Converged RAN/MEC Slicing in Beyond 5G (B5G) Networks

UPC

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

PhD Research Thesis

PhD Candidate:
Behnam Ojaghi

Supervisors:
Prof. Dr. Ferran Adelantado
Universitat Oberta de Catalunya (UOC)
Prof. Dr. Christos Verikoukis
University of Patras and ISI/ATHENA

Tutor:
Prof. Dr. Luis Alonso
Universitat Politècnica de Catalunya (UPC)

Department of Signal Theory and Communications
School of Telecommunications Engineering

May 2023

MARIE CURIE ACTIONS

Abstract

Mobile communication technologies are evolving towards Beyond Fifth-Generation (B5G) wireless communication networks, aiming to fulfill the new use cases with ever-increasing demands with diverse Quality-of-Service (QoS) requirements. The diversity of performance requirements, ranging from 1 Gbps peak rate to 1ms end-to-end latency or some thousands of low-power devices per sector, renders a classical network architecture unfeasible for new services. This has raised the need for novel mobile network architecture. In that sense, 5G and B5G have been conceived as softwarized networks able to provide service-tailored connectivity by using the principle of network softwarization such as network slicing, virtualization, and edge computing to serve efficiently the diversified service requirements of 5G and B5G networks. Multi-access Edge Computing (MEC) and Radio Access Network (RAN) slicing are considered vital mechanisms of upcoming B5G systems as they allow the creation of end-to-end isolated RAN slices and increase of serving mobile data traffic on edge. The efficient deployment of RAN functions promises flexible and disaggregated architectures for the diversified needs of B5G networks. Based on the current challenges and expected future requirements, the main objective of this thesis is to propose solutions for implementing dynamic RAN slicing and Functional Split (FS) along with MEC placements in 5G/B5G. In particular, this thesis is divided into three parts. In the first part (**Chapter 3**), we model a joint slicing and FS optimization in the 5G RAN with the objectives of optimizing the centralization degree and throughput destined to tackle the aforementioned challenges. In this work, the RAN slicing allowed a customized FS deployment per slice, thus optimizing the available resources, *e.g.,* transport network capacity and Remote Radio Head (RRH) or Central Unit (CU) computational capacity.

Next, we present the second part in **Chapter 4** by extending the first work by proposing SlicedRAN: service-aware network slicing framework for 5G RAN to create isolated RAN slices based on the service requirements with customized functional splits per slice. The proposed framework investigates the bottlenecks in the capacity of RRHs, Fronthaul/Backhaul (FH/BH) network capacity along with a minimum level of Service Level Agreement (SLA) for each slice imposed by the different service types. The broad implication of the present research demonstrates a strong trade-off between SLA and the FH/BH network between CU and RRHs which provide a basis for designing a virtualized network infrastructure with a cost-efficient FH/BH network whilst guaranteeing SLA of different slices.

Finally, in the last part presented in **Chapter 5**, we investigate dynamic RAN/MEC slicing framework in Open-RAN (O-RAN) architecture to dynamically place the RAN protocol stack of Virtual Network Functions (VNFs) and MEC server per slice. This framework contains the bottlenecks in the capacity of Open-RAN Radio Units (O-RUs), MEC server computation capacity, together with a customized FS per slice, to jointly solve the challenge of operating cost-efficient edge networks and maintaining the served traffic with various QoS criteria. We use a robust Benders decomposition algorithm, which reduces the computation complexity while ensuring an exact and optimal global solution. The proposed algorithm successfully optimizes the joint throughput and system cost in various traffic scenarios while satisfying QoS criteria, as shown by trace-driven simulation results.

Hence, in order to determine the right MEC settings for on-demand traffic and alter the MEC type to satisfy the QoS requirements of various User Equipment (UEs) belonging to different slice types, we explore the compute and storage capacity for MEC services such as Enhanced Mobile Broadband (eMBB) and ultra-Reliable and Low-Latency Communications (uRLLC). The overall conclusion of the present findings demonstrates a trade-off between the throughput attained and the cost incurred to the network.

As a result, we investigate multi-objective optimization to construct slices while optimizing throughput and decreasing computational cost objectives, and we compare its performance to that of a single objective (maximizing throughput). The findings demonstrate that a throughput increase of up to 160% can be made possible by increasing 78% in the computation cost for a single objective when compared with multi-objective without prioritization. In addition, comparing a single objective with a multi-objective with priority in throughput, it increases throughput by up to 82% and adds 17% to computation costs. Consequently, a single objective of maximizing throughput can result in high throughput at the expense of high cost. It is possible to achieve almost half the amount of throughput using multi-objective with prioritization in throughput, whereas costs can be reduced five-fold.

Resumen

Las tecnologías de comunicación móvil están evolucionando hacia las redes de comunicaciones inalámbricas *Beyond Fifth-Generation (B5G)*, con el objetivo de satisfacer los nuevos casos de uso con demandas cada vez mayores y con diversos Requisitos de Calidad de Servicio (QoS). La diversidad de requisitos de rendimiento, que van desde velocidad máxima de 1 Gbps a latencia de extremo a extremo de 1 ms o algunos miles de dispositivos de bajo consumo por sector, hace inviable una arquitecturas de red clásica para nuevos servicios. Esto ha planteado la necesidad de nuevas arquitectura de red. En ese sentido, 5G y B5G han sido concebidas como redes baasadas en software (*"softwarized networks"*) capaces de proporcionar conectividad adaptada al servicio mediante el uso de principios de las redes basadas en software tales como *"network slicing"*, virtualización y *"edge computing"* para servir eficientemente la diversidad de requisitos de los servoicios de las redes 5G y B5G. Tanto *Multi-access Edge Computing (MEC)* como slicing de la Red de Acceso (*Radio Access Network -RAN- slicing*) se consideran mecanismos vitales de los próximos sistemas B5G, ya que permiten la creación de segmentos de la red de acceso aislados de extremo a extremo y el aumento del servicio de tráfico de datos móviles en el borde de la red (*"network edge"*). El despliegue eficiente de funciones RAN permite arquitecturas flexibles y desagregadas para responder a las distintas necesidades de las redes B5G. De acuerdo con los desafíos actuales y los requisitos futuros esperados, el principal objetivo de esta tesis es proponer soluciones para implementar RAN slicing dinámico y División funcional (FS) junto con ubicaciones MEC en 5G/B5G. En particular, esta tesis se divide en tres partes. En la primera parte (**Capítulo 3**), modelamos la optimización conjunta de slicing y FS en 5G RAN con los objetivos de optimizar el grado de centralización y el rendimiento necesario para abordar los retos antes mencionados. En este trabajo, *RAN slicing* permitió una implementación personalizada de FS por slice, optimizando así los recursos disponibles, por ejemplo, la capacidad de la red de transporte y de los *Remote Radio Heads (RRH)* o de la Unidad Central (CU).

A continuación, presentamos la segunda parte en el **Capítulo 4**, que es una extensión del primer trabajo y propone SlicedRAN, un marco para el slicing de redes de acceso 5G basado en los servicios, que crea FSs personalizados para cada slice. El marco propuesto investiga la cuellos de botella en la capacidad de los RRH, la capacidad de la red Fronthaul/Backhaul (FH/BH) junto con una nivel mínimo de *Service Level Agreement (SLA)* para cada slice. Las amplias implicaciones de la presente investigación muestran un fuerte compromiso entre SLA y la red FH/BH entre CU y RRH que proporciona una base para diseñar una infraestrcutura de red virtualizada con una red FH/BH rentable al mismo tiempo que garantiza el SLA de diferentes slices.

Finalmente, en la última parte presentada en el **Capítulo 5**, investigamos el marco para el *RAN/MEC slicing* dinámico en la arquitectura Open-RAN (O-RAN) y así colocar dinámicamente la pila de protocolos RAN de Funciones de Red Virtuales (VNF) y el servidor MEC por slice. Este marco contiene los cuellos de botella en la capacidad de las unidades de radio Open-RAN (O-RU), la capacidad de cómputo del servidor MEC, y el FS personalizado por slice, y permite resolver conjuntamente un doble desafío: operar la red de forma eficiente en términos de coste; y mantener el tráfico servido con distintos criterios de calidad de servicio. Utilizamos un algoritmo robusto de descomposición de Benders, que reduce la complejidad de cálculo al tiempo que garantiza una solución global exacta y óptima. Él algoritmo propuesto optimiza con éxito el rendimiento conjunto y el coste del sistema para varios tipos de escenarios de tráfico mientras se satisfacen los criterios de QoS, tal y como se muestra en los resultados de la simulación.

Por lo tanto, para determinar la configuración correcta de MEC para el tráfico bajo demanda y modificar el tipo de MEC para satisfacer los requisitos de QoS de varios Equipos de Usuario (UE) pertenecientes a diferentes slices, exploramos la capacidad de cómputo y almacenamiento para varios servicios MEC. La conclusión general de los hallazgos demuestra compromiso entre el rendimiento obtenido y el coste incurrido por la red.

Como resultado, investigamos la optimización multiobjetivo para construir slices mientras optimizamos el rendimiento y disminuimos los objetivos de coste, y comparamos su rendimiento con el de un solo objetivo (maximizar el rendimiento). Los hallazgos demuestran que, cuando las prioridades para los dos objetivos son iguales, se puede lograr un aumento del rendimiento de hasta un 160 % aumentando el coste de computación en un 78%. Además, cuando se prioriza el rendimiento, el aumento en el rendimiento es de aproximadamente un 82%, mientras que el coste de cómputo aumenta un 17%.

Acknowledgements

I, *Behnam Ojaghi* declare that this thesis titled "Converged RAN/MEC Slicing in Beyond 5G (B5G) Networks" and the work presented in it is my own and has been generated by me as a result of my own original research. "No pain, no gain"; the pain of difficulties comes with the respective gains based on the effort that has been done to achieve its goals. Fortunately, this will be the last one, as I could achieve not only the highest Education degree in Telecommunication Engineering but also some of my wishlists with additional unexpected good moments. This journey could not be possible without the support and help of so many people. First of all, I would like to express my special thanks to my supervisors, Dr. Ferran Adelantado and Dr. Christos Verikoukis, for giving me a chance to join a prestigious ITN Marie Skłodowska-Curie Fellowship program, SPOTLIGHT project, take this challenging experience, and for their support, guidance, criticisms, and comments to conduct my research and end up with PhD work.

During my PhD, I have had a collaboration with my fellow project team-mates in UOC, CTTC, and Iquadrat, Dr. Elli Kartsakli (currently in BSC), Dr. Angelos Antonopoulos (currently in Nearby Computing S.L.), for the assistance they provided me since I first started my studies, for their ongoing support, and the brainstorming meetings, whose collaboration was important in accomplishing this journey. Special thanks to Dr. Angelos Antonopoulos's kind comments and fruitful discussions.

I also want to express my gratitude to my former advisors Bulent Tavli and Erdogan Dogdu from TOBB ETU. Indeed, they were the ones who helped me take my initial steps in research and encouraged me to embark on a career in academia. Last but not least appreciations go to my family, and friends. My friends Santiago, Maria, Mohammad, and my parents, my wife, and my son, Ayhan.

Table of Contents

List of Figures

List of Tables

List of Acronyms

List of frequently used Acronyms

3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation Network
API	Application Programming Interface
AR	Augmented Reality
ARQ	Automatic Repeat Request
B5G	Beyond 5G
BBU	BaseBand Unit
BS	Base Station
BH	BackHaul
COMP	Coordinated MultiPoint
COTS	Commercial-Off-The-Shelf
CP	Control Plane
CPRI	Common Public Radio Interface
C-RAN	Cloud RAN
CU	Central Unit
DP	Data Plane
D-RAN	Distributed RAN
DU	Distributed Unit
eNB	evolved Node B
EPC	Evolved Packet Core

eMBB	Enhanced Mobile Broadband
ETSI	European Telecommunications Standards Institute
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
FFT	Fast Fourier Transform
FH	FrontHaul
FPGA	Field-Programmable Gate Array
FS	Functional Split
HARQ	Hybrid Automatic Repeat Request
HD	High Definition
gNB	Next Generation Node B
GPU	Graphics Processing Unit
IFFT	Inverse Fast Fourier Transform
IoT	Internet of Things
IP	Internet Protocol
ITU	The International Telecommunication Union
OBASI	Open Base Station Architecture Initiative
O-CU	Open-Central Unit
O-DU	Open-Distributed Unit
O-RAN	Open-Radio Access Network
O-RU	Open-Remote Unit
LTE	Long-Term Evolution
MAC	Medium Access Control
MEC	Multi-access Edge Computing
mMTC	massive Machine Type Communications
MNO	Mobile Network Operator
NFV	Network Function Virtualization
NG	Next Generation Network
NR	New Radio

PDCP	Packet Data Convergence Protocol
PDN	Packet Data Network
PHY	Physical Layer
QoS	Quality-of-Service
RAN	Radio Access Network
RAT	Radio Access Technology
RF	Radio Frequency
RLC	Radio Link Control
RRH	Remote Radio Head
RRM	Radio Resource Management
RU	Radio Unit
SDN	Software Defined Networking
SLA	Service Level Agreement
UE	User Equipment
UP	User Plane
uRLLC	ultra-Reliable and Low-Latency Communications
V2X	Vehicle-to-everything
VNF	Virtual Network Function
V-RAN	virtualized RAN
VR	Virtual Reality

Chapter 1

Introduction

This chapter introduces the motivation underneath our thesis proposal in Section 1.1. The main challenges and the contributions are presented in Section 1.2. Next, the structure and outline of the thesis are described in Section 1.3. Finally, our contributions are presented at the end.

1.1 Motivation

The mobile communication technologies are evolving towards Beyond Fifth-Generation (B5G) of wireless communication networks, aiming to fulfill the new use cases with ever-increasing demands for higher data rates and with diverse Quality-of-Service (QoS) requirements. The unprecedented surge in data traffic experienced over the last decade has stretched telecommunications networks to their capacity. As the mobile data traffic grows along with the quantity and diversity of offered services, it becomes increasingly important to understand the demands and the respective complexity generated by them. According to Cisco's report [1], global IP traffic has increased 127-fold from 2005 to 2021, and it is expected that the total mobile network traffic will exceed 368EB per month in 2027 [2]. It is precisely in this context of traffic explosion in which the requirements for B5G have been defined [3]. In a nutshell, very high data rates (*i.e.,* aggregate data rate, edge data rate or peak data rate), extensive coverage and massive connectivity, sub-ms round-trip time delays, a reduction of the cost and energy consumption are just a few of the performance metrics that 5G and B5G networks are expected to support [4]. The diversity of performance requirements, ranging from 1 Gbps peak data rate to 1ms end-to-end latency or some thousands of low power devices per sector, renders a classical Fourth-Generation (4G) network architecture unfeasible for 5G. The International Telecommunication Union (ITU) classifies 5G mobile network services into three main types [5, 6]: i) Enhanced Mobile Broadband (eMBB) refers to the services with high-bandwidth requirements (*e.g.,* High Definition (HD) and 3D videos), ii) ultra-Reliable and Low-Latency Communications (uRLLC) is for the services demanding low-latency and high-reliability (*e.g.,* Vehicle-to-everything (V2X) and automated driving), and iii) massive Machine Type Communications (mMTC) is for the services that demand high-connection density yet with relaxed latency and throughput requirements (*e.g.,* smart city). Given that each 5G service may need different requirements as shown in Fig. 1.1, the traditional *one-size-fits-all* approach to mobile network architecture, equivalently 4G, is ineffectual to handle the complex services with different QoS requirements [7].

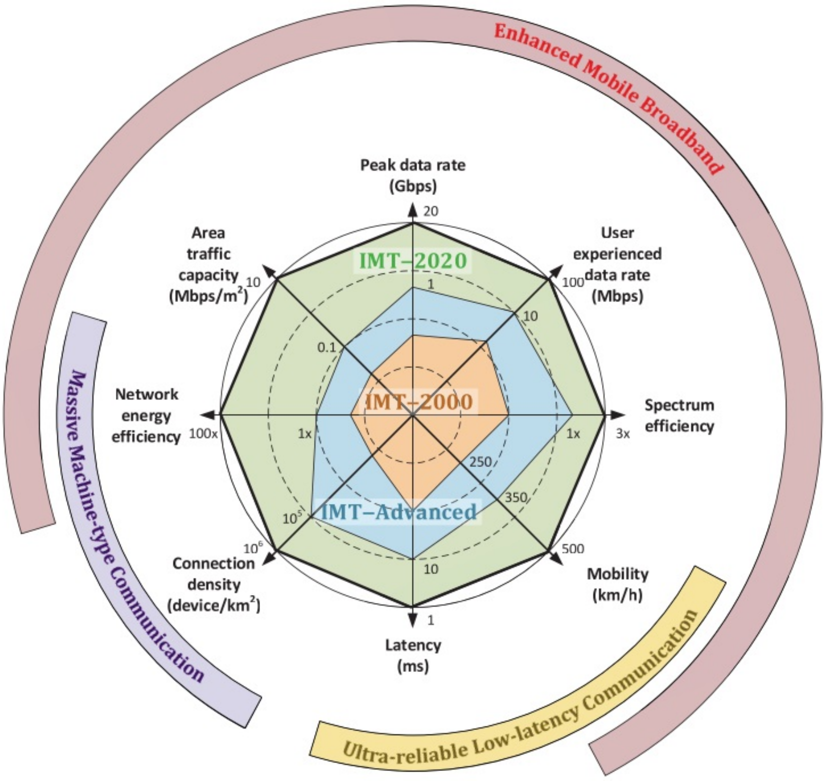


Figure 1.1: 5G requirements [6].

Fig. 1.2 provides a comparison of the performance characteristics and technical specifications of 4G and 5G technology.

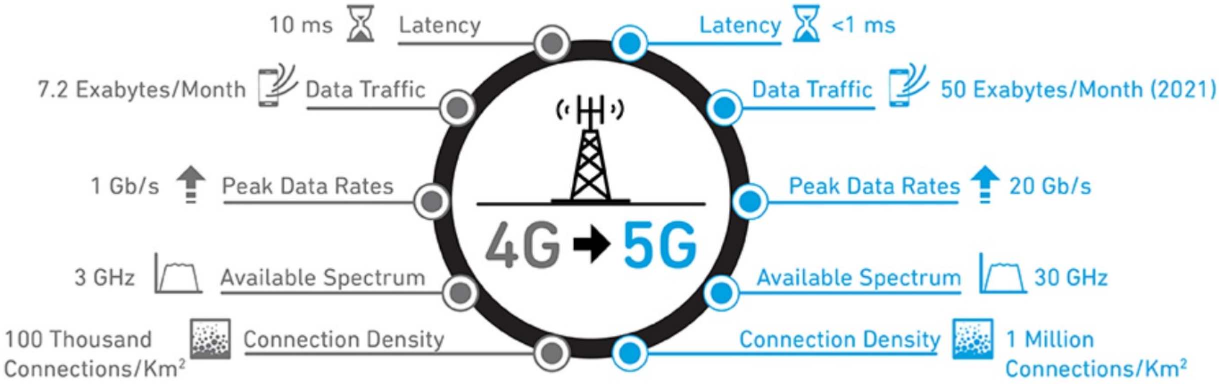


Figure 1.2: 4G vs 5G Comparisons [8]

To address these complexities, various technologies are emerging. First, the European Telecommunications Standards Institute (ETSI) aims for dynamic network management and service provisioning with latency reduction via bringing cloud-computing capabilities into the edge side of mobile networks and within the Radio Access Network (RAN), that is known as Multi-access Edge Computing (MEC) platform [9, 10]. MEC provides computing resources at the edge, enabling faster and more efficient data processing and reducing latency to support end-users with critical applications [3].

Second, network slicing enables the deployment of several logical networks on a common physical infrastructure to achieve the QoS required by different services. The slicing process includes the 5G RAN, 5G core, and 5G transport network. In particular, RAN slicing emerges as an essential component at the network edge, built upon a single rigid mobile network infrastructure that creates on-demand isolated slices on top of the physical network, thus enabling dynamic use of RAN resources and opening up the potential for different types of supported services such as eMBB, uRLLC, and mMTC services [11].

Thus, 5G and B5G networks are required to re-design current network functionalities, able to provide service-tailored connectivity, by leveraging network virtualization techniques based on Software Defined Networking (SDN) and Network Function Virtualization (NFV) [12]. Specifically, the aforementioned softwarization process is aimed to provide the network with enough flexibility to create isolated slices on-demand on top of the physical network.

This is particularly relevant in 5G, where the RAN node, known as gNB, has been split up into three units: the Centralized Unit (CU), the Distributed Unit (DU) and the Radio Unit (RU), connected through a packet-based integrated FrontHaul/BackHaul (FH/BH), also known as *cross-haul* network. The whole protocol stack is thus partially allocated in the CU, and partially allocated in the DU and RU [13], however, the optimal distribution of functions among gNB units (known as *functional split*) has attracted the attention of the research community, which still remains as an open research problem [14–17].

1.2 Main Challenges and Contributions

Recapping the detailed discussion of challenges mentioned in 1.1, in this thesis study, we focus on the challenges in the RAN domain, where RAN slicing process is required to adapt the RAN architecture to support 5G users' performance requirements. Remarkably, the main challenges of RAN slicing are efficient spectrum sharing, the algorithmic aspects of resource allocation, and creating and managing several slices on the same shared infrastructure in an efficient and isolated manner. The isolation of slices should have minimal impact on the services of this slice or other slices, *i.e.,* guaranteeing the agreed Service Level Agreement (SLA) for each slice. Hence, two slices are isolated as long as the actions performed on one slice do not result in an SLA violation on the other slice.

The combination of those challenges mentioned in 1.1 and this section lead to the motivation of the current thesis that addresses the need for joint RAN slicing and FS optimization solutions at gNB along with MEC placements solutions for efficient management of radio resources shared among different Mobile Network Operators (MNOs) and guaranteeing agreed SLAs between distinct types of services.

This thesis attempts to fill the gap in the literature regarding the study of joint RAN slicing and FS optimization solutions and MEC placement per slice in different network scenarios. A dynamic RAN slicing framework is presented to dynamically place the RAN protocol stack functions and MEC server and allocate radio spectrum and computing resources for the slices. This framework creates isolated RAN slices by selecting optimal FS and MEC placement per slice, solving the problem of running cost-effective edge networks and increasing the amount of traffic served with diverse QoS requirements.

Fig. 1.3 illustrates the global view for dynamic RAN slicing that has been studied in this thesis. In particular, we consider that RAN resources and nodes can be shared in isolation based on their particular service demands. As shown in this figure (bottom), all physical RAN resources and respective 5G demands are represented in Resource& Functional Level, where SDN controllers can be programmed to efficiently conduct policies and rules based on demand needs. Indeed, the fundamental technological enablers include softwarization, *e.g.,* VNFs, as well as SDN and NFVs. Accordingly, isolated slices are created for the demands based on their traffic types in the Network Level (middle) which are customized per slice. On the Service Level (top), lifecycle management automation realizes SLA fulfillment and QoS Assurance, wherein each service with a different SLA is controlled with Service Deployments/Operations to guarantee their respective QoS and SLA fulfilments.

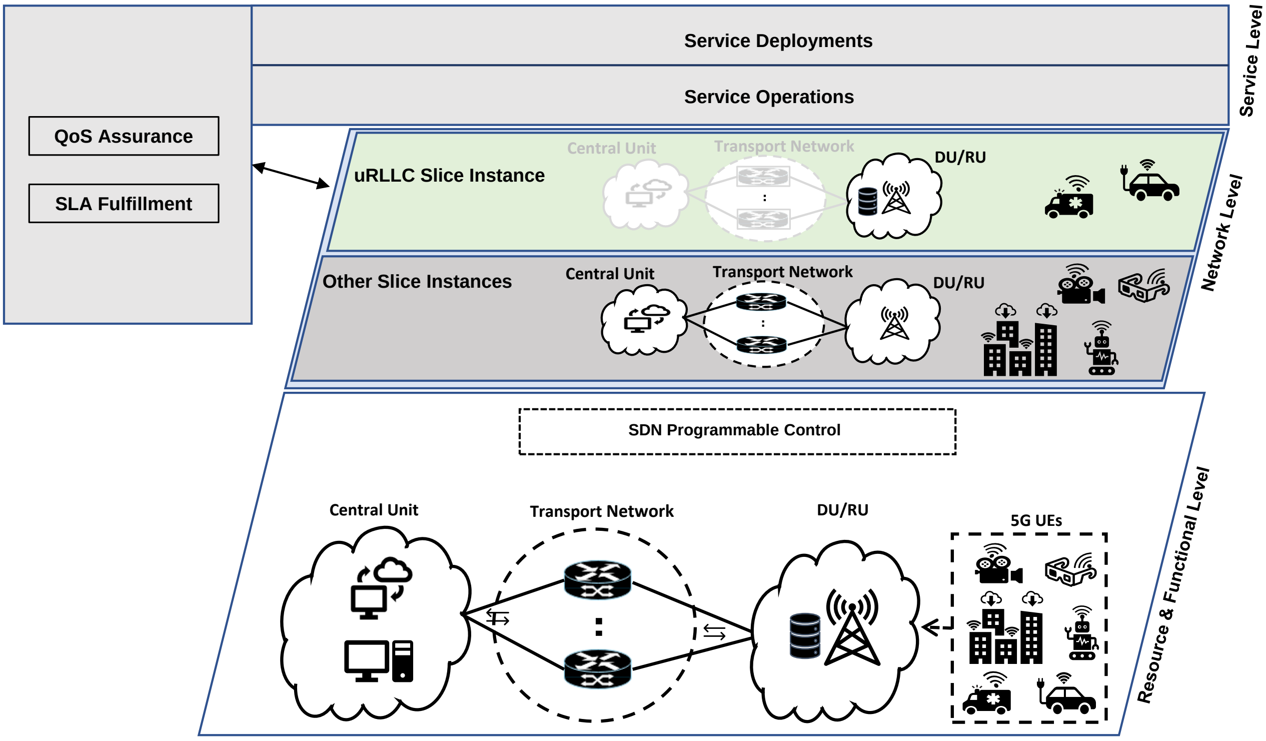


Figure 1.3: Thesis Vision.

1.3 Thesis Outline

This thesis is divided into 6 chapters, plus the references. The chosen structure facilitates reading comprehension and the achievement of the distinct objectives according to the following order:

- **Chapter 1: Introduction.** In this chapter, the current trends, motivations, technologies, challenges, contributions, and the thesis outline are described.

- **Chapter 2: Background.** This chapter addresses the global framework for network slicing, providing the necessary background information on the reference legacy and the expected future architectures that we explore within our thesis. It contains an explanation of the RAN architecture and its evolution, RAN functional splits, network softwarization, and enablers for 5G networks, as well as network slicing strategies for 5G RAN, core and transport networks with relevant information about the B5G slicing challenges and open issues in this research.

  After studying the State of the Art (SoA), we have realized that there are numerous challenges in sharing RAN resources and fulfilling upcoming 5G services. To this aim, our novel contributions to corresponding challenges in this thesis are organized into three chapters:

- **Chapter 3: Joint Slicing and Functional Split in 5G RAN.**
  Challenge. The diversity of 5G performance requirements entails new upgrades based on network virtualization and slicing to meet the 5G and beyond network requirements. Besides, network functional splits need to be well-tuned when meeting 5G services.

  Contribution . In our first approach to contribute to RAN slicing considering the 5G service requirements, we study radio resource management optimizations in C-RAN architecture to slice RAN resources and computation for the main 5G services defined by ITU in order to efficiently fulfill the 5G requirements. Accordingly, assuming C-RAN architecture, we present a framework for the joint RAN slicing and functional split with optimization of Centralization Degree (CD) and throughput.

- **Chapter 4: Service-Aware Network Slicing Framework for 5G RAN.**
  Challenge. Following our contribution in chapter 3 for 5G services, an efficient RAN slicing is needed to meet the respective requirements where RAN slices need to be adjusted with the so-called SLA enforced by the 5G services. In addition, based on what has been heeded in the contribution of chapter 3, in the new architecture of RAN, some RAN functionalities could be decoupled and centralized, where the optimal distribution of RAN functions still remains an open research issue.

  Contribution . Our second contribution which is aligned with the previous chapter is to address the principles of RAN slicing by using the concept of RAN slicing and virtualization based on the SLA of different services. In this way, the slicing approach would be service-aware since slice creation is done based on different use cases. Meanwhile, we aim to find optimal functional split based on taking advantage of the newly introduced concept of vitalization of functional splits to offload functions from RRHs flexibly. To this aim, this chapter provides an optimization framework for 5G RAN, which creates isolated RAN slices based on the service requirements with customized FSs per slice on top of a network composed of a CU, a FH/BH network, and a set of RRHs. This framework maximizes the throughput by jointly selecting the optimal routing paths from a connected User Equipment (UE) to CU, and FS while satisfying the QoS requirements.

- **Chapter 5: SO-RAN: Dynamic RAN Slicing and MEC Placement.**
  Challenge. Optimization of RAN resources along with creating isolated slices are the challenges we will address in this chapter, where each slice needs to be tailored to distinct services based on their needs. Besides, new services in 5G and B5G need high computation and low latency

next to the end users, where MEC placement within RAN architecture is another challenge that needs to be integrated with the RAN slicing to better meet the needs of services.

Contribution . In order to address the challenges mentioned above, this chapter aims to bridge the gap between the RAN slicing and MEC placements and proposes a novel optimization approach that minimizes RAN/MEC economic costs and maximizes served traffic. This framework creates isolated RAN slices by optimal placement of RAN and MEC functions per slice, which solves the problem of operating cost-efficient edge networks and increases the served traffic with diverse QoS requirements.

- **Chapter 6: Conclusions.** Finally, this chapter concludes the thesis and identifies the potential improvements based on our findings in the previous chapters for future research directions.

1.3.1 Research Contribution

This thesis investigates the dynamic 5G RAN slicing and MEC placement solutions for next-generation mobile networks. The main contributions of this doctoral thesis have been published in the following peer-reviewed Journals (**J**), flagship international Conferences (**C**) and Book chapters (**B**). The list of publications follows:

[C] **B. Ojaghi**, F. Adelantado, E. Kartsakli, A. Antonopoulos, and C. Verikoukis, “Sliced-ran: Joint slicing and functional split in future 5g radio access networks,” in ICC 2019 - 2019 IEEE International Conference on Communications (ICC), May 2019, pp. 1–6. URL: $https://doi.org/10.1109/ICC.2019.8761081$

[J1] **B. Ojaghi**, F. Adelantado, A. Antonopoulos, and C. Verikoukis, “SlicedRAN: Service-Aware Network Slicing Framework for 5G Radio Access Networks,” IEEE Systems Journal, pp. 1–12, 2021. URL: $https://doi.org/10.1109/JSYST.2021.3064398$

[B] Maule, M., **B. Ojaghi**, Rezazadeh, F. (2022). Advanced Cloud-Based Network Management for 5G C-RAN. In: Rodriguez, J., Verikoukis, C., Vardakas, J.S., Passas, N. (eds) Enabling 6G Mobile Networks. Springer, Cham. URL: $https://doi.org/10.1007/978-3-030-74648-3_11$

[J2] **B. Ojaghi**, F. Adelantado, A. Antonopoulos and C. Verikoukis, ”Impact of Network Densification on Joint Slicing and Functional Splitting in 5G,” in IEEE Communications Magazine, URL: $https://doi.org/10.1109/MCOM.001.2100680$

[J3] **B. Ojaghi**, F. Adelantado and C. Verikoukis, ”SO-RAN: Dynamic RAN Slicing Via Joint Functional Splitting and MEC Placement,” in IEEE Transactions on Vehicular Technology, 2022, URL: $https://doi.org/10.1109/TVT.2022.3209069$

[J4] **B. Ojaghi**, F. Adelantado, and C. Verikoukis, ”On The Benefits of vDU Standardization in Softwerized NG-RAN: Enabling Technologies, Challenges, and Opportunities,” in IEEE Communications Magazine, URL: $https://doi.org/10.1109/MCOM.001.2200390$

Chapter 2

Background

In this chapter, we discuss the most relevant technologies and the related research work on this topic. This section starts with an overview of the Long-Term Evolution (LTE) Radio Access Network (RAN) architecture evolution. Then the need for the new RAN architecture along with the latest features of 5G New Radio (5G NR), the capabilities of *functional split* and *edge computing* are discussed. Furthermore, this section explains the major enabling technologies of 5G networks such as Software Defined Networking (SDN), Network Function Virtualization (NFV), containerization, and 5G network slicing.

2.1 RAN Architecture

In this section, a literature review on RAN architecture and its evolution is explained. First, we briefly explain the traditional RAN architecture, then give an explanation of Cloud-RAN, and Open-RAN architectures as evolved technology, and finally describe the features of 5G NR.

2.1.1 Overview of LTE RAN Architecture

Long Term Evolution (LTE) has been designed to support only packet-switched services and aims to provide seamless Internet Protocol (IP) connectivity between User Equipment (UE) and the Packet Data Network (PDN). Third Generation Partnership Project (3GPP) has released the standards for LTE, and succeeding the LTE-Advanced [18–20]. The high-level network architecture of LTE is comprised of the following three main components:

- The User Equipment (UE).
- The Evolved UMTS Terrestrial Radio Access Network (E-UTRAN).
- The Evolved Packet Core (EPC).

Fig. 2.1 shows the schematic architecture of LTE in which the EPC communicates with PDNs in the outside environment such as the internet or the IP multimedia subsystem. The interfaces between the different parts of the system are denoted Uu, S1 and SGi as shown below:

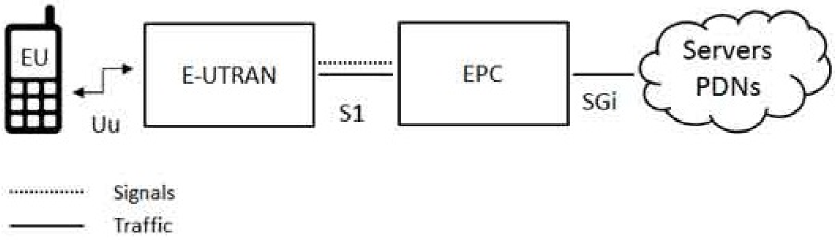


Figure 2.1: The illustration of LTE components [21].

The access network of the LTE-A [22–24], E-UTRAN, simply consists of a network of evolved NodeBs (eNBs) which has been illustrated below (Fig. 2.2):

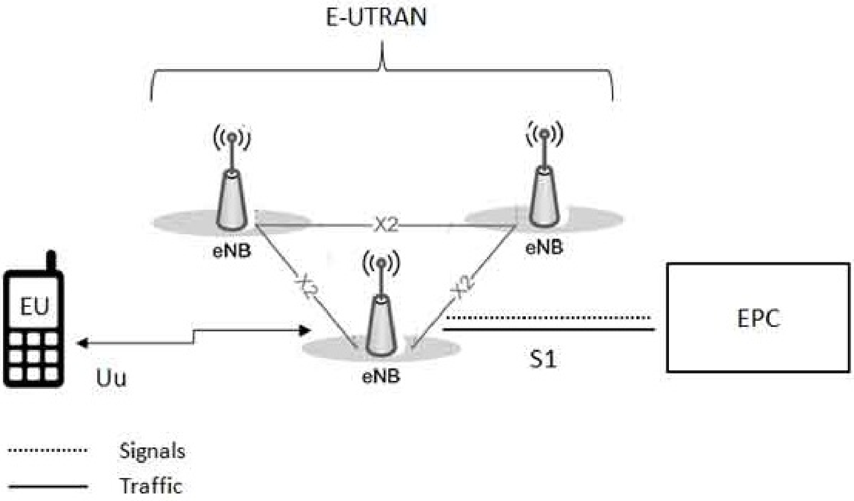


Figure 2.2: The schematic view of E-UTRAN [21].

The E-UTRAN handles the radio communications between the UE and the EPC and has one component known as eNB. The E-UTRAN is in charge of all radio-related functions, which are summarized briefly as:

- **Radio Resource Management (RRM)** - This covers all functions related to the radio bearers, such as radio bearer control, radio admission control, radio mobility control, scheduling and dynamic allocation of resources to UEs in both uplink and downlink.
- **Header Compression** - This helps to ensure efficient use of the radio interface by compressing the IP packet headers that could otherwise represent a significant overhead, especially for small packets such as Voice over Internet Protocol (VoIP).
- **Security** - All data sent over the radio interface is required to be encrypted according to the LTE standard, however, it is ultimately up to the Mobile Network Operator (MNO) to enforce this measure.
- **Connectivity to the EPC** - This consists of the signaling toward the mobility management entity (MME) and the bearer path toward the serving gateway (S-GW).

On the network side, all of these functions reside in the eNBs, each of which can be responsible for managing multiple cells. The eNBs are connected to nearby eNBs by means of an interface known as 'X2' principally used for signaling and packet forwarding during handover, and to the EPC by means of the 'S1' interface. Unlike the previous 2G/3G technologies, LTE integrates the radio controller function into the eNB. This allows tight interaction between the different protocol layers of the RAN, thus reducing latency and improving efficiency. One consequence of the lack of a centralized controller node is that, as the UE moves, the network must transfer all information related to a UE, that is, the UE context, together with any buffered data, from one eNB to another. Mechanisms are therefore needed to avoid data loss during handover. The X2 interface as a solution to this issue is established between one eNB and some of its neighbor eNBs. However, there exist other main issues such as the higher deployment cost of eNBs in dense scenarios, and limitations in the coordination of different eNBs which are needed to be considered in the new architecture of RAN.

2.1.2 RAN Evolution

The RAN has evolved over Distributed RAN (D-RAN) deployments with radios, *i.e.,* Remote Radio Head (RRH), and BaseBand Units (BBU). These BBUs are connected via Backhaul (BH) networks to the core. In 5G, RAN deployment architectures evolved into i) Cloud RAN (C-RAN), ii) virtualized RAN (vRAN), and finally, Open RAN (O-RAN).

C-RAN: firstly proposed by China Mobile [25] that has gained momentum technologies in the 5G era. As proposed by 3GPP in [13], the C-RAN architecture decomposed into three main parts:

- **Central Units (CUs)** - composed of higher flexible and programmable processors;
- **Remote Radio Heads (RRHs)** - located at the remote site and controlled by CU;
- **FrontHaul (FH) network** - low-latency high bandwidth optical or wireless network, connecting CU and RRHs;

In this network architecture, the Base Station (BS), known as eNB within the LTE network architecture, is fully shifted from the RRHs into a centralized CU [26]. CUs are connected to the evolved packet core (EPC) through a BH network, and all RRHs are connected to CUs through a FH network, typically transmitting radio signals using current specifications of serial line interfaces like Common Public Radio Interface (CPRI) or Open Base Station Architecture Initiative (OBSAI) [26].

This architecture is capable of improving spectrum efficiency, more energy efficiency, and reducing deployment costs (due to pooling gains) [27, 28]. Nevertheless, its advantages do not guarantee realistic large-scale deployments due to the stringent requirements of the FH for 5G. For example, in C-RAN where all BS functionality is centralized except the Radio Frequency (RF) function which is located at the RRH, thus transmitting IQ samples through the FH that requires a bandwidth of around 2.5 Gbps and a very low delay of 0.25 ms (with a 20 MHz bandwidth) [29].

The main advantages of C-RAN architecture are as follows:

- Simplifies the structure of RRHs

- Lower operation and deployment costs
- Enables virtualization and slicing of RAN
- Allows the flexible allocation of a pool of radio and computational resources
- Controls the transitions from distributed BSs to a centralized RAN
- Migrates a hardware-defined infrastructure to a software-defined environment
- Maximizes spectrum efficiency and hardware usage.

Both D-RAN and C-RAN have been implemented utilizing vendor-proprietary hardware-based appliances from RAN incumbents such as Ericsson, Huawei, and Nokia.

vRAN: incorporates SDN/NFV virtualization techniques into the RAN, hence, virtualizing all network functions (*i.e.,* Virtualized Network Functions (VNF)), and resources as well as decoupling Data Plane (DP) and Control Plane (CP) among CU and Distributed Unit (DU). The DU places near RRHs and facilitates real-time processing, such as communication between the user and cell site. It is, however, still in its early stages of development. Spectrum virtualization, air interface and infrastructure virtualization, multi-Radio Access Technologies (RATs) virtualization, and computing resources are only a few examples of various levels of virtualization that are feasible. For example, BBU processing can be virtualized, allowing to run of Virtualized Network Functions (VNF) to be implemented on standardized, Commercial-Off-The-Shelf (COTS) hardware such as x86 or Arm-based servers. It facilitates the shift of traditional network architecture from hardware-based to software-based, as well as the concept of proprietary hardware-based BSs, into a more flexible, adaptable, and affordable solution. Hence, vRAN, through the use of commodity hardware, offers more flexibility and reduces RAN costs. 5G New Radio (5G NR) is the evolution of LTE Advanced and LTE Advanced Pro wireless technologies defined in 3GPP Release 15 and beyond and current RAN technology is presented as a hardware- and software-integrated platform, aiming for the disaggregation between hardware and software elements, where Open RAN (O-RAN) is among key pillars toward this evolving RAN technology.

