nous avons adopté une approche de partitionnement en sous-zones de la zone à couvrir. Au lieu de fournir des blocs radio (BR) aux utilisateurs, on essaie de fournir des BR à chaque sous-zone. Plusieurs contraintes supplémentaires ont été présentées pour satisfaire les exigences de couverture : une contrainte pour s'assurer que les BR réservés ne dépassent pas la capacité des têtes radio ; une contrainte pour satisfaire la demande moyenne minimale des utilisateurs et la demande totale de ressources radio par tranche pour le trafic de liaison montante (*uplink*) et de liaison descendante (*downlink*) ; et enfin une contrainte pour assurer la proportionnalité entre les ressources radio réservées pour la liaison montante et la liaison descendante. Ces contraintes supplémentaires conduisent à un problème d'optimisation complexe lorsque l'on considère le problème de réservation conjoint des ressources radio et cœur. Le problème de réservation conjoint (appelé *réservation en une étape*) devient insoluble lorsque le nombre de tranches augmente. Pour faire face à ce problème, nous avons introduit une approche alternative (appelée *réservation en deux étapes*), dans laquelle la réservation des ressources radio et des ressources du réseau cœur sont effectués séquentiellement. Nous avons montré que cette approche a une complexité inférieure à celle de l'approche conjointe, tout en offrant de bonnes performances en termes de coût de la réservation et d'efficacité d'utilisation des ressources de l'infrastructure.

Défi 3 Comme indiqué à chapitre 7, la dynamique des trafics dans les tranches (arrivées/départs des flux), ainsi que la disponibilité des ressources de l'infrastructure, peuvent conduire à une QdS offerte par la tranche inférieure au niveau attendu par le fournisseur de services gérant la tranche. L'approche traditionnelle, dans laquelle les ressources allouées/réservées sont adaptées aux demandes crêtes, peut conduire à une *surallocation* des ressources, diminuant ainsi l'efficacité de l'utilisation des ressources de l'infrastructure.

Dans le chapitre 7, une méthode de réservation des ressources de tranches robuste au caractère aléatoire des demandes de ressources a été proposée. Le caractère aléatoire est dû à un nombre en partie inconnu d'utilisateurs avec une utilisation aléatoire des ressources de tranche. La robustesse est obtenue en fournissant une garantie probabiliste que la quantité de ressources réseau réservée pour une tranche satisfera aux exigences de cette tranche. La méthode proposée essaie également de maintenir l'impact de la réservation de ressources sur les services en arrière-plan (qui varient également dans le temps) à un niveau prescrit.

Défi 4 Ce défi est abordé au chapitre 8. Les demandes d'une tranche sont caractérisées par des délais variables entre leur soumission et le temps d'activation et par différents niveaux de priorité (par exemple, *Premium* et *Standard*). Nous avons conçu un mécanisme de contrôle d'admission des tranches et de réservation des ressources priorisé. Les décisions d'admission sont fournies et les ressources nécessaires aux tranches admises sont réservées avec un délai de réponse dépendant de la priorité des tranches et du temps restant avant leur activation. De plus, différentes stratégies de traitement ont été proposées, dont chacune a un impact différent sur le traitement des requêtes de tranches selon différents niveaux de priorité.

Les résultats numériques montrent que la proportion de tranches admises peut être ajustée efficacement en fonction de la différence de délai de traitement entre les tranches Premium et Standard. Lorsque la différence de délai augmente, les demandes de tranches Premium sont accordées beaucoup plus fréquemment, avec moins d'ajustements au fil du temps dans le schéma de réservation. Cela impacte directement les coûts de réservation, qui sont réduits pour les tranches Premium par rapport aux tranches Standard lorsque la différence de délai est importante.

A.4.3 Publications

Les résultats de cette thèse ont fait l'objet des publications suivantes.

Revues

(R_1) [Luu et al., 2021a] Q.-T. Luu, S. Kerboeuf, and M. Kieffer, "Admission Control and Resource Provisioning for Prioritized Slice Requests with Uncertainties," *soumis à IEEE Transactions on Network and Service Management*, 2021.

(R_2) [Luu et al., 2021c] Q.-T. Luu, S. Kerboeuf, and M. Kieffer, "Uncertainty-Aware Resource Provisioning for Network Slicing," in *IEEE Transactions on Network*

and Service Management, vol. 18, no. 1, pp. 79-93, Mar. 2021.

(R_3) [Luu et al., 2020a] Q.-T. Luu, M. Kieffer, A. Mouradian, and S. Kerboeuf, "Coverage-Aware Resource Provisioning Method for Network Slicing," in *IEEE/ ACM Transactions on Networking*, vol. 28, no. 6, pp. 2393-2406, Dec. 2020

Conférences/Workshops

(C_1) [Luu et al., 2021b] Q.-T. Luu, S. Kerboeuf, and M. Kieffer, "Foresighted Resource Provisioning for Network Slicing," *IEEE International Conference on High Performance Switching and Routing (HPSR)*, Paris, June 2021, pp. 1-8.

(C_2) [Luu et al., 2020b] Q.-T. Luu, S. Kerboeuf, A. Mouradian, and M. Kieffer, "Radio Resource Provisioning for Network Slicing with Coverage Constraints," in *Proc. IEEE International Conference on Communications (ICC)*, Dublin, Ireland, June, 2020, pp. 1-6.

(C_3) [Luu et al., 2018] Q.-T. Luu, M. Kieffer, and A. Mouradian, and S. Kerboeuf, "Aggregated Resource Provisioning for Network Slices," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, Dec. 2018, pp. 1-6.