O-RAN: In particular, O-RAN Alliance [30], which is a community of MNOs, is committed to evolving RAN architecture that breaks down what was once a *one-size-fits-all*, hardware-centric RAN, making it more open, interoperable interfaces and elastic than currently deployed networks. In that sense, O-RAN Alliance is working toward realizing the vision of Next Generation (NG) cellular networks, where MNOs use standardized interfaces to control multi-vendor infrastructure and provide high-performance services to their subscribers [30].

The key principles of the O-RAN Alliance include:

- Leading the industry towards open, interoperable interfaces, RAN virtualization, and big data and AI-enabled RAN intelligence;
- Maximizing the use of COTS hardware and merchant silicon and minimizing proprietary hardware;
- Specifying APIs and interfaces, driving standards to adopt them as appropriate, and exploring open source where appropriate.

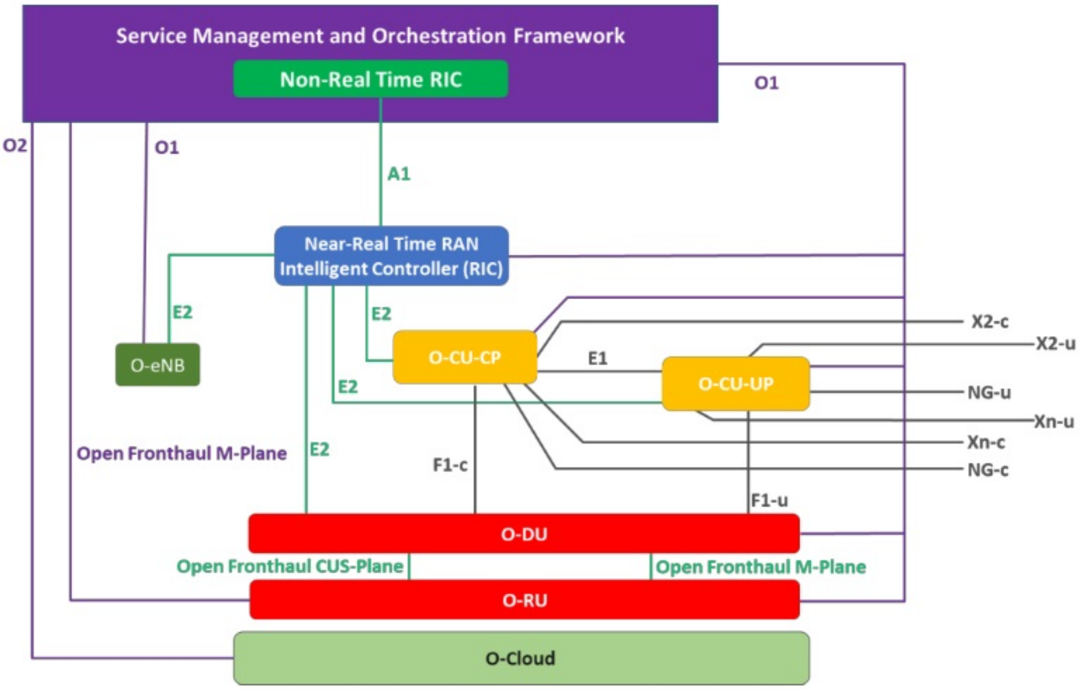


Figure 2.3: The schematic overview of 5G NR with the core network and internet [30].

In this network architecture, as shown in Fig. 2.3, the O-RAN consists of three main components:

O-RAN Central Unit (O-CU): is the centralized and virtualized component of RAN that is in charge of the Packet Data Convergence Protocol (PDCP) layer. Its northbound interface is the BH network to the core; its southbound interface is the F1 interface. F1 is referred to as midhaul as defined by 3GPP - sits between the O-CU and O-DU.

O-RAN Distributed Unit (O-DU): is the component responsible for all baseband processing, scheduling, Radio Link Control (RLC), Medium Access Control (MAC), and the upper part of the Physical layer (PHY). The F1 is the northbound interface, and the O-RAN FH is the southbound interface. The virtualization of this component requires some hardware assistance in the form of accelerators such as Field-Programmable Gate Arrays (FPGAs) or Graphics Processing Units (GPUs).

O-RAN Radio Unit (O-RU): is the component responsible for the lower part of the PHY layer processing (*e.g.,* Fast Fourier Transform (FFT) / Inverse Fast Fourier Transform (IFFT), beamforming). There is a remote possibility of virtualization of the O-RU; however, one working group in the O-RAN Alliance plans a “white box” radio implementation using off-the-shelf components. It enables anyone to construct a radio without proprietary components, which differs from virtualization.

In O-RAN architecture, Service Management and Orchestration (SMO) framework contains the Non-Real Time RAN Intelligent Controller (Non-RT RIC) function, which has the goal of supporting intelligent RAN optimization in non-real-time (*i.e.,* greater than one second) by providing policy-based guidance using data analytics and AI/ML training/inference. Non-RT RIC can leverage SMO services such as data collection and provisioning services of O-RAN nodes. Near-RT RIC, O-CU-Control Plane (O-CU-CP), O-CU-User Place (O-CU-UP), O-DU, and O-RU are the Network Functions (NF) for the radio access side. Near-RT RIC enables near real-time control and optimization of O-RAN (O-CU and O-DU) nodes and resources over the E2 interface with near real-time control loops (*i.e.,* 10ms to 1s). The Near-RT RIC is used to monitor, suspend/stop, override and/or control primitives to control the behaviors of O-RAN nodes. The Near-RT RIC hosts xApps that use the E2 interface to collect near real-time RAN information to provide value-added services using these primitives, guided by the policies and the enrichment data provided by the A1 interface from the Non-RT RIC.

This architecture helps MNOs to support the various demands in 5G networks due to open and interoperable interfaces, and the flexibility of deploying network NFs and enabling the customized functional split per use cases which brings advantages in both network systems and user experience part.

2.2 RAN Functional Splits

5G networks are expected to support various applications with high flexibility meeting a diversity of requirements in terms of latency, data rates, and massive connectivity. A 5G NF supplies a particular capability to support communication through a 5G network. Generally, the whole operation of the 5G gNB can be modeled as a chain of NFs [17, 31], where these NFs are normally virtualized, known as VNFs implemented in CU and DUs that could increase the performance of gNB, simplify the network operation, *etc.* The low-layer NFs are implemented on dedicated hardware, namely RU, which is generally realized as Physical Network Function (PNF). In general, NFs can be the functions that are common NFs that are essential for all applications, for example, authentication and identity management NFs. On the other hand, there exist some other functions which might not be useful for all the use cases. For instance, for enhanced Mobile Broadband Communication (eMBB) applications requiring higher gNBs coordination, centralized NFs (located in CU) provide better coordination, resource sharing [3] or an ultra-Reliable Low Latency Communication (uRLLC) application need higher decentralized NFs to reduce the Hybrid Automatic Repeat Request (HARQ) delay and guarantee low delay.

Indeed, an MNO, based on different application requirements is able to decide dynamically for each function module to be realized in either the CU or DU, known as Functional Split (FS). To reduce the FH traffic amount, some modules can be migrated to the DU side and other NFs shifted to CU. However, the NFs at DU can be just as basic signal and analog processing known as Distributed RAN (D-RAN) [32].

In that sense, 3GPP [13] proposes eight different FS options for the distribution of these functions among RAN components. The brief overview of different FS options is illustrated in Fig. 2.4:

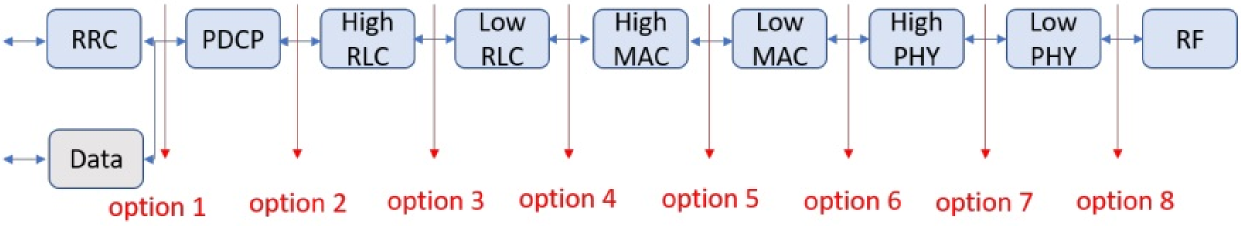


Figure 2.4: Functional split options [13].

- *Option 1:* The FS in this option locates RRC in the CU and PDCP, RLC, MAC, PHY and RF are in the DU.
- *Option 2:* RRC, PDCP are in the CU and may use any type of FH network. RLC, MAC, PHY and RF are in the DU. The main advantage of this option is the possibility to have an aggregation of different DU technologies (*e.g.,* 5G, LTE, and Wifi).
- *Option 3 (Intra RLC split):* Low RLC (partial function of RLC), MAC, PHY and RF are in the DU. PDCP and high RLC (the other partial function of RLC) are in the CU. The failure over the transport network may also be recovered using the end-to-end Automatic Repeat Request (ARQ) mechanism at CU. This may provide protection for critical data. This option also reduces the FH latency constraints as real-time scheduling is performed locally in the DU.
- *Option 4 (RLC-MAC split):* MAC, PHY and RF are in DU. PDCP and RLC are in the CU. This split allows synchronized multi-cell coordination for Coordinated MultiPoint (CoMP) and enhanced time-domain Inter-Cell Interference Coordination scheme (eICIC), but requires a low-latency FH and has significant traffic overheads.
- *Option 5 (Intra MAC split):* RF, PHY and some parts of the MAC layer (e.g. HARQ) are in the DU. The upper layer is in the CU.
- *Option 6 (MAC-PHY split):* PHY and RF are in the DU. The upper layers are in the CU. This split known as C-RAN achieves the highest centralization and coordination which enables more efficient resource management and can be realized only with an ideal FH which consumes very high bandwidth and has very low delay bounds.
- *Option 7 (Intra PHY split):* Part of the PHY function and RF are in the DU. The upper layers are in the CU.
- *Option 8 (PHY-RF split):* RF functionality is in the DU and the upper layers are in the CU.

Fig. 2.5 shows an example of cloud-assisted flexible FS in C-RAN. As previously explained, C-RAN is the highest centralized architecture: most of the processing, control, and management functionalities are migrated into the BBU pool (or equivalently, CU), and the basic RF functionality remains in RRHs. However, due to the various demands of applications, a fully centralized system is not optimal in all scenarios. For example, uRLLC users need more decentralized FSs to reduce the HARQ delay. As seen in this figure, with the software-defined environment (*e.g.,* SDN/NFV), the C-RAN operator implements a flexible FS instead of a fully centralized system which is directly related to the application requirements.

Likewise, O-RAN offers specific FS for the Lower-Layer Splits (LLS) and the Higher-Layer Splits (HLS) for locating NFs. These LLS and HLS mainly rely on network services and available transport links. Indeed each FS has its own set of requirements and, as a result, various alternatives for connecting CU, DUs, and RUs have been recommended, each one with its own set of advantages and disadvantages. Most commonly, option-2 is recognized as a FS between CU and DU, whilst option-7 is perceived between DU and RU.

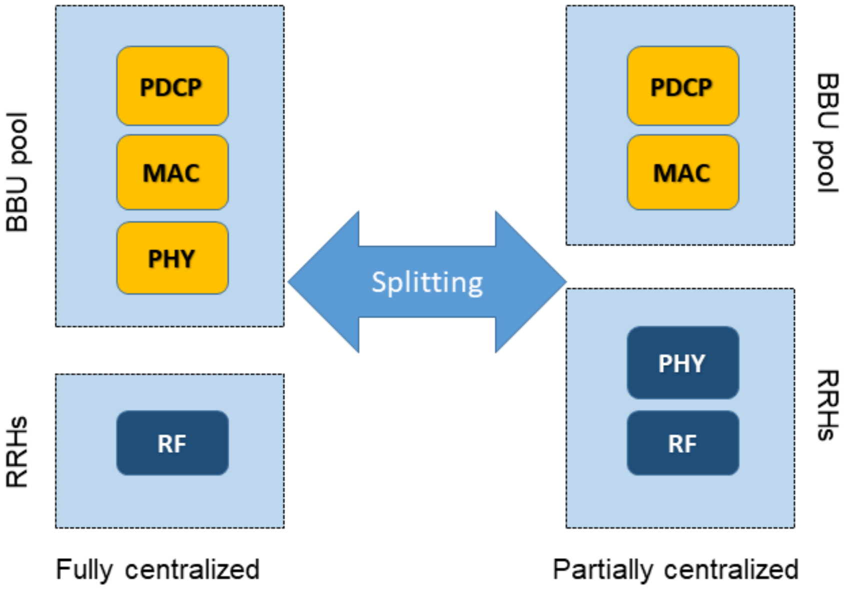


Figure 2.5: Cloud-RAN Flexible Functionality Splitting.

Various organizations are looking into the standardization of FSs to define the proper FS option. Nevertheless, there is no single, ideal FS agreed upon among standardization bodies as different FSs will accommodate different applications. However, it is unlikely that the industry will support all eight options. Among the standardization bodies, 3GPP has supported Option 2 for highly centralized applications. Meanwhile, Option 6 is being supported by SCF as the optimal FS for low-cost, low-capacity deployments by focusing on its 5G network Functional Application Platform Interface (nFAPI) efforts. See Fig. 2.6. O-RAN Alliance recommends option 7.2 for networks with high-capacity and high-reliability requirements.

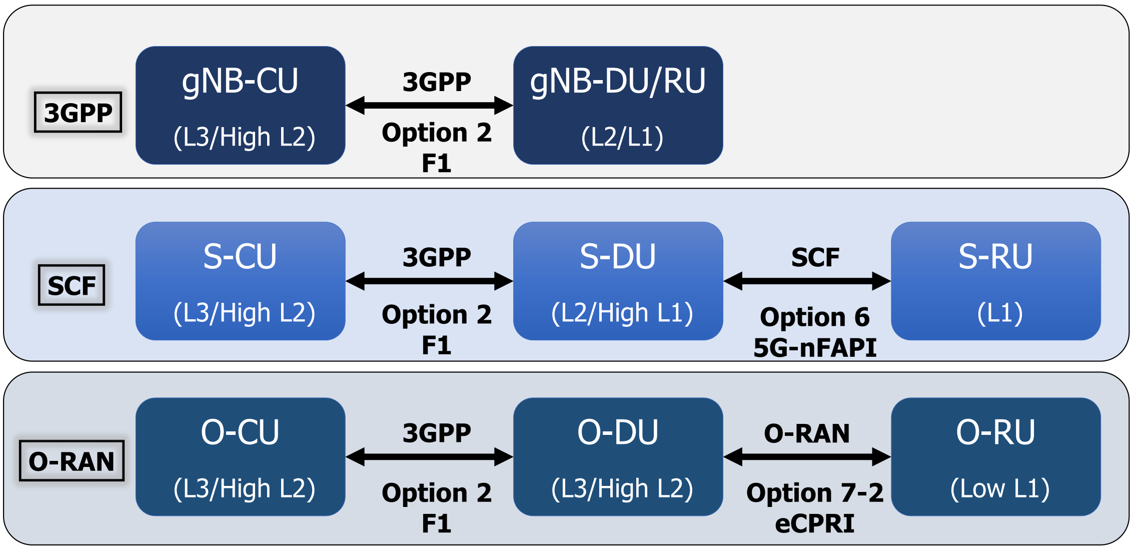


Figure 2.6: The RAN split options recommended by 3GPP, SCF, and O-RAN.

Fig. 2.7 shows eight 3gpp options along with their supported standardization bodies.

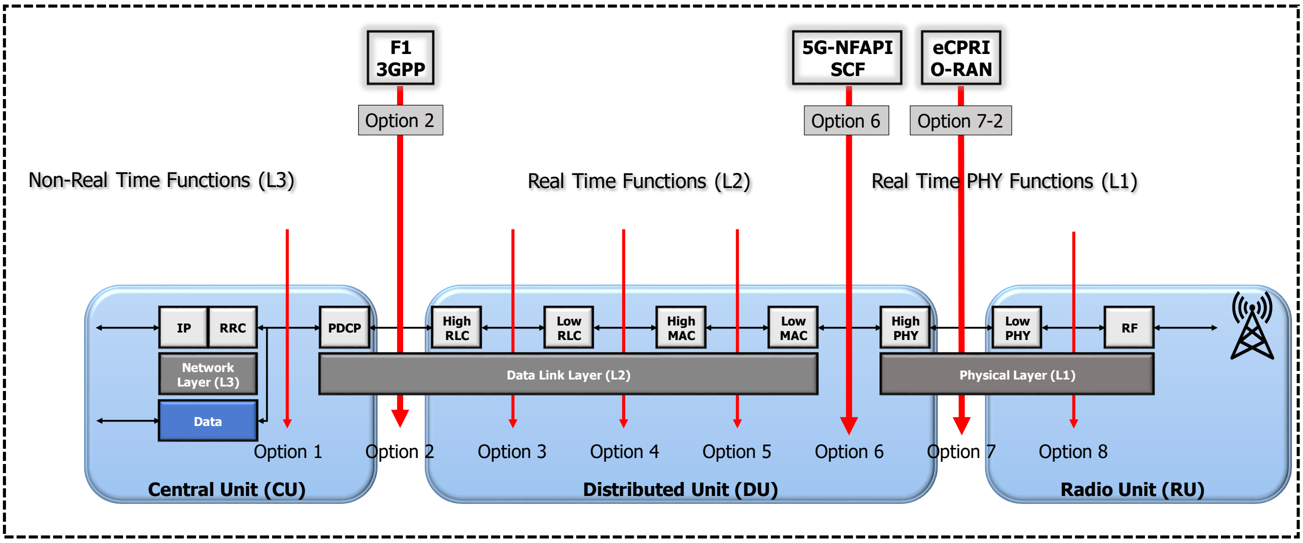


Figure 2.7: The 3GPP RAN splits: towards standardization

SCF's nFAPI contributes significantly to the O-RAN platform by offering the most deployable implementation of an open FH link based on Split 6, which is particularly well-suited to small cell, commercial networks. nFAPI covers SCF's established FAPI interfaces in a transport layer, allowing open connections between components at PHY layer that can expand the architecture to the MAC layer, allowing it to be virtualized and an open FH link between a Small cell-RU (S-RU) and a Small cell-DU (S-DU). This enables MNOs to select these elements from several vendors if it results in the best solution for their particular use case. The kind of network that will be used determines which splits will be chosen (among others). Split 6 does not need a high-quality connection such as fiber for every link between the RU and DU, whereas Split 7.2 requires it because the PHY operations are concentrated in one location. This can be more practical and cost-effective in a variety of enterprise contexts because it eliminates the need to purchase and install optimized fiber to every RU and might even provide more freedom to locate cells where required.

RAN disaggregation or centralization, which relies on adapting to a particular FS also enables virtualization of many NFs, with the software hosted on servers.

In B5G, the virtualized RAN resources are shared dynamically among the gNBs and across gNBs' components based on user needs. Thus, CU is assumed to be fully virtualized, namely vCU. This allows developing software and implementation of network slicing in RAN architecture. The next section will thoroughly discuss the new technologies that can be adapted in the B5G RAN architecture.

2.3 Network Softwarization & Enablers

Network Softwarization is a technology that allows a certain NF to run in software rather than hardware. The softwarization approach converts traditional network appliances with non-standard hardware into software-based Virtual Machines (VM) installed in standard equipment. This technology has the potential of efficiently handling heterogeneous resources spanning network and cloud domains and easily and flexibly deploying services with significant Capital EXpenditures (CAPEX) and OPerating EXpenses (OPEX) reduction.

In particular, *network softwarization* is attributed to two main factors: SDN and NFV. SDN separates the network's CP and DP in order to enable centralized network control, more automated provisioning, rapid innovation through a programmable network, and policy-based management of network resources. NFV replaces dedicated network hardware devices with software-based VNFs. The VNFS can run as software on commodity hardware, providing high flexibility, scalability, and cost-effectiveness.

In this section, we will discuss the promising enablers for network softwarization such as virtualization techniques (SDN/NFV), containerization, edge computing, and network slicing.

2.3.1 Software Defined Networking (SDN)

The emergence of Software Defined Networking (SDN) [12] as one of the key technologies provides a basis for introducing a uniform QoS networking approach in the context of evolving mobile networks domain. The SDN architecture includes an intermediate CP that dynamically configures and abstracts the underlying forwarding plane resources in order to deliver customized services to clients located in the application plane [33]. Any SDN service is built on a set of resources, NFs, and interfaces that are tailored to the specific need. Resources may be physical or virtual, active or passive, and in many cases, may be created, scaled, or destroyed by the client or the server on their own initiative or at their request. Resources available to SDN include VNFs, as defined by the ETSI [34]. The main idea of SDN is the separation of the CP from DP through a well-defined API (*e.g.,* OpenFlow). The devices that support OpenFlow are composed of two logical parts: (i) an exposed OpenFlow API that handles the exchanges between switch/router and controller, and (ii) a flow table that defines how to process and forward packets within the network. In this approach, a software control program, called the controller, has an overview of the whole network and is responsible for the decision-making, while the hardware is simply in charge of forwarding packets to their destination as a set of packet-handling rules. The SDN controller abstracts and aggregates/partitions the underlying resources while carrying out the virtualization function.

Fig. 2.8 depicts the reference SDN architecture model, which is split into three main layers:

- The SDN applications can specify their requirements for traffic management in the underlying networks through northbound APIs.
- The SDN controller, which is in charge of the CP, connects the application and DPs. It translates application requirements into appropriate forwarding rules that are enforced by the underlying network switches. The SDN controller can use the southbound API to access NFs provided by

SDN-enabled switching devices.

- The DP refers to network elements (*e.g.,* routers and switches) that collect network status information and process packets based on SDN controller-supplied rules, such as traffic statistics and network topology.

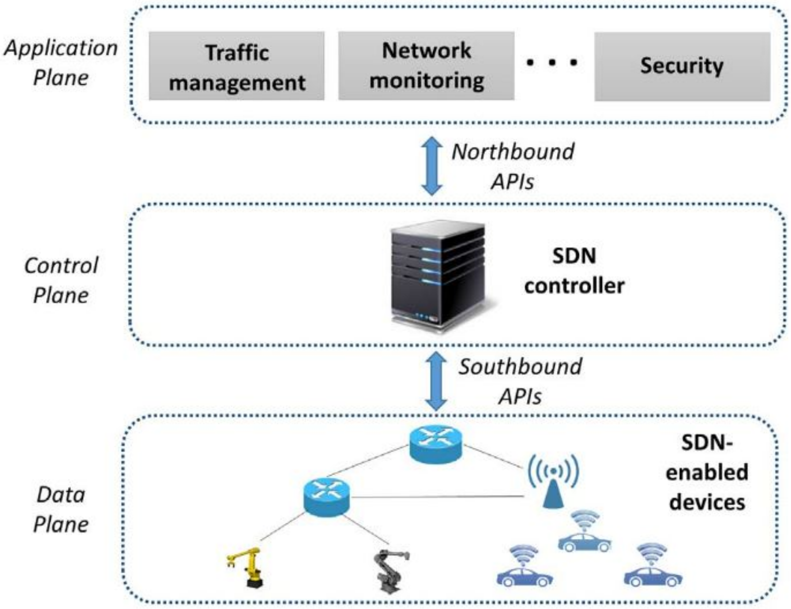


Figure 2.8: The three layers in SDN architecture [35].

Due to virtualization, each client context provides a unique Resource Group that the client connected with that context can use to carry out its service(s). The SDN controller optimally assigns the selected resources to such discrete Resource Groups via orchestration. The interaction of both controller functions allows for the fulfillment of diverging service needs from all clients while maintaining their isolation. These preliminaries of SDN ideas can contribute towards addressing different challenges faced by current and future mobile networks. In particular, this is in line with the objectives of 5G network slicing, which must meet a diverse set of service demands in a flexible and cost-effective manner [36]. The Open Networking Foundation (ONF) vision states that the SDN architecture naturally supports slicing [37] because the client context offers the whole abstract set of resources (as a Resource Group) and the associated control logic that forms up a slice, including the entire collection of relevant client service characteristics.

Given the importance of SDN as an enabler for both virtualization [38] and slicing [39], there has been great research interest in software-defined mobile networks in recent years mostly concentrated on the core network due to its similarity to wired networks [40, 41]. The Radio Access Network (RAN) as a complex and costly part of mobile networks infrastructure tenders more profits from SDN concept. In RAN, using multiple radio access technologies such as LTE and WiFi, utilizing advanced PHY techniques like Coordinated Multi-point (CoMP), etc., requires a high level of coordination among BSs, thus the utilization of SDN enables these functionalities. Furthermore, softwarization in the RAN through programmability enables flexibility and authorizes a broad range of applications and novel technologies such as virtualization [38] and slicing [39] which will be further discussed.

2.3.2 Network Function Virtualization (NFV)

The SDN architecture outlined above provides a comprehensive insight into the CP functionalities that enable slicing, but it is deficient under certain capabilities that are essential for effectively managing the life cycle of network slices and the resources that build them up. In that sense, the Network Function Virtualization (NFV) architecture [42] is the best fit to perform this function as it manages the infrastructure resources and orchestrates their allocation for the realization of VNFs and network services. The NFV technology is a carrier-driven initiative with the goal to adapt the way that operators design networks by using virtualization technologies to virtualize network functions. In particular, NFV is responsible for forwarding NFs as software, capable of running as virtualized rules and allowing them to be deployed at required locations in the network without needing to install equipment for each new rule. NFV is applicable to any network function in both mobile and fixed networks. To utilize NFV's management and orchestration capabilities, a proper collaboration between SDN and NFV is essential. In a nutshell, SDN forms a concept related to NFV, but they refer to different domains. SDN is focused on the separation of the network control layer from its forwarding layer, while NFV is focused on porting NFs to virtual environments to enable the migration from proprietary appliance-based embodiments to a standard hardware and cloud-based infrastructure.

The NFV architecture contains the following entities [36]:

- **Network Functions Virtualization Infrastructure (NFVI):** a set of resources for hosting and connecting VNFs that includes network connectivity between locations, such as between data centers and public or private hybrid clouds. Physical resources in general comprise computing, storage, and network hardware, which provide processing, storage, and communication for NFVs via the virtualization layer placed in the virtualization layer. The NFV architecture can use an existing virtualization layer, including a hypervisor, with conventional functionality that simply isolates hardware resources and assigns them to VNFs.

- **VNFs:** Software-based implementations of NFs that run over the NFVI.

- **NFV-Management and Orchestration (NFV-MANO):** handles all orchestration, and life-cycle management tasks of physical or virtual resources pertaining to virtualization in the NFV architecture (*i.e.,* VNFs). The MANO framework is composed of three functional components [43]:

- **Virtualized Infrastructure Manager (VIM):** in charge of controlling and managing the NFVI resources.

- **VNF Manager (VNFM):** conducts VNF(s) configuration and life cycle management on its domain,

- **Orchestrator:** According to ETSI, it contains two sets of NFs that are handled by the Resource Orchestrator (RO) and the Network Service Orchestrator (NSO). The RO orchestrates NFVI resources across VIMs. The NSO executes the life cycle management of network services using the capabilities given by the RO and the (possibly distinct) VNFMs.

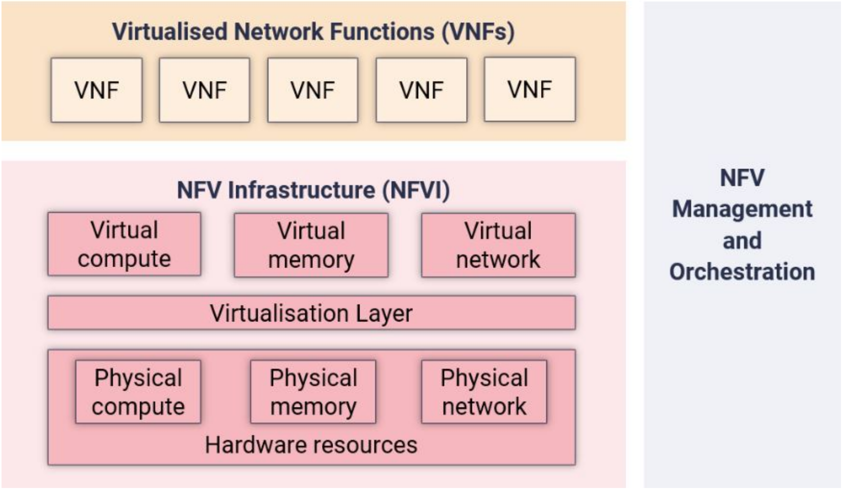


Figure 2.9: High-level ETSI NFV architecture [43].

Fig. 2.9 presents a high-level view of the NFV architecture, as established by ETSI [43].

NFV enables the underlying clouds and infrastructures to be virtualized to construct customized UP and CP processing across several domains. Moreover, NFV technology can be used to implement RRH/RU upgrades for lower splits in software [44]. This solution can be an enabler for a network with flexible FSs, where the NFs are adapted according to a certain set of requirements and enabled when required by NFV.

2.3.3 Containerization

Containerization offers a new type of virtualization, in which a single Operating System (OS) kernel can create multiple isolated user-spaces and run different applications from the CU. Both VMs (conventional approach) and containers can assist in network softwarization and resource optimization. Each VM needs its own OS running on top of a virtualized resource. It includes the hypervisor which places between hardware (*i.e.,* infrastructure), and OS and is a piece of software and firmware for creating and managing VMs. However, VMs can use a significant amount of system resources. As shown in Fig. 2.10, each VM runs not only a full copy of an OS but also a virtual copy of every piece of hardware required for the OS to function, where the amount of Random Access Memory (RAM) and Central Processing Unit (CPU) cycles consumed by this approach quickly grows.

A container is a standardized software component that packages up code and all of its dependencies to ensure that an application will run swiftly and consistently in different computing environments. A container image is a lightweight, standalone, executable file that contains all the components required to run an application, including the code, run-time, system tools, system libraries, and settings. A container's lifecycle is handled by what is generally referred to as a container runtime. Containers provide an abstraction to the operating system, whereas VMs provide an abstraction to the hardware. Containers are thus more efficient because each container shares the host OS kernel as well as, in most cases, binaries and libraries. Finally, containers only use the hardware resources they require at the time of execution, therefore there is no resource reservation as there is with VMs.

Containerization is an important technology that B5G will embrace to define how NFs and computing

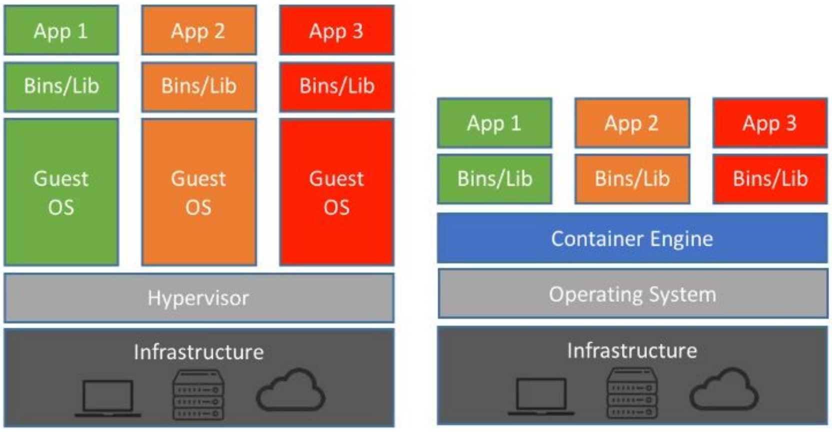


Figure 2.10: VMs (Left) vs Container (Right) architecture [45].

infrastructures should be distributed among RAN elements. Network functionalities that were previously developed as monolithic programs are now split down into smaller micro-services and delivered as containers in CU using the cloud-native method. It is premised on the basis of decomposing an application into a set of micro-services that can be created and deployed separately. The micro-services are packaged into light-weight containers which are scheduled to run on compute nodes by a container orchestrator. In this regard, the most popular containerization platform, Docker[1] can create, deploy and manage different micro-services to increase the density, scaling, and speed of deployment, portability, and decrease cost.

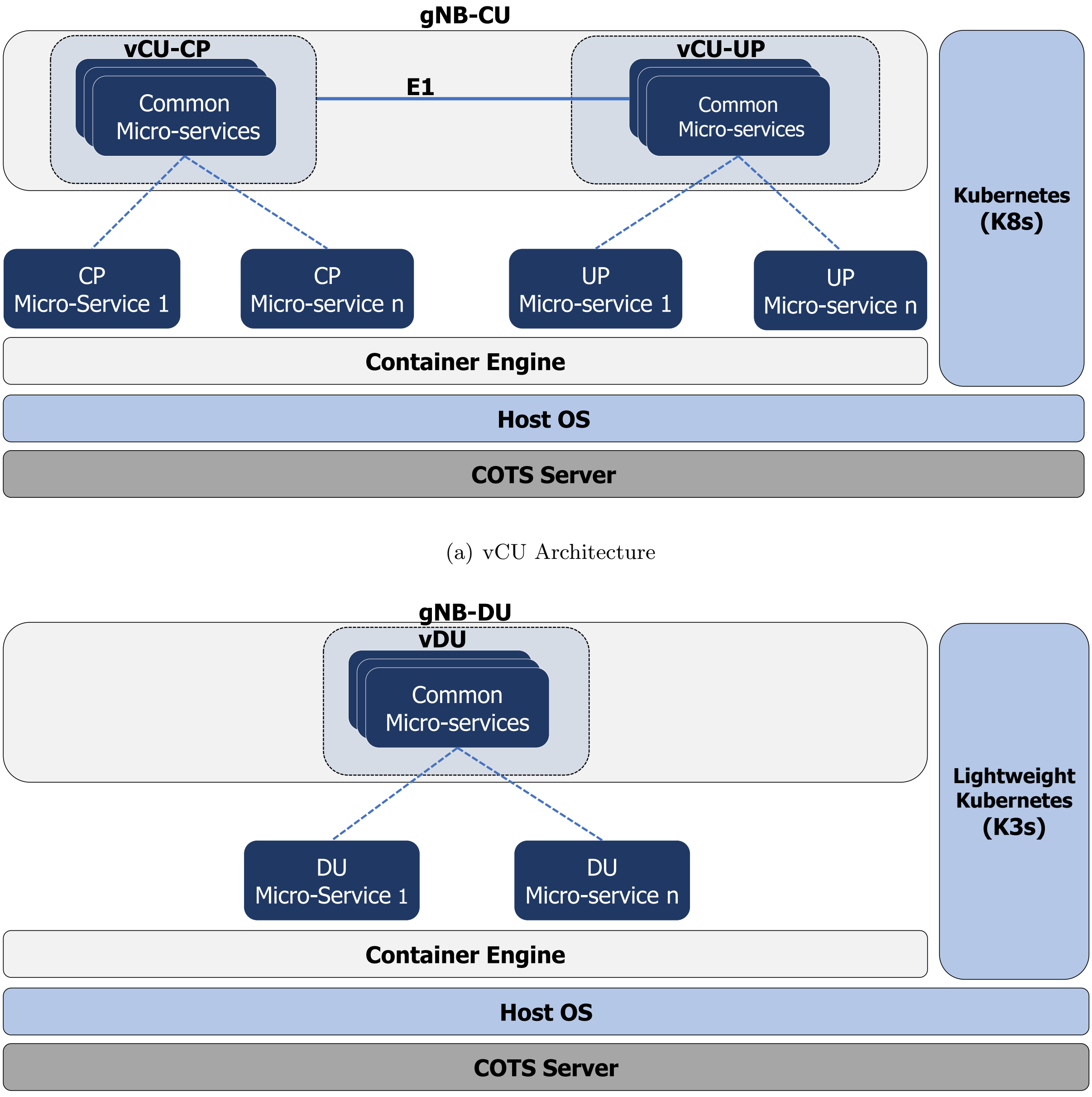
Containerization in B5G is indeed acting as a superior backend for applications running on accessing devices. In this way, it provides both manufacturers and end users the foundation for NG network slicing, on-demand provisioning of virtual resources, self-service functionality, resource pooling, network hypervisors, scalability, high flexibility, and smooth evolution to support future standards, and easy support for new revenue-generating services. For example, it enables placing MEC inside RAN architecture that provides ultra-low latency, reliability, and scalability to meet the service demands of a large number of IoT devices.

In addition to the CU/DU split and the CU-CP/CU-UP split, container technology can improve scalability, flexibility, and resource efficiency. Each micro-service in the virtualized CU/DU (*i.e.,* vCU and vDU) can have its own flavor (size) for flexible dimensioning. For example, each micro-service of vDU can be scaled out if additional cells are deployed. In vCU, traffic loads for CP and UP are balanced between each plane's micro-service separately to maximize resource usage efficiency. Each micro-service can be scaled on-demand or automatically based on the current load status. In this regard, VNFs of vDU that are detached from the hardware resources can run on a device with COTS server. In vDU, the PHY and MAC layers demand extremely complicated computational tasks such as scheduling algorithms and Forward Error Correction (FEC). These tasks may cause a burden on a COTS server's processing power and reduce vDU performance. As a result, some computationally demanding processes with repeating structures, like FEC, could be offloaded to different hardware chips for acceleration and implemented as an option on a COTS server.

The MANO of vCU containers can be supported by the service orchestration tools such as Kubernetes

[1] https://www.docker.com/

(K8s), and its lightweight versions (*e.g.,* K3S, KubeEdge) for vDU, which are open-source container orchestration systems for automated deployment of services, featuring high flexibility and scalability. Fig. 2.11 shows the virtualized architecture of vCU and vDU using containerization technology [46].



(a) vCU Architecture

(b) vDU Architecture

Figure 2.11: The containerized vCU and vDU Architecture [46].

2.3.4 Edge Computing

In recent years, a new trend in computing is arising with the functionality of clouds but moving towards the network edges [47]. This paradigm is named Multi-access Edge Computing (MEC) [48]. In fact, MEC refers to computing at the edge of a network. The edge is similar to a distributed cloud with proximity close to the end user that delivers ultra-low latency, reliability, and scalability.

While the issue of the delay remains a key drawback for Cloud Computing, MEC is widely agreed to be a key technology for enabling various services, especially hard delay requirements such as Tactile Internet (with millisecond reaction time) [49–51] or Internet of Things (IoT) [52]. Edge computing reduces latency to milliseconds and allows for constant connectivity. Furthermore, when the edge network experiences high traffic, the edge may offload data to the cloud to maintain a quick and reliable connection. RAN and MEC are highly complementary technologies as a RAN deployment necessitates a significant amount of processing power, each such site instantly becomes a MEC site - readily expandable to handle emerging B5G services, particularly those demanding low latency or high bandwidth. Moreover, how close a RAN/MEC site is located to cell-sites will often define how well it can support particular services – or whether it can support them at all. The location of a C-RAN/MEC site in a Central Office (CO) or, equivalently data center is often less expensive than deploying so in the field, but the penalty is higher latency. Hence, there is a trade-off between cost and performance, and a meticulous understanding of the use-cases – *i.e.,* which services are expected to fulfill at such a site – is essential [53].

Given the distinct service types and needs that may be present at a RAN/MEC site, it is reasonable that the infrastructure is portioned into multiple domains, as shown in Fig. 2.12.

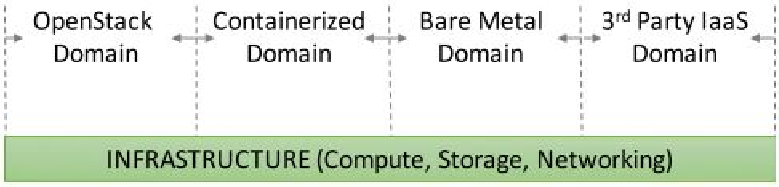


Figure 2.12: RAN/MEC site architecture example [53].

As proposed in [54] a MEC can be placed close to RRHs/RUs to effectively enable different applications such as services with very high data flows (*e.g.,* video streaming), while others have hard latency requirements. Indeed there is an intricate coupling between the design of the new architecture of RAN and the deployment of MEC services. This platform would open new opportunities for the MNO to reduce their system costs and gain more benefits due to serving hard delay/throughput requirement applications.

2.4 Network Slicing in 5G and Beyond (B5G)

Network slicing is a new technology that shows a promising component in the 5G and B5G that is defined by the Next Generation Mobile Network (NGMN) Alliance [3] that handles the deployment of multiple logical networks as independent business operations on a common physical infrastructure. It enables the network elements and NFs to be flexibly customized and reused in each network slice to fulfill specific requirements. The MNO of a network slice views the network slice as a distinct virtualized network shared with other network slices using NFV and SDN principles that enable MNOs to establish different capabilities, deployments, and architectural flavors for each service and run multiple network instances in parallel. Network slicing is designed to be an end-to-end solution that incorporates both the core network and the RAN, where each slice can have its own network design, network engineering, and network provisioning. In the core network, each slice's network components and NFs are virtualized via NFV and SDN to accommodate its unique requirements. Slicing in the RAN can be based on physical radio resources (*e.g.,* transmission point, spectrum, frequency, or time) or logical resources abstracted from physical radio resources such as computing resources. They introduce significant technical challenges such as network slice instantiating and maintenance, slicing over multi-domains, and allocation of computing, storage, and radio resources along with supporting algorithms and mechanisms.

Given that 5G is devised as a network capable of providing service-tailored connectivity, by utilizing network virtualization methodologies based on SDN and NFV [12] which enables the network with enough flexibility, programmability and automation mainly to create isolated slices on-demand on top of the physical network. In the following, we will discuss the introduction of these technologies into RAN and the core part of 5G mobile networks. Network slicing enables the separation of a network in an isolated way, and each slice provides unique connectivity while running on the same shared infrastructure. In this way, 5G virtualization offers a novel level of flexibility, enabling operators, to allocate a customized slice to certain kinds of services. This will allow operators to more efficiently support different sets of services, thus each slice will be able to access different types of resources, such as VNFs and shared infrastructure (*e.g.,* VPNs, cloud services).

2.4.1 5G RAN Slicing

Network slicing is an important component of NG-RAN architecture, enabling MNOs to build virtualized networks that can be tailored to satisfy a variety of demands in terms of functionality and isolation. Virtualization will be an essential component at the network edge namely, the virtual partitioning of the mobile RAN. Also through RAN slicing, MNOs will be able to create unique services that are customized for various use cases such as IoT, automated cars, streaming video, remote health care, etc. They can create virtual networks for those applications that boast separate blends of performance, capacity, latency, security, reliability, and coverage. With the ability to deploy a wide array of sliced networked services from a single physical network, MNOs will be able to diversify, expand, and increase their revenue streams in a highly cost-effective way. A RAN slice requires both physical and virtual resources. Virtual resources are managed by the ETSI NFV-MANO, while physical resources are managed by the 3GPP network slicing management system. To allocate the virtual/physical resources of a Network Slice Instance (NSI), the Network Slice Management Function (NSMF) receives slice-related requirements from the Communication Service Management Function (CSMF). The NSMF divides slice-related requirements into Transport Network (TN), RAN, and Core Network (CN) slice subnets. It is the responsibility of the NSMF to manage the life cycle and resources required for three of the subnets for the corresponding Network Slice Subnet Management Function (NSSMF), namely the CN, TN, and RAN NSSMFs. NSMF requests 5GC, TN, and NG-RAN NSSMFs to create the necessary 5GC, TN, and NG-RAN NSSs. It is the responsibility of the Network Function Management Function (NFMF) to manage every component in each subdomain. In Fig. 2.13, the NG-RAN subdomain is illustrated, where each of its components (namely the CU, DU, and the RU) are managed by its NFMF. In the 3GPP network slicing management system, the NFMF, the NSSMF, and the NSMF manage the physical resources of the networks. In addition to physical resources, NSIs or RAN slices also contain virtual resources. Virtual resource management and orchestration fall outside of the 3GPP's scope but within the ETSI NFV.

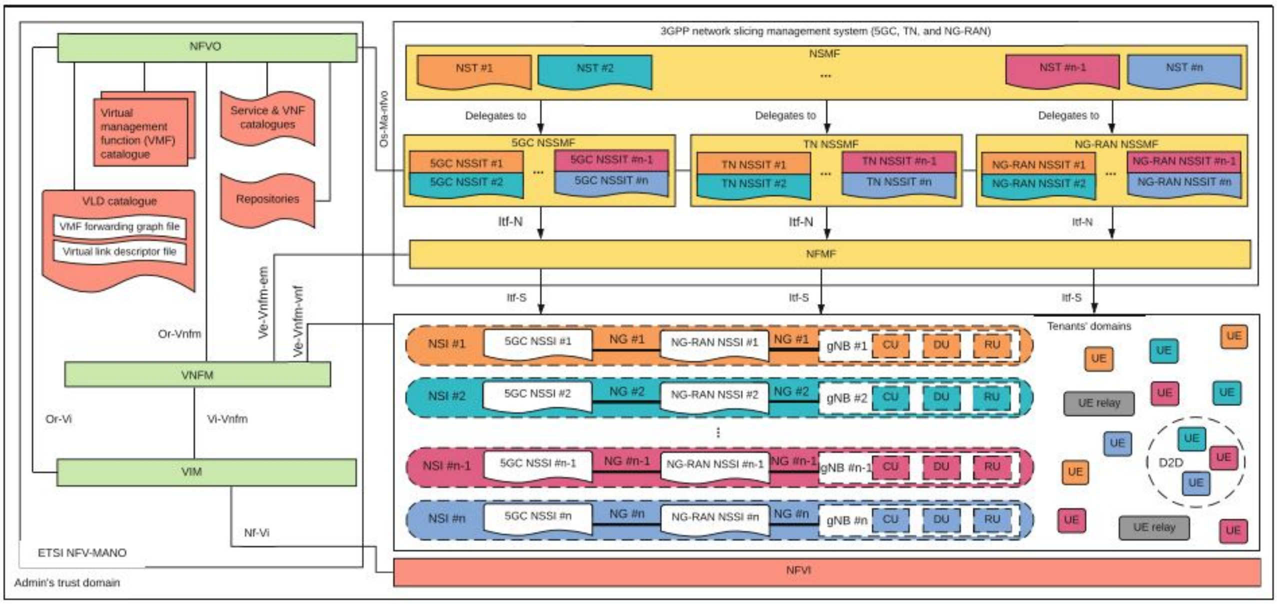


Figure 2.13: The 3GPP and ETSI unified framework for RAN slicing in NG-RAN [55].

The Open Networking Foundation (ONF), a non-profit operator-led consortium in the SDN movement, has contributed its SDN platforms to O-RAN, expanding efforts to develop RAN control standards called Software-Defined RAN (SD-RAN) [2]. Likewise, RAN architecture in B5G can apply the SDN/NFV to decouple CU into a User Plane (UP) and Control Plane (CP) part, where customized UP and CP processing can be tailored to specific services such as scheduling multiple applications within the PDCP layer, and CU-CP hosts control functions for various services that need to be isolated from one another. However, in DU, there is no CP and UP separation despite the fact that the MAC level scheduling has a significant impact on the RAN performance and service-specific aspects of scheduling and selection of PHY layer functionality and that the F1 midhaul is unable to operate the DU on a real-time. The virtualization could shift further to DU, namely as vDU, to support the aforementioned aspects in the NG-RAN.

Given the importance of the service-specific design of RAN architecture, the traditional *one-size-fits-all* approach to mobile network infrastructure is unable to deal with the expected wide range of services and the extremely different requirements of NG [7]. To efficiently serve this traffic, virtualization and slicing emerge as essential components within NG-RAN architecture to create on-demand isolated slices for different types of supported services such as eMBB, uRLLC, and mMTC. However, implementing network slicing and virtualization causes additional complexities that need to be adopted into RAN architecture. In 5G, virtualization will do away with the notion of communications built upon a single mobile network infrastructure, opening up the potential for limitless numbers and types of supported services.

The International Telecommunication Union (ITU) categorized 5G mobile network services into three use cases ([5]):

- Enhanced Mobile Broadband (eMBB) which tries to meet the services that need high-bandwidth,

[2] https://opennetworking.org/open-ran/

such as High Definition (HD) videos, Virtual Reality (VR), and Augmented Reality (AR) (Fig. 2.14).

- ultra-reliable and Low-latency Communications (uRLLC) focus on the demanding digital industry to meet low latency services, such as assisted and automated driving (Fig. 2.15).
- Massive Machine Type Communications (mMTC) meet demands for services that include high requirements for connection density, such as smart city and smart agriculture.

Figure 2.14: The real-time mixed reality with AR and VR services [56].

Figure 2.15: The assisted and automated driving services [57].

To fully meet these application demands, RAN architecture entails being flexible enough in order to support improved resource pooling and to have higher spectral efficiency over various transport network configurations.

Indeed the virtualization and the slicing process required to adapt the RAN to users' performance requirements pose significant challenges in 5G networks. In the future, RAN will be composed of a CU and a set of geographically distributed RRHs connected through a packet-based network, as proposed by 3GPP in [13] and in [58]. Given the importance of SDN as an enabler for both virtualization [38] and slicing [39], some recent works have been focused on RAN virtualization platforms and slicing designs. Foukas et al. propose FlexRAN in [59], a flexible and programmable software-defined RAN (SD-RAN) platform. The platform is composed of a centralized controller and one agent per eNB that separates control and data planes and allows a flexible control plane design. This architecture enables the dynamic allocation of control functions between the centralized controller and the decentralized agents, thus tailoring the RAN to meet the performance requirements. However, despite making a step forward in the direction of the virtualization of the RAN, the proposal still lacks a slicing design. The same authors in [60] proposed Orion, which is a RAN slicing design running on the FlexRAN platform while guaranteeing functional isolation among slices. Isolation of functions among slices is of paramount importance since it allows a slice-custom FS within a single shared eNB. In other words, two different slices sharing the same physical node can be configured with different FSs. For instance, in a specific gNB, a slice serving a high-speed user better suits a centralized FS, so that the coordination among neighboring cells is tighter and the handover performance can be simplified. Conversely, the slice serving a low-latency user requires more decentralized FSs to reduce the HARQ delay. This is the main weakness of [14, 15, 61] where, for simplicity, either no slicing is considered or all slices are assumed to suit the same FS. Two notable recent works, [58] and [54], address the optimization of the FS. Specifically, WizHaul is proposed in [58] as a joint routing and FS optimization to achieve maximum centralization. Similarly, FluidRAN follows the same rationale but targets the monetary cost minimization [54].

Up to now, only using a single FS is allowed to be implemented in gNB components. There is a need to support and standardize multiple FSs per unit to meet NG service requirements. The NG-RAN slicing is a complementary technology that enables running different FSs per unit (*i.e.*, vCU or vDU), thus guaranteeing functional isolation between slices that is crucial as it enables a slice-custom FS within a single unit [62]. For instance, in a given gNB, for a tele-driving service that requires a strict latency (10 ms), a slice using virtual PDCP in vDU (close to vehicle) with reduced HARQ delay can support this service. On the other hand, for traffic efficiency service, which has tolerable delay requirements, a slice better suits with a centralized FS with PDCP in vCU such that neighboring cells can coordinate more accurately and the handover process can be made simpler for these services. (See Fig. 2.16).

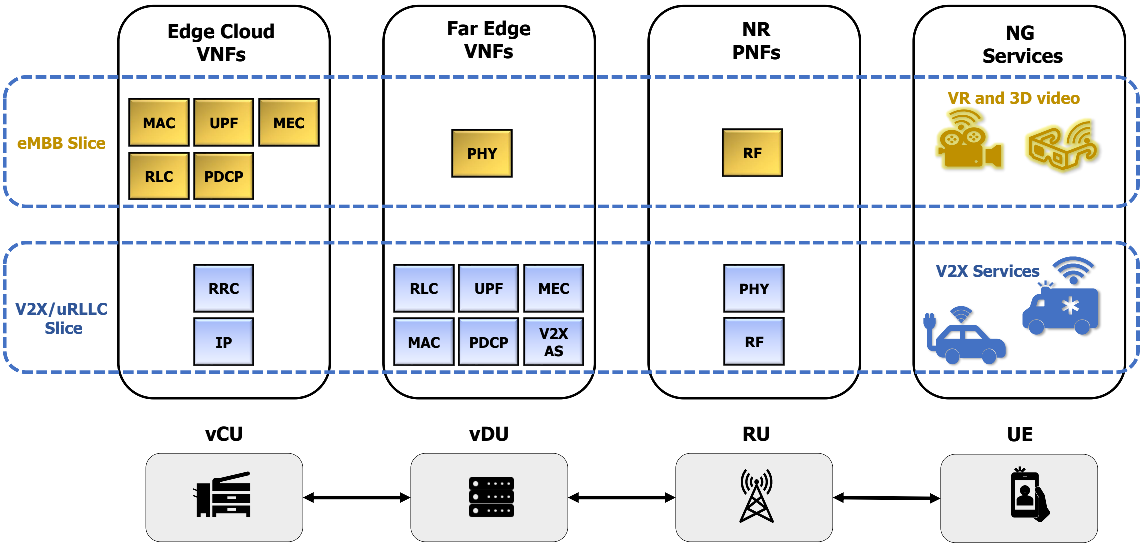


Figure 2.16: The NG-RAN slicing with a slice-custom FS within a single unit [63]

2.4.2 5G Core Slicing

Generally, core networks have been created as a single network architecture that can be used for a variety of functions, while still preserving backward compatibility and interoperability. Due to the fact that all functionality is provided by a single set of vertically integrated nodes, this one-size-fits-all strategy has lowered costs to a reasonable level. The recent technological shift towards softwarization (thanks to virtualization, SDN/NFV, containerization/cloudification, and advanced automation and orchestration) enables to create more scalable, flexible, and dynamic networks. Network slicing in 5G supports multiple virtual networks over one physical network infrastructure and allows core networks to be logically separated with each slice providing customized connectivity, and all slices running on the same or shared infrastructure relying on the MNO's needs. There has been a significant study on mobile core slicing in the literature and virtualization of core network functions combined with the use of mature virtualization technologies (*e.g.,* KVM [64], Docker [65]) have led to systems that realize core network slicing [66–68]. Some research works leveraging NFV in the core appeared in [69] which is focusing on scalability, in [70] studied the customization for specific services, and finally, in [71] augmenting flexibility. The flexibility and elasticity in service provisioning imply that a mobile network operator is able to deploy several instances of the EPC, all at the same time, to serve different services. Moreover, as explored in the literature, EPC can be offered as a service over the cloud [72] and carrier cloud work which is studied in [73] shows how to expand and enlarge the service model of cloud infrastructure providers from only providing computing and storage capabilities as a service in data centers to also enabling end-to-end mobile connectivity as a service. The carrier cloud work explained how LTE, EPC can be offered as a service (*i.e.,* LTEaaS, EPCaaS), in fact, anything can be offered also as a service [74] over the cloud which enables the support of diverse use cases and, in turn, allows the introduction of 5G network slicing [75].

2.5 B5G Slicing Challenges & Open Issues

Creating and managing several RAN slices in an isolated way is challenging. Besides, providing radio resource isolation has the disadvantage of inefficient usage of radio resources. One solution is to provide dynamic spectrum-sharing radio resources, however, this increases the need for shared or coordinated scheduling mechanisms between slices and access channels while ensuring isolation of resources for each slice is challenging. This problem can be addressed by adopting a programmable Software-Defined RAN (SD-RAN) [60] platform that separates CP and DP and allows a flexible CP design. However, as NG networks are expected to support multi- Radio Access Technologies (RATs), RAN slicing solutions need to keep up with this concept. This adds another challenge as it is unclear whether multiple RATs can be multiplexed over the same possibly specialized hardware, or whether each RAT needs its own dedicated hardware.

Chapter 3

Joint Slicing and Functional Split in 5G RAN

As mentioned in the previous chapter, the diverse and wide range of services in 5G along with their corresponding different performance requirements, has raised the need for new mobile network architecture, where the current network architectures need to be upgraded based on promising technologies such as network virtualization and slicing to meet the 5G and beyond network requirements. In that sense, 5G has been conceived as a softwarized network, using SDN/NFV technologies, able to provide service-tailored connectivity. Network slicing as a key mechanism is being widely attracted by both the research community and standardization bodies to serve the diverse services in 5G. In particular, cloudification and centralization of network functions in the Radio access network (RAN), known as Cloud RAN (C-RAN), have been enabled using virtualization technologies such as SDN and NFV. In C-RAN, the Base Band processing unit (BBU), also known as the Central Unit (CU), is decoupled from the distributed Remote Radio Head (RRH) and pooled in a central cloud. To support the 5G requirements efficiently, it is essential to benefit from C-RAN architecture to share RAN resources and computation for 5G services. The goal of this chapter is to bridge the gap between functional split and network slicing by efficiently sharing RAN computation and resources. In that sense, we investigate joint network slicing and functional split in the C-RAN-based network architectures that optimize the network performance in terms of centralization degree, and throughput, which will be further discussed.

3.1 Introduction

The unprecedented surge in data traffic experienced over the last decade has stretched telecommunications networks to their capacity. According to Cisco's forecast [1], global IP traffic has increased 127-fold from 2005 to 2021. This demand rise is exacerbated by wireless and mobile traffic to account for more than 60% of the total IP traffic in 2021, growing twice as fast as fixed IP traffic. It is precisely in this context of traffic explosion in which the requirements for 5G have been defined [3]. In a nutshell, 5G must support high data rates, low latency targeting about 1 ms round-trip time, a reduction of the cost and energy consumption, and massive connectivity [4].

The diversity of performance requirements, ranging from 1 Gbps peak rate to 1ms end-to-end latency, renders a classical static network architecture unfeasible for 5G. Thus, 5G is conceived as a network able to provide service-tailored connectivity, by leveraging network virtualization techniques based on Software Defined Networking (SDN) [12] which provides the network with enough flexibility, mainly to create isolated slices on-demand on top of the physical network.

The slicing process required to adapt the Radio Access Network (RAN) to users' performance requirements poses significant challenges in 5G networks. In the future, RAN will be composed of a Central Unit (CU) and a set of geographically distributed Remote Radio Heads (RRH) connected through a packet-based network, as proposed by 3GPP in [13] and in [58][1]. Although the optimal distribution of layers PDCP/RLC/MAC/PHY between the CU and the RRHs (known as *functional split*) has attracted the attention of the research community, it still remains an open research problem [14–17].

3.2 State of the Art

Virtualization and slicing have become two major concepts for future 5G networks. Given the importance of SDN as an enabler for both virtualization [38] and slicing [39], some recent works have been focused on RAN virtualization platforms and slicing designs. Foukas et al. propose FlexRAN in [59], a flexible and programmable software-defined RAN (SD-RAN) platform. The platform is composed of a centralized controller and one agent per evolved Node B (eNB) that separates control and data planes and allows a flexible control plane design. This architecture enables the dynamic allocation of control functions between the centralized controller and the decentralized agents, thus tailoring the RAN to meet the performance requirements. However, despite making a step forward in the direction of the virtualization of the RAN, the proposal still lacks a slicing design. The same authors in [60] proposed Orion, which is a RAN slicing design running on the FlexRAN platform while guaranteeing functional isolation among slices. Isolation of functions among slices is of paramount importance since it allows a slice-custom functional split within a single shared eNB. In other words, two different slices sharing the same physical node can be configured with different functional splits. For instance, in a specific eNB, a slice serving a high-speed user better suits a centralized functional split, so that the coordination among neighboring cells is tighter and the handover performance can be simplified. Conversely, the slice serving a low-latency user requires more decentralized functional splits to reduce the HARQ delay. This is the main weakness of [14, 15, 61] where, for simplicity, either no slicing is considered, or all slices are assumed to suit the same functional split. Two notable recent works, [58] and [54], address the optimization of the functional split. Specifically, WizHaul is proposed in [58] as a joint routing and functional split optimization to achieve maximum centralization. Similarly, FluidRAN follows the same rationale but targets monetary cost minimization [54]. However, none of them consider the slicing of the RAN.

[1] The packet-based network connecting the CU and RRHs is an integrated FrontHaul/BackHaul (FH/BH) network, also known as *cross-haul* in [58].

3.3 Contribution

This chapter is aimed at designing a joint routing (from user to CU) and functional split optimization while considering different slices. As shown schematically in Fig. 3.1, the slicing of the RAN allows a customized functional split deployment per slice, thus optimizing the available resources, *e.g.,* transport network capacity and RRH or CU computational capacity. Fig. 3.1 conveys how different control functions are allocated either at the RRH or at the CU for each slice.

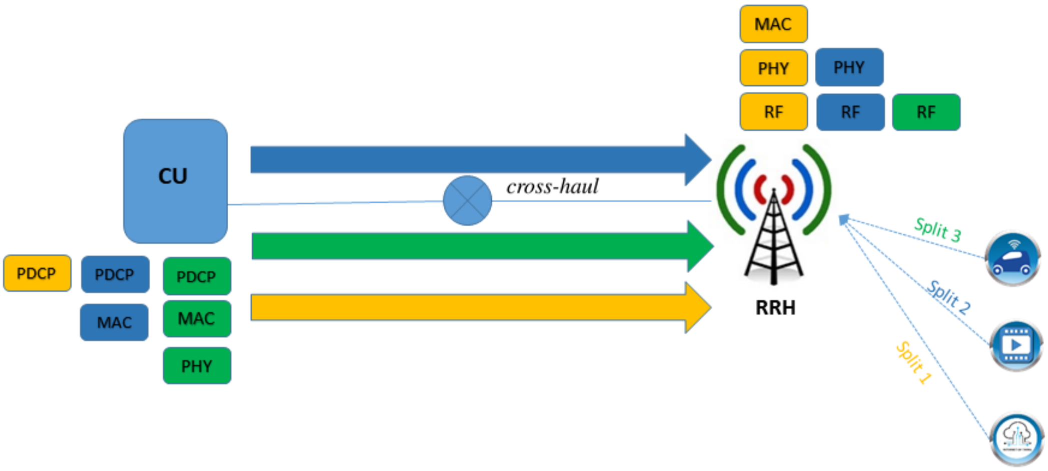


Figure 3.1: Slicing based on service requirements.

Specifically, the contributions of this chapter are the following:

- We propose a joint slicing and functional split RAN optimization. As described in [13] and implemented in [60], functional isolation is assumed at the eNB. This means that each slice of an eNB can have a different functional split.
- The slice creation is extended up to the user. In general, the slicing only considers the RAN, [54, 58]. Instead, this work aims to exploit the high density of RRHs by jointly analyzing routing in the RAN and user association.
- Not all services support the whole range of possible functional splits. For instance, high transmission rates usually require a high degree of centralization to implement efficient Coordinated Multipoint (CoMP). We take this into account in the functional split optimization.

3.4 System Model

As thoroughly described in Section 3.2, the RAN can be modeled as a CU, referred to as node 0, and a set of RRHs, $\mathcal{R} = \{1, \ldots, R\}$. Connecting the RRHs and the CU, there is a transport network (FH/BH) composed of a set of forwarding nodes, $\mathcal{Q} = \{1, \ldots, Q\}$, connected among them and with the CU and the RRHs through a set of links $\mathcal{L} = \{l_{i,j} : i, j \in \mathcal{R} \cup \mathcal{Q} \cup \{0\}\}$. Each link $l_{i,j} \in \mathcal{L}$ has a capacity equal to $\omega_{i,j} \geq 0$ (in bps) and a delay $d_{i,j} \geq 0$ (in sec). The connection between the CU and an RRH can be realized through multiple paths [54]. Let us then define the ith path $P^{r,i}$ from RRH

r to the CU as the set of links between them. Given that there might exist multiple paths between RRH r and CU, the set of all possible paths from RRH r to CU is denoted by P^r. Additionally, the set of users served by the network is denoted by $\mathcal{U}$ and is characterized by their required data rate λ_u^s, where $u \in \mathcal{U}$ and $s \in \mathcal{S}$ is the service type of the user.

3GPP has proposed in [13] a wide range of possible granularities of the functional split, from the coarsest granularity (the functional split is determined by the computational capacity of the RRH and CU computational capacities and the transport network capacity) to the finest granularity (the functional split is decided on a user, bearer or slice basis). Without precluding any of the granularity levels proposed by 3GPP, in the sequel, we focus on the traffic type-based functional split. As shown in [13], the control functions of layers PDCP, RLC (high and low), MAC (high and low), and PHY (high and low) can be allocated either in the CU or in the RRH. Accordingly, a specific allocation (*i.e.,* functional split) can be determined per slice. Given that slicing will be done on a service-type basis, hereafter service and slice concepts will be interchangeable. In the sequel, we will assume a set of four RAN functions, denoted as $\mathcal{F} = \{f_0, f_1, f_2, f_3\}$, where f_0 is the low layer network function (RF, signal and analog processing, etc.) which is always placed in the RRH and f_3 is the high layer network function (*e.g.,* PDCP and above layers). Depending on the functional split, these functions will be allocated either at the RRH or at the CU. That is, a completely centralized functional split will allocate f_0 at the RRH and the rest of the functions at the CU. Conversely, a completely decentralized functional split will accommodate all functions at the RRH. Initially, f_0 is always placed in RRH and f_3 is in CU, no matter which split we use. Based on the aforementioned definitions, three different functional splits are considered in the following, as also assumed in [58]. In split 1, namely ϕ_1, f_1 and f_2 are located in RRH. This split is useful to serve users with low latency requirements. Split 2, ϕ_2, which enables better utilization of hardware, only allocates f_1 in RRH while f_2 is moved to CU. In split 3, ϕ_3, all functions are moved to CU (complete centralization, also known as Centralized-RAN, C-RAN). This split allows a higher degree of coordination among eNBs or gNBs.

Each functional split imposes different maximum latency and minimum throughput constraints to the FH/BH. For instance, in functional split 1, the decentralization reduces traffic overhead and the required BH capacity can be approximated by the aggregate users' traffic. Instead, for functional split 3, only the RF function is located at the RRH, thus transmitting IQ samples through the FH. In this case, samples are usually encapsulated with CPRI [76] and the required FH capacity depends on the bandwidth of the eNB, the number of antennas, etc. That is, the FH capacity requirement does not depend on the users' traffic for ϕ_3. This can be observed in Table 3.1, where FH/BH bandwidth and latency requirements for each split are shown as described in [54].

Table 3.1: Bandwidth and latency requirements of functional splits [54]
(λ_u^s is the datarate of user u with service type s).

Split	Bandwidth (bps)	Latency (ms)	Functions allocated at the RRH
ϕ_1	λ_u^s	30	f_0, f_1, f_2
ϕ_2	$1.02\lambda_u^s$	2	f_0, f_1
ϕ_3	$2.5 \cdot 10^9$	0.25	f_0

Given the described scenario, the network can create different slices on top of the physical RAN, each one with a specific functional split tailored to provide the required QoS, while guaranteeing the data rate and latency constraints of each functional split.

3.4.1 Optimal Slicing and Functional Split

The management and operation of the dynamic functional split in the RAN pose significant challenges with several trade-offs. On the one hand, a reduction in the transport network's load can be achieved by locating RAN functions at RRHs. On the other hand, offloading the RAN functions and pooling at the CU benefits from a computing cost reduction and offers centralized control that can improve the network's performance. However, while some splits have very tight delay constraints and create high FH traffic, in other splits the CU might not have enough computation power to accommodate all RAN functions. In this context, a convenient network slicing algorithm provides the network with a higher degree of flexibility, thus enabling the adaptation of the functional split of each slice to traffic requirements and network limitations (RRH and/or CU computing capacity, transport network capacity, and delay, etc). Thereby, in this chapter, we aim to find the optimal joint functional split and network slicing for future 5G networks.

3.4.2 User Associations

As pointed out in previous sections, the type of service of each user can determine a minimum degree of centralization (*i.e.,* the set of possible functional splits). For instance, a high-speed moving user requires a high degree of centralization to coordinate the handover process better, or, in other words, it can only be supported by split ϕ_3. On the contrary, an Ultra-Reliable Low Latency (URLLC) user higher decentralization to guarantee low delay. Therefore, not all services can be supported with all functional splits.

Let us define the spectrum allocated to RRH r as W_r. The spectrum is divided into Physical Resources Blocks (PRB), each one with a bandwidth equal to w (in Hz). Based on the definitions, if the user u with type s and transmission rate λ_u^s (in bps), requires a number of PRBs per subframe equal to $\rho_u^{r,s}$ when served by RRH r, the user will be served by the RRH with the highest SINR (*i.e.,* the smaller $\rho_u^{r,s}$) as long as there are enough resources (See Fig. 3.2). Thus, the user association must hold the condition $\sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}} x_u^{r,s}\rho_u^{r,s} \leq \rho^r$, where $x_u^{r,s} \in \{0,1\}$ is a binary variable to check whether user u with type s is connected to RRH r or not, and $\rho^r = W_r/w$ is the number of available PRBs in that RRH.

3.4.3 Routing and Delay Constraints

Connected users inject traffic (bps) into the network, which is transmitted through the transport network over path $p \in P^{r,i}$ between CU and RRH r. This is schematically shown in Fig. 3.3. Each RRH has connections to CU via sliced paths. These paths are used based on different demands of users' traffic. In other words, each RRH is responsible for transmitting flows of connected users to CU based on their service types. Hence, for each RRH there are different paths from CU. We define the

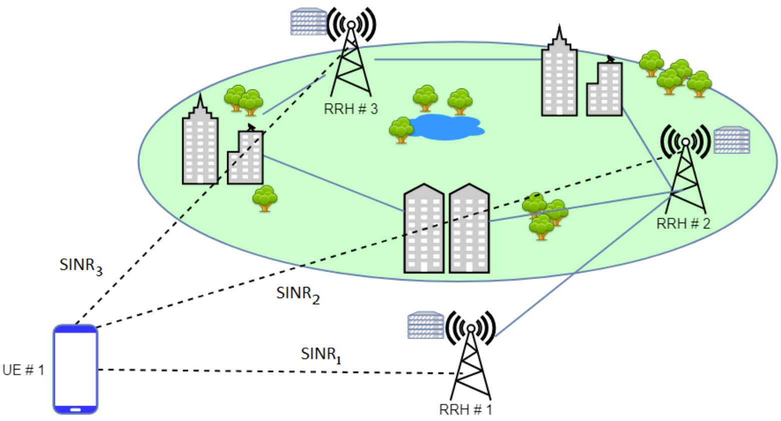


Figure 3.2: Illustration of user associations with SINR

traffic of service type s served by RRH r through path p as $t_p^{r,s}$, which must hold $\sum_{p \in P^r} t_p^{r,s} = T^{r,s}$, where $T^{r,s}$ is the total traffic of service type s served by RRH r (in bps).

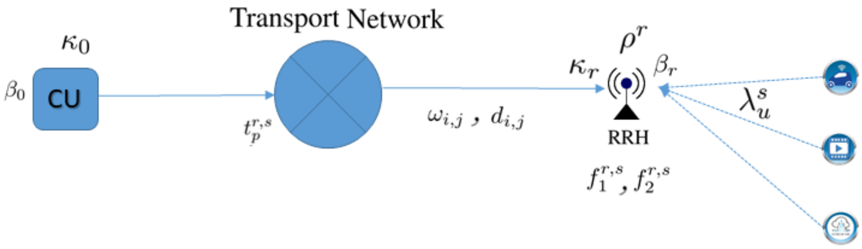


Figure 3.3: System model scheme

Based on Table 3.1 and the results found in [54], it can be seen that

$$T^{r,s} = f_1^{r,s} Q_1 - f_2^{r,s} Q_2 + (1 - f_1^{r,s}) Q_3 \tag{3.4.1}$$

where $f_1^{r,s}$ and $f_2^{r,s}$ are equal to 1 when function f_1 and f_2, respectively, are located at the RRH r for service s and they are equal to 0 otherwise, and

$$\begin{cases} Q_1 = \sum_{u \in U} 1.02 \lambda_u^s . x_u^{r,s} + 1.5 \\ Q_2 = \sum_{u \in U} 0.2 \lambda_u^s . x_u^{r,s} + 1.5 \\ Q_3 = 2500 \end{cases}$$

Equation (3.4.1) provides the bandwidth requirements for a given functional split. Thus, when functional split ϕ_1 is used, $f_1^{r,s} = f_2^{r,s} = 1$, and so $T^{r,s} = Q_1 - Q_2$. Conversely, for functional split ϕ_3, $f_1^{r,s} = f_2^{r,s} = 0$ and consequently $T^{r,s} = Q_3$. Therefore, the set of paths from CU to RRH r must be able to forward traffic of type s of at least $T^{r,s}$.

Routing also takes into account the delay of each link. In particular, slices with a functional split ϕ_n, with $n = \{1, 2, 3\}$, can only forward traffic over paths with an aggregate delay below the constraints shown in Table 3.1.

3.4.4 Computational Cost Model

The deployment of network functions either at the RRH or at the CU incurs a computational burden or cost. In the following, the computational cost in the RRH and in the CU is stated. Thus, the computational cost required at the RRH r is given by:

$$C_r = \beta_r \sum_{u \in U} \sum_{s \in \mathcal{S}} \lambda_u^s . x_u^{r,s} \left(f_1^{r,s} . c_1^{r,s} + f_2^{r,s} . c_2^{r,s} \right) \tag{3.4.2}$$

where $c_1^{r,s}$ and $c_2^{r,s}$ are the computational cost of each function located at the RRH in CPU operations per bit per second, and λ_u^s is the user transmission rate. We also use (monetary units per cycle) which is the average cost for serving each request; hence, for RRHs we set $\beta_r = 0.017$ and for CU $\beta_0 = 1$ as detailed in [54]. As for the CU,

$$C_0 = \beta_0 \sum_{r \in R} \sum_{s \in S} \sum_{u \in U} \lambda_u^s . x_u^{r,s} \left(c_1^{r,s}(1 - f_1^{r,s}) + c_2^{r,s}(1 - f_2^{r,s}) \right) \tag{3.4.3}$$

3.5 Problem Formulation

3.5.1 Optimization of Computational Cost

The main objective of this optimization model is to maximize the centralization degree, which is defined as the inverse of the computational cost, $CD = (\sum_{r \in \mathcal{R}} C_r + C_0)^{-1}$. As discussed in [54], the computational costs of RRHs and CU can not be directly compared. In general, deploying additional capacity in a RRH is more costly than doing it in the CU. This effect is considered in (3.4.2) and (3.4.3) by assuming $\beta_r > \beta_0$. Therefore, the minimization of the total cost is equivalent to the maximization of the CD. This is the objective function of the optimization problem, as shown in (3.5.1) as long as it holds the capacity constraints in the integrated FH/BH. With regard to constraints, constraint (3.5.2) ensures the computation capacity needed to process $f_1^{r,s}$ and $f_2^{r,s}$ is less than the available capacity in RRH r (κ_r).

Equation (3.5.3) is used to bound the maximum computational capacity supported by the CU (κ_0). Equation (3.5.4) is defined for the traffic of service type s served by RRH r through path p as $t_p^{r,s}$. Constraint (3.5.5) states that the flow from each RRH r to CU is bounded by the capacity of the links of the paths, denoted as $\omega_{i,j}$ (in bps). The binary variable $y_{i,j}^p$ is used to check if the path p includes link $l_{i,j}$ or not. Equations (3.5.6), (3.5.7), (3.5.8) as described in Section 3.4.3, are used for routing paths. Indeed each path is considered for the specific slices, and these paths have constraints in terms of delay requirements (ms). Note that M is a large positive number, to zeroize the paths with unacceptable delays for each configuration. Constraint (3.5.9) is explained in Section 3.4.2, and constraint (3.5.10) ensures that each user is allowed to be connected to only one RRH. As discussed, not all paths are feasible solutions for a given functional split. In particular, only paths with an aggregate delay below the maximum delay supported by the functional split will be considered. Thus, based on Table 3.1, we define for each RRH r the set of paths with a delay above 30 ms (the maximum delay allowed by split ϕ_1) as $P_r^{\phi_1}$. Analogously, $P_r^{\phi_2}$ as the set of paths with a delay larger than 2 ms and $P_r^{\phi_3}$ as the set of paths with a delay larger than 0.25 ms.

$$\max CD = \left(\sum_{r\in R} C_r + C_0\right)^{-1} \tag{3.5.1}$$

subject to:

$$\sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}} x_u^{r,s}\lambda_u^s\left(f_1^{r,s}c_1^{r,s} + f_2^{r,s}c_2^{r,s}\right) \leq \kappa_r,\, \forall r\in\mathcal{R} \tag{3.5.2}$$

$$\sum_{r\in R}\sum_{s\in S}\sum_{u\in U}\sum_{n=1}^{2} \lambda_u^s.x_u^{r,s}c_n^{r,s}(1-f_n^{r,s}) \leq \kappa_0 \tag{3.5.3}$$

$$\sum_{p\in P^r} t_p^{r,s} = T^{r,s},\, \forall r\in\mathcal{R}, \forall s\in\mathcal{S} \tag{3.5.4}$$

$$\sum_{r\in R}\sum_{s\in S}\sum_{p\in P^r} t_p^{r,s}.y_{i,j}^p \leq \omega_{i,j},\, \forall j\neq i\in\mathcal{Q} \tag{3.5.5}$$

$$\sum_{p\in P_r^{\phi_1}} t_p^{r,s} \leq M(2 - f_1^{r,s} - f_2^{r,s}),\, \forall r\in\mathcal{R}, \forall s\in\mathcal{S} \tag{3.5.6}$$

$$\sum_{p\in P_r^{\phi_2}} t_p^{r,s} \leq M(1 - f_1^{r,s} + f_2^{r,s}),\, \forall r\in\mathcal{R}, \forall s\in\mathcal{S} \tag{3.5.7}$$

$$\sum_{p\in P_r^{\phi_3}} t_p^{r,s} \leq M(f_1^{r,s} + f_2^{r,s}),\, \forall r\in\mathcal{R}, \forall s\in\mathcal{S} \tag{3.5.8}$$

$$\sum_{s\in S}\sum_{u\in U} x_u^{r,s}\rho_u^{r,s} \leq \rho^r,\, \forall r\in\mathcal{R} \tag{3.5.9}$$

$$\sum_{r\in R} x_u^{r,s} = 1,\, \forall u\in\mathcal{U}, \forall s\in\mathcal{S} \tag{3.5.10}$$

$$f_1^{r,s}, f_2^{r,s} \in \{0,1\},\, \forall r\in\mathcal{R}, \forall s\in\mathcal{S} \tag{3.5.11}$$

3.5.2 Optimization of Throughput

As explained in Section 3.4.3, the aggregation of the traffic in each RRH defines the throughput of the whole network. The problem defined in Section 3.5.1 can be converted from the maximization of the CD into the maximization of the throughput by changing the objective function (3.5.1) for constraint (3.5.4).

3.5.3 Linearization of the Problem

In general, the method to linearize non-linear problems, where constraints include multiplication of variables, depends on the kinds of variables involved in the constraints. For the case of binary variables, for instance, if we have a multiplication of binary variables, say x, y, it can be solved by adding another binary variable, z, that holds: $z \geq 0, z \geq x + y - 1$ and $z \leq x, z \leq y$. Since our optimization problem is a Mixed Integer Non-Linear Programming (MINLP) problem, it can be converted into a linear problem (i.e, MIP), by defining new binary variables $z_1^{u,r,s}$ and $z_2^{u,r,s}$ to model linearly the multiplication of binary variables of $f_1^{r,s}.x_u^{r,s}$, $f_2^{r,s}.x_u^{r,s}$ stated in (3.4.1), (3.5.2), and (3.5.3). This formulation extends naturally to more indices, and it must hold the following constraints:

$$x_u^{r,s}, f_1^{r,s}, f_2^{r,s}, z_1^{u,r,s}, z_2^{u,r,s} \in \{0, 1\} \tag{3.5.12}$$

$$\begin{cases} z_1^{u,r,s} \leq x_u^{r,s} \\ z_1^{u,r,s} \leq f_1^{r,s} \\ z_1^{u,r,s} \geq x_u^{r,s} + f_1^{r,s} - 1 \end{cases} \tag{3.5.13}$$

$$\begin{cases} z_2^{u,r,s} \leq x_u^{r,s} \\ z_2^{u,r,s} \leq f_2^{r,s} \\ z_2^{u,r,s} \geq x_u^{r,s} + f_2^{r,s} - 1 \end{cases} \tag{3.5.14}$$

3.6 Performance Evaluation

In this section, we explore joint optimization of functional split and slicing with maximization of centralization degree (*i.e.,* minimization of computational cost). For the numerical analysis, Monte Carlo simulations have been run for the averages of 100 times to get statistical significance. In order to highlight the impact of joint slicing and functional split optimization, users of three different services have been considered. The first service has a data rate uniformly distributed in the range of 10-1000 kbps. This service can only be supported with functional split ϕ_1. The second service has a data rate in the range of 10-50 Mbps and can be supported by any of the three functional splits. Finally, the last service has a data rate within the range of 70-200 Mbps, and it can only be served with functional split ϕ_3. The scenario is composed of a single CU connected to a set of 10 RRHs. As for the computational capacity, we utilized the values used in [54], with $\kappa_0 = 100$, $\kappa_r = 1$ CPU Reference Core per Gbps. Regarding the computational cost, $c_1^{r,s} = 3.25$ and $c_2^{r,s} = 0.75$ CPU reference core per Gbps. We obtain results after running 100 times for two metrics of minimizing the computation cost and maximizing the throughput. Results are compared where no slicing is considered. Therefore, a single functional split is applied to each RRH. In the sequel, the optimization proposed in this chapter is denoted by $Multi - Splits$ and the average cost and the average throughput for the first, second, and third splits, when no slicing is considered, are denoted by $split_1$, $split_2$, $split_3$, respectively. It is worth mentioning that in each scenario we set all RRHs to a single functional split in order to compare them with $Multi - Splits$.

In Fig. 3.4, we first present the computational cost. The proposed algorithm presents a higher computational cost than $split_1$, $split_2$, $split_3$ (*i.e.,* a lower centralization degree). This increase in the computational cost can be justified with the inherent flexibility of $Multi-Splits$. In particular, $Multi-Splits$ aims to centralize network functions as long as network constraints (FH/BH capacity and delay, and available computational capacity in RRHs and CU) permit it. This decision is made on a slice basis. On the contrary, each of $split_1$, $split_2$, $split_3$, is configured for RRHs. With this, users that can not be served with the functional split deployed in the RRH will have to be dropped. For $split_3$, it is expected to have the computation cost lower than $Multi-Splits$ as shown in Fig. 3.4. However, for $split_1$ and $split_2$, for all of the numerical evaluations, we found that both splits give close results and with a big difference with $Multi-Splits$. The existing difference between the computation cost of these splits with $Multi-Splits$ is due to dropping more users for $split_1$ and $split_2$.