Livrables de projet

(L_1) [Perrot et al., 2020] N. Perrot, M. Antonia, S. Kerboeuf, Q.-T. Luu, *et al.*, "Virtual Network Orchestration Framework and Algorithms," *ANR MAESTRO-5G Project Deliverable D3.1*, 2020.

Démonstrations

(D_1) [Orlandi et al., 2018] B. Orlandi, S. Kerboeuf, F. Faucheux, J.-L. Lafragette, A. Boubendir, and Q.-T. Luu, "Network Slicing Made Easy! From Graph-based Design to Automated Deployment of Network Slices in 5G," *Nokia 5G Smart Campus Event*, Nozay, 2018 (en collaboration avec Orange Labs).

Séminaires

(S_1) [Luu et al., 2020c] Q.-T. Luu, S. Kerboeuf, A. Mouradian, and M. Kieffer, "Resource Provisioning for Network Slices with Coverage Constraints," *ANR MAESTRO-5G Workshop on Orchestration of 5G Networks and Beyond*, CentraleSupélec, Gif-sur-Yvette, Dec. 2020.

Brevets

(B_1) [Kerboeuf et al., 2018] S. Kerboeuf, Q.-T. Luu, M. Kieffer, and A. Mouradian, Method and Apparatus for Mapping Network Slices Onto Network Infrastructures With SLA Guarantee, *WIPO Patent No. WO202011460 8A1* (déposé le 7 décembre, 2018 par Nokia Solutions and Networks).

Appendix B

Index

Appendix C

About the Author

Quang-Trung Luu

L2S, CentraleSupélec, Université Paris-Saclay

Email: luuquangtrung.vn@gmail.com • Personal page: luuquangtrung.github.io

Education

2017–2021	Ph.D. in Networking and Telecommunications *CentraleSupélec, Université Paris-Saclay*, France
2016–2017	M.Sc. in Multimedia Networking *Télécom Paris & Université Paris-Saclay*, France
2015–2016	M.Sc. in Devices and Antennas for Telecommunications, *Université Paris-Saclay*, France
2008–2013	Eng. Dipl. in Electronics and Telecommunications *Hanoi University of Science and Technology*, Vietnam

Experiences

2021–2022	Postdoctoral Fellow, *Univ. of Avignon & Univ. of Toulouse*, France
2017–2020	Research Engineer, *Nokia Bell Labs*, Nozay, France
2017	Intern, *Inria & Ecole Normale Supérieure*, Lyon, France
2016	Intern, *GeePS, CentraleSupélec*, Gif-sur-Yvette, France
2015	Engineer, *Samsung Vietnam Mobile R&D Center*, Hanoi, Vietnam
2013–2014	Engineer, *Viettel Network*, Hanoi, Vietnam

Awards and Honors

Oct. 2020	ENSA Publication Award, by Nokia Bell Labs
July 2019	Nokia France Student Award (finalist), by Nokia France
Dec. 2018	Student Travel Grant, by IEEE Communications Society
2017–2020	CIFRE Fellowship, by French National Association for Technical Research
2015–2016	International Master's Scholarship, by Université Paris-Saclay
May 2013	Student Research Prize (first runner-up), by Hanoi University of Science and Technology

Personal

Born in 1990, Lâm Thao, Phu Tho, Vietnam

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Contrôle et optimisation des réseaux virtuels sans fil

Mots clés: tranche de réseau, réservation de ressources, allocation de ressources, virtualisation des réseaux sans fil, 5G, contrôle d'admission de tranches

Résumé: Le découpage du réseau est une technologie clé des réseaux 5G, grâce à laquelle les opérateurs de réseaux mobiles peuvent créer des tranches de réseau indépendantes. Chaque tranche permet à des fournisseurs d'offrir des services personnalisés. Comme les tranches sont opérées sur une infrastructure de réseau commune gérée par un fournisseur d'infrastructure, il est essentiel de développer des méthodes de partage efficace des ressources.

Cette thèse adopte le point de vue du fournisseur d'infrastructure et propose plusieurs méthodes de réservation de ressources pour les tranches de réseau. Actuellement, les chaines de fonctions appartenant à une tranche sont déployées séquentiellement sur l'infrastructure, sans avoir de garantie quant à la disponibilité des ressources. Afin d'aller au-delà de cette approche, nous considérons dans cette thèse des approches de réservation des ressources pour les tranches en considérant les besoins agrégés des chaines de fonctions avant le déploiement effectif des chaines de fonctions. Lorsque la réservation a abouti, les chaines de fonctions ont l'assurance de disposer de suffisamment de ressources lors de leur déploiement et de leur mise en service afin de satisfaire les exigences de qualité de service de la tranche. La réservation de ressources permet également d'accélérer la phase d'allocation de ressources des chaines de fonctions.

Title: Dynamic control and optimization of wireless virtual networks

Keywords: network slicing, resource provisioning, resource allocation, wireless network virtualization, 5G, slice admission control

Abstract: Network slicing is a key enabler for 5G networks. With network slicing, Mobile Network Operators (MNO) create various slices for Service Providers (SP) to accommodate customized services. As network slices are operated on a common network infrastructure owned by some Infrastructure Provider (InP), efficiently sharing the resources across various slices is very important.

In this thesis, taking the InP perspective, we propose several methods for provisioning resources for network slices. Previous best-effort approaches deploy the various Service Function Chains (SFCs) of a given slice sequentially in the infrastructure network. In this thesis, we provision aggregate resources to accommodate slice demands. Once provisioning is successful, the SFCs of the slice are ensured to get enough resources to be properly operated. This facilitates the satisfaction of the slice quality of service requirements. The proposed provisioning solutions also yield a reduction of the computational resources needed to deploy the SFCs.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France