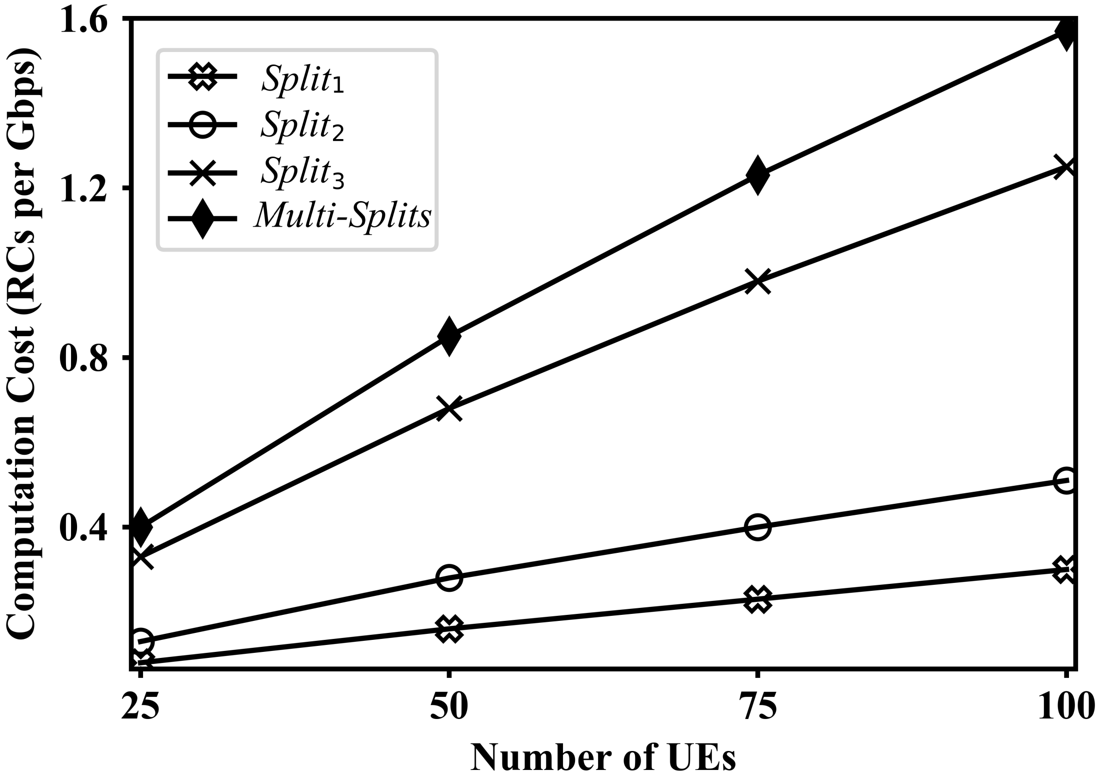


Figure 3.4: Computation Cost analysis as a function of the number of users.

The behavior explained above is translated into a decrease of the throughput in the $split_1$, $split_2$, $split_3$, as it can be observed in Fig. 3.5. It shows that with $Multi-Splits$ more throughput is achieved due to serving more users with different types of services.

Finally, Fig. 3.6 shows the throughput/computation cost for 30, 60, 90, 120 and 150 users. In particular, for a given algorithm (*i.e.,* $Multi-Splits$, $split_1$, $split_2$, and $split_3$), the number of users increases from left to right. This figure shows how the proposed algorithm can achieve different trade-offs between the computation cost (*i.e.,* the inverse of centralization degree) and the throughput when the range of services covers completely different Quality of Service requirements.

The trade-off between these two metrics can be clearly seen: as expected, as the number of users

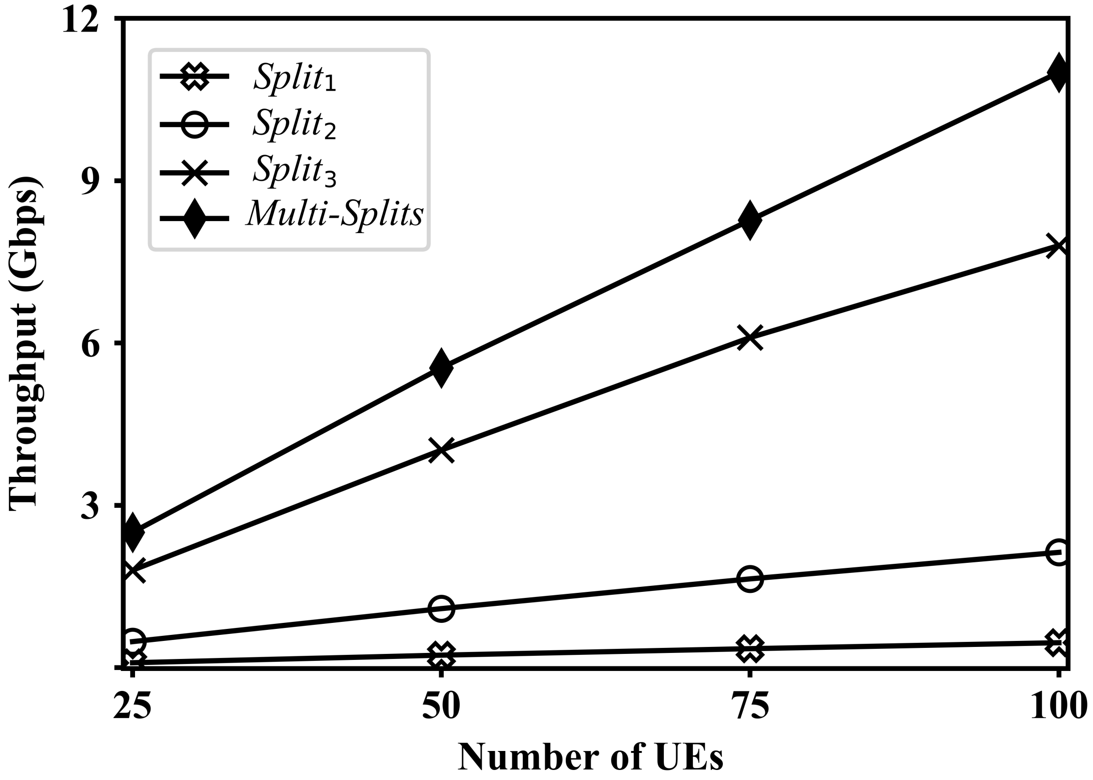


Figure 3.5: Throughput analysis as a function of the number of users.

increases, the computation cost also increases. Likewise, throughput can only be improved at the expense of additional computational costs (decentralization). This improvement is directly related to the type of services of users which means the traffic type with a higher amount, the more throughput can be achieved. As it is shown in this figure, the slope of $Multi - Splits$ is much higher than the slope of the *split*$_1$, *split*$_2$, *split*$_3$. This means that the computation cost increase required to increase the throughput is smaller for $Multi - Splits$. For example, for *split*$_2$, the increase in throughput is about 400 (Mbps) (*i.e.,* 0.4 Gbps) whereas this increase is about 12 (Gbps) for $Multi-Splits$. Hence, $Multi - Splits$ outperforms the increase in throughput when compared with single functional split cases (*i.e., split*$_1$, *split*$_2$, *split*$_3$).

3.7 Discussion

In the aforementioned chapter, the joint RAN slicing and functional splitting, we see a trade-off between the computation cost and the throughput. However, the density of the RAN — defined as the number of eNBs per area unit — impacts the cost of deployment (larger number of complex RAN nodes). In other words, is the network performance improvement in terms of throughput and served users achieved by joint slicing and functional splitting per slice worth the added cost?

The advantage of instantiating multiple slices and functional splits on a shared RRH lies in reducing the distance between users and serving RRHs. As services can only be served with a subset of

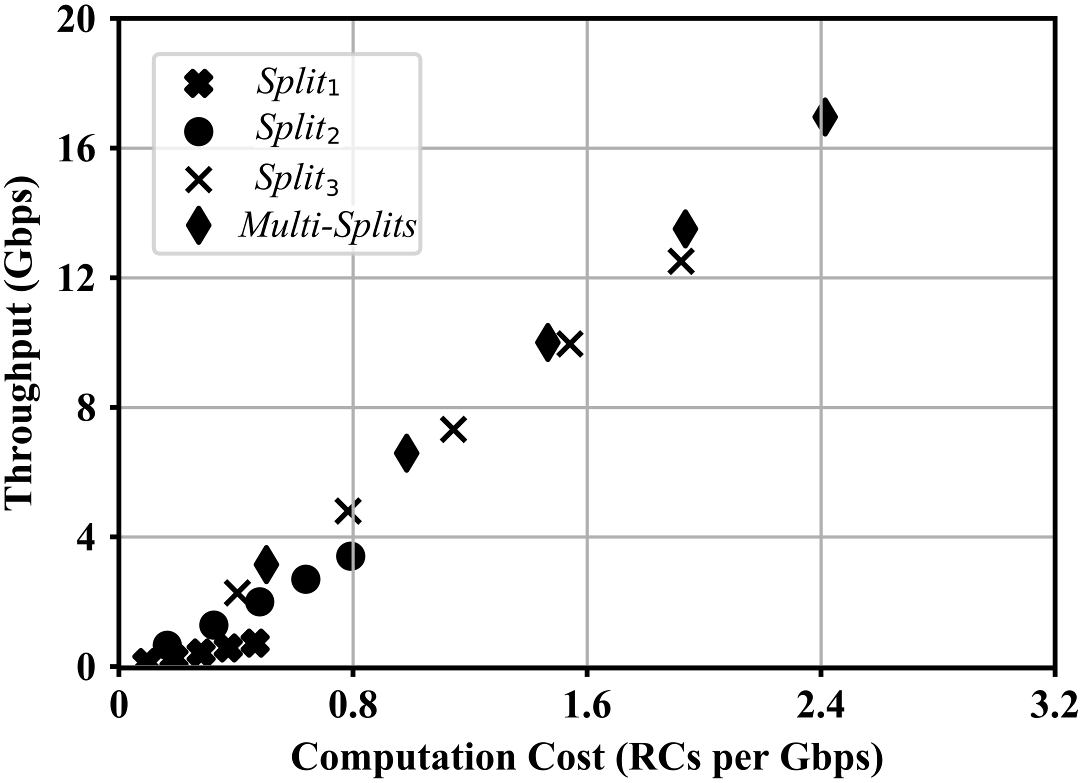


Figure 3.6: Throughput vs Computation Cost trade-off for 30, 60, 90, 120 and 150 users respectively.

functional splits, only the subset of RRHs with a proper functional split can accommodate the slice. Modeling this effect is equivalent to removing the subset of RRHs with an inappropriate functional split from the set of feasible RRHs for instantiating the slice.

We will investigate the impact of an increasing number of RRHs, *i.e.,* densifying network on joint functional split and RAN slicing in the next chapter.

3.8 Conclusion

RAN slicing along with functional split are two major concepts of future 5G networks. In this chapter, we proposed $Multi-Splits$, which is a joint slicing and functional split optimization framework for 5G not yet fully investigated. We formulated this problem as a Mixed Integer Programming (MIP) to jointly optimize the centralization degree and throughput. Our results show that there is a trade-off between the centralization degree and the throughput. According to our results, computational cost in $Multi-Splits$ is higher than the one in $split_1$, $split_2$, $split_3$, which means that the centralization degree is lower than in $split_1$, $split_2$, $split_3$. However, this is compensated by the increase in the throughput. Furthermore, in $Multi-Splits$ the throughput increase is much higher and needs less increase in the computation cost. Based on our observations throughout this work, in the next chapter, we will aim to extend this work by assessment of the system performance in terms of throughput through various experimental results and considering different users' Service Level Agreements (SLAs). Another direction of the next chapter is to analyze the performance of the dynamic joint functional split and RAN slicing algorithm in scenarios with different RAN node densities.

Chapter 4

Service-Aware Network Slicing Framework for 5G RAN

According to the State of the Art (SoA) in RAN architecture explored in chapter 2 and our finding in chapter 3 with respect to the 5G service requirements, an efficient RAN slicing is needed to meet the 5G respective requirements where RAN slices need to be adjusted with the so-called users' Service Level Agreement (SLA), which brings about paramount challenges in 5G networks. Furthermore, it has been observed in the investigation of the functional split contributed in chapter 3, in the new architecture of RAN, some network functionalities can be decoupled and pooled at CUs, where the optimal distribution of RAN functions still remains an open research issue.

In this chapter in line with the theme of this thesis, the concept of RAN slicing and virtualization based on the SLA of different services have been utilized to address the principles of RAN slicing. In this way, the slicing approach would be service-aware slicing since slice creation is done based on different use cases. Meanwhile, we aim to find optimal functional split based on taking advantage of the newly introduced concept of small cells vitalization functional splits to offload functions from RRHs flexibly [17].

4.1 Introduction

The fifth generation (5G) of mobile communications is designed to serve various types of demanding services with extremely different Quality of Service (QoS) requirements. The International Telecommunication Union (ITU) categorizes 5G mobile network services into three main types [5]: i) Enhanced Mobile Broadband (eMBB) is the set of services that need higher-bandwidth, such as High Definition (HD) videos, Virtual Reality (VR), and Augmented Reality (AR), ii) ultra-Reliable and Low Latency Communications (uRLLC) characterizes the range of services demanding low latency and more reliable mobile services, such as industrial Internet, remote surgery, assisted or automated driving, and iii) massive Machine Type Communications (mMTC) is designated for the services that require high connection density though with relaxed latency and throughput requirements, such as smart city and smart agriculture applications. It has been largely proven in the literature that the traditional *one-size-fits-all* approach to mobile network infrastructure is unable to deal with the expected wide range of services and the extremely different QoS requirements of 5G [7]. In order to be able to serve this traffic, virtualization emerges as an essential component at the network edge, namely the virtual partitioning of the mobile Radio Access Network (RAN). In 5G, virtualization will do away with the notion of communications built upon a single rigid mobile network infrastructure, opening up the potential for different types of supported services such as eMBB, uRLLC, and mMTC services [11, 32]. Through virtualization, Mobile Network Operators (MNOs) will be able to create on-demand isolated slices on top of the physical network to support various use cases, such as the Internet of Things (IoT), automated cars, streaming video, remote health care, etc [77, 78]. Furthermore, virtualization enables MNOs to create virtual networks for those applications that possess separate blends of performance, capacity, latency, security, reliability, and coverage.

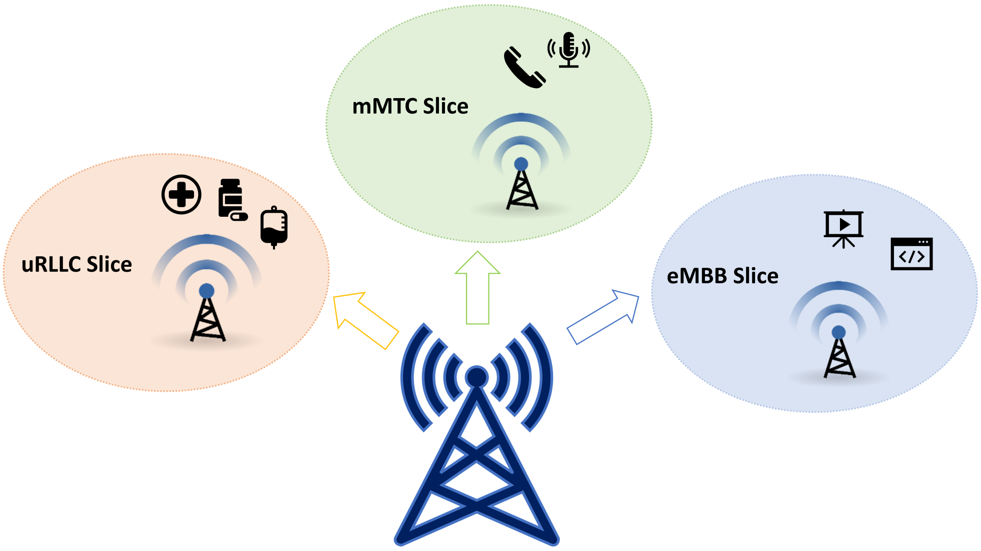


Figure 4.1: RAN Slicing to meet different application requirements.

Indeed virtualization and slicing cover two main parts of 5G networks, namely core slicing and RAN slicing. The study on mobile core slicing focuses on the virtualization of core network functions [79, 80], combined with the use of mature virtualization technologies (*e.g.,* KVM [64], Docker [81]), thus enabling scalability [82] and [83] augmenting flexibility. While the concept of RAN slicing focuses on virtualizing RAN functions and resources, thus facilitating the sharing of eNodeBs (eNBs) among different slices and serving different functionalities with a customized slice [59, 60]. The process of RAN slicing required to adapt the RAN to UEs' performance requirements poses significant challenges in 5G networks. To fully meet these application demands, RAN architecture must be flexible enough to adapt the network to such diverse requirements (See Figure 4.1). Centralized/Cloud RAN (C-RAN) has emerged as an architecture to improve performance thanks to its ability to coordinate between access nodes, while it is cost-efficient due to resource pooling. In 5G, C-RAN will be composed of a Central/Cloud Unit (CU), and a set of geographically distributed Remote Radio Heads (RRHs) connected through a packet-based network (*i.e.,* integrated Fronthaul/Backhaul (FH/BH)) as proposed by 3GPP [13].

4.2 State of the Art

More recently, a flexible design approach is suggested for C-RAN, where the optimal distribution of BS functions between the CU and the RRHs (known as functional split) are challenging [84–86]. This architecture determines the amount of functions left locally at the RRHs, and the amount of functions centralized at a high-processing CU. A proper choice of functional split depends on the capacity of the FH/BH network, as the centralization of the RAN functions imposes strict capacity requirements in the FH/BH network. This renders the design of the FH/BH network even more complicated due to the virtualization and capability of having multiple split choices per RRH.

Given the importance of Software Defined Networking (SDN) as an enabler for both virtualization [38] and slicing [39], some recent works have focused on RAN virtualization platforms and slicing designs. Foukas et al. propose FlexRAN [59], a flexible and programmable Software-Defined RAN (SD-RAN) platform. The platform is composed of a centralized controller and one agent per eNodeB (eNB) that separates control and data planes and allows a flexible control plane design. This architecture enables the dynamic allocation of control functions between the centralized controller and the decentralized agents, thus tailoring the RAN to meet the application performance requirements. However, despite making a step forward in the direction of the virtualization of the RAN, the proposal still lacks a slicing design. The same authors have also proposed Orion [60], which is a RAN slicing design running on the FlexRAN platform that guarantees functional isolation among slices. Isolation of functions among slices is of paramount importance since it allows a slice-custom functional split within a single shared eNB, *i.e.,* different slices sharing the same physical node can be configured with different functional splits. For instance, in a given eNB, a slice serving a high-speed UE better suits a centralized functional split, so that the coordination among neighboring cells is tighter and the handover performance can be simplified. Conversely, the slice serving a low latency UE would require decentralized functional splits to reduce the HARQ delay. This is the main weakness of [14, 15, 61], where for simplicity, either no slicing is considered, or all slices adopt the same functional split. Two notable recent works propose Wizhaul [58] and FluidRAN [54] to address the functional split optimization. More specifically, WizHaul [58] formulates a joint routing and functional split optimization to maximize the Centralization Degree (CD) of the network, *i.e.,* the network functions placed at the CU according to the availability of the network resources. Similarly, FluidRAN [54] follows the same rationale but targeting at monetary cost minimization. Table 4.1 presents the abstract of the most related works and compares their contributions, methodologies, and characteristics.

However, despite their insightful conclusions, the slicing option in the RAN is neglected in both of these works. Given the complex and diverse set of QoS requirements of services that 5G will have to serve, the approaches proposed in [54, 58] that optimized the centralization degree, tend to prioritize high throughput services. This prioritization, which made sense in previous mobile technologies, is no longer convenient for 5G, where the diverse requirements must be met. In order to overcome this aspect, different minimum Service Level Agreements (SLAs) for each service have to be investigated.

In this context, our recent work in [89] proposes a joint routing (from UE to CU) and functional split optimization while considering different slices, and shows that there is a trade-off between CD and the throughput in the network.

Table 4.1: The Comparison of Related Work

Literature	Methodology & Contribution	Characteristics				
		C-RAN	FS	FH/BH	Slicing	SLA
[84]	Flexible FS and a FH transport protocol	✓	✓	-	-	-
[85]	Integrated transport solution to find FS	✓	✓	✓	-	-
[86]	FS, MEC placement, and optimal routing	✓	✓	✓	-	-
[59]	Flexible SD-RAN platform with a centralized controller	✓	✓	✓	-	-
[60]	RAN slicing design running on the FlexRAN platform	✓	✓	-	✓	-
[14, 15, 61]	Flexible FS and a FH transport network	✓	✓	✓	-	-
[87, 88]	Network slicing for C-RAN resources	✓	-	-	✓	-
[11, 32]	Maximize C-RAN operator's revenue	✓	-	-	✓	-
[77]	Optimizing the capacity and traffic allocation	-	-	-	✓	✓
[58]	Joint optimal FS, and routing	✓	✓	✓	-	-
[54]	Joint RAN/MEC solution	✓	✓	✓	-	-
[89]	Joint slicing and optimal FS	✓	✓	✓	✓	-
SlicedRAN	Service-aware network slicing framework	✓	✓	✓	✓	✓

4.3 Contribution

In this work, we extend the work presented in chapter 3 and design SlicedRAN, a service-aware network slicing framework for 5G RAN, which covers the optimization of routing and functional split while maximizing the throughput and setting minimum SLA thresholds for each service. The work not only proposes a joint slicing, functional split, and routing solution, but it is also intended to gain insight into how the RAN should be designed and the interweaving of the different RAN aspects.

Our main contributions are summarized as follows:

- We propose MIP optimization problem that is already defined in Chapter 3. This problem is known as NP-complete problem, hence, we further propose an effective heuristic, named SlicedRAN, which is based on Relaxation Induced Neighborhood Search (RINS) heuristic, that obtains near-optimal solutions in a very short computing time.

- We design a heuristic algorithm for RAN slicing, SlicedRAN, which covers optimal slicing and customized functional split per slice by the optimization of the throughput. To the best of our knowledge, this is the first work proposing an analytical framework for the SlicedRAN design by considering the bottlenecks in the capacity of RRHs, FH/BH network capacity and along with a minimum level of SLA for each slice imposed by the main type of uRLLC, mMTC, and eMBB services with different QoS requirements.

- We elucidate how to solve the problem of providing isolated and tailored slices for different services with customized functional splits per slice when CU and RRHs are connected through a FH/BH network. Whereas existing proposals assume a single functional split per RRH [58], SlicedRAN leverages virtualization to create multiple slices in RRHs, each one with the most appropriate functional split to meet the requirements of the slice. Slices are created based on the QoS of traffic demand and the set of RAN constraints, such as RRHs' computational capacity, the capacity of the FH/BH network, spectrum availability per RRH, etc.

- Given the diversity of QoS requirements, it is shown that the maximization of the throughput could degrade the performance of the slices with the lowest throughput requirements. In order to overcome this issue, SlicedRAN analyses the cross effects between different slices when minimum SLAs are imposed in each slice.
- We perform an extensive simulation study to investigate the limits of the network in terms of the capacity of RRHs and the capacity of FH/BH network, while we also evaluate the impact of imposing minimum SLAs on the network where these limits remain.

4.4 System Model

In this section, we model the traffic and the RAN, including the CU, the RRHs, and the integrated FH/BH network connecting them. Likewise, the FSs and the traffic routing in the network are described, and finally, the associated constraints are defined.

4.4.1 Radio Access Network

The initial C-RAN concept is to apply a single direct FH link to connect each RRH to the BBU pool (equivalently, CU). However, due to concerns to scalability, CAPEX, and multiplexing, it is expected that the FH will evolve towards more complex and shared topologies which have been comprehensively explored in [15, 90]. In this work, we focus our discussion on fully connected network topology and present a simple but realistic deployment of the C-RAN network topology which is composed of a CU, a set of RRHs, and an integrated packet-based FH/BH network (often known as crosshaul [91]), which is a set of forwarding nodes (*i.e.,* routers) connecting CU and RRHs, as introduced in [13]. However, our framework can be applied to different network topologies by modifying the capacity of specific links in the network.

Following the model presented in the previous chapter, we define a C-RAN architecture as a graph topology $G = (\mathcal{I}, \mathcal{Q}, \mathcal{L})$, where $\mathcal{I}$ is the superset of CU (node 0) and the set of R RRHs, $\mathcal{Q}$ is the set of forwarding nodes (*i.e.,* routers), and $\mathcal{L}$ is the set of links $\mathcal{L} = \{l_{i,j} : i, j \in \mathcal{I} \cup \mathcal{Q}\}$ connecting these elements whose vertices can be divided into two disjoint sets $\mathcal{I}$ and $\mathcal{Q}$, that is, $\mathcal{I}$ and $\mathcal{Q}$ are each independent sets such that every edge connects a vertex in $\mathcal{I}$ to one in $\mathcal{Q}$ (see Fig. 4.2 (a)). Vertex sets $\mathcal{I}$ and $\mathcal{Q}$ are often known as bipartite [92] sets. Accordingly, the set of forwarding nodes is defined as $\mathcal{Q} = \{1, \dots, Q\}$; and the set of RRHs, $\mathcal{R} = \{Q+1, \dots, Q+R\}$. Each link $l_{i,j} \in \mathcal{L}$ has a capacity equal to $\omega_{i,j} \geq 0$ (in b/s). Fig. 4.2 (a) shows the layout of the RAN, where forwarding nodes are organized as a matrix of m rows and Q/m columns. In our analysis of dimensioning network, we consider two network topology models: i) A bipartite graph network topology: where $\mathcal{I}$ sets are connected via one layer of $\mathcal{Q}$ sets in which each CU-RRH path is comprised of a set of $CU - Q - R$ nodes, hence, having a column of forwarding nodes. ii) A k-partite graph network topology: where $\mathcal{I}$ sets are connected via k layers of $\mathcal{Q}$ sets and each CU-RRH path is comprised of a set of $CU - Q_1 - Q_2, ..., Q_k - R$ nodes, thus, as shown in Fig. 4.2 (a), including k-layers of forwarding nodes. Note that this general architecture does not preclude other architectures. The FH/BH architecture considered in the model is the more general case that can be modified by setting different link capacity values (e.g. setting a link capacity to $\omega_{i,j} = 0$), thus the topology can be changed without modifying the model. That is, the model also supports these other topologies. The connection between the CU and an RRH can be realized through multiple paths [54], each path including several links. Given that there might exist multiple paths between CU and RRH r, the set of all possible paths from CU to RRH r is denoted by $\mathcal{P}^r$. Note that this general architecture does not preclude other architectures. The FH/BH architecture considered in the model is the more general case that can be modified by setting different link capacity values (e.g. setting a link capacity to $\omega_{i,j} = 0$), thus the topology can be changed without modifying the model. That is, the model also supports the other topologies.

The computational capacity of the CU and the RRHs is limited and expressed as κ_r for $\forall r \in \mathcal{R}$ and κ_0 for CU. As for the bandwidth allocated to RRHs, we define ρ^r as the number of Physical Resource Blocks (PRB) allocated to RRH r (See Table 4.2). For the sake of simplicity, as used in [93], in the following, we assume equal transmitted power per PRB with a distance-dependent path-loss model, as for the Signal-to-Noise Ratio (SNR) and for the Modulation and Coding Scheme (MCS), we adopt the models used in [94].

4.4.2 Traffic model

In our system model, we focus on the DownLink (DL) traffic, however, our study could be extended to include UpLink (UL). The set of UEs is denoted by $\mathcal{U}$, and the cardinality of the set is expressed by U. Each UE demands a service type $s \in \mathcal{S}$, which is mainly characterized by a required data rate. Thus, the data rate required by UE u with service s is denoted by λ_u^s. We also denote the total number of UEs with service of s as η^s. These demands at each RRH create an aggregate flow emanating from the CU routed to RRH. Hence, the RAN operation can be modeled as a multi-commodity flow problem where the flows rely on the FS at each RRH.

4.4.3 Functional Splits

The protocol stack in an eNB consists of several layers, each one responsible for a specific function or a set of functions [17]. Indeed, the whole operation of the eNB can be modeled as a chain of these functions. In this context, the FS can be defined as the distribution of functions/layers between the CU and the RRH. 3GPP has proposed in [13] a wide range of possible granularities for the FS, from the coarsest granularity (the FS is determined based on the computational capacity of the RRH and the CU, as well as on the FH/BH network capacity) to the finest granularity (the FS is decided on a UE, bearer or slice basis).

Without precluding any of the granularity levels proposed by 3GPP, in our work, we focus on the slice-based FS, assuming that one slice is created for each service[1]. As shown in [13], the network layers Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC) (high and low sublayers), Medium Access Control (MAC) (high and low sublayers), and Physical Layer (PHY) (high and low sublayers) can be allocated either in the CU or in the RRH. Accordingly, each FS will be defined by the set of functions allocated in the CU and the set of functions allocated in the RRH.

In the sequel, we will assume a set of four network functions, denoted as $\mathcal{F} = \{f_0, f_1, f_2, f_3\}$, where

[1] Given that slicing will be done on a service type basis as highlighted by 3GPP [10], hereafter service and slice concepts will be interchangeable.

f_0 is the low layer network function (RF, signal and analog processing, etc.), which is always placed in the RRH; f_1 serves all PHY functions except for function f_0; f_2 corresponds to RLC and MAC; and f_3 is the high layer network function (*e.g.,* PDCP and above layers). Depending on the FS, these functions will be allocated either at the RRH or at the CU, and thus defining the FH/BH bandwidth requirements between the CU and RRHs. Note that we focused on the main types of FS options, which are the key splits as discussed in [17]. Thus, the addition of other FS options in our work would not affect the system model. We use Table 3.1 presented in the previous chapter that describes the allocation of functions and the associated FH/BH bandwidth requirements for each split [54]. In principle, regardless of the adopted FS, f_0 is always placed in RRH and f_3 is in CU, thus generating three different FS options, namely split 1, split 2, and split 3. Split 1 is a completely decentralized FS that accommodates all functions except f_3 at the RRH. That is, all layers below PDCP run in the RRH. Given the allocation of functions, this split does not have traffic overhead and the required FH/BH capacity can be approximated by the aggregate UEs' traffic. In split 2, f_2 is moved from the RRH to the CU, thus leaving only f_0 and f_1 in the RRH (see Fig. 4.2 (b)). This allows a higher degree of coordination among eNBs sharing the same CU, thus enabling better utilization of resources with techniques such as Coordinated MultiPoint (CoMP), frame alignment, and centralized HARQ. However, split 2 allocation imposes higher traffic overhead than split 1. Finally, in split 3, only the RF function is located at the RRH, while the rest of functions are moved to CU (complete centralization), thus transmitting In-Phase and Quadrature (IQ) samples through the FH/BH. In this case, samples are usually encapsulated with Common Public Radio Interface (CPRI) [76] and the required fronthaul capacity depends on the bandwidth allocated to the eNB, the number of antennas, etc. That is, fronthaul capacity requirement does not depend on the UEs' traffic for split 3. The main advantage of split 3 is that the centralization achieves the highest coordination degree among eNBs. Note that processing functions has a cost and needs Central Processing Unit (CPU) processing resources. We use c_1 and c_2 as the CPU computational costs for f_1 and f_2 (CPU reference core per Gb/s).

Given the described scenario, the network creates different slices on top of the physical RAN to serve the traffic. Fig. 4.2 (b) conveys an example of a slice created to serve a UE u with service s. The slice is created across the FH/BH network (through one or several paths) and an RRH r, from the CU to the UE. Depending on the service and the required QoS, the FS will be 1, 2, or 3. Note, however, that the set of forwarding nodes, the links, and the RRH can be shared with other slices.

4.4.4 Traffic Routing in the RAN

Given that the traffic served by RRH r can be forwarded through any of the paths in $\mathcal{P}^r$, we define the traffic over one of these paths as t_p^r, where $p \in \mathcal{P}^r$. Therefore, the total traffic served by RRH r can be expressed as $\sum_{p\in\mathcal{P}^r} t_p^r$. Similarly, as discussed in subsection 4.4.3, the traffic that traverses the FH/BH network depends not only on the traffic received/transmitted by/from the UEs but also on the FS. Thus, a UE served by RRH r generating a traffic λ_u^s causes a traffic through the FH/BH network equal to $T_u^{r,s} = \alpha_{\phi^{r,s}}\lambda_u^s + \beta_{\phi^{r,s}}$, where λ_u^s is the traffic generated by UE u with service s and $\phi^{r,s}$ is the FS used in RRH r for service s. In general, $\phi^{r,s}$ can take values in $\{1,2,3\}$. However, due

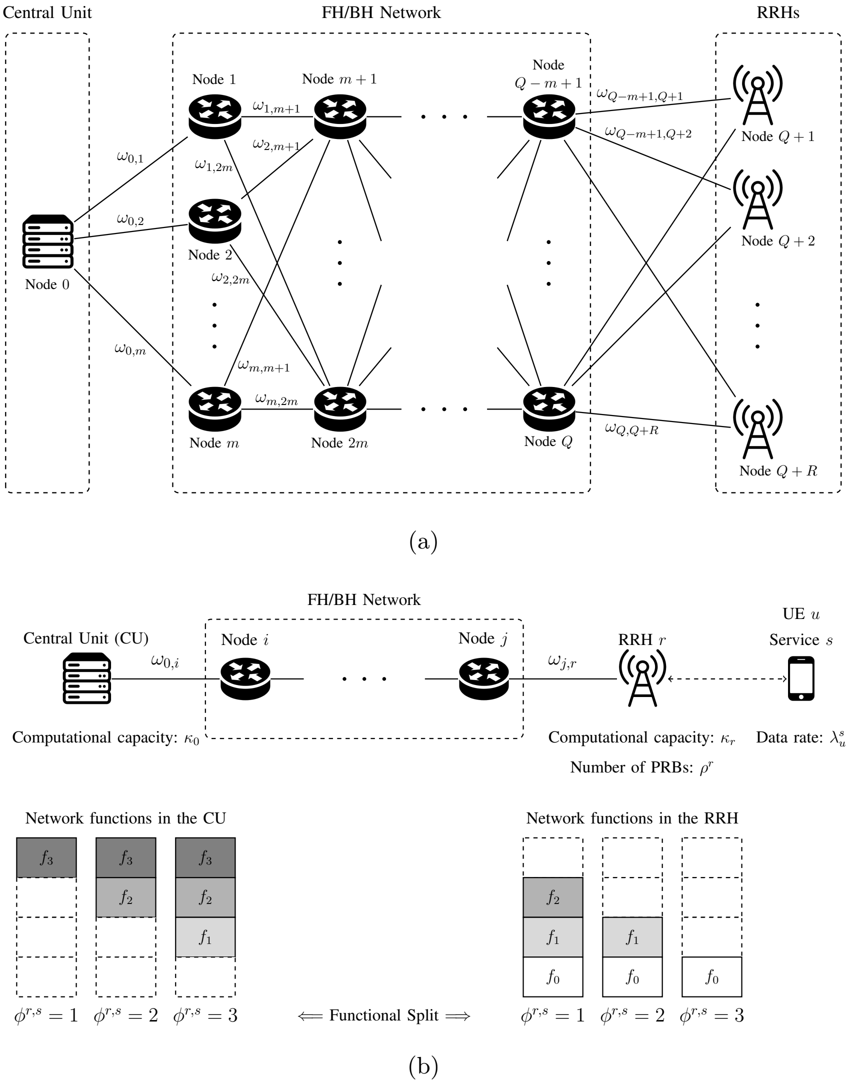


Figure 4.2: (a) Radio Access Network model; (b) Scheme of a slice created over a path across the FH/BH network from the CU to the UE u with service s.

Table 4.2: Summary of Notations

Symbol	Description
Sets	
$\mathcal{R}$	Set of RRHs
$\mathcal{Q}$	Set of forwarding nodes
$\mathcal{U}$	Set of UEs
$\mathcal{F}$	Set of network functions
Parameters	
$\omega_{i,j}$	Total bit-rate capacity of link $l_{i,j}$ (b/s)
ρ^r	Available physical resource block (PRB) at RRH_r
$\rho_u^{r,s}$	Required PRB of UEs to connect to RRH r
λ_u^s	Transmission rate of UE u with type s (b/s)
$T^{r,s}$	Traffic served by slice s of RRH r
c_1	CPU consumption to compile f_1 (RCs per Gb/s)
c_2	CPU consumption to compile f_2 (RCs per Gb/s)
κ_r	Computation capacity of each RRH (RCs per Gb/s)
κ_0	Computation capacity of CU (RCs per Gb/s)
τ^s	Proportion of UEs (*i.e.,* SLA) for slice/service type s
η^s	Number of UEs with slice/service type s
w	Bandwidth of a PRB (KHz)
Variables	
$x_u^{r,s}$	Binary variable to associate UE u with type s to RRH r
t_p^r	The variable to show the traffic routing from CU to RRH r
$f_n^{r,s}$	The variable to indicate the placement of functions
$y_{i,j}^p$	The variable to indicate if the path p includes link $l_{i,j}$

to the constraints imposed by QoS requirements, each service s can only use a subset of FSs, denoted as Φ^s. Thus, $\phi^{r,s} \in \Phi^s \subseteq \{1,2,3\}$. As for $\alpha_{\phi^{r,s}}$ and $\beta_{\phi^{r,s}}$, they are coefficients and used to properly calculate the traffic load in the FH/BH network, and depend on the FS used in RRH r for service s, i.e. $\phi^{r,s}$. As observed in Table 3.1, when service/slice s uses FS 1, i.e. $\phi^{r,s} = 1$, we have $\alpha_1 = 1$ and $\beta_1 = 0$. In the case of split 2, $\alpha_2 = 1.02$ and $\beta_2 = 1.5 \cdot 10^6$ b/s. Finally, in split 3, $\alpha_3 = 0$ and $\beta_3 = 2.5 \cdot 10^9 \cdot \frac{\rho_u^r}{100}$ b/s[2], where ρ_u^r is the bandwidth allocated to UE u at RRH r expressed in the number of PRBs. Accordingly, the FH/BH transmission rate of slice 3 depends on the bandwidth allocated in the RRH for this slice.

According to the definitions stated above, the traffic traversing the FH/BH network to serve UEs with service s connected to RRH r is given by

$$T^{r,s} = \sum_{u \in \mathcal{U}} x_u^{r,s} \cdot T_u^{r,s}, \tag{4.4.1}$$

where $x_u^{r,s} \in \{0,1\}$ is a binary variable equal to 1 when UE u requires service s and is served by RRH r, and 0 otherwise. Each RRH can run different slices and serve different services[3] simultaneously. Thus, if we define the traffic served by slice s of RRH r as $T^{r,s}$, the total traffic served by RRH r can be expressed as $\sum_{s \in \mathcal{S}} T^{r,s}$. Therefore, it holds that

$$\sum_{p \in \mathcal{P}^r} t_p^r = \sum_{s \in \mathcal{S}} T^{r,s}. \tag{4.4.2}$$

[2] According to literature, the required transmission rate required for 20 MHz bandwidth (i.e. 100 PRBs) is around 2.5 Gb/s. This is the reason why the number of PRBs is normalized with respect to 100 PRBs

[3] Please, recall that *service* and *slice* are used interchangeably, and we assume that the network creates a slice per each service.

The accommodation of functions f_0, f_1, f_2 and f_3 in the RRH r or in the CU depends exclusively on the adopted FS. We define the set of variables $f_n^{r,s} \in \{0,1\}$ for $n = \{0,1,2,3\}$. If function f_n runs in the RRH r for service/slice s, then $f_n^{r,s} = 1$. Conversely, if it runs in the CU, then $f_n^{r,s} = 0$. By inspecting Fig. 4.2 (b) and Table 3.1, it can be shown that, when service s uses split 1, then $f_0^{r,s} = f_1^{r,s} = f_2^{r,s} = 1$ and $f_3^{r,s} = 0$. If it uses split 2, then $f_0^{r,s} = f_1^{r,s} = 1$ and $f_2^{r,s} = f_3^{r,s} = 0$. Finally, if it uses split 3, then $f_0^{r,s} = 1$ and $f_1^{r,s} = f_2^{r,s} = f_3^{r,s} = 0$. Thus, in general, when service/slice s uses FS $\phi^{r,s}$,

$$f_n^{r,s} = \begin{cases} 1 & \text{if } n \leq 3 - \phi^{r,s} \\ 0 & \text{otherwise} \end{cases}, \tag{4.4.3}$$

where $n = \{0,1,2,3\}$ and $\phi^{r,s} = \{1,2,3\}$. Note that allocating a function either in the CU or in the RRH has a computational cost. In the sequel, as described in Section 4.4.3, the computational cost of function f_n, for $n = \{0,1,2,3\}$, is denoted by c_n.

4.5 SlicedRAN: Service-Aware Network Slicing Framework for Base Station

The management and the operation of dynamic FS in the RAN pose significant challenges with several trade-offs. On the one hand, a reduction in the FH/BH network's load can be achieved by locating RAN functions at RRHs, though at the expense of increasing the computational needs in the RRHs. On the other hand, offloading the RAN functions and pooling them at the CU benefits from a reduction of the computational capacity required at the RRHs and offers centralized control that can improve the network's performance, but with higher FH/BH bandwidth requirements. In the same vein, not all FSs meet the requirements of all services, and as proposed by 3GPP [13] it is expected that each slice would have diverse QoS requirements. Regardless of how exactly a slice is implemented within the RAN, different functionality mapping (*i.e.,* FS selection) may be suitable for each slice. The QoS requirements of each service have been taken into account when adapting the FS for each service. For instance, eMBB traffic requires a high degree of coordination among eNBs to achieve high data rates. This suggests a scenario in which eMBB UEs require high bandwidth along with high-speed execution for these bandwidth-intensive applications, processing of a vast amount of data in a cloud (equivalently CU) [3]. This means centralizing network functions towards the CU, (e.g. split 3). Conversely, uRLLC needs fast retransmissions to guarantee low latency and high reliability. In that sense, decentralized FSs are needed (e.g. split 1), which means the experienced delay for this service is minimized since the most of functions are decentralized and located in the RRHs. mMTC, such as IoT applications, is a service with which intermediate splits would work [95].

In this context, a convenient network slicing algorithm provides the network with a higher degree of flexibility, thus enabling the adaptation of the FS of each slice to traffic requirements and network limitations (RRH and/or CU computing capacity, FH/BH network capacity, etc.). Thereby, we propose a MIP framework to formulate joint FS and network slicing for future 5G networks, and describe the proposed MIP optimization problem formulation. We next propose an effective heuristic method SlicedRAN based on the Relaxation Induced Neighborhood Search (RINS) heuristic and then explain its performance.

4.6 Problem Formulation

As already stated, in the following we propose an optimization solution aimed at maximizing the throughput of the 5G network by jointly selecting the most convenient and efficient FS and routing per slice.

Accordingly, the objective of the solution is to maximize the throughput by determining i) the UE association to the RRH and ii) the path through which each type of traffic is forwarded. In parallel, the solution decides the most appropriate FS for each service based on the constraints of the network, such as the capacity of the links and the computational capacity of the CU and the RRHs.

Hence, the maximization of the network throughput, which is our objective function, can be written as in (4.6.1). The constraints of the optimization model are defined in (4.6.2) - (4.6.10).

$$\text{max.} \sum_{r \in R} \sum_{s \in S} \sum_{u \in U} \lambda_u^s \cdot x_u^{r,s} \tag{4.6.1}$$

Subject to:

$$\sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}} \sum_{n=0}^{3} x_u^{r,s} \cdot \lambda_u^s \cdot c_n \cdot f_n^{r,s} \leq \kappa_r, \; \forall r \in \mathcal{R} \tag{4.6.2}$$

$$\sum_{r \in R} \sum_{s \in S} \sum_{u \in U} \sum_{n=0}^{3} \lambda_u^s \cdot x_u^{r,s} \cdot c_n (1 - f_n^{r,s}) \leq \kappa_0 \tag{4.6.3}$$

$$\sum_{p \in P^r} t_p^r = \sum_{s \in S} T^{r,s}, \; \forall r \in \mathcal{R} \tag{4.6.4}$$

$$\sum_{r \in R} \sum_{p \in P^r} t_p^r . y_{i,j}^p \leq \omega_{i,j}, \; \forall j \neq i \in \mathcal{Q} \tag{4.6.5}$$

$$\sum_{s \in S} \sum_{u \in U} x_u^{r,s} . \rho_u^{r,s} \leq \rho^r, \; \forall r \in \mathcal{R} \tag{4.6.6}$$

$$\sum_{r \in R} x_u^{r,s} = 1, \; \forall u \in \mathcal{U}, \forall s \in \mathcal{S} \tag{4.6.7}$$

$$\sum_{u \in U} \sum_{r \in R} x_u^{r,s} \geq \tau^s . \eta^s, \; \forall s \in \mathcal{S} \tag{4.6.8}$$

$$x_u^{r,s} \in \{0, 1\}, \; \forall r \in \mathcal{R}, \forall s \in \mathcal{S} \tag{4.6.9}$$

$$\phi^{r,s} \in \Phi^s, \forall r \in \mathcal{R}, \forall s \in \mathcal{S} \tag{4.6.10}$$

First, (4.6.2) ensures that the computational cost of the functions run in each RRH does not exceed the RRH computational capacity κ_r. Similarly, constraint (4.6.3) is used to bound the maximum computational capacity supported by the CU (κ_0). Constraint (4.6.4) guarantees that all the traffic served by a RRH, regardless of the slice to which the traffic belongs, equals the traffic forwarded over the paths from the CU to the RRH, as shown in (4.4.2). Constraint (4.6.5) states that the flow from each RRH r to CU is bounded by the capacity of the links of the paths, denoted as $\omega_{i,j}$ (b/s). The $y^p_{i,j} \in \{0,1\}$ indicates if the path p includes link $l_{i,j}$ or not. As for the number of PRBs allocated by each RRH, constraint (4.6.6) ensures that the number of PRB allocated to UEs served by the RRH can not exceed the maximum number of PRBs. Moreover, UEs are not served by more than a single RRH simultaneously with constraint (4.6.7). We impose a minimum SLA for each slice in constraint (4.6.8) to guarantee the SLA of a particular slice. Specifically, $\tau^s \in [0,1]$ stands for the minimum proportion of UEs with service type s that must be served with the required QoS and η^s is the total number of UEs with service type s. Thus, the minimum number of UEs that should meet the required QoS is $\tau^s \cdot \eta^s$. Indeed, this constraint solves the problem of prioritization for the services with higher bandwidth requirements and creates a balance for the diversity of accepted services/slices, where its impact is analyzed in Section 4.7. Constraint (4.6.9) defines $x^{r,s}_u$ as a binary variable. Finally, (4.6.10) defines the set of allowed FSs for a slice/service. For illustrative purposes, let us assume that a given service s must be served always with FS 1. In that case, the set of allowed FSs, denoted in (4.6.10) as Φ^s, would be $\Phi^s = \{1\}$. Thus, (4.6.10) forces all RRHs to serve service s with FS $\phi^{r,s} = 1$. Instead, if we assume that a service can be delivered with splits 2 and 3, then $\Phi^s = \{2,3\}$. In this case, the FS for service s can take values $\phi^{r,s} = \{2,3\}$ in the different RRHs. It is worth noting that the value of $\phi^{r,s}$ determines the value of $f^{r,s}_n$ for $n = \{0,1,2,3\}$ according to (4.4.3).

Theorem 1. *The optimization model turns out to be a MIP problem, which is an NP-complete problem and has a complexity of $\mathcal{O}(2^N)$.*

Proof. The optimization model that maximizes the network throughput is a MIP problem, for which commercial and free solvers can be used. Generally, a MIP problem is known to be an NP-complete problem, and as the computation time for NP-complete problems is high [96], the number of nodes (UEs and RRHs) in the network negatively affects the computation time due to the increment of the search space of the variables. We solved the MIP optimization using IBM ILOG CPLEX Optimization Studio [97]. This optimizer has a high-performance solver which uses algorithms such as branch-and-bound, branch-and-cut, etc. Jeroslow [98] proved that the complexity of branch-and-bound for a MIP problem is $\mathcal{O}(2^N)$, where N is the number of variables in the optimization. In our optimization problem, we have binary ($x^{r,s}_u$) and continuous (t^r_p) variables that require the branch-and-bound method.

We next propose a solution algorithm for the MIP problem. Note that our MIP optimization model and the provided solution algorithm are generic and can be easily extended for various scenarios where the routing and computation cost functions are strictly convex and linear on the UEs' traffic load.

4.6.1 Solution Method: RINS Heuristic for SlicedRAN

The computational complexity of the MIP problem increases substantially for large-scale networks, and the number of nodes in the network negatively affects this computation time. To overcome this issue, one solution could be using Reinforcement Learning (RL) algorithms or developing a heuristic approach. Indeed there is a trade-off between selecting the heuristic approach and RL algorithms. In particular, the training time in RL algorithms is high whereas in the heuristic approach is null. Conversely, once trained, the computational time is lower for RL than for the heuristic method. The heuristic approach is able to face changes better than RL as long as it can be executed fast enough (*i.e.,* reduced computational complexity). To this aim, we develop a heuristic solution, SlicedRAN, using a heuristic method to handle it in a shorter computing time. One of these heuristic methods is known as RINS [99] that separates a MIP problem into sub-problems and explores a neighborhood of the current incumbent solution and solves reduced problems at some nodes of a branch-and-cut tree and obtains a good solution among the incumbent solution. The proposed solution algorithm is summarized in Algorithm 1.

Algorithm 1: SlicedRAN

```
Input: $G = (\mathcal{I}, \mathcal{Q}, \mathcal{L})$: a network topology graph
$\mathcal{U}$: the set of UEs
$\tau^s$: the proportion of UEs (i.e., SLA %)
Initialize: Compute SNR and create candidate list
            of RRHs for each UE
            Compute $\rho_u^{r,s}$ for each UE and create a candidate list of RRHs based on $\rho_u^{r,s}$
Output: Sol: the solution for all UEs of each slice
1  repeat
2      $\forall u \epsilon \mathcal{U}$
3  foreach $s \epsilon S$ do
4      for $\tau^s$ = 0 to 100 do
5          tmp ← Solve SlicedRAN based on RINS heuristic in sub-problems
6          if SLA constraint holds then
                                                        ▷ (11)
7              Sol.s ← tmp
                                                        ▷ Feasible solution
8          end
9          else
10             Sol.s ← Inf.
                                  ▷ Infeasible solution due to lack of resources
11         end
12     end
13     Sol ← Sol.s
14 end
15 return Sol
```

According to this algorithm, we take a network topology, the set of all UEs, and the proportion of UEs $\tau^s \in [0, 1]$ as an input and then initialize the SNR and $\rho_u^{r,s}$ of each UE to the list of RRHs. Next, for all UEs belonging to each slice we solve SlicedRAN based on RINS (lines 1-5). Then, we check the constraint (4.6.8) to check the SLA threshold for each service; if the network guarantees the current SLA (*i.e.,* τ^s) (line 6), then the feasible solution is sought for the current slice s, where the solution is stored in $Sol.s$ (line 7). Otherwise, the solution is infeasible due to insufficient resources in the network (line 10). Finally, we store the final solution of the current slice s in Sol (line 13) and return this solution as an output of our algorithm (line 15).

Theorem 2. *The run-time complexity of SlicedRAN is* $\mathcal{O}\big((|S|.|U|.|T|).(2^{|S|.|U|.|R|+|P|.|R|})\big)$.

Proof. SlicedRAN starts at line 2 and ends at 15. In this loop, we iterate on the number of UEs $|U|$ and with the number of services $|S|$. For each SLA threshold τ^s, it takes the number of $|T|$ steps. We next run RINS in subproblems, exploring the convex hull of all the feasible solutions for binary variable $x_u^{r,s}$ and continuous variable t_p^r. The binary variable $x_u^{r,s}$ has a maximum number of $|\mathcal{S}|.|\mathcal{U}|.|\mathcal{R}|$, and continuous variable t_p^r has a maximum number of $|\mathcal{P}|.|\mathcal{R}|$ in the network. Thus, the run-time complexity of SlicedRAN is $\mathcal{O}\big((|S|.|U|.|T|).(2^{|S|.|U|.|R|+|P|.|R|})\big)$.

4.7 Performance Evaluation

In this section, we present the effectiveness of the proposed solution from the overall system and slice viewpoint by investigating numerical results for the performance of SlicedRAN.

4.7.1 Simulation Scenario

In our analysis, we consider three main types of uRLLC, mMTC, and eMBB services with different QoS requirements. Table 4.3 summarizes our simulation setup, where we assume a bandwidth of 20 MHz for each BS (i.e, $\rho^r = 100$ PRBs) with 4 forwarding nodes ($Q = 4$ and $m = 2$ in Fig. 4.2(a)) and a link capacity ranging from $\omega_{i,j} = 100$ Mb/s to $\omega_{i,j} = 25$ Gb/s. We study a scenario composed of a single CU connected to a set of RRHs from 4 to 16 ($R = 4$ to $R = 16$) and adopt the values of [100–102] to define three types of applications, *i.e.,* medical, IoT, and video streaming applications. We consider $s = 1$ for medical applications (uRLLC) which use split 1 with $\lambda_u^1 = 120$ Kb/s, $s = 2$ for IoT messages (mMTC) which use split 2 with $\lambda_u^2 = 30$ Kb/s, and finally $s = 3$ for video streaming applications (eMBB) which need a higher degree of centralization (*i.e.,* split 3) with $\lambda_u^3 = 20$ Mb/s. As for the computational capacity, we utilize the values used in [54], with $\kappa_0 = 100$, $\kappa_r = 1$ CPU reference core per Gb/s. Regarding the computational cost, $c_1 = 3.25$ and $c_2 = 0.75$ CPU reference core per Gb/s. Note that we exclude the computation costs for $f_0^{r,s}$ which is always placed in RRHs and $f_3^{r,s}$ since it is always in the CU. We consider a distance-dependent path-loss model with a transmission power 30 dBm and for the MCS calculation, we adopt the values used in [94]. We then explore two configurations for the evaluation of SlicedRAN.

Configuration 1 (C.1): This configuration is a uniform distribution where 33%, 34%, and 33% are set to UEs with the type of medical applications, IoT messages, and video streaming, respectively. This configuration has a balanced number of different UEs with different QoS requirements, totally 1000 UEs (*i.e.,* $U = 1000$). Indeed, eMBB UEs which need higher bandwidth (*i.e.,* PRB) in RRHs have the same distribution of mMTC applications (*i.e.,* IoT UEs) which require less bandwidth while injecting extra overheads into the FH/BH network.

Configuration 2 (C.2): In general, the distribution of traffic (each type of UEs) is not necessarily uniform, as a massive number of IoT connections is expected. Therefore, we consider a scenario with 6380 UEs (*i.e.,* $U = 6380$)[4] where 80% of the connections correspond to IoT applications (mMTC), while 15%, and 5% are set to UEs with medical applications (uRLLC) and video streaming (eMBB), respectively. Apparently, this configuration has fewer eMBB UEs, thus less requirements in terms of PRBs in RRHs. On the other hand, it has more IoT applications that add huge overheads into the network, thus higher requirements in terms of the capacity of FH/BH networks.

Considering these different configurations with diverse QoS requirements, it is of prominent importance to:

- Examine different behavior and requirements of these configurations in terms of capacity requirements in RRHs and FH/BH network, which gives the knowledge for designing a virtualized network infrastructure for each configuration.

- Explore a proper split per BS, which provides a basis on how to design a cost-efficient FH/BH network for each configuration.

We have conducted extensive Monte-Carlo simulations implemented in Java, while the optimization model is built and solved with IBM ILOG CPLEX Optimization Studio [97]. This optimizer has a high-performance solver which uses algorithms such as branch-and-bound, branch-and-cut, etc. Hence, the optimizer explores the convex hull of the optimization problem and then enumerates all the feasible solutions. Note that the computing time needed to obtain the optimal solution (*i.e.,* MIP) with a CPU processor of Core i7-8550U, a RAM of 16 GB for a scenario composed of a single CU, $Q = 4$, $R = 10$ and for C.1, is the matter of hours while the proposed algorithm (*i.e.,* SlicedRAN) is able to obtain the near-optimal solution around 9 seconds.

We obtain results by maximizing the throughput for each configuration for three metrics of interest:

- *Served Traffic:* to identify the portion of traffic served by applying SlicedRAN on the existing network infrastructure;

- *Link Usage:* to explore the minimum capacity required in the FH/BH network for each configuration;

- *Spectrum Usage:* to study a cost-efficient (*i.e.,* the minimum number of RRHs) set-up required in RRHs per configuration.

All results are compared with [58] where the main objective is maximizing CD and no slicing is considered. Therefore, a single FS is allowed to use in each RRH. In the following, the MIP optimization is labeled as *Optimal*, the proposed heuristic as *SlicedRAN*, and the state of the art [58] as *SoA*.

Within the simulation, we compare the performance of SlicedRAN and Optimal with the benchmark scheme of SoA ([58]). We analyze the capacity requirements of RRHs in Section 4.7.2, where we only evaluate the network performance without imposing constraints in the FH/BH network; then, we

[4] In order to have a fair comparison, we consider the same offered traffic of 6.65 Gb/s for both C.1 and C.2. This is why for C.1 totally $U = 1000$ UEs and for C.2 $U = 6380$ have been chosen.

Table 4.3: Simulation Setup

System Bandwidth	20 MHz
Number of PRBs (ρ^r)	100
Number of RRHs (R)	4 - 16
Number of Forwarding Nodes (Q)	4
Number of UEs (U)	1000 - 6380
The capacity of links ($\omega_{i,j}$)	0.1 - 25 Gb/s
Transmission rate of uRLLC (Medical apps)	120 Kb/s
Transmission rate of mMTC (IoT msg)	30 Kb/s
Transmission rate of eMBB (Video Streaming)	20 Mb/s
Transmitted power	30 dBm
CPU consumption to compile f_1 (c_1)	3.25 RCs per Gb/s
CPU consumption to compile f_2 (c_2)	0.75 RCs per Gb/s
Computation capacity of each RRH (κ_r)	1 RCs per Gb/s
Computation capacity of CU (κ_0)	100 RCs per Gb/s
Value of SLA for each slice (τ^s)	5% - 100%

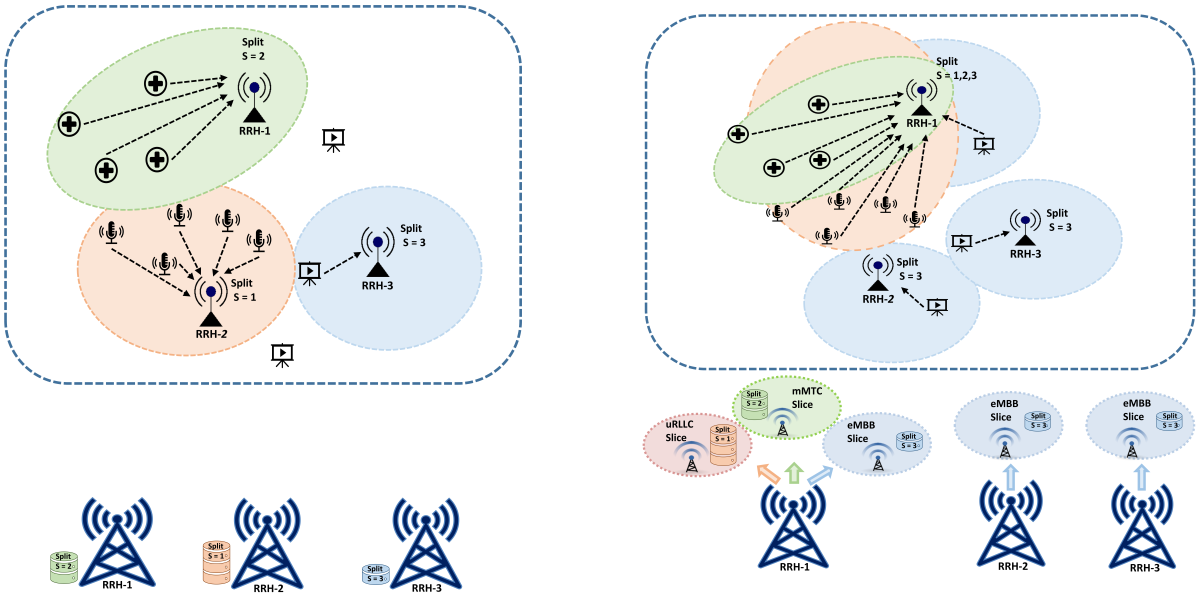
enforce the FH/BH constraints in Section 4.7.3 to explore the impact of the limitation on the FH/BH network in order to provide the guidelines on the design of the FH/BH network. As stated before, maximizing the throughput has a great impact on the QoS of those services/slices which have less bandwidth requirements. To this end, we analyze the impact of imposing a minimum SLA for the network performance in Section 4.7.4.

4.7.2 Analysis of Capacity of RRHs

As mentioned previously, the aim of investigating the capacities of RRHs is to find a proper set-up of flexible FS along with the minimum capacity of requirements per RRH. We thus explore the network performance where only constraints on the capacities of RRHs are imposed. For the sake of simplicity, we first present the analysis of a small network (*i.e.,* toy example) with 3 RRHs (*i.e.,* $R = 3$) and 12 UEs (*i.e.,* $U = 12$) (see Fig. 4.3), to clarify the advantages of leveraging virtualization in SlicedRAN. As can be seen in Fig. 4.3.a, SoA fails to support the QoS of all UEs due to the restriction in the capacity of RRHs (here RRH-3) and also the limitation in using only a single FS in RRHs. Conversely, in Fig. 4.3.b, SlicedRAN serves the QoS of those UEs who were not satisfied in SoA since in SlicedRAN RRHs are virtualized, which enables them to use different functional splits in order to meet the QoS of different UEs. For example, RRH-1 serves three-different UEs by supporting three-different functional splits as illustrated in Fig. 4.3.b.

We next present the results of our numerical analysis in Fig. 4.4, where we assume that $\omega_{i,j} = \infty$ and $\tau^s = 0, \forall s \in \mathcal{S}$ for a set of RRHs from 4 to 16 BSs ($R = 4$ to $R = 16$) where the offered load is 6.65 Gb/s for C.1.

As can be seen in Fig. 4.4, it is evident that the increase in the number of RRHs is consistent with the increase in the average of served traffic in Optimal, SlicedRAN, and SoA. The heuristic



(a)

(b)

Figure 4.3: a). Scheme of UE association in SoA, where only a single functional split is allowed to be set per BS. b). Scheme of UE association in SlicedRAN, thanks to slicing, which allows multiple virtual functional splits to be placed on each BS.

SlicedRAN has higher performance compared to SoA while lower than Optimal. However, it performs very close to Optimal with a lower computation time. For example, when $R = 6$ SlicedRAN achieves 5.54 Gb/s throughput while SoA can reach up to 2.95 Gb/s in the throughput, which indicates 85% of gain in the performance of SlicedRAN when compared with SoA performance. The main reason for outperforming SlicedRAN is strongly linked to the benefits of virtualization, which allows RRHs to use different FSs to serve diverse traffic demands; while in SoA (without slicing) each RRH is allowed to use only a single FS in order to serve the corresponding service. Another reason for this behavior lies in the mean distance between the UE and the serving RRH. By further analysis of this figure, it can be identified that increasing the number of RRHs (densification) has more impact on SoA performance. We observe that a smaller number of RRHs results in more traffic rejection in SoA and increasing the number of RRHs adds more diversity in terms of FS for SoA, thus supporting plenty of different services. Indeed when the density of the RAN (*i.e.,* the number of RRHs) increases, the gap between the two alternatives is reduced. For example, by increasing RRHs from $R = 6$ to $R = 10$ (*i.e.,* the addition of four RRHs) in SoA, the mean served traffic increases up to 68% (from 2.95 Gb/s to 4.98 Gb/s). Whereas this increase of traffic in SlicedRAN is $\sim 14\%$ (from 5.54 Gb/s to 6.30 Gb/s). As stated above, there is clear evidence that having a network with a small number of RRHs (each RRH with a single FS) fails to support plenty of distinct services. Hence, at the beginning the UEs who were not able to be served with nearby RRHs (due to different FS requirements), with increasing the number of RRHs, the diversity of BSs with different FSs allows them to find a BS with the corresponding FS to meet their requirements.

Similar results were found after evaluating C.2 in Fig. 4.5, wherein the same settings of Fig. 4.4 is applied. It is apparent that in all cases SlicedRAN outperforms SoA with a significant difference

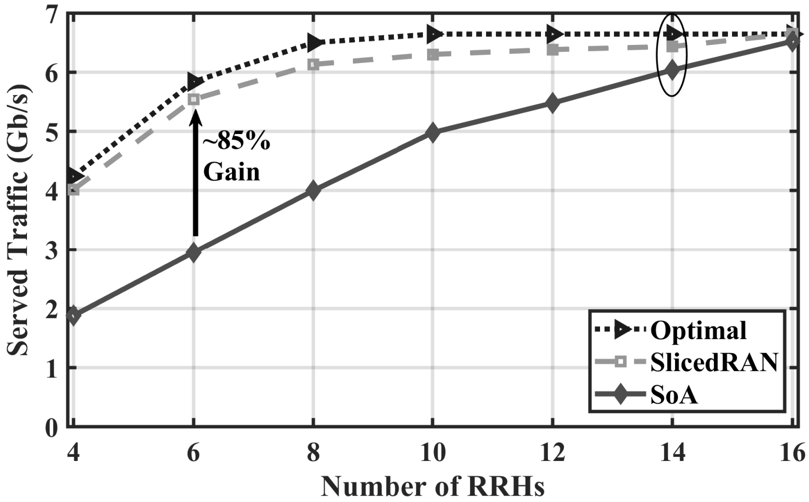


Figure 4.4: Average served traffic w.r.t number of RRHs from $R = 4$ to $R = 16$ where $\omega_{i,j} = \infty$ with offered load = 6.65 Gb/s for C.1

in the cases of a small number of RRHs. For example, SlicedRAN achieves 5.47 Gb/s in throughput when $R = 6$, while SoA reaches only 2.57 Gb/s, hence, we have a considerably higher gain in the performance of SlicedRAN, that is $\sim 112\%$ gain in throughput.

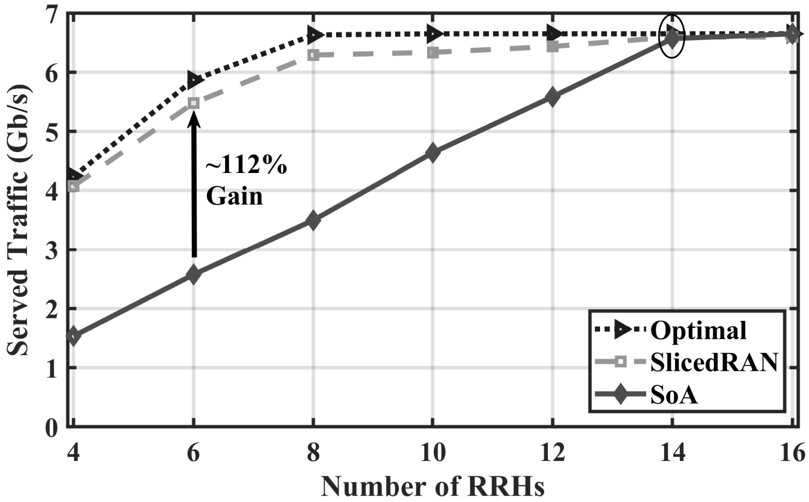


Figure 4.5: Average served traffic w.r.t number of RRHs from $R = 4$ to $R = 16$ where $\omega_{i,j} = \infty$ with offered load = 6.65 Gb/s for C.2

Comparing Figs. 4.4 and 4.5 show that a significant improvement was obtained in the majority of cases. However, the main inspection of Fig. 4.5 indicates that SlicedRAN serves almost all traffic when $R = 14$, while in Fig. 4.4 in order to reach this performance, two more RRHs are needed (*i.e.,* $R = 16$), which leads to extra deployment costs for MNOs. This is because in C.1 we have a uniform distribution of traffic *i.e.,* with the same number per each type of UEs, which leads to higher bandwidth requirements in RRHs (due to existing more eMBB UEs). In addition, we observe that SoA has a better performance in Fig. 4.4 which is evaluated for C.1. For example, the average of served traffic in Fig. 4.4 for $R = 8$ is higher than 4 Gb/s, while in Fig. 4.5 (*i.e.,* C.2) the achieved value is less than 3.5 Gb/s. The main reason for this behavior is that the distribution of traffic in C.2 is not uniform, and we have fewer eMBB UEs that need higher bandwidth requirements (*i.e.,* PRB) in RRHs. Indeed, C.2 is mainly composed of 80% of mMTC UEs and 15% of uRLLC UEs (*i.e.,* in total 95 % of all offered traffic), and only 5% of all UEs are eMBB UEs. Hence, the usage of capacities in RRHs (*i.e.,* PRB, computation cost, spectrum) in C.2 is fewer than C.1, where it has a uniform distribution of traffic (*i.e.,* the same number per each type of UEs). Furthermore, SoA in C.2 serves almost all traffic with $R = 14$ while C.1 needs at least two more RRHs (*i.e.,* $R = 16$) in order to achieve the same throughput. The main reason for this contradicting behavior is that as we have fewer number of eMBB UEs in C.2, the diversity requirements of RRHs with different FS is lower, which means fewer number of RRHs are needed to meet the requirement of eMBB UEs. On the other hand, in C.1, the uniform distribution entails distinct FSs for each type of service and due to using only a single FS in SoA, this uniform traffic needs more RRHs with different FS in order to serve all traffic.

From these results, we can conclude that with slicing we manage to better use the resources in terms of spectrum usage in RRHs by creating different slices in each RRH. Thus, to achieve the same performance in SoA, MNOs need to deploy more RRHs, which adds more costs for them. This is more important for MNOs to decrease the hardware deployment costs where they are in quest of a cost-efficient design of the BS framework.

4.7.3 Analysis of FH/BH Network

The results of subsection 4.7.2 showed the gain achieved by slicing in RRHs, when no constraints were imposed on the FH/BH network. However, the design of the FH/BH network has an impact on this gain. Note that the degree of centralization depends on the design and the available capacity in FH/BH networks. Indeed as we increase the capacity of links, higher traffic can be served in the network, and the impact of the FH/BH network on the benefits of slicing can be diminished.

In this regard, we analyze the FH/BH networks for k-partite network topologies by increasing the capacity of links (*i.e.,* $\omega_{i,j}$). First, we explore this analysis for C.2, where more mMTC UEs are deployed. As you see in Fig. 4.6, the remarkable point is that imposing constraints in the FH/BH networks leads to the loss in the throughput gained in Section 4.7.2. The results show that the heuristic SlicedRAN and Optimal have tight and close results, and, as we increase the capacity of the FH/BH links, more throughput is achieved due to having more capacities in the FH/BH networks. Thereafter, the impact of limitation in FH/BH networks is reduced. However, increasing the capacities of FH/BH networks makes this loss and limitation negligible in SlicedRAN while it remains suffering in SoA. Fig. 4.6 depicts that only 16% of the offered traffic is actually served when $\omega_{i,j}$ = 100 Mb/s (a loss equal to 84%) for SlicedRAN while this loss is $\sim$ 97% for SoA. Remarkably, this reduces as we increase the capacities in FH/BH networks. For example, the loss of traffic by SlicedRAN is $\sim$ 25% when we have $\omega_{i,j}$ = 1 Gb/s while in SoA it is close to 65%, and when we increase the capacity to $\omega_{i,j}$ = 2 Gb/s, the average of traffic served by SlicedRAN is higher (*i.e.,* 6.55 Gb/s) which means under 1% of offered traffic is lost while SoA achieves 4 Gb/s which means 40% of offered traffic is dropped. This effect is pronounced for the utilization of slicing in SlicedRAN, which better uses the resources in the FH/BH network by employing different FSs, which has an impact on the FH/BH network in order to serve different services with various QoS requirements. Whereas in SoA, on one side, only a single FS is used in each RRH to serve the corresponding traffic of services. On the other side, SoA suffers from the limitation in the capacities of RRHs since it achieves the same throughput of SlicedRAN when we have more number of RRHs, that is, at least $R = 14$ RRHs for SoA (See Fig. 4.5).

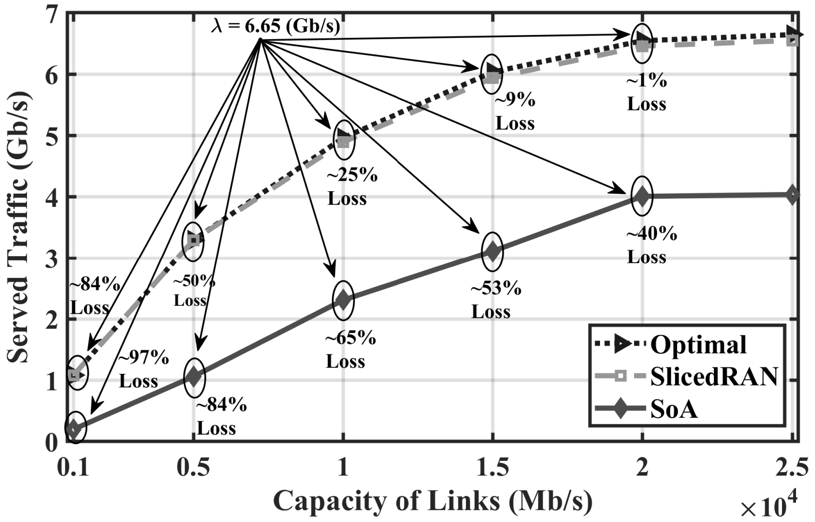


Figure 4.6: Average served traffic w.r.t the capacities in FH/BH network (*i.e.*, $\omega_{i,j}$ (in Mb/s)) with offered load = 6.65 Gb/s for C.2

In Fig. 4.7 which is obtained for C.1, the results demonstrate relatively the same behavior as Fig. 4.6. Comparing these figures shows that when we have more capacities in FH/BH network, SoA in C.1 performs better when compared to SoA performance in C.2. This is because in C.2 we have more deployed eMBB and mMTC UEs that inject more overhead and capacity requirements in FH/BH network; hence, more traffic is rejected.

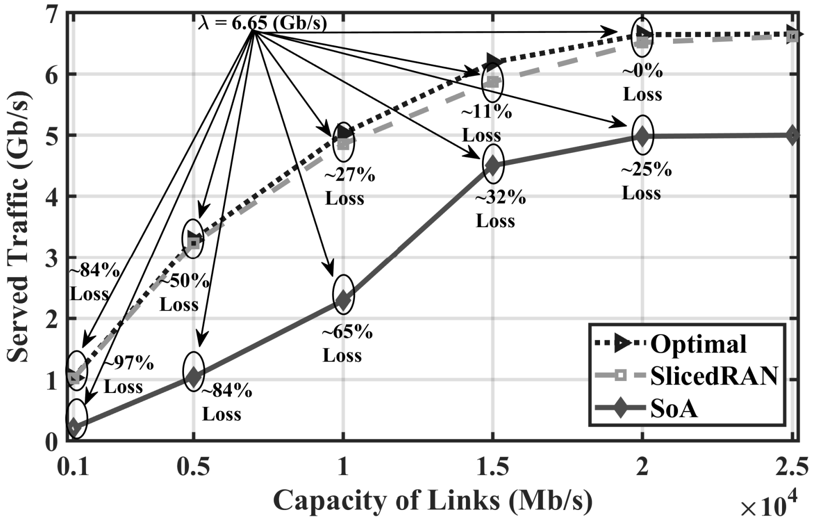


Figure 4.7: Average served traffic w.r.t the capacities in FH/BH network (*i.e.*, $\omega_{i,j}$ (in Mb/s)) with offered load = 6.65 Gb/s for C.1

Table 4.4: Resource usage for C.2

Optimal (A) / SlicedRAN (B) / SoA (C) performance with 10 RRH									
$\omega_{i,j}$ (Gb/s)	Served Traffic (%)			Link Usage (%)			Spectrum Usage (%)		
	A	B	C	A	B	C	A	B	C
1	16.3	16.3	3	99.9	99.5	99.9	17.3	17.3	7.7
5	49.6	49.6	15.9	99.8	99.5	99.9	62.6	62	27.5
10	74.7	74	34.7	99.7	98	99.9	89.9	95.9	54.2
15	90.8	89.3	46.8	99.8	96.2	99.7	95.1	99	72.6
20	98.5	91.1	60.3	99.3	98.2	91.3	97.1	98.7	85.9
25	100	91.6	60.7	88.1	87.2	73.6	98.5	98.5	85.8

In Table 4.4, we assess the resource usages of the network in terms of *Served Traffic, Link Usage, Spectrum Usage* for C.2. It must be pointed out that both SlicedRAN and Optimal better use the resources, and more traffic is served with the same resources in the network when compared to SoA. For example, SlicedRAN achieves superior results respectively up to 74% of total traffic when the limitation is imposed into the FH/BH network with $\omega_{i,j} = 10$ Gb/s whereas with the same capacity in the FH/BH network achieved traffic is about 34.7% in SoA which is less than half of the traffic which is served by SlicedRAN. A similar pattern was obtained when capacity is increased to $\omega_{i,j} =$ 15, 20 Gb/s, where SlicedRAN achieves 89.3%, 91.1%, respectively while SoA reaches only 46.8%, 60.3% sequentially of the total traffic offered.

A similar conclusion was reached by Table 4.5 where we have 10 RRHs in the network for C.1. The findings are directly in line with previous findings particularly when the capacity of the FH/BH links is small. From this table, it is evident that the performance of SlicedRAN is substantially better than the SoA performance. For instance, when $\omega_{i,j} = 1$ Gb/s SlicedRAN performs almost five times better than SoA in the percentage of traffic served while using 53% more in the spectrum. It is essential to highlight the fact that in almost all the cases, the usage of the FH/BH links is close to 100%, while in the case of spectrum usage, SlicedRAN performs better when compared to SoA performance.

From these results in tables 4.4 and 4.5, we conclude that dimensioning the FH/BH network, impacts on all explored metrics (*Served Traffic, Link Usage, Spectrum Usage*) of the network. Indeed as much as we increase the capacity of links in the FH/BH networks, it allows us to serve more traffic, especially with leveraging virtualization in SlicedRAN we can have higher gains in terms of *Served Traffic* and efficient resource usage (*i.e., Link Usage, Spectrum Usage*) when compared with SoA. This gives an overview of the minimum capacity requirements in terms of both in RRHs and FH/BH network, thus, providing a basis for MNOs to design a cost-efficient virtualized network architecture covering both RRHs and FH/BH networks.

Furthermore, we next study and compare two network topologies known as bipartite and k-partite networks to get a more comprehensive overview of dimensioning FH/BH network, and compare the performance of these topologies in Table 4.6. As explained in Section 4.4.1, k-partite networks include

Table 4.5: Resource usage for C.1

Optimal (A) / SlicedRAN (B) / SoA (C) performance with 10 RRH									
$\omega_{i,j}$ (Gb/s)	Served Traffic (%)			Link Usage (%)			Spectrum Usage (%)		
	A	B	C	A	B	C	A	B	C
1	15.5	15.4	3.2	99.6	99.8	99.9	11.4	15.3	7.3
5	49.4	48.5	15.6	99.8	99	99.9	56.2	55.6	28.4
10	75.6	70	34.5	99.5	98.2	99.9	81.9	87.4	50.9
15	93.1	82.3	67.7	99.6	93.8	99.9	92.1	91.8	85.5
20	99.9	90.7	74.9	89.1	91.8	78.6	93.4	90.5	90.5
25	100	90.9	75.2	88.7	75.7	74.3	95.2	85.5	90.5

Table 4.6: Resource usage of SlicedRAN with bipartite and k-partite topologies

k-partite (bipartite) performance			
$\omega_{i,j}$ (Gb/s)	Served Traffic (%)	Link Usage (%)	Spectrum Usage (%)
1	16.3 (16.2)	99.9 (99.7)	17.3 (17.4)
5	49.6 (49.6)	99.8 (99.9)	62.6 (59.5)
10	74.7 (74.7)	99.7 (99.6)	89.9 (88.1)
15	90.8 (90.8)	99.8 (99.8)	95.1 (95.6)
20	98.5 (98.4)	99.3 (99.1)	97.1 (97.6)
25	100 (100)	88.1 (88.1)	98.5 (98.3)

k-layers of forwarding nodes whereas bipartite topology is composed of only one column (*i.e.*, layer) of forwarding nodes. The results in Table 4.6 show that increasing the capacity of links for both network topologies gives the same performance for all metrics of interest. This investigation on bipartite and k-partite network topologies along with dimensioning of these networks gives an overview of the minimum capacity requirements in terms of both in RRHs and FH/BH network, and more importantly, the minimum number of Q required in the FH/BH network, which gives the hint over scalability, CAPEX, and thus, provides a basis for MNOs to design a cost-efficient virtualized network architecture covering both RRHs and FH/BH networks.

We further explore dimensioning of these network topologies by fixing the capacity of links from CU to forwarding nodes ($CU - Q$), and relaxing the capacity of links from the forwarding nodes to the RRHs ($Q - R$). Fig. 4.8 shows the results of this dimensioning for the bipartite and k-partite topologies where the capacity of $CU - Q$ links increases from 5 Gb/s to 20 Gb/s. In this figure, with successive increases in the capacity of $Q - R$ links, as expected, the served traffic further increases. However, the rapid increase in bipartite topology is due to existing ample routes from $Q - R$ links.

While in k-partite topology where there are more layers among forwarding nodes, which negatively affects the routing of traffic, especially where the links have small capacity. For example, in Fig. 4.8 (b), with the capacity of $CU - Q$ links = 10 Gb/s, when the capacity of $Q - R$ links increases from 0.1 to 1 Gb/s, the served traffic increases more than 3.6 Gb/s for bipartite while this amount is about 1.4 Gb/s for k-partite topology. As shown in Fig. 4.8 (c), having the capacity of $CU - Q$ links = 20 Gb/s leads to approximately 99% of serving all traffic. However, this happens for k-partite topology when the capacity of $Q - R$ links is more than 10 Gb/s while bipartite topology achieves the same performance with less than 5 Gb/s in the capacity of $Q - R$ links. This behavior may give a clue on designing a cost-efficient network depending on MNOs' interests. For example, for an MNO routing the traffic for a specific customer with minimum usage in routers and links in the substrate network has paramount importance (e.g. saving routing costs), thus reserving the resources of the network for other customers. In practice, MNOs shall share the network (due to virtualization) and forward the traffic of different customers with the unused routers in the FH/BH network.

Having different QoS requirements in 5G, especially for three main types of eMBB, uRLLC, and mMTC services need to adopt by MNOs to meet the diversity of these QoS requirements. Indeed, maximizing the throughput is challenging to satisfy the QoS of different services, and could have an impact on the QoS of those services which have less bandwidth requirements on the network. Fig. 4.9 illustrates this challenge for the proposed SlicedRAN where the objective is maximizing the throughput and all links have the same capacity in the FH/BH network. As can be observed from this figure, SlicedRAN prioritizes the uRLLC and eMBB services, and after it satisfies all of these services (i,e., $\omega_{i,j}$ = 20 Gb/s), then starts to increase serving mMTC services which have less bandwidth requirements. For example, in this figure, almost all eMBB and uRLLC services are served when $\omega_{i,j}$ = 20 Gb/s. On the other hand, served traffic for mMTC services remains under 50% with the same capacity. Indeed, up to 50% of mMTC services are dropped as they have less bandwidth requirements on the FH/BH network. To this end, in the next subsection, we analyze the imposing of different SLAs to guarantee the QoS of mMTC services.

4.7.4 Analysis of Imposing SLA for each Slice

As observed in Section 4.7.3, the extreme demands with different QoS requirements in 5G show that maximizing the throughput could have an impact on the QoS of those services which have less bandwidth requirements on the network. To overcome this, a constraint to guarantee a minimum percentage of UEs per slice that meet the required QoS is needed. As explained in Section 4.6.1, we denote this as SLA. In the following, we analyze the impact of imposing the SLA for each slice. Due to the limitation of resources, it is not always feasible to achieve the SLA. In that case, the solution will be infeasible.

In Table 4.7, infeasible solutions have been labeled as ”Inf.”. We initially simulated an SLA for mMTC and eMBB ranging from 75% to 100% for C.1 and C.2.

From this table, it can be observed that imposing SLA on eMBB slice increases the throughput, and increasing the value of SLA results in infeasibility in most cases, while the effect of mMTC slice is not as higher and not causes infeasibility. Let us first focus on the impact of mMTC slice wherein increasing the capacity in the FH/BH network leads to the increase in the throughput as expected.

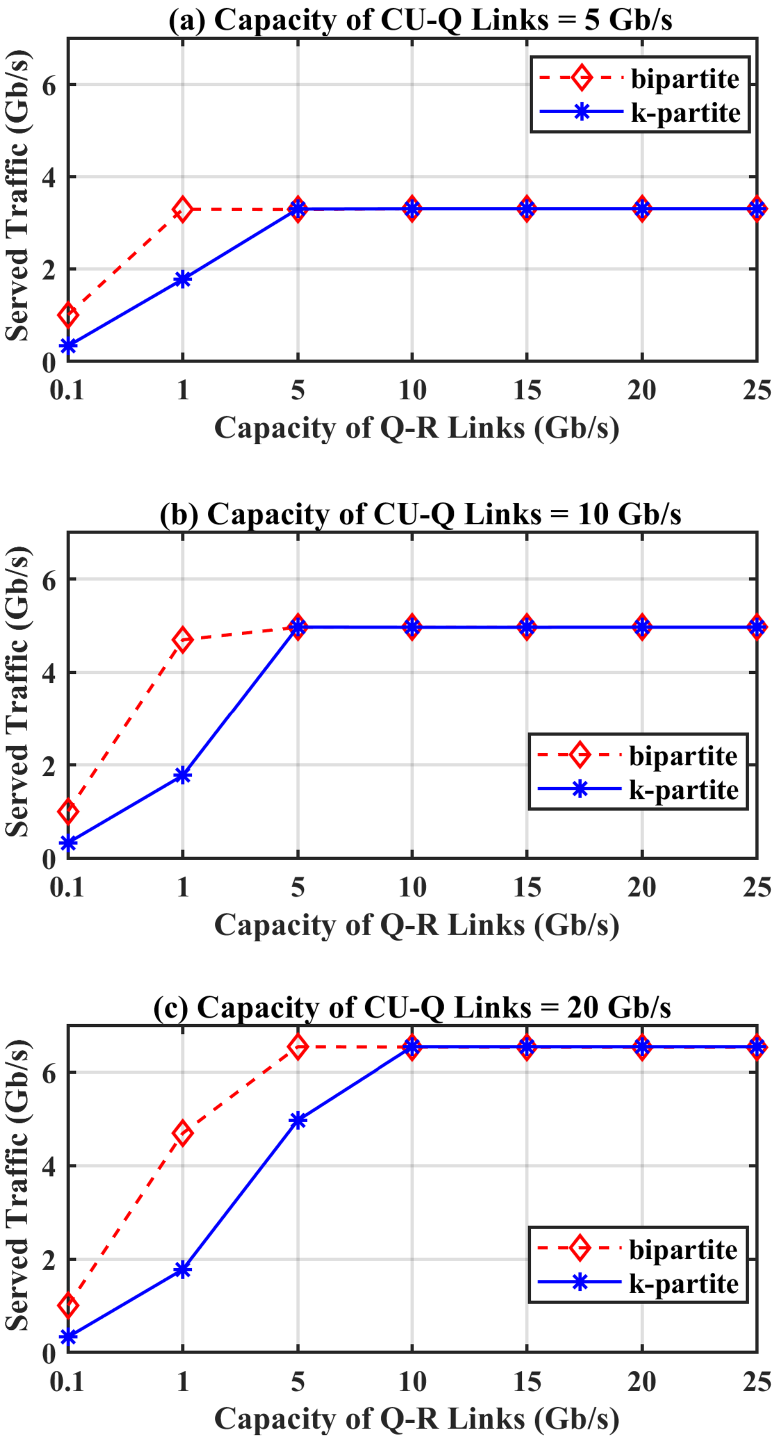


Figure 4.8: Served traffic (%) for bipartite and k-partite network topologies w.r.t the different capacities in FH/BH network with $R = 10$

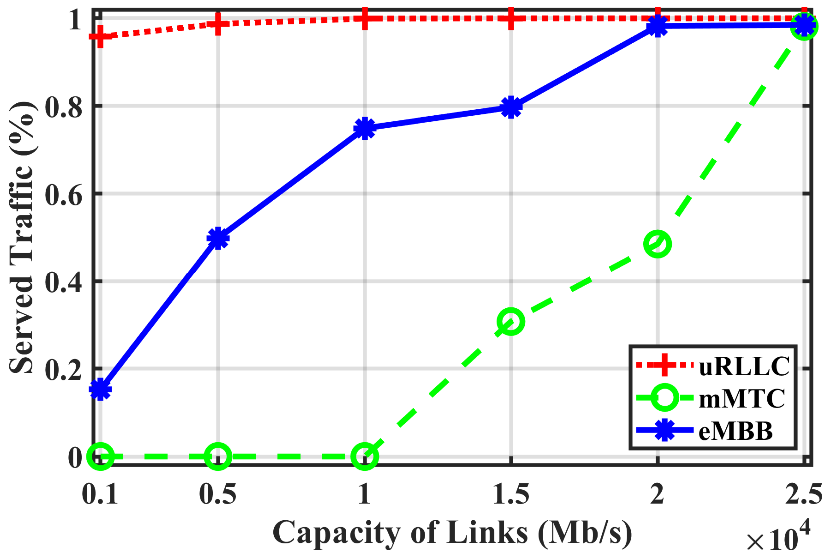


Figure 4.9: Served traffic (%) w.r.t the capacities in FH/BH network (*i.e.*, $\omega_{i,j}$ (in Mb/s)) when $R = 10$ with offered load = 6.65 Gb/s for C.2

Table 4.7: The performance analysis of SlicedRAN w.r.t the different SLAs (from 75% to 100%) imposed for mMTC and eMBB slices

$\omega_{i,j}$ (Mb/s)	The performance of C.1						The performance of C.2					
	SLA = 75%		**SLA = 95%**		**SLA = 100%**		**SLA = 75%**		**SLA = 95%**		**SLA = 100%**	
	mMTC	eMBB	mMTC	eMBB	mMTC	eMBB	mMTC	eMBB	mMTC	eMBB	mMTC	eMBB
100	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.
200	15.3	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.
300	174.8	Inf.	106.1	Inf.	85.8	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.
400	300.1	Inf.	238.9	Inf.	223.7	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.
500	410.7	Inf.	355.4	Inf.	341.7	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.
600	513.0	Inf.	462.1	Inf.	449.1	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.
700	607.6	Inf.	560.9	Inf.	546.8	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.
800	700.2	Inf.	655.9	Inf.	644.1	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.
900	787.6	Inf.	744.4	Inf.	734.8	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.
1000	867.2	Inf.	828.8	Inf.	819.9	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.
5000	3143.3	Inf.	3143.3	Inf.	3117.8	Inf.	1929.3	Inf.	1447.1	Inf.	1308.9	Inf.
10000	4751.5	Inf.	4735.3	Inf.	4745.8	Inf.	4169.5	Inf.	3916.3	Inf.	3863.9	Inf.
15000	4938.8	6073.8	5133.5	Inf.	5645.8	Inf.	4868.9	6042.8	4630.1	Inf.	4621.8	Inf.
20000	6319.0	6597.4	6311.6	6609.5	6317.8	Inf.	6124.1	6533.1	5905.9	6537.2	5702.1	Inf.

Completely different results were observed with increasing the SLA of mMTC slice which leads to the decreases in the amount of throughput. For example, in this table, increasing SLA from 75% to 100% with $\omega_{i,j}$ = 500 Mb/s for C.1. leads to decrease of $\sim$ 17% of throughput from 410.7 Mb/s to 341.7 Mb/s. The main reason for this behavior is that mMTC slice has more overhead than uRLLC slice (due to using split type 2); hence, more load is occupied in the FH/BH networks when the value of τ^s is increased to guarantee SLA of this slice. On the other hand, imposing SLA for eMBB slice results in different behavior, especially when we have fewer resources (capacity in FH/BH networks). As can be seen in the table, having more SLA leads to infeasibility if we do not have enough resources in the FH/BH networks. For example, we can guarantee the SLA = 75% for both C.1 and C.2 when $\omega_{i,j}$ = 15 Gb/s, but when we increase the SLA to 95% and 100%, then it leads to infeasibility and we can not achieve even the minimum throughput for other slices as well. Comparing the results for these configurations indicates that the minimum capacity of links required to guarantee the minimum SLA in C.2 is much higher than C.1 especially due to mMTC services. For example, when SLA = 75% the minimum capacity required to assure this amount of SLA in C.2 is more than $\omega_{i,j}$ = 1 Gb/s, while in C.1, $\omega_{i,j}$ = 200 Mb/s satisfies this SLA. Furthermore, similar to the results of C.1, as we increase SLA for mMTC slice the throughput decreases for C.2. For example, when $\omega_{i,j}$ = 5 Gb/s, increasing the SLA from 75% to 100% means decreasing 621 Mb/s in the throughput, hence, dropping $\sim$ 32% of traffic.

After analyzing imposing SLA on the throughput, we now assess this impact on the performance of other slices. Fig. 4.10 shows the analysis of enforcing SLA for mMTC slice and its impact on other slices. For a fair comparison, we choose the value of capacities in the FH/BH networks where the solutions are feasible for these scenarios (as can be identified from Table 4.7). It can be seen in this figure that mMTC slice has a minimum priority to be served in all cases, and the served traffic for this slice is in the minimum percentage compared with other slices. This behavior is linked to the objective function (*i.e.,* maximizing throughput) which gives higher priority to the slices with higher data rate requirements (*i.e.,* eMBB and uRLLC slices).

In Fig. 4.11, we analyze the impact of enforcing SLA for eMBB slice and its impact on the other slices. As can be seen in this figure and as already explained, eMBB slice always has a higher priority to be served. Thus, ensuring the SLA for this slice has few impacts on the total throughput, and its effect is not as high in the performance of other slices. For example, when $\omega_{i,j}$ = 15 Gb/s, increasing the SLA from 50% to 75% for eMBB slice leads to reduce devices in the mMTC slice from 18% to 9%. Furthermore, from this figure we can observe that as far as the capacity in the FH/BH network is not enough to serve eMBB slice, it forces to drop devices in the mMTC slice which has a huge overhead in the FH/BH network. Note that when a higher SLA is imposed it results in infeasible solutions, for example in this figure when $\omega_{i,j}$ = 5 Gb/s, imposing a SLA above 50% makes the solution infeasible.

We now analyze the impact of enforcing SLA on the performance of eMBB and uRLLC slices in C.2. Fig. 4.12 shows the analysis of enforcing SLA for mMTC slice and its impact on eMBB and uRLLC slices. As can be seen in this figure, mMTC slice has the same behavior as in C.1. This slice has a lower priority (because of lower data requirements); hence it is considered as the last slice to allocate resources. For instance, when $\omega_{i,j}$ = 15 Gb/s increasing SLA from 5% to 100% for mMTC slice yields a reduction of near to 66% for eMBB slice while an increase up to 95% for mMTC slice.

Fig. 4.13 evidence the analysis of enforcing SLA for eMBB slice and its impact on other slices.

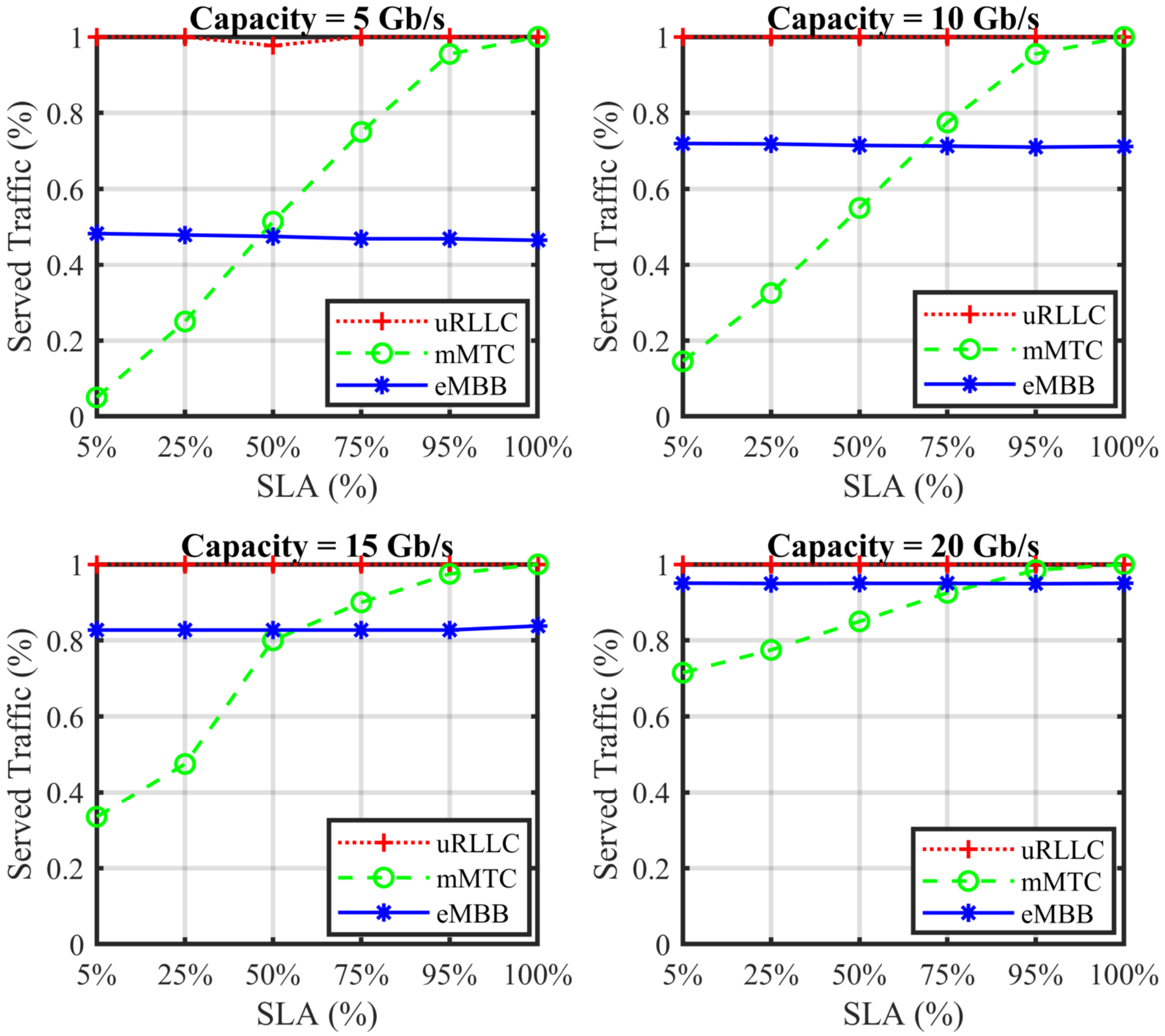


Figure 4.10: Served traffic (%) w.r.t the different SLAs imposed for mMTC slice with $R = 10$ for C.1

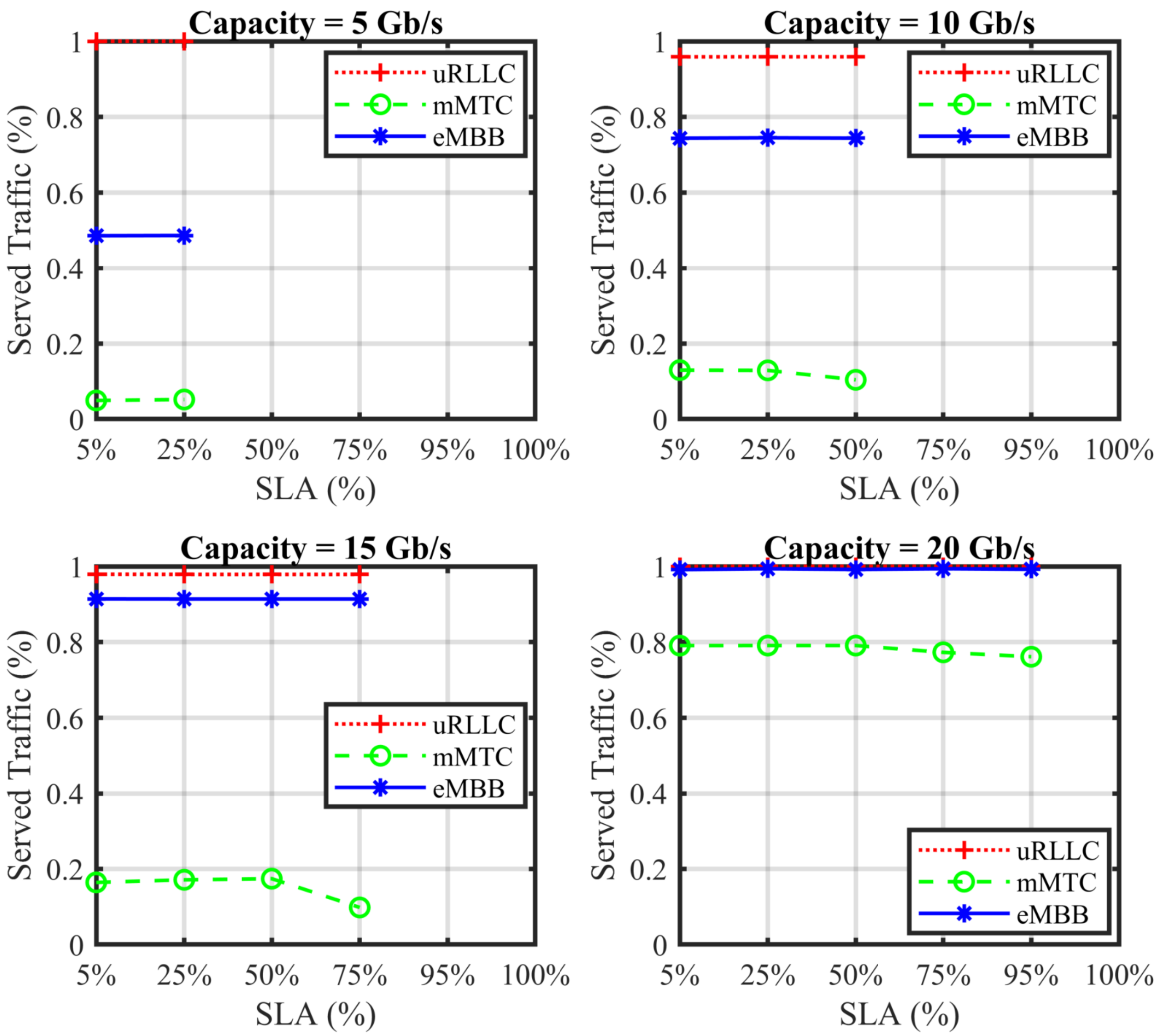


Figure 4.11: Served traffic (%) w.r.t the different SLAs imposed for eMBB slice with $R = 10$ for C.1

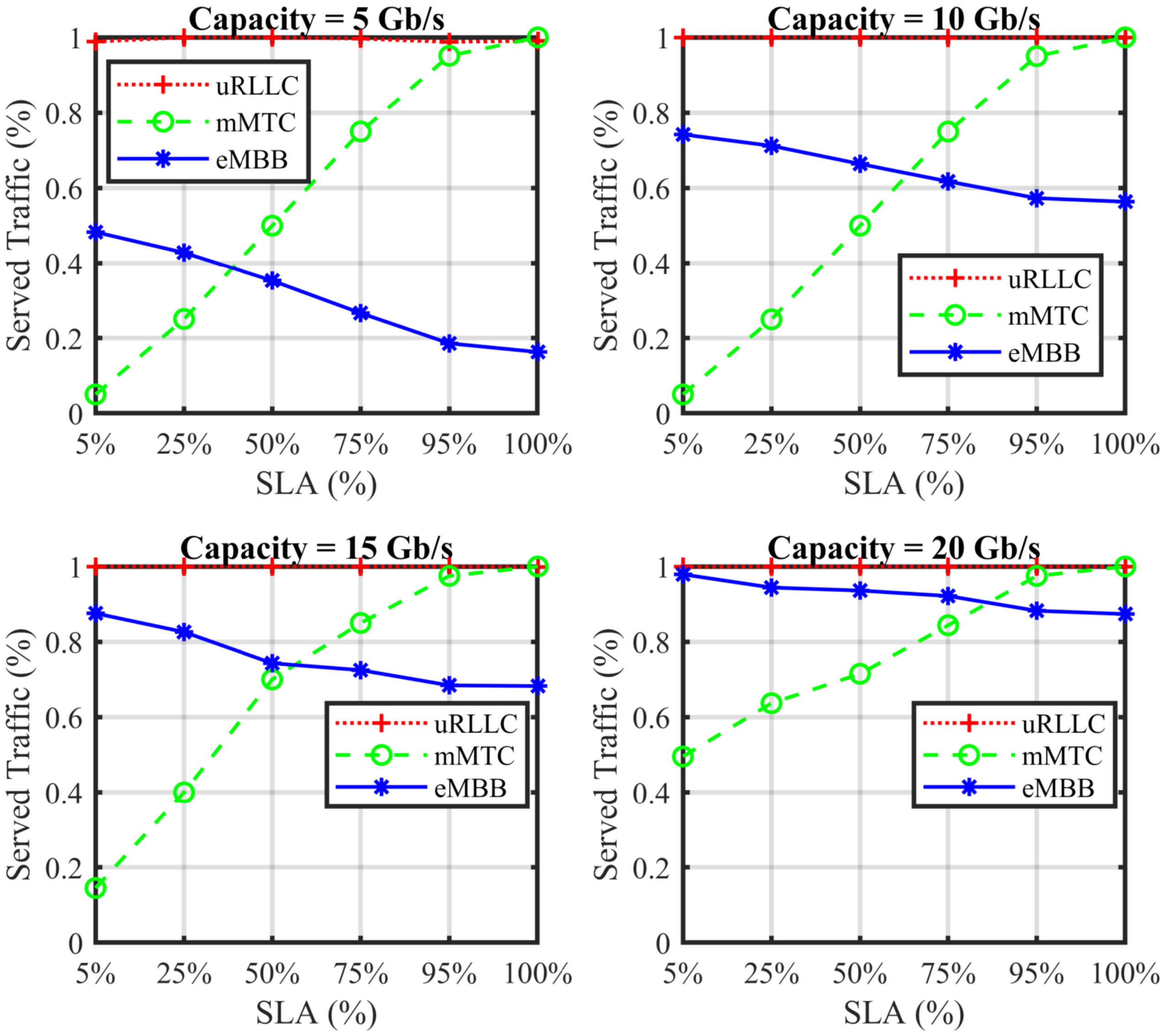


Figure 4.12: Served traffic (%) w.r.t the different SLAs imposed for mMTC slice with $R = 10$ for C.2

As already explained eMBB slice, always has a higher priority to be served. Thus, its effect is not as high for the performance of other slices and presents some limitations. For example, when $\omega_{i,j} =$ 10 Gb/s enforcing SLA for eMBB keeps mMTC slices always below 3%, and when $\omega_{i,j} = 20$ Gb/s since we have more resources to serve almost all data traffic of eMBB slices, hence, mMTC allowed to be served up to 48.5%. Note that uRLLC slice is at the prime queue to be served since it does not have extra overhead in the FH/BH network and also has more data requirements when compared to mMTC slice. Hence, uRLLC is always the top priority to be served.

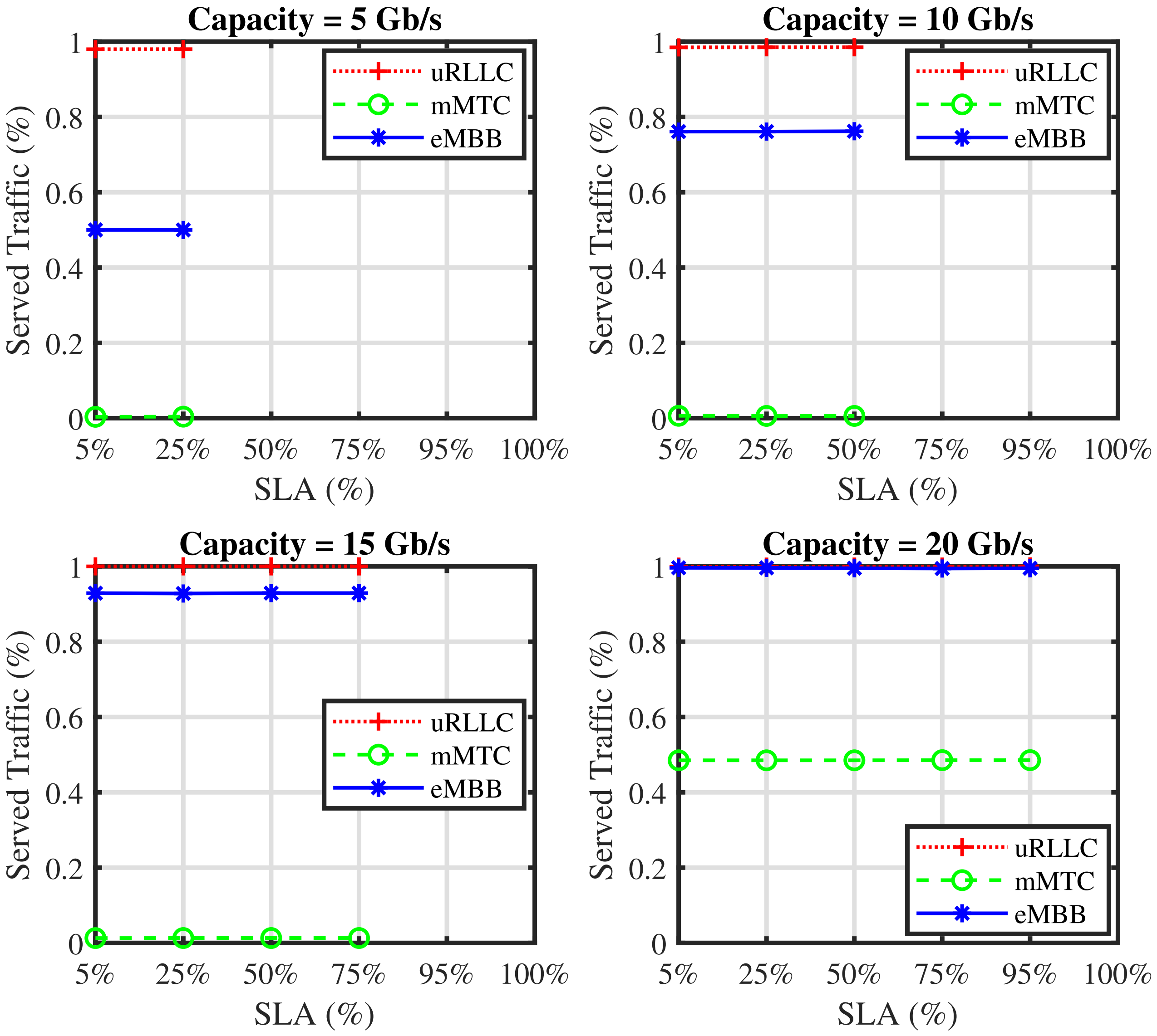


Figure 4.13: Served traffic (%) w.r.t the different SLAs imposed for eMBB slice with $R = 10$ for C.2

4.8 Conclusions

RAN slicing needs to cover two main and significant aspects namely, performing a dynamic FS of RAN and creating isolated and efficient slices based on the QoS requirements. In this work, we proposed SlicedRAN: service-aware network slicing framework for 5G RAN, which creates isolated RAN slices based on the service requirements with customized FSs per slice on top of a network composed of a CU, a FH/BH network and a set of RRHs. We first formulate a MIP framework, which maximizes the throughput by jointly selecting the optimal routing paths from a connected UE to CU, and FS while satisfying the QoS requirements. We further provide an effective heuristic method, SlicedRAN, that computes near-optimal solutions in a short computing time compared to the optimal one (*i.e.,* MIP). Our framework considers the bottlenecks in the capacity of RRHs, FH/BH network capacity along with a minimum level of SLA for each slice imposed by the different service types. The broad implication of the present research demonstrates a strong trade-off between SLA and the FH/BH network between CU and RRHs which provide a basis for designing a virtualized network infrastructure with a cost-efficient FH/BH network whilst guaranteeing SLA of different slices. The key findings of this chapter are itemized as follows:

- SlicedRAN has shown to efficiently allocate network functions (that is, select the FS) when traffic with different QoS requirements is served. This analysis found evidence for SlicedRAN model in which it prioritizes eMBB and uRLLC services sequentially. Hence, we studied imposing different SLAs to overcome this problem.

- Our findings on imposing of SLA shows that there is a trade-off between SLA and the FH/BH network between CU and RRHs. Indeed, it impacts more on the FH/BH networks rather than on the air interface. This is an important finding in the understanding of the minimum capacity required to guarantee a particular SLA and serve different QoS of slices.

- SlicedRAN mitigates the impact of limits of the network (up to 95% of traffic for mMTC slice) and guarantees on the QoS requirements of those services which have fewer bandwidth requirements (mMTC services) with a sacrifice on the other services which have a higher bandwidth requirement (up to 66% of traffic for eMBB services).

- Another promising finding was that imposing different SLAs not only affects the QoS of other slices but also has an impact on the total throughput performance. This is mainly because mMTC slice has a huge overhead and costs in the FH/BH network; hence, it leads to a drop in the eMBB and uRLLC UEs. Indeed, eMBB slice requires more spectrum resources and has a higher data rate requirement, and uRLLC slice has no overhead in the FH/BH network. Thus, a higher SLA in mMTC slice means rejecting eMBB and uRLLC slices and, thus less throughput in the network.

After analyzing joint RAN slicing and FS considering different QoS requirements imposed as SLAs, another key enabler of 5G and beyond networks, which got more attention is Multi-access Edge Computing (MEC). It can be deployed within RAN architecture in close proximity to UE to provide a computing capability with lower latency in the edge to support critical applications. The next chapter will study and explore dynamic RAN slicing by selecting optimal FS and MEC placement per slice, to solve the problem of running cost-effective edge networks and increasing the amount of traffic served with diverse QoS requirements.

Chapter 5

SO-RAN: Dynamic RAN Slicing and MEC Placement

Following our contribution in the previous chapter, RAN slicing and FS optimization need to carefully cover the isolation issue when creating RAN slices. In particular, each RAN slice needs to be tailored to distinct services based on their needs which is challenging. Likewise, Next-Generation (NG) mobile networks aim to evolve RAN from hardware-centric towards virtualized and elastic RAN.

Open RAN (O-RAN), a novel RAN architecture, is an open, and interoperable interface that embraces this evolution. Besides, new services in NG need high computation and low latency next to the end users, where Multi-access Edge Computing (MEC) placement within RAN architecture is another challenge that needs to be integrated with the RAN slicing to facilitate the deployment of critical applications with stringent Quality of Service (QoS) of NG services. Joint MEC and RAN slicing concepts are considered vital mechanisms of upcoming NG systems as they allow the creation of end-to-end isolated RAN slices and the increase of serving mobile data traffic on edge. However, this solution should be dynamic and isolated when allocating resources to different slices. In order to address the challenges mentioned above, this chapter aims to bridge the gap between the RAN slicing and MEC placements and proposes a novel optimization framework that minimizes RAN/MEC economic costs and maximizes served traffic. This framework creates isolated RAN slices by optimal placement of RAN and MEC functions per slice, which solves the problem of operating cost-efficient edge networks and increases the served traffic with diverse QoS requirements. To this aim, we first formulate our framework as a convex and linear problem and then decompose it using a distributed method that accelerates the performance and guarantees the optimal global solution.

5.1 Introduction

Next-Generations (NG) of mobile communications will undoubtedly drive the transition from inflexible and homogeneous networks to dynamic and disaggregated architectures based on softwarization, virtualization, and re-programmability of network components. These novel architectures are expected to enable new functionalities such as: (i) to provide on-demand virtual network slices that, while sharing the same physical infrastructure, are tailored to various mobile virtual network operators, network services, and traffic requirements; (ii) to split Network Functions (NF) across multiple software and hardware components; (iii) to guarantee and serve various services with significantly different Quality of Service (QoS) requirements. The International Telecommunication Union (ITU) classifies NG mobile network services into three main types [5]: i) Enhanced Mobile Broadband (eMBB) refers to the services with high-bandwidth requirements (*e.g.,* High Definition (HD) and 3D videos), ii) ultra-Reliable and Low-Latency Communications (uRLLC) is for the services demanding low-latency and high-reliability (*e.g.,* automated driving), and iii) massive Machine Type Communications (mMTC) is for the services that demand high-connection density yet with relaxed latency and throughput requirements (*e.g.,* smart city). The traditional *one-size-fits-all* approach to mobile network architecture is ineffectual to handle the complex services with different QoS requirements of NG [7]. To address such constraints, various technologies are emerging. First, the European Telecommunications Standards Institute (ETSI) [9] aims for dynamic network management and service provisioning with latency reduction via bringing cloud-computing capabilities into the edge side of mobile networks and within the Radio Access Network (RAN), which is known as Multi-access Edge Computing (MEC) platform. Second, RAN slicing emerges as an essential component at the network edge, built upon a single rigid mobile network infrastructure that creates on-demand isolated slices on top of the physical network, thus enabling dynamic use of RAN resources and opening up the potential for different types of supported services such as eMBB, uRLLC, and mMTC services [11]. The concept of MEC inside the RAN architecture utilized in this chapter is a novelty with respect to the previous work presented in chapter 4.

Furthermore, Open RAN (O-RAN) Alliance [103], which is a community of Mobile Network Operators (MNO), is committed to evolving RAN architecture that breaks down what was once a *one-size-fits-all*, hardware-centric RAN, making it more open, interoperable interfaces and elastic than currently deployed networks. The O-RAN, known as the set of gNBs in NG, consists of three main components: O-RAN Central Unit (O-CU) is the centralized and virtualized component of RAN that is in charge of the Packet Data Convergence Protocol (PDCP) layer. Its northbound interface is the BH network to the core; its southbound interface is the F1 interface. O-RAN Distributed Unit (O-DU) is the component responsible for all baseband processing, scheduling, Radio Link Control (RLC), Medium Access Control (MAC), and the upper part of the Physical layer (PHY). The F1 is the northbound interface, and the O-RAN FH is the southbound interface. O-RAN Radio Unit (O-RU) is the component responsible for the lower part of the PHY layer processing (*e.g.,* Fast Fourier Transform (FFT) / Inverse Fast Fourier Transform (IFFT), beamforming).

3GPP [13] proposes 8 different options, known as Functional Split (FS), for the distribution of these functions among RAN components. O-RAN embraces and promotes 3GPP FS options, where the O-RAN protocol stack runs as Virtual Network Functions (VNF) in a Virtual Machine (VM) divided across O-CU, O-DU and O-RUs [103]. Each FS describes how the logical nodes interact with one another and their specific tasks. A promising concept is to implement most parts of the O-RAN protocol stack as VNF in O-CU; However, it leads to stringent capacity requirements in the southbound interface. Furthermore, critical applications such as uRLLC need lower latency meantime higher computation to be served in the edge networks. The MEC platform is among the key enablers of NG, which is deployed within O-RAN in close proximity to User Equipment (UE). It provides a computing capability with lower latency in the edge to support critical applications [9]. In such an architecture, RAN slicing arises as a critical part of enabling MNOs to build virtualized networks that can be tailored to meet a variety of demands and QoS criteria in terms of functionality and isolation [89, 104]. Isolation of functions between slices is important because it allows a slice-custom FS to be configured in a shared gNB, running several slices in a physical node to have multiple FSs. The main challenge of RAN slicing is creating and managing several slices on the same shared infrastructure in an efficient and isolated manner, with minimal impact on the services of this slice or other slices, *i.e.,* guaranteeing the agreed Service Level Agreement (SLA) for each slice. Hence, two slices are isolated as long as the actions performed on one slice do not result in an SLA violation on the other slice.

As proposed by 3GPP [13] it is expected that each slice would have diverse QoS requirements. Hence, a customized functionality mapping (*i.e.,* FS selection) may be suitable for each slice to meet the QoS requirements of different services. For instance, eMBB traffic requires a high degree of coordination among gNBs to achieve high datarates. This suggests a scenario in which eMBB UEs require high bandwidth along with high-speed execution for these bandwidth-intensive applications, processing of a vast amount of data in a cloud (equivalently O-CU) [95]. This means centralizing VNFs towards the O-CU. Conversely, uRLLC needs fast retransmissions to guarantee low latency and high reliability. In that sense, decentralized FSs are needed, which means the experienced delay for this service is minimized since most of the functions are decentralized and located in the O-DUs [95]. Hence, each service gets support with the most appropriate FS, virtualized in the same gNB that in turn enhances UEs' Quality of Experience (QoE). MNOs need to set sufficient resources for each slice using scaling actions (*i.e.,* scaling up/down/out/in) depending on the load of adapted FS in that slice, up to the maximum allocated capacity. However, this scaling process should be dynamic and isolated when allocating resources to different slices, such that scaling resources for one slice does not affect the cost of resources for another slice.

5.2 State of the Art

Network slicing covers two main parts of NG networks, namely core slicing and RAN slicing. The study on mobile core slicing focuses on the virtualization of core NFs [79], combined with the use of mature virtualization technologies (*e.g.,* Docker [81]), thus enabling scalability [82] and augmenting flexibility [83]. RAN slicing, on the other hand, focuses on virtualizing RAN functions and resources, thus facilitating the sharing of gNBs among different slices. RAN slicing has gained much attention from both industry and academia recently. The 3GPP has motivated the need for RAN slicing in the NG network, which is discussed in [105]. Several proofs-of-concept RAN slicing systems have been studied in [59, 60, 106] to enable the dynamic allocation of control functions between the centralized controller and the decentralized agents and offer a cost-efficient solution for running the NG RAN as a VNF. However, despite making a step forward in the direction of RAN virtualization, the proposal still lacks a slicing design. The same authors have also proposed Orion [60], which is a RAN slicing design running on the FlexRAN platform that guarantees functional isolation among slices. A detailed study of the FS optimization can be found in Wizhaul [58] and FluidRAN [54] where they formulate a joint routing and FS optimization to maximize the Centralization Degree (CD) of the network, *i.e.,* the NFs placed at the CU, according to the availability of the network resources. Similarly, FluidRAN [54] follows the same rationale but targeting at monetary cost minimization. However, despite their insightful conclusions, the slicing option in the RAN is neglected in both of these works.

In this context, our contributions in chapters 3 and 4 which are also published in [89, 104] present a joint routing and FS optimization while considering different slices. However, these contributions lack dynamic RAN/MEC slicing that considers the impact of MEC placement inside O-RAN architecture on both throughput and network costs, as well as an assessment of slice creation costs along with UE QoE (*i.e.,* UEs' satisfaction), are missing in the previous works.

The MEC placement is crucial for RAN slicing due to various critical applications with very low latency and high computation requirements that have to be served in the edge side within RAN architecture. Different from the existing studies [54, 60, 104], our work unifies previous works and covers comprehensively joint optimization of RAN, MEC, and slice costs, serving traffic and performance of different MEC models targeting satisfying coupled constraints, which is formulated as a convex and linear problem and then is decomposed it into a distributed problem using Benders decomposition that accelerates the performance and guarantees the optimal global solution.

5.3 Contribution

In this chapter, we propose a comprehensive optimization approach with an analytical framework that jointly minimizes RAN/MEC economic costs and maximizes served traffic. A dynamic NG RAN/MEC slicing framework is presented to dynamically place the RAN protocol stack of VNFs and MEC server and allocate radio spectrum and computing resources for the slices. This framework creates isolated RAN slices by selecting optimal FS and MEC placement per slice, solving the problem of running cost-effective edge networks and increasing the amount of traffic served with diverse QoS requirements.

The design of our framework yields multiple novelties, summarized as follows:

- We provide an analytical framework that covers dynamic RAN slicing, customized FS and MEC server placement per slice. To the best of our knowledge, this is the first work proposed for O-RAN architecture to jointly optimize the throughput and cost by considering the bottlenecks in the capacity of O-RUs, MEC server computation capacity in both O-CU and O-RU along with a customized FS per slice and a SLA for each slice imposed by the main type of uRLLC, mMTC, and eMBB services.

- The proposed optimization problem is an NP-complete problem with high computation complexity. Hence, we apply an effective Benders decomposition algorithm, which reduces the complexity and meantime guarantees an exact and optimal global solution. This algorithm is general and scalable, and it can be simply modified for diverse scenarios where the respective variables and functions are convex and linear on the traffic demands.

- We present how to solve the problem of dynamic RAN/MEC slicing with tailored slices for different services with customized FSs per slice and investigate spectrum efficiency and the impact of network densification on the computing cost in O-RUs.

- We assess the computation capacities to find proper MEC settings to meet QoS requirements of different services belonging to specific slice types.

- Unlike prior approaches, such as [58] that presume a single FS per gNB, our approach leverages virtualization to create multiple RAN slices, each one with the most appropriate FS and MEC placement per slice to meet the requirements of that slice.

5.4 System Model

In this section, we present the traffic model, the O-RAN architecture, including the O-CU, O-DU, O-RU, and the transport network connecting them. Likewise, the FSs and MEC placements, the traffic model, and the problem statement are described.

5.4.1 Radio Access Network

In NG RAN architecture, gNBs are segregated into different parts: O-CU, O-DU, and O-RU, and Forwarding Nodes (FNs) in the transport network connecting (O-CU)-(O-DU)-(O-RU). The protocol stack in a gNB consists of several layers, each one responsible for a specific function or a set of functions [17]. In this context, the FS can be defined as the distribution of functions/layers between the O-CU and the O-DU. The O-RU only has radio/signaling functions; the O-CU and O-DU are computing elements, but the O-CUs are normally bigger servers while the O-DUs are smaller units placed close to the O-RUs in the network edge. As proposed in 3GPP [13], we assume that O-DUs are co-located with O-RUs. Hence, hereafter, O-RU units will also include O-DU units.

As for the bandwidth, we define ρ^r as the number of Physical Resource Blocks (PRB) allocated to O-RU r. For the sake of simplicity, as used in [93], in the following, we assume equal transmitted power per PRB with a distance-dependent path-loss model. As for the Signal-to-Noise Ratio (SNR) and the Modulation and Coding Scheme (MCS), we adopt the models used in [94].

Without loss of generality, we focus on a completely connected mesh network topology, which is however the characteristic of this envisioned evolution described in [6, 13]. Accordingly, we consider our network topology as a bipartite graph[1] [92], and define an O-RAN architecture with one O-CU, R O-RUs and a transport network composed of V FNs. Hence, the O-RAN architecture can be represented as a complete bipartite graph $G = (\mathcal{I}, \mathcal{V}, \mathcal{L})$, where $\mathcal{I}$ is the superset of O-CU (node 0) and the set of R O-DUs/O-RUs, $\mathcal{V}$ is the set of FNs (*i.e.*, routers), and $\mathcal{L}$ is the set of links connecting these elements. Accordingly, the set of all nodes is defined as $\mathcal{N} = \{0, 1, \dots, V + R\}$ in which: O-CU, referred to as node 0; the set of FNs, namely $\mathcal{V} = \{1, \dots, V\}$; and the set of O-DUs/O-RUs, $\mathcal{R} = \{V + 1, \dots, V + R\}$. Therefore, $\mathcal{N} = \mathcal{I} \cup \mathcal{V}$. The set of links $\mathcal{L} = \{l_{i,j} : i, j \in \mathcal{N}\}$, and each link $l_{i,j} \in \mathcal{L}$ has a capacity equal to $\omega_{i,j} \geq 0$ (b/s) and the respective delay $d_{i,j}$ (See Fig.5.1). The considered network topology presents a simple but realistic deployment of the RAN network topology [6, 13, 15], which does not preclude other architectures. That is, our model can directly adapt to any changes without modifying the model.

Functional splits and MEC placements. 3GPP has proposed in [13] a wide range of possible granularities for the FS, from the coarsest granularity (the FS is determined based on the computa-

[1] In the mathematical field of graph theory, a bipartite graph (*i.e.,* bigraph) is a graph whose vertices can be divided into two disjoint sets I and V (that is, I and V are each independent sets such that every edge connects a vertex in I to one in V. Vertex set I and V are often denoted as partite sets

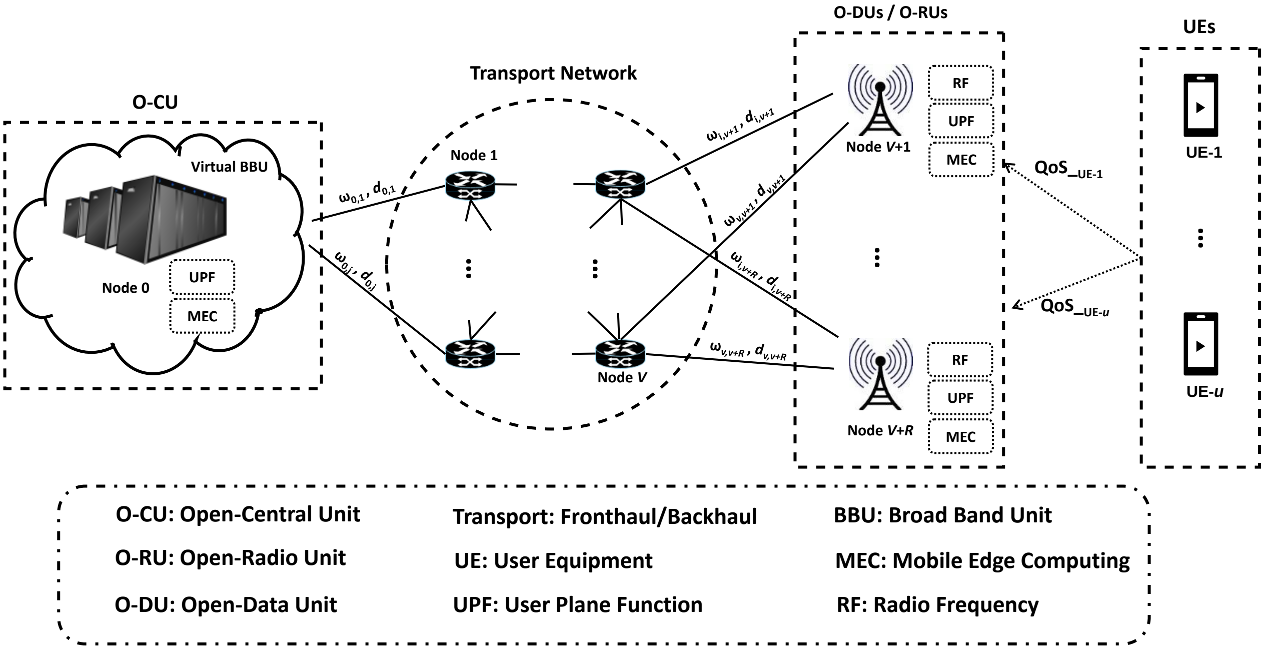


Figure 5.1: The system model.

tional capacity of the O-RU and the O-CU, as well as on the transport network capacity) to the finest granularity (the FS is decided on a UE, bearer or slice basis). As shown in [13], the network layers Radio Resource Control (RRC), Internet Protocol (IP), Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC) (high and low sublayers), Medium Access Control (MAC) (high and low sublayers), and Physical Layer (PHY) (high and low sublayers) can be allocated either in the O-CU or in the O-RU. Accordingly, each FS will be defined by the set of functions allocated in the O-CU and the set of functions allocated in the O-RU. The selection of each split is conceived as the bottleneck, indicating the minimum FS option to enable accommodation of a MEC service in O-RU or in O-CU [9]. In particular, the placement of PDCP and all layers below PDCP (*i.e.,* data link and physical layers) in O-RU enables placement of MEC functionality in O-RU as well.

Note that processing functions have a cost and need Central Processing Unit (CPU) processing resources for the placements of functions (CPU reference core per Gb/s). The computational capacity of O-CU and O-RUs is limited and expressed as κ_r for $\forall r \in \mathcal{R}$ and κ_0 for O-CU. Fig. 5.1 depicts the system model, where O-DUs/O-RUs are connected to O-CU through possible paths $\mathcal{P}^r$, wherein each path consists of a set of links $l_{i,j}$ in the transport network, which are normally designed as a mesh network [6]. The flow of these paths relies on the adapted FS.

5.4.2 Traffic Model

In our system model, we focus on the DownLink (DL) traffic, however, our study could be extended to include UpLink (UL)[2]. The set of UEs is denoted by $\mathcal{U}$, and the cardinality of the set is expressed by U. Each UE u demands a service type $s \in \mathcal{S}$, which is characterized by the packet size (payload

[2] Note that for extending our model to cover UL, it needs to consider the service and functional split requirements for the considered model.

in bits), required datarate, and latency by q_u^s, λ_u^s, d_u^s respectively. We also denote the total number of UEs and CPU computational costs of all the protocol stack of service s as n^s and c^s respectively. These demands at each O-RU create an aggregate flow emanating from the O-CU routed to O-RU.

A first radio feature that is considered as an essential enabler for slicing at the RAN level is the tiling scheme [107], which is a practical implementation of the so-called NG flexible numerology concept. This offers the opportunity of serving different services using different Sub Carrier Spacing (SCS) or Transmission Time Interval (TTI) lengths with a flexible transmission time in the PHY layer. The principle is that time-frequency resources with the same numerology (γ) are grouped together within a tile, where UEs are classified into specific tiles (per slice) based on the requirements of their service types.

Spectrum sharing through flexible numerology for each slice. We consider three main use cases of ITU, namely eMBB, mMTC, uRLLC, which are defined as $s \in \{0, 1, 2\}$ respectively. We consider the following assumption for each slice based on their QoS requirements [108]:

- uRLLC: uses short TTIs (*e.g.,* 0.25 ms) to meet latency requirements, hence, larger SCS is useful for this use case.

- mMTC: uses a lower bandwidth with a more extended TTI size to save device energy and increase coverage (similar to the Narrow Band IoT). Hence, a TTI of 0.5 ms is beneficial for this use case.

- eMBB: uses a TTI of 1 ms to minimize control overhead.

Hence, we assume each tile is served by one slice instance and uses one type of numerology, which can be calculated by:

$$\gamma_s = \frac{\gamma_0}{2^{\iota_s}}, \tag{5.4.1}$$

where the slot length is defined as $\gamma_s \in \{1ms, 0.5ms, 0.25ms\}$ and $\gamma_0 = 1$ and $\iota_s \in \mathbb{Z}^{\geq}$ takes specific values per slice (non-negative integer numbers). For eMBB slice (*i.e.,* $s = 0$) $\gamma_0 = 1$, mMTC slice (*i.e.,* $s = 1$) $\gamma_1 = 0.5$, and finally uRLLC slice (*i.e.,* $s = 2$) $\gamma_2 = 0.25$.

With pre-fixed slot length, the SCS of each type of numerology is varied as $\mu_s \in \{15kHz, 30kHz, 60kHz\}$.

Accordingly, for eMBB slice $\mu_o = 15$, mMTC slice $\mu_1 = 30$, and finally uRLLC slice $\mu_2 = 60$.

5.4.3 Problem Statement

The main objective of the MNO is to select the flexible FS of RAN configuration and optimal MEC placements either in O-CU or O-RU that will satisfy the UE's demand while minimizing the total monetary cost, which arises significant challenges with several trade-offs. On the one hand, placing VNFs at O-RUs enables to place MEC at O-RUs and meet low delay requirements on the edge side, next to UEs. This leads to increasing the computational needs in the O-RUs, and high cost (both monetary and computation) since the capacity of MEC servers at O-RU is limited. On the other hand, offloading NFs and pooling them at the O-CU (*i.e.,* centralization), benefits from a reduction of the computational capacity/cost required at the O-RUs and offers centralized control to better coordination. This can improve the network's performance and meet the high-computation requirements of UEs in the cloud center (*i.e.,* O-CU) with higher transport bandwidth and low delay requirements using high-capacity MEC platforms, which has a lower cost of installation.

To this aim, MNOs could use the benefits of network slicing by creating isolated slices with MEC-enabled servers and VNFs both in O-CU and O-RUs, which reduce MNOs' costs and provide a virtualized and flexible architecture with scalable resources to meet QoS requirements of UEs imposed as SLA contracts/objectives. The MNOs' decisions need to be fine-grained, *i.e.,* per O-RU, and consider all the above aspects using the capabilities of RAN slicing.

5.5 Dynamic RAN/MEC slicing design.

In this context, a dynamic network slicing algorithm provides the network with a higher degree of flexibility to address issues discussed above.

5.5.1 Slice-Specific and Dynamic FS/MEC Placements

Without precluding any of the granularity levels proposed by 3GPP, in our work, we focus on the slice-based FS, assuming that one slice is created for each service[3]. For the slice-specific approach, we consider two key FS options of 3GPP option 7-2 and option 1 to find the joint dynamic FS and MEC server placement based on QoS requirements of UEs.

Split 7-2, only RF and low-PHY functions are located at O-RU, while the rest of functions are moved to O-CU (complete centralization), thus transmitting IQ samples through the transport. In this case, samples are usually encapsulated with Common Public Radio Interface (CPRI) [76] and the required FH capacity depends on the bandwidth allocated to the gNB, the number of antennas, etc. That is, transport capacity requirement does not depend on the UEs' traffic for this split option of 3GPP. The main advantage of this split is that the centralization achieves the highest coordination degree among gNBs.

Split 1 is a completely decentralized FS that accommodates all functions except RRC-IP at O-RU. That is, PDCP and all layers below PDCP run in the O-RU. Given the allocation of functions, this split does not have traffic overhead and the required transport capacity can be approximated by the aggregate UEs' traffic. This split is also known as the Control-/User Plane Separation (CUPS) split, as RRC contains the Control Plane Functions (CPF), and the User Plane Functions (UPF) are operated from the PDCP and above [31]. Note that low-PHY and RF functionalities are known as cell-specific functions, which are always placed in O-RU. Table 5.1 includes a summary of the allocation of functions and the associated transport bandwidth and delay requirements for each split [54].

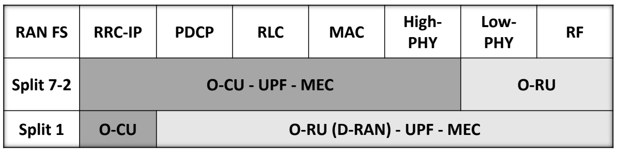
Accordingly, if split 1 is selected, then MEC would be placed in O-RU, but for split 7-2, MEC could be placed in O-CU (See Fig. 5.2).

However, this architecture needs to be flexible enough to cover the QoS requirements of NG by

[3] Given that slicing will be done on service type basis as highlighted by 3GPP [10], hereafter service and slice concepts will be interchangeable.

Table 5.1: Functions' allocation and transport bandwidth and delay requirements for a traffic denoted by λ_u^s, for UE u and service s, with 20 MHz bandwidth; Downlink: MCS index 28, 2x2 MIMO replicated from [54].

Split Type	Traffic Load (b/s)	Delay (ms)	MEC Function at O-RU
7-2	$2.5 \cdot 10^9$	0.25	$f = 0$
1	λ_u^s	30	$f = 1$



RAN FS	RRC-IP	PDCP	RLC	MAC	High-PHY	Low-PHY	RF
Split 7-2	O-CU - UPF - MEC					O-RU	
Split 1	O-CU	O-RU (D-RAN) - UPF - MEC					

Figure 5.2: FS options for MEC and UPF placement.

virtualization of RAN NFs. This means the functions above the PHY layer could be virtualized and run in a VM. In this way, using VM could help to support both FS options within RAN architecture, which enables co-location of MEC and UPF (protocol stack of RAN NFs).

Table 5.2: Summary of Notations

Symbol	Description
Sets	
$\mathcal{R}$	Set of O-RUs
$\mathcal{Q}$	Set of FNs
$\mathcal{U}$	Set of UEs
$\mathcal{F}$	Set of network functions
Parameters	
$\omega_{i,j}$	Total bit-rate capacity of link $l_{i,j}$ (b/s)
$\rho_u^{r,s}$	Required PRB of UEs to connect to O-RU r
λ_u^s	Datarate of UE u with type s (b/s)
$T^{r,s}$	Traffic served by slice s of O-RU r
c^s	CPU consumption to compile RAN functions (RCs per Mb/s)
κ_r	Computation capacity of each O-RU (RCs per Gb/s)
κ_0	Computation capacity of O-CU (RCs per Gb/s)
τ^s	Proportion of UEs (*i.e.*, SLA) for slice/service type s
n^s	Number of UEs with slice/service type s
W	Bandwidth of a PRB (kHz)
Variables	
$\rho^{r,s}$	The variable to automate of PRB allocation per slice
$x_u^{r,s}$	Binary variable to associate UE u with type s to O-RU r
t_p^r	The variable to show the traffic routing from O-CU to O-RU r
$f^{r,s}$	The variable to indicate the placement of functions
$y_{i,j}^p$	The variable to indicate if the path p includes link $l_{i,j}$

5.6 Problem Formulation

To consider all the above aspects, we formulate a novel optimization framework for dynamic RAN/MEC slicing, abstracting slice-customized FS, isolated slices, dynamic spectrum sharing, and describe the proposed optimization problem formulation.

UE association. When creating a slice, we assume that each UE is connected to a single O-RU. The UE association can be defined as:

$$\sum_{r \in \mathcal{R}} x_u^{r,s} \leq 1, \forall u \in \mathcal{U}, \forall s \in \mathcal{S}, \tag{5.6.1}$$

where $x_u^{r,s} \in \{0,1\}$ is a binary variable to check whether UE u of s service type is connected to O-RU r or not.

Dynamic spectrum sharing per slice. We can calculate the spectrum usage of each slice in terms of PRB, which has 12 sub-carriers. Let's define $\rho^{r,s}$ as a variable be the number of PRBs dynamically allocated to the s-th slice at O-RU r, as:

$$\sum_{s \in \mathcal{S}} 12 \cdot \mu_s \cdot \rho^{r,s} \leq W, \forall r \in \mathcal{R}, \tag{5.6.2}$$

where W is the total bandwidth.

Moreover, we use Shannon theory to define the transmission rate per PRB:

$$\eta_u^{r,s} = 12 \cdot \mu_s \cdot \log_2(1 + SNR_u^{r,s}), \tag{5.6.3}$$

where $SNR_u^{r,s}$ is the SNR between UE u with service s and O-RU r. Moreover, the multiplication of SCS μ_s with 12 sub-carriers defines the spectrum usage of each slice.

The datarate of UE u with type s defined by λ_u^s (b/s) and $\rho_u^{r,s}$ is the PRB required by UE u with type s which can be computed as $\rho_u^{r,s} = \lceil \frac{\lambda_u^s}{\eta_u^{r,s}} \rceil$. Hence, the total PRB required by all the associated UEs should be less than or equal to the available PRB in O-RU r for slice s:

$$\sum_{u \in \mathcal{U}} x_u^{r,s} \cdot \rho_u^{r,s} \leq \rho^{r,s}, \forall s \in S, \forall r \in \mathcal{R}, \tag{5.6.4}$$

where $\rho^{r,s}$ is the available PRBs in O-RU r, which will be dynamically scaled based on the traffic load of each slice/service s.

RAN routing decisions. Given that the traffic served by O-RU r can be forwarded through any of the paths in $\mathcal{P}^r$, we define the traffic over one of these paths as t_p^r, where $p \in \mathcal{P}^r$. Therefore, the total traffic served by O-RU r can be expressed as $\sum_{p \in \mathcal{P}^r} t_p^r$. Similarly, the traffic that traverses the transport network depends not only on the traffic received/transmitted by/from the UEs but also on the FS. Thus, a UE served by O-RU r generating a traffic λ_u^s causes a traffic through the transport network equal to $T_u^{r,s} = a_{\phi^{r,s}} \lambda_u^s + b_{\phi^{r,s}}$, where λ_u^s is the traffic generated by UE u with service s and $\phi^{r,s}$ is the FS used in O-RU r for service s, which can choose either FS 1 or 7-2. As for $a_{\phi^{r,s}}$ and $b_{\phi^{r,s}}$, they are coefficients and used to properly calculate the traffic load in the transport network, and depend on the FS used in O-RU r for service s (*i.e.,* $\phi^{r,s}$). As observed in Table 5.1, when service/slice s uses FS 1, i.e. $\phi^{r,s} = 1$, we have $a_1 = 1$ and $b_1 = 0$. In the case of split 7-2, *i.e.,* $\phi^{r,s} = 7$, $a_7 = 0$ and $b_7 = 2.5 \cdot 10^9 \cdot \frac{\rho_u^r}{100}$ b/s[4], where ρ_u^r is the bandwidth allocated to UE u at O-RU r expressed in number of PRBs. Notice that the bandwidth requirements in the transport network depend on the

[4] According to literature, the required transmission rate required for 20 MHz bandwidth (*i.e.,* 100 PRBs) is around 2.5 Gb/s. This is the reason why the number of PRBs is normalized with respect to 100 PRBs

FS type. For split 7-2, it relies solely on the O-RU bandwidth and is load-independent. That is, the capacity requirement in the transport network does not depend on the UEs' traffic [54] since In-Phase and Quadrature (IQ) samples of the allocated bandwidth are transmitted, no matter if there is traffic or not. For split 1, λ_u^s is the throughput of a single UE s with type s, and if more UEs are served with the available bandwidth of O-RU, then the traffic load is the $\sum_{u\in\mathcal{U}}\sum_{s\in\mathcal{S}}\lambda_u^s$, for all UEs served by the O-RU.

According to the definitions stated above, the traffic traversing the transport network to serve UEs with service s connected to O-RU r is given by

$$T^{r,s} = \sum_{u\in\mathcal{U}} x_u^{r,s}\cdot T_u^{r,s}, \tag{5.6.5}$$

Each O-RU can run different slices and serve different services[5] simultaneously. Thus, if we define the traffic served by slice s of O-RU r as $T^{r,s}$, the total traffic served by O-RU r can be expressed as $\sum_{s\in\mathcal{S}} T^{r,s}$. Therefore, it holds that

$$\sum_{p\in\mathcal{P}^r} t_p^r = \sum_{s\in\mathcal{S}} T^{r,s}, \tag{5.6.6}$$

Hence, (5.6.6) gives the total traffic traversing the transport network to serve UEs with service s served by O-RU r (b/s). The routing decisions need to respect the link capacities:

$$\sum_{r\in\mathcal{R}}\sum_{p\in P^r} t_p^r\cdot y_{i,j}^p \leq \omega_{i,j},\ \forall j\neq i\in\mathcal{N}. \tag{5.6.7}$$

Constraint (5.6.7) states that the flow from each O-RU r to O-CU is bounded by the capacity of the links of the paths, denoted as $\omega_{i,j}$ (b/s). The indicator parameter $y_{i,j}^p\in\{0,1\}$ is used to to check if the path p includes link $l_{i,j}$ or not.

The accommodation of cell-processing and MEC functions in the O-RU r or in the O-CU depends exclusively on the adopted FS. We define the variable $f^{r,s}\in\{0,1\}$; If UPF and MEC functions runs in the O-RU r for service/slice s, then $f^{r,s}=1$. Conversely, if they run in the O-CU, then $f^{r,s}=0$. By inspecting Table 5.1, it can be shown that, when service s uses split 1, then $f^{r,s}=1$. If it uses split 7-2, then $f^{r,s}=0$. Thus, in general, when service/slice s uses FS $\phi^{r,s}$,

$$f^{r,s} = \begin{cases} 0 & \text{if } \phi^{r,s}=1 \\ 1 & \text{if } \phi^{r,s}=7 \end{cases}, \tag{5.6.8}$$

where $\phi^{r,s}=\{1,7\}$.

Network function computation. The deployment of the protocol stack of UPF and MEC functions at O-RU or at O-CU incurs a computational cost. In the following, the computational cost at O-RU and O-CU is stated. For O-RU r is given by:

$$F_r = \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}} x_u^{r,s}\cdot\lambda_u^s\cdot(c^s\cdot f^{r,s}) \leq \kappa_r,\ \forall r\in\mathcal{R}, \tag{5.6.9}$$

[5] Please, recall that *service* and *slice* are used interchangeably, and we assume that the network creates a slice per each service.

where c^s is the computational cost of functions (UPF and MEC) of slice s located at O-RU in CPU operations per bit per second. It ensures the computation capacity needed to process NFs per slice s located in O-RU is less than the available computation capacity in O-RU r (κ_r). As for O-CU:

$$F_0 = \sum_{r\in\mathcal{R}}\sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}} x_u^{r,s}\cdot\lambda_u^s\cdot(c^s\cdot(1-f^{r,s}))\leq\kappa_0, \tag{5.6.10}$$

This constraint is used to bound the maximum computational capacity supported by O-CU (κ_0).

Delay constraints. The total delay for serving different UEs mainly depends on the traffic load, the execution time of NFs, and MEC computation requirements, as:

$$D_{tot}^{u,s} = \sum_{r\in\mathcal{R}} x_u^{r,s}\cdot\Big(D_{prc}^{u,s}+D_{trn}^{u,s}+D_{prp}^{u,s}\Big), \tag{5.6.11}$$

To obtain the overall processing delay experienced $D_{prc}^{u,s}$ for UE u with type s we formulate it as follows:

$$D_{prc}^{u,s} = \Big(\frac{F_0^s}{\kappa_0}\Big)+\Big(\frac{F_r^s}{\kappa_r}\Big), \forall u\in\mathcal{U},\forall s\in\mathcal{S}, \tag{5.6.12}$$

where $F_0^s = \sum_{r\in\mathcal{R}}\lambda_u^s\cdot x_u^{r,s}\cdot c^s(1-f^{r,s})$, and $F_r^s = x_u^{r,s}\cdot\lambda_u^s\cdot c^s\cdot f^{r,s}$, which are defined above.

The transmission delay for UE u of type s can be defined as:

$$D_{trn}^{u,s} = \sum_{p\in\mathcal{P}^r}\sum_{l_{i,j}\in\mathcal{L},j\neq i}\Big(\frac{q_u^s}{\omega_{i,j}}\Big)\cdot y_{i,j}^p, \forall u\in\mathcal{U},\forall s\in\mathcal{S}. \tag{5.6.13}$$

In general, propagation time $D_{prp}^{u,s}$ is proportional to the physical distance between the transmitter and the receiver, however, the difference between the propagation time of different O-RUs is negligible. For all O-RUs and O-CU which induces a per-hop latency equal to the propagation delay-distance $10\mu/km$ for fiber connectivity [58, 109].

As for routing paths between O-CU and O-RUs, not all paths are feasible solutions for a given FS used for specific slice. Hence, we partition paths to a set of sub-paths, each one eligible per slice, which depends on the FS selection. In particular, only paths with an aggregate delay below the maximum delay supported by the FS will be considered. Thus, based on Table 5.1, we define for each O-RU r the set of sub-paths with a delay above 30 ms (the maximum delay allowed by split 1) as P_r^1. Similarly, P_r^7 as the set of sub-paths with a delay larger than 0.25 ms. The following constraints are used to disconnect (make them zero) the paths with larger than the selected split delay requirements. For instance, equation (5.6.14) is used for uRLLC slice, which uses split 1, and zeroize the flow paths which have a larger than 30ms delay. Similarly, equation (5.6.15) is used to zeroize the paths with a higher delay than splits requirements.

$$\sum_{p\in P_r^1} t_p^r \leq M(1-x_u^{r,s}), \forall r\in\mathcal{R}, s=2 \tag{5.6.14}$$

$$\sum_{p\in P_r^7} t_p^r \leq M(1-x_u^{r,s}), \forall r\in\mathcal{R}, \forall s\in\{0,1\} \tag{5.6.15}$$

where M is a big-number.

Slice-customized SLAs. We need to cover different QoS requirement of UEs as SLA objectives, and we need to add SLA constraints to the network. Since each network slice may have different SLA contracts with MNOs, hence, we consider specific SLA per slice based on SLA objective. Accordingly, we impose a minimum SLA for each slice in constraint (5.6.16) - (5.6.17) to guarantee the SLA of a particular slice.

Imposing delay SLA as a threshold for each slice.

$$\sum_{u\in U} D_{tot}^{u,s} \leq d_t^s, \forall s \in \mathcal{S}, \tag{5.6.16}$$

where d_t^s is the delay threshold imposed as SLA objective for service/slice s.

Imposing UE's percentage SLA as a threshold for each slice.

$$\sum_{u\in U}\sum_{r\in R} x_u^{r,s} \geq \tau^s.n^s, \forall s \in \mathcal{S}, \tag{5.6.17}$$

where $\tau^s \in [0,1]$ stands for the minimum proportion of UEs with service type s that must be served with the required QoS and n^s is the total number of UEs with service type s. Thus, the minimum number of UEs that should meet the required QoS is $\tau^s \cdot n^s$. In the following, we propose an optimization solution aimed to maximize the throughput of the NG network by selecting the slice-specific FS, routing, and dynamic spectrum sharing per slice while satisfying the SLA objectives of each slice. Hence, the network throughput, which is our objective function, can be written as:

$$O = \sum_{r\in R}\sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}} \lambda_u^s \cdot x_u^{r,s} \tag{5.6.18}$$

Putting the above constraints together, we can introduce the mathematical problem $\mathbb{P}1$, where the objective function is maximizing the throughput. The constraints of the optimization model are defined in (5.4.1) - (5.6.17).

Constraint (5.6.19) defines $x_u^{r,s}$ as a binary variable. Finally, (5.6.20) defines the set of allowed FSs for a slice/service.

Lemma 1. The constraints (5.6.7), (5.6.9), and (5.6.10) show non-linearity due to the product of two integer/binary variables, however, these constraints are linearized to simplify and maintain the problem as a convex and linear problem.

Proof. In order to simplify and linearize the problem, we define $y_{i,j}^p$ and $f^{r,s}$ as indicator parameters that are controlled offline. This means the indication of each one varies based on the conditions of network, where $f^{r,s}$ indicating the placement of network functions (*i.e.,* FS selection), which relies on the association of decision binary variable $x_u^{r,s}$ and mainly depends on the type of services with associated QoS requirements of each service that connects to different O-RUs. That is, due to the chain of UE's association and network functions, x cannot be associated with O-RU r

$$\mathbb{P}1\text{: } \max_{x,t,\rho} \quad O(x,t,\rho)$$

s.t:

$$(5.6.1)-(5.6.17)$$

$$x_u^{r,s} \in \{0,1\}, \forall r \in \mathcal{R}, \forall s \in \mathcal{S}, \forall u \in \mathcal{U} \tag{5.6.19}$$

$$\phi^{r,s} \in \{1,7\}, \forall r \in \mathcal{R}, \forall s \in \mathcal{S} \tag{5.6.20}$$

unless f is deployed there, hence, $x \leq f$. Similarly, the binary variable (*i.e.,* indicative parameter) $y_{i,j}^{p}$ based on the existing links in the (O-CU)-(O-RUs) paths indicate whether the link $l_{i,j}$ belongs to the path p or not. Hence, if $y_{i,j}^{p}$ exists in path p, then $y_{i,j}^{p} = 1$, otherwise, $y_{i,j}^{p} = 0$.

This problem outputs the UEs' association x, the UPF and MEC placements f, the dynamic spectrum per slice ρ, and the routing variables t for the transport network.

Theorem 1. *The optimization model* $\mathbb{P}1$ *turns out to be a MIP problem, which is an NP-complete problem, and has a complexity of* $\mathcal{O}(2^N)$.

Proof. The optimization model $\mathbb{P}1$ that maximizes the network throughput is a MIP problem, for which commercial and free solvers can be used. Generally, a MIP problem is known to be an NP-complete problem, and as the computation time for NP-complete problems is high [96], the number of nodes (UEs and O-RUs) in the network negatively affects the computation time due to the increment of the search space of variables. Jeroslow [98] proved that the complexity of branch-and-bound for a MIP problem is $\mathcal{O}(2^N)$, where N is the number of variables in the optimization. In our optimization problem, we have binary ($x_u^{r,s}$) and continuous (t_p^r) variables that require the branch-and-bound method.

Proposition 1. We have proved that $\mathbb{P}1$ is recognized as an NP-complete problem and has high computation complexity to solve it. Hence, we follow a solution algorithm for the problem $\mathbb{P}1$ to reduce its complexity and meantime guarantee an optimal global solution. However, the solution is generic and scalable, which can be easily extended for various scenarios where their corresponding variables and functions are strictly convex and linear [110] on the UEs' traffic load.

5.6.1 Solution: Decomposed and Distributed Algorithm

The computational complexity of the $\mathbb{P}1$ problem increases substantially for large scale networks, and the number of nodes in the network negatively affects this computation time. To solve this problem, commercial and available solvers such as CPLEX [97] or Gurobi [6] can be used, as we will explain in the Performance Evaluation section. The common standard techniques used in these solvers are

[6] https://www.gurobi.com/

the simplex and branch and bound algorithms, which are usually treated as black-box solvers, and sometimes these problems are unsolvable using the aforementioned algorithms. Given that $\mathbb{P}1$ includes complicated binary variables and is recognized as MIP problem (See **Theorem 1.**). For these kinds of problems, with the increase of data load, solving them in commercial solvers using simplex or branch and bound algorithms take a long time to find the global optimal solution. Benders' decomposition algorithm is one of the popular decomposition schemes [111] that exploits the structure of a given MIP problem and decentralizes the overall computational burden. Benders algorithm is well-suited for network design and network optimization problems [112, 113], where it decomposes the problem to be solved into two simpler problems, namely the Master-Problem (MP) and the Sub-Problem (SP). It has already been proved in [114, 115] that Benders decomposition outperforms existing algorithms in commercial solvers like a simple branch and bound or branch and cut significantly, especially as the number of nodes increases. Hence, to overcome the issue of $\mathbb{P}1$ problem, we use Benders' decomposition algorithm that allows us the distributed solution of $\mathbb{P}1$ problem and finding an exact optimal point. In Benders, the MP is a relaxed version of the Original-Problem (OP), *i.e.,* $\mathbb{P}1$, containing only the integer/binary variables, and the associated constraints. For maximization problems, its solution gives an Upper Bound (UB) on the objective function of the OP. The SP is the OP with the variables obtained in the MP fixed, and includes all continuous variables and the associated constraints. Solving the dual of the SP (DSP) provides information about the SP portion of the objective function of OP, which yields a Lower Bound (LB) on the objective function of the problem and is used to generate cuts for the MP. The MP and SP are solved iteratively until the LB and UB are sufficiently close. Note that Benders guarantees convergence to a globally optimal solution as far as the OP is convex and linear. In our problem, Benders algorithm optimizes the binary variables for the UEs' association, and the UPF and MEC placements in MP $\mathbb{P}1_{MP}$ for fixed routing; and in SP $\mathbb{P}1_{SP}$ solves the routing variables for transport network, and the dynamic spectrum for slices for fixed UEs' assignments and FS. Accordingly, we fix the binary variables of UEs' association to $\bar{x}$ to solve $\mathbb{P}1_{SP}$:

$$\mathbb{P}1_{SP}\colon \underset{t,\rho\geq 0}{\max.}\ \ O(\bar{x},t,\rho)$$

s.t:

(5.6.2) - (5.6.4) - (5.6.6) - (5.6.7) - (5.6.14) - (5.6.15)

In the Benders framework, rather than solving the SP, we solve its dual (*i.e.,* DSP), which can be computed as

$$\mathbb{P}1_{DSP}\colon \underset{\nu,\sigma}{\min.}\ \ Z(\bar{x},\nu,\sigma) s.t.\ \ \mathcal{H}^{\top}\cdot\nu\cdot\sigma\geq\zeta, \tag{5.6.21}$$

where ζ is the co-efficient of t and $\rho, \mathcal{H}$ is set by the objective and constraints $\mathbb{P}1_{SP}$, ν and σ are the matrices of the dual variables (one for each constraint in $\mathbb{P}1_{SP}$).

The MP $\mathbb{P}1_{MP}$ solves the integer/binary variables if we fix the routing and the spectrum continuous variables to $\bar{t}, \bar{\rho}$:

$$\mathbb{P}1_{MP}\text{: } \max_{x,\Theta\geq 0}. \; O(x,\bar{t},\bar{\rho}) + \Theta$$

s.t:

(5.6.1) - (5.6.4) - (5.6.5) - (5.6.9) - (5.6.10) - (5.6.11) - (5.6.17)

$$\Theta \leq Z(x,\nu^{\varepsilon},\sigma^{\varepsilon}), \forall \nu^{\varepsilon},\sigma^{\varepsilon} \in \xi_{opt}, \tag{5.6.22}$$

$$0 \leq Z(x,\nu^{\varepsilon},\sigma^{\varepsilon}), \forall \nu^{\varepsilon},\sigma^{\varepsilon} \in \xi_{fes}, \tag{5.6.23}$$

where (5.6.22) and (5.6.23) are the optimality and feasibility cuts, respectively, which gradually construct the entire constraint set of $\mathbb{P}1_{MP}$. The intuition behind this method is that the optimal solution can be found before a full re-construction is built. The optimization steps of proposed solution are summarized in Algorithm 2. We use the set ξ_{opt} to obtain the optimality cuts, and the set ξ_{fes} to get the feasibility cuts. We first solve the MP ($\mathbb{P}1_{MP}$) in each iteration k to get the currently optimum configurations x^k, and the surrogate parameter Θ^k (Line 2). The current UB UB^k is set using this assignment (Line 3-5). We next use the current variable x^k and parameter Θ^k to solve the DSP $\mathbb{P}1_{DSP}$ (Line 6-7). Then, using the value of the relaxed MP, we compute the new LB LB^k (Line 8-9). For each iteration k, we update the sets ξ_{opt} and ξ_{fes}, by adding the respective values ν^r in ξ_{opt} if the dual optimal value is bounded. Otherwise, if unbounded, we add the value to the set of feasible cuts in ξ_{fes} as it gives us the information of feasible solutions (Line 11-15); these will be used in solving $\mathbb{P}1_{MP}$ in the next iteration. In each iteration, the new cuts generate new constraints, and this shortens the solution space. These stages continue until UB and LB's difference becomes less than ϵ (Line 19). Note that the precision of the result can be tweaked by choosing ϵ, and if the result is unfeasible, then we will obtain an unbounded value for the slave problem in the first iteration [111].

Theorem 2. Convergence of Algorithm 2. *Algorithm 2 converges to a global optimal solution of* $\mathbb{P}1$ *within a finite number of iterations.*

Proof. The algorithm only stops when it has determined an optimal solution, and it has proven it to be such. We refer our proof to Proposition 2.19 in [116] and [111], which precisely apply for Algorithm 2 and supports that there always exists the feasibility cuts and finitely many (dual) feasible basis which can produce the optimality cuts. Indeed at each iteration of Algorithm 2, the set of (x, Θ) in $\mathbb{P}1_{MP}$ is shrinked by introducing one extreme halfline of the polyhedral cone C (*i.e.,* either the constraint (5.6.22) or (5.6.23)) until to reach an optimal solution, thus the number of optimality and feasibility cuts is finite. Let (x^*, Θ^*) be an optimal solution to the reformulated original problem.

- 1. The feasibility set of the master problem $\mathbb{P}1_{MP}$ is always contained in the feasibility set of the master problem with cuts (5.6.23), *i.e.,* no feasible solutions are cut in Algorithm 1.
- 2. The optimal solution $(\bar{x}, \bar{\Theta})$ obtained by the algorithm is feasible for the master problem

$\mathbb{P}1_{MP}$ due to the constraint (5.6.22). It means Algorithm 1 only admits a solution with bounded value.

Thus, from 1. and 2. we can obtain at which point/iteration master and subproblem values ensures the convergence of Algorithm 1 to optimum:

$$O(x^*, \bar{t}, \bar{\rho}) + \Theta^* \leq O(\bar{x}, \bar{t}, \bar{\rho}) + \bar{\Theta} \leq O(x^*, \bar{t}, \bar{\rho}) + \Theta^* \tag{5.6.24}$$

Algorithm 2: Benders Decomposition Algorithm

```
Initialize: $\epsilon, k \leftarrow 1, \xi_{opt}^k, \xi_{fes}^k,\ UB^k \leftarrow +\infty, LB^k \leftarrow -\infty$
repeat
    Solve $\mathbb{P}1_{MP}(\xi_{fes}^k, \xi_{opt}^k)$ to obtain $\Theta^k, x^k$
    if $(\Theta^k < UB^k)$ then
        $UB^k \leftarrow \Theta^k$
    end
    For fixed $\bar{\Theta}^k, \bar{x}^k$
    Solve $\mathbb{P}1_{DSP}$ to obtain $\nu^k, \sigma^k$
    if $(LB^k > LB^{k-1})$ then
        $LB^k \leftarrow F_0^k + \sum_{r \in \mathcal{R}} F_r^k + Z(x^k, \nu^k, \sigma^k)$
    end
    if $(Z(x^k, \nu^k, \sigma^k) < \infty$ then
        $\xi_{opt}^{k+1} \leftarrow \xi_{opt}^k \cup \{\nu^r\}$
        else
            $\xi_{fes}^{k+1} \leftarrow \xi_{fes}^k \cup \{\nu^r\}$
        end
    end
    $k = k + 1.$
until
    $(UB^k - LB^k)/LB^k \leq \epsilon$
Set the optimal configuration: $\rho^* = \rho^k; x^* = x^k$
Obtain optimal routing $t^*$ from $\mathbb{P}1_{DSP}$
```

5.6.2 Slice Creation Costs

Besides the objective of $\mathbb{P}1$ to maximize UE's satisfaction, another goal of MNOs is to minimize the system cost incurred by each slice ($\mathbb{P}2$). Hence, we extend the previous optimization framework to include the constraints and objective functions related to system costs (monetary costs) for slice creation minimization.

The total cost for creating each slice incurred by a O-RAN is the cost of O-CU and O-RU for VMs running on the server to execute corresponding VNFs, MEC cost (in O-RU and O-CU).

$$C_{tot}^{r,s} = C_r^s + C_0^s, \forall s \in \mathcal{S}, \forall r \in \mathcal{R}. \tag{5.6.25}$$

In particular, the VMs and MEC servers can be installed either in O-RU or O-CU. The processing of VNFs (*i.e.,* UPF) and MEC servers deployed at O-CU is more cost-efficient, as opposed to O-RU due to the larger capacity of the MEC servers and computation capacity of O-CU compared to the smaller O-RU. Similarly, the deployment cost of the MEC server, which depends on the rental site cost, is cheaper at O-CU than the rental site cost of the MEC server at O-RU. We consider the above factors as the total cost to decide on the economic benefit of each candidate location [117, 118].

For O-RU r is given by:

$$C_r^s = \varphi_r^s + \sum_{u \in \mathcal{U}} \alpha_r^s \cdot x_u^{r,s} \cdot f^{r,s} + \sum_{u \in \mathcal{U}} \beta_r^s \cdot x_u^{r,s} \cdot \lambda_u^s \cdot (c^s \cdot f^{r,s}), \tag{5.6.26}$$

where the first term is for MEC deployment cost, the second for VM instantiating, and the last term is for computing cost per slice. We use φ_r^s as a MEC deployment cost, α_r^s as a VM installation cost, and β_r^s (monetary units per cycle) as the average cost for serving each request at O-RU r.

As for O-CU:

$$C_0^s = \varphi_0^s + \sum_{r \in \mathcal{R}} \sum_{u \in \mathcal{U}} \alpha_0^s \cdot x_u^{r,s} \cdot (1 - f^{r,s}) + \sum_{r \in \mathcal{R}} \sum_{u \in \mathcal{U}} \beta_0^s \cdot x_u^{r,s} \cdot \lambda_u^s \cdot (c^s \cdot (1 - f^{r,s})) \tag{5.6.27}$$

5.6.3 Slice Creation Costs and UE's Satisfaction: Trade-Offs

MNOs aim to solve the previous problems at the same time. We extend the previous optimization frameworks to simultaneously maximize the satisfaction of UEs (translated into throughout) together with system costs (for slice creation) minimization. Hence, considering the maximization of the network throughput for UE's satisfaction and minimization of slice costs, it can be written as a multi-objective optimization framework $\mathbb{P}2$ as:

$$\mathbb{P}2\text{: } \min_{x,t,\rho} \; C_{tot}^{r,s}(x,t,\rho) - O(x,t,\rho)$$

s.t:

(5.4.1)–(5.6.17), (5.6.19)–(5.6.20)

The minimization objective forces $\mathbb{P}2$ to have its minimum value. It is evident that MNOs have different objectives for their networks, and the appropriate assignment of weights can adapt to this; one can prioritize the individual (or a subset of) objectives for MNOs over the other objectives. For example, tuning a high weight to the first term (*i.e.,* $C_{tot}^{r,s}(x,t,\rho)$) will prioritize minimizing slice costs while setting a high weight to the second term (*i.e.,* $O(x,t,\rho)$) implies the focus on maximization of the network throughput.

Theorem 3. *The optimization problem $\mathbb{P}2$ has similar complexities of $\mathbb{P}1$ ($\mathcal{O}(2^N)$) and is an NP-complete.*

Proof. The optimization model $\mathbb{P}2$ that minimizes the system costs is a MIP problem due to existing the same variables and linear constraints as of $\mathbb{P}1$. Thus, by using Lemma 1, Theorems 1 and 2, and Proposition 1 and Proposition 2.19 in [116], we can easily obtain Theorem 3, and obtain the respective optimal solution $\mathbb{P}2$ in polynomial time. To this aim, only minor changes are needed by adding extra bounded conditions for slice costs that count the slice minimization costs' objective. As a result, the algorithm's steps will remain unchanged, except for the definition of $\mathbb{P}1_{MP}$ as Benders' decomposition impacts only the definition of the MP. Accordingly, a solution $(\bar{x}, \bar{\Theta})$ to problem P2 obtained by the algorithm is a ϵ-optimal solution if $(\bar{x}, \bar{\Theta})$ satisfies all the constraints of problem P2 and $0 \leq \left(O(x^*, \bar{t}, \bar{\rho}) + \Theta^*\right) - \left(O(\bar{x}, \bar{t}, \bar{\rho}) + \bar{\Theta}\right) \leq \epsilon$. This also proves the capability of the proposed framework in terms of scalability, flexibility and being generic which is stated in Proposition 1.

5.7 Performance Evaluation

In this section, we present the effectiveness of the proposed solution from the overall system and slice viewpoint by investigation of numerical results for the performance of the proposed algorithm.

5.7.1 Simulation Scenario

Our analysis considers three main types of eMBB, mMTC, and uRLLC services with different QoS requirements. Table 5.3 summarizes our simulation setup, where we assume a bandwidth of 20 MHz for each gNB (i.e, $\rho^r = 100$ PRBs) with mesh type FNs and a link capacity $\omega_{i,j} = \infty$ in the transport network.

We study a network scenario with area of 4000 m^2 composed of a single O-CU connected to a set of O-RUs $R = 10$ and adopt the values of [100–102] to define three types of applications, video streaming, IoT, and medical applications. We consider $s = 0$ for video streaming applications (eMBB) with $\lambda_u^0 = 20$ Mb/s and $d_u^0 = 100$ ms, $s = 1$ for IoT messages (mMTC) with $\lambda_u^1 = 1$ Mb/s, $d_u^1 = 100$ ms and finally $s = 2$ for medical applications (uRLLC) which use split 1 with $\lambda_u^2 = 5$ Mb/s, and delay of $d_u^2 = 10$ ms. As for computational capacity, we utilize the values used in [54], with $\kappa_0 = 32$, $\kappa_r = 4$ CPU reference core per Mb/s. Regarding the computational cost, $c^1 = 2$, $c^2 = 1$ and $c^3 = 4$ CPU reference core per Mb/s. We use split 7-2 for $s = 0$ and 1 which have a tolerable delay and higher computational requirements. The number of UEs are variable from 500 to 4000 UEs where 30%, 40%, and 30% are set to eMBB, mMTC, and uRLLC, respectively. We consider a distance-dependent path-loss model with a transmission power 30 dBm and for MCS calculation, we adopt the values used in [94].

We have conducted extensive Monte-Carlo simulations implemented in Java, while the optimization model is built and solved with IBM ILOG CPLEX Optimization Studio [97]. Note that the computing time needed to obtain the optimal solution using Benders distributed solution with a CPU processor of Core i7-8550U, a RAM of 16 GB for a scenario composed of a single CU, $Q = 4$, $R = 10$ is the matter of seconds. All results are compared with the scenario where no slicing is considered. Hence, a single FS (split 7-2) is allowed to use in each O-RU, having a MEC server computation capacity in O-CU. Indeed, in order to host a MEC server in O-RU, it has to implement all full-stack of gNB functions [9]; and this limits its eligible FSs. In the following, the proposed solution is labeled as *SO-RAN*, and state of the art with *SoA*.

Table 5.3: Simulation Setup

System Bandwidth	20 MHz
Number of PRBs (ρ^r)	100
Number of O-RUs (R)	10
Number of FNs (Q)	4
Number of UEs (U)	500 - 4000
The capacity of links ($\omega_{i,j}$)	infinite Gb/s
Transmission rate of uRLLC (Medical apps)	5 Mb/s
Transmission rate of mMTC (IoT msg)	1 Mb/s
Transmission rate of eMBB (Video Streaming)	20 Mb/s
Delay of uRLLC (Medical apps d_u^2)	10 ms
Delay of mMTC (IoT msg d_u^1)	100 ms
Delay of eMBB (Video Streaming d_u^0)	100 ms
Transmitted power	30 dBm
CPU consumption to compile c^1	2 RCs per Mb/s
CPU consumption to compile c^2	1 RCs per Mb/s
CPU consumption to compile c^3	4 RCs per Mb/s
Computation capacity of each O-RU (κ_r)	4 RCs per Gb/s
Computation capacity of O-CU (κ_0)	32 RCs per Gb/s
VM installation cost at O-RU (α_r^s)	1
VM installation cost at O-CU (α_0^s)	$0.5\alpha_r^s$
Rental cite (deployment) cost at O-RU (φ_r^s)	1
Rental cite (deployment) cost at O-CU (φ_0^s)	$0.5\varphi_r^s$
Computation cost at O-RU (β_r^s)	1
Computation cost at O-CU (β_0^s)	$0.017\beta_r^s$

We present the results of our numerical analysis in Fig. 5.3, where we want to explore the insights of the spectrum and functional computation (UPF and MEC) limitations, thus we assume that $\omega_{i,j} = \infty$, $\forall s \in \mathcal{S}$ for $R = 10$.

As can be seen, the increase in the number of UEs is consistent with the increase in the average of served traffic in *SO-RAN* and *SoA* due to the addition of more UEs with extra diversity close to O-RUs, which allows to serve those services which have less spectrum requirements, hence, more traffic could be served. The *SO-RAN* has a higher performance compared to SoA due to using slicing (having MEC capability in both O-CU and O-RUs) and variable numerology based on the profile of services, which allows to better use the resources and serve more traffic. For example, when number of UEs = 2000 (15.8 Gb/s offer load), *SO-RAN* achieves 11.6 Gb/s throughput while *SoA* can reach up to 9.84 Gb/s in the throughput, respectively. Indeed increasing SCS based on the profile of services makes them to better use the spectrum and meet their requirements.

Fig. 5.4 represents the total served number of UEs w.r.t offered number of UEs.

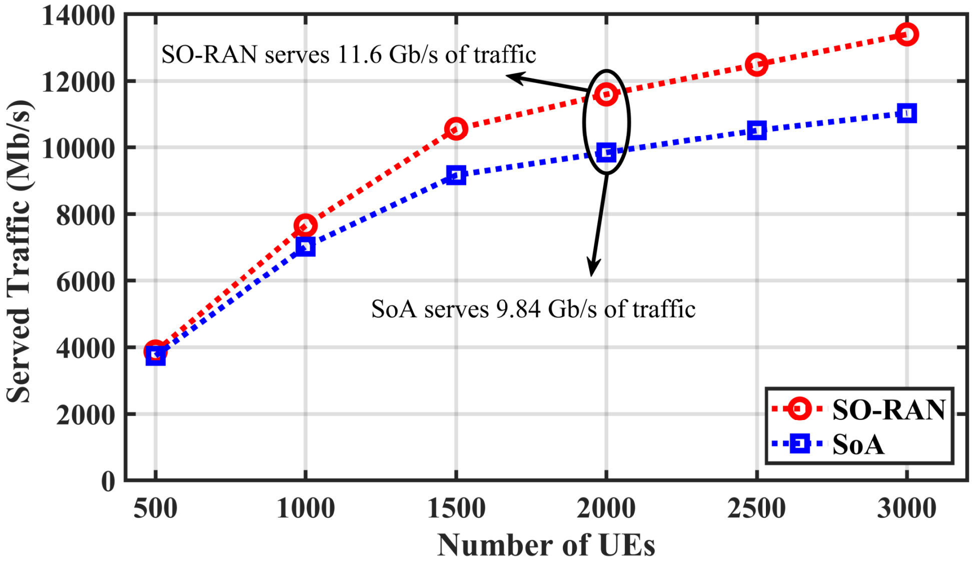


Figure 5.3: Average served traffic w.r.t number of UEs for $R = 10$

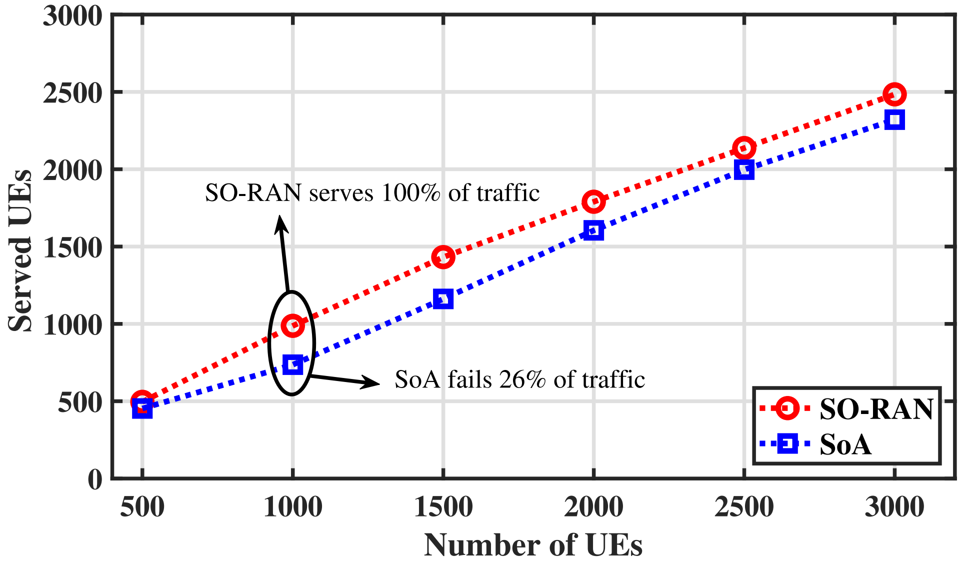


Figure 5.4: Served UEs w.r.t number of UEs for $R = 10$

By analysis of this figure, it can be identified that with a small number of UEs, *SO-RAN* can serve all the traffic load while SoA fails to fully serve them. For example, when the number of UEs = 1000, *SO-RAN* serves all of them while *SoA* misses serving 264 UEs (26%) respectively. However, by increasing the number of UEs none of them could serve all traffic due to limited radio resources (*i.e.,* spectrum). However, in all results, *SO-RAN* outperforms *SoA* due to using slicing design.

Fig. 5.5 shows the performance of *SO-RAN* and represents the total served traffic per slice w.r.t offered a number of UEs.

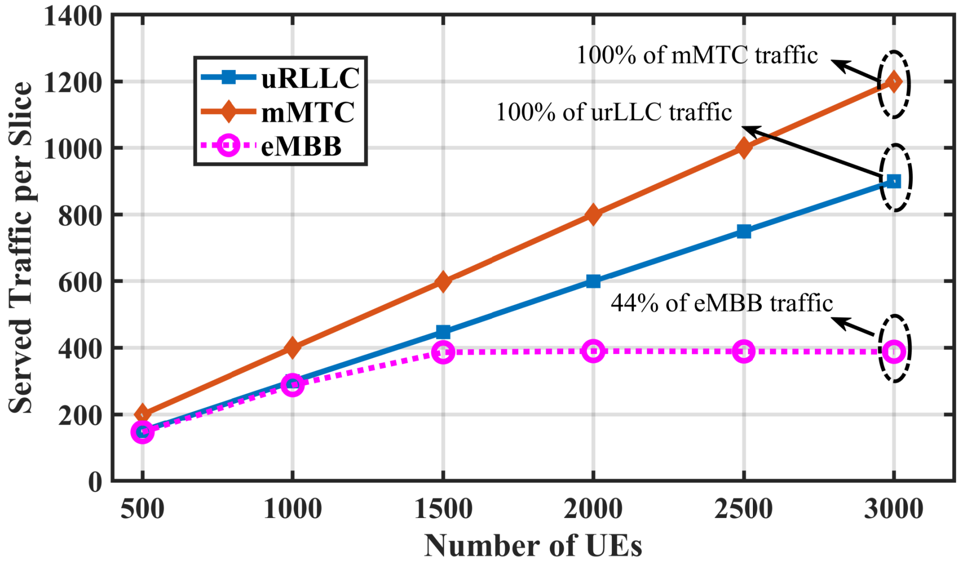


Figure 5.5: Average served traffic per slice w.r.t number of UEs for $R = 10$

As you see in this figure, the mMTC slice is always fully served due to lower datarate requirements and following that fewer PRB requirements. However, eMBB slice after an increase from UEs = 500 to UEs = 1500, has an almost fixed behavior as we increase the number of UEs. This is mainly due to higher datarate, computation, and spectrum requirements of eMBB slice which reaches the maximum allowed resources of the network for this slice. However, uRLLC has incremental behavior with respect to the number of UEs. Once we increase the number of UEs, more uRLLC services are served as its spectrum and datarate requirements are not as high as eMBB slice, hence, serving them is not exceeding the available network resources.

Fig. 5.6 depicts the total served traffic in terms of throughput per slice w.r.t offered a number of UEs and it behaves similarly to Fig. 5.5. According to Fig. 5.6, both uRLLC and mMTC slices are fully served as their requirements are not exceeding the available network resources. On the other hand, the eMBB slice which has the higher datarate and spectrum requirements obtain the most resources, nevertheless, once the traffic load surpasses the maximum available resources, (*i.e.,* UEs = 1500) the rest of eMBB traffic is rejected due to the lack of available resources.

The results of Fig. 5.6 showed the served traffic per slice when no constraints were imposed on the transport network. However, the design of the transport network has an impact on the served traffic. Accordingly, we explore the impact of limiting the transport network capacity for the achieved traffic per slice. Fig. 5.7 illustrates this impact with $\omega_{i,j}$ = 2 to 20 Gb/s with a load of 3000 UEs. All the

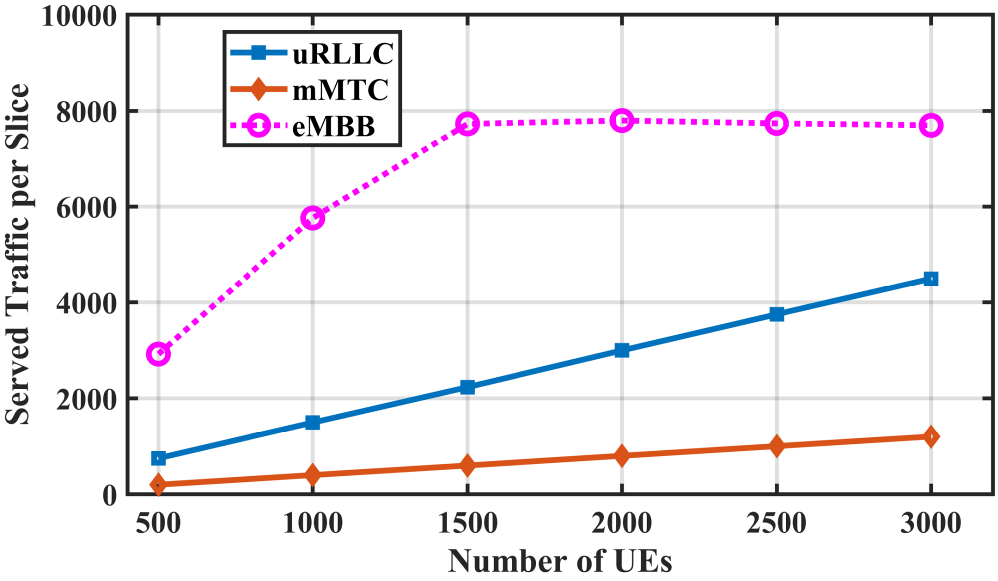


Figure 5.6: Average served traffic per slice (Mb/s) w.r.t number of UEs for $R = 10$

results are compared with the last x-tick in Fig. 5.6. As you see in this figure, the remarkable point is that enforcing constraints in the transport networks leads to the loss in the throughput gained in Fig. 6 with the number of UEs = 3000. The results show that as we increase the capacity of the transport links, more throughput is achieved. For example, around 84% of the uRLLC traffic is lost with $\omega_{i,j} =$ 6 Gb/s compared to the traffic served in Fig. 5.6 with the number of UEs = 3000. Remarkably, this reduces as we increase the capacities of transport networks. For instance, this loss is under 1% when $\omega_{i,j} = 14$ Gb/s. Similar behavior is achieved for eMBB and mMTC slices, where the served traffic for eMBB is reduced to 42% with $\omega_{i,j} = 6$ Gb/s and 23% with $\omega_{i,j} = 14$ Gb/s. Similarly, the impact for mMTC is 97% to 0.5% with $\omega_{i,j} = 6$ Gb/s and $\omega_{i,j} = 14$ Gb/s respectively.

5.7.2 Spectral Efficiency Analysis

In this section, we analyze the spectrum efficiency of O-RUs and evaluate the impact of the number of O-RUs to find a proper set-up of flexible FS, with the minimum capacity requirements per O-RU. As previously explored in Figs. 5.3 and 5.4 in a network with 10 O-RUs, we could not meet the required demands, especially when we had demands of more than 1500 UEs (11.85 Gb/s traffic load). We present the results of our numerical analysis in Fig. 5.8, where we assume that $\omega_{i,j} = \infty$, and MEC computation capacities are initialized with $\kappa_0 = 32$, $\kappa_r = 4$ CPU reference core per Gb/s for a set of O-RUs from $R = 10$ to 20.

By seeing this figure, we can inspect that even though the achieved throughput for the left side figure (with an offered load of 8 Gb/s) is very close for both *SO-RAN* and *SoA*, however, it is evident that the increase of the served traffic is consistent with the increase of the number of O-RUs in *SO-RAN* while it is almost fixed for *SoA* as we increase the number of O-RUs. This means network densification (*i.e.*, an increasing number of O-RUs) allows *SO-RAN* to serve more traffic due to its virtualized nature and the capability of using multiple FSs, which allows to better use the spectrum resources and meet the diversified requirements of UEs. By seeing the middle figure (with 16 Gb/s

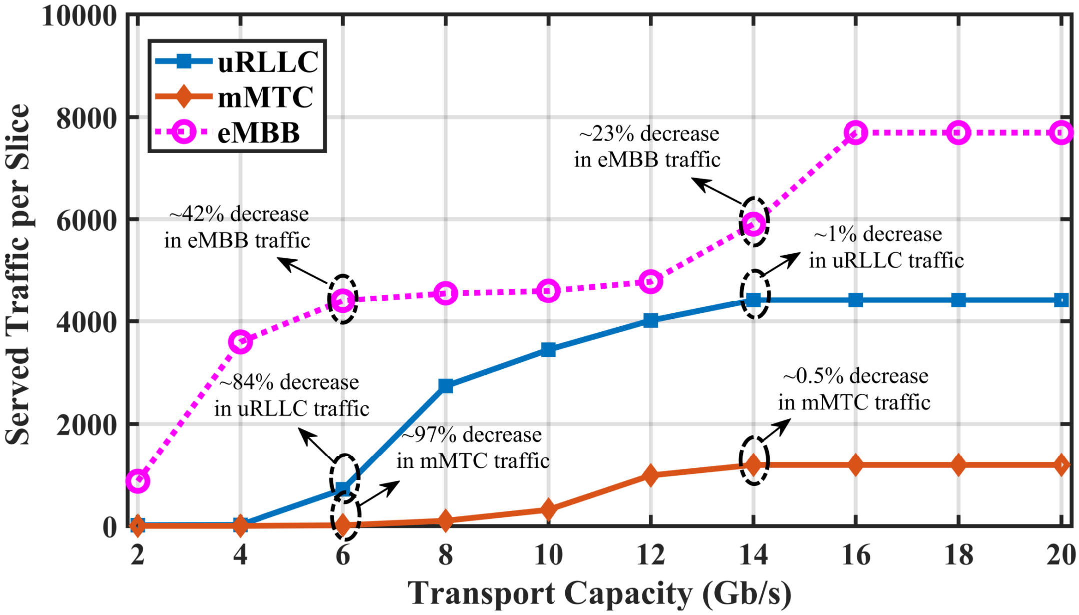


Figure 5.7: Average served traffic per slice (Mb/s) w.r.t transport capacity (Gb/s) for number of UEs = 3000.

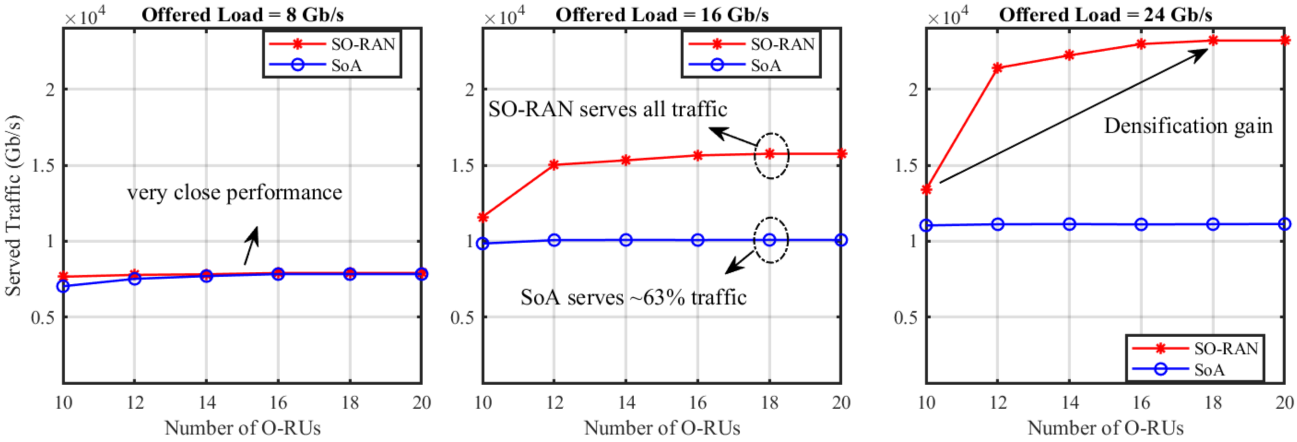


Figure 5.8: Served traffic w.r.t number of O-RUs

offered load) *SO-RAN* is able to serve all traffic with $R = 18$ while *SoA* with the same number Of O-RUs can serve traffic up to 10 Gb/s.

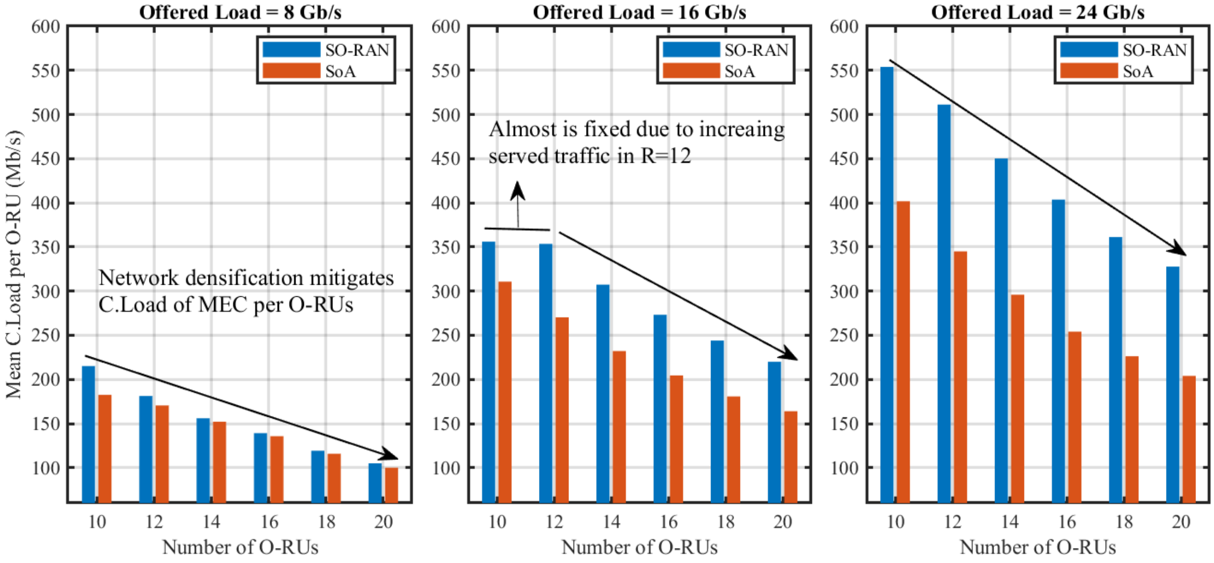


Figure 5.9: C.Load w.r.t number of O-RUs

We now evaluate the impact of network densification (equivalently increasing the number of O-RUs) on the computation load incurred in O-RUs. Fig. 5.9 represents the mean computation load per O-RU for different network scenarios with an offered load of 8, 16, and 24 Gb/s, respectively. As observed from this figure, the network densification diminishes the mean computation load of MEC to be distributed among newly installed O-RUs. This behaviour is more evident in the scenario with offer load = 24 Gb/s due to having a higher demand for services to be connected to the network. For example, when comparing the reduction in the computation load of the first scenario with offer load = 8 Gb/s to the last one with offer load = 24 Gb/s, we observe that this reduction from $R = 10$ to 20 is about 52% for the first scenario while it is around 40% for the last scenario.

5.7.3 MEC Capacity Analysis

The results of subsection 5.7.2 show the gain achieved by slicing in O-RUs, when MEC computation costs were imposed to the initial capacities $\kappa_0 = 32$, $\kappa_r = 4$. However, they do not offer any insight into finding proper MEC settings as MEC prices vary significantly depending on their computation capacities, storage, etc. This part evaluates three forms of main edge computing platforms that cover the whole spectrum spanning the UEs to the O-CU. The first MEC type is based on a single board computer built on either ARM64 and AMD64 architecture such as *NVIDIA Jetson Nano* module or an X86 board with *Intel Movidius Myriad X Vision Processing Unit (VPU)*, that are ideal for protocol translation, data aggregation, and Artificial Intelligence (AI) inference, this type known as *Tiny Edge* model. The second type is a medium edge deployment model representing a cluster of moderate-cost machines running at the edge computing layer. The compute cluster is powered by an internal Graphics Processing Unit (GPU), Field Programmable Gate Arrays (FPGA), VPU, or an Application Specific Integrated Circuit (ASIC). In this model, a cluster manager like Kubernetes is used for the orchestration of the workloads and resources in the clusters. This type is called as *Light Edge* model. The last type of MEC includes expensive machines with high computational resources that could run heavy with high datarate services in the edge layer. This MEC type is called as *Heavy Edge* model [119]. We translate the aforementioned MEC models to the computation capacities per O-RUs and set $\kappa_r = 4$, 8, and 16 for *Tiny Edge*, *Light Edge*, and *Heavy Edge* models, respectively. We then explore the aforementioned MEC platforms in terms of computation capacities in Fig. 5.10 to obtain practical insights on proper MEC settings to assess their impact on supporting different traffic demands while keeping the network costs as low as possible. According to the results of Fig. 5.8, it can be inspected that network densification mitigates the loss of traffic demand for *SO-RAN*; nevertheless, this loss still remains without any changes for *SoA*. We thus explore the network performance as a function of different MEC platforms to assess their impact on provisioning hungry-computational services by choosing a proper MEC setting with affordable prices. The Fig. 5.10 depicts this impact on increasing the served traffic for both *SO-RAN* and *SoA*. For example, for the traffic load of 24 Gb/s when we double the MEC computation capacities (*i.e.,* using *Light Edge*), the served traffic increases from 13.4 Gb/s to 20 Gb/s (49% increase) for *SO-RAN* while this increase is 11 to 16 Gb/s (45% increase) for *SoA*. However, increasing MEC capacity, *i.e.,* changing MEC type from Light to Heavy is not cost-efficient since the current MEC type (*Light Edge*) is able to serve all traffic demand. The main outcome of this figure is that the MNOs can choose a proper MEC type for the on-demand traffic to optimize their costs.

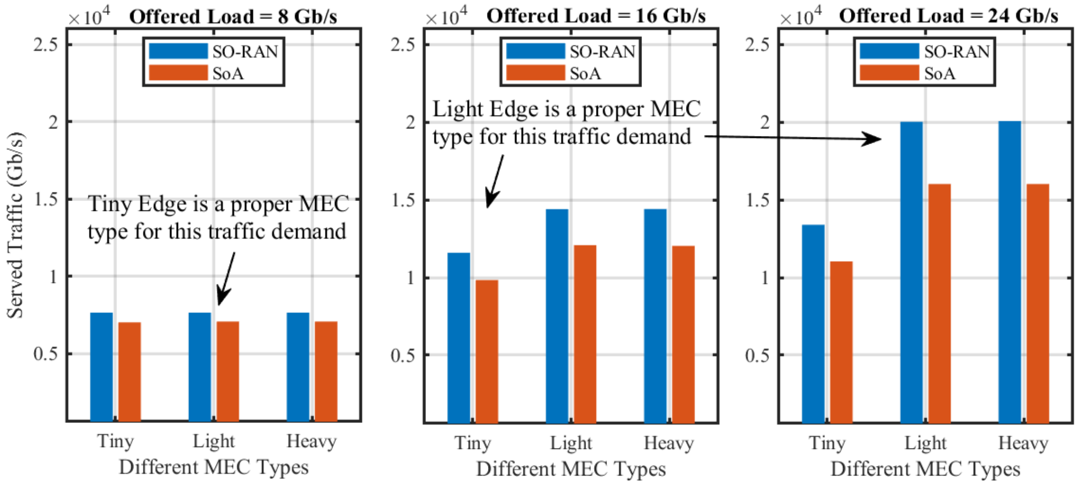


Figure 5.10: Served traffic w.r.t MEC computation capacity

We now analyze the performance of slices in terms of QoS satisfaction as a function of different MEC platforms to assess their impact on supporting different slices by choosing a proper MEC setting. The Fig. 5.11 represents the performance of each slice per chosen MEC model. According to this figure, for slices with small traffic load *Tiny Edge* as a cost-efficient model can meet the requirement of slices. However, as the offered load increases the need to upgrade the computation capacity of MEC server increases as well. For example, when the offered load = 16 Gb/s, the *Tiny Edge* model losses around 27% traffic of eMBB slice. Similarly, this loss reaches up to 47% with offered load = 16 Gb/s when using *Tiny Edge*. This means customers will be unsatisfied if this amount of losses happens. Hence, MNOs may need to adjust the MEC type to reach QoS satisfaction of UEs (here eMBB slice). As it is clear in this figure, adjusting MEC setting to *Light Edge* mitigates previous losses and serves all slices with their QoS requirements.

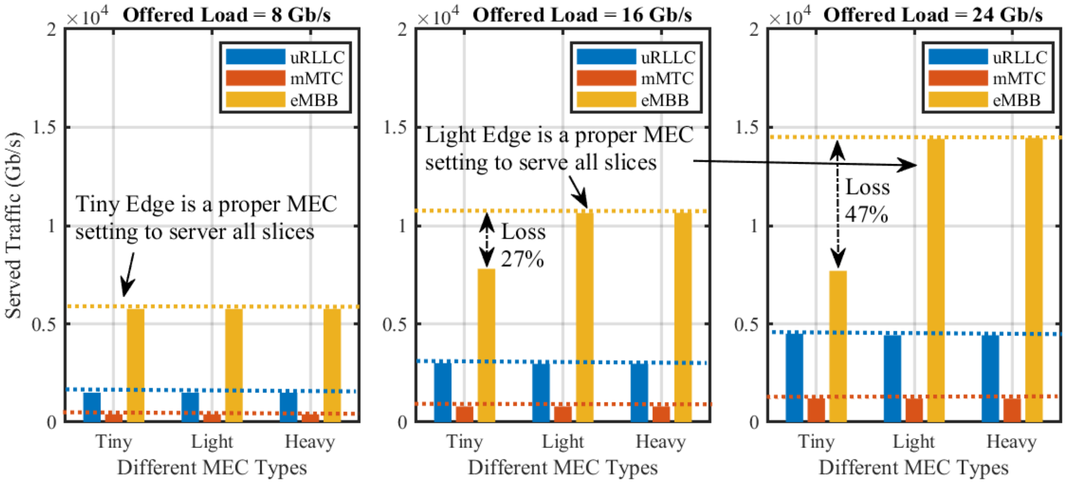


Figure 5.11: Served traffic per slice w.r.t different MEC settings

5.7.4 Multi-Objective (Joint Throughput and Cost) Analysis

In the previous subsections, we analyzed network performance in terms of served traffic that showed the gain achieved by slicing in O-RUs. However, as we already mentioned creating slices to serve traffic has a cost, and there is a trade-off on the achieved throughput and the cost incurred. To this aim, in this part, we explore the joint objective of maximizing the achieved throughput and minimizing the cost to create slices. Fig. 5.12 and Fig. 5.13 represent this trade-off and show how this affects the throughput and cost metrics. For this experiment, we compare i) the single objective (*i.e.,* Single-Obj) with ii) multiple objectives with the same weight/priority ((*i.e.,* Multi-pr=1,1) for both objectives and iii) multiple objectives with the high weight/priority of throughput (*i.e.,* Multi-pr=1,2) where the value 1 is used as a low priority for the cost and the value 2 is used as a high priority of throughput. According to Fig. 5.12, using a single objective of maximizing throughput for $R = 10$ has 35% increase of throughput when compared to the performance of joint objectives (Multi-pr=1,1) and 12% increase of throughput when compared to the performance of Multi-pr=1,2. Indeed using a single objective of maximizing throughput is more beneficial when the network is densified ($R = 20$) due to the extra added cost of O-RUs and, accordingly extra cost of creating slices. Hence, the network tries to keep the same achieved throughput to prevent forcing additional costs to the system.

A similar conclusion was reached by Fig. 5.13 where our metric of interest is computation cost. For example, when $R = 20$, the computation cost of a single objective has decreased about 78% when compared to multi objectives Multi-pr=1,1 and 17% for Multi-pr=1,2. Comparing Fig. 5.13 and Fig. 5.12 shows that 160% throughput could be achieved with the addition of 78% in the computation cost if we have the same priority for both objectives. Furthermore, prioritizing throughput (*i.e.,* Multi-pr=1,2) reaches up to 82% throughput with the addition of 17% in the computation cost.

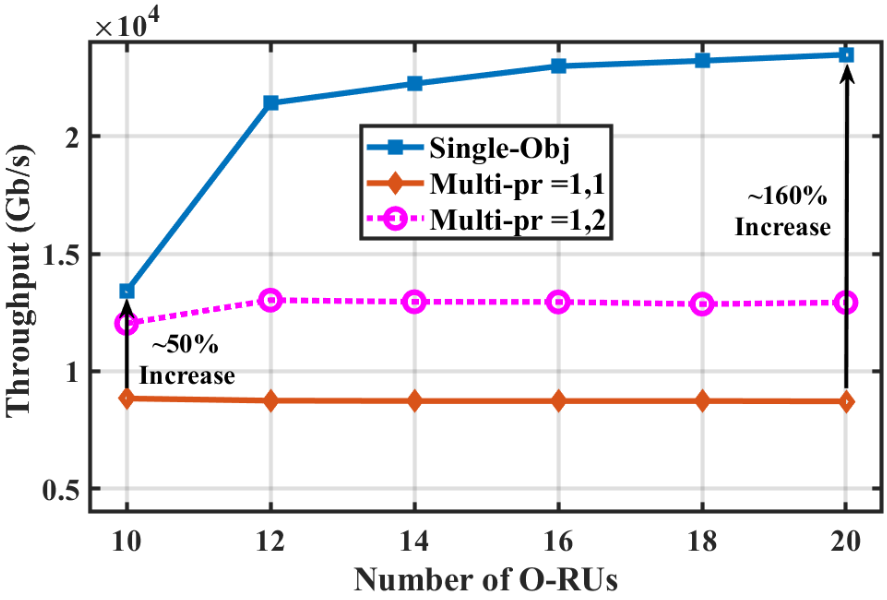


Figure 5.12: Throughput w.r.t number of O-RUs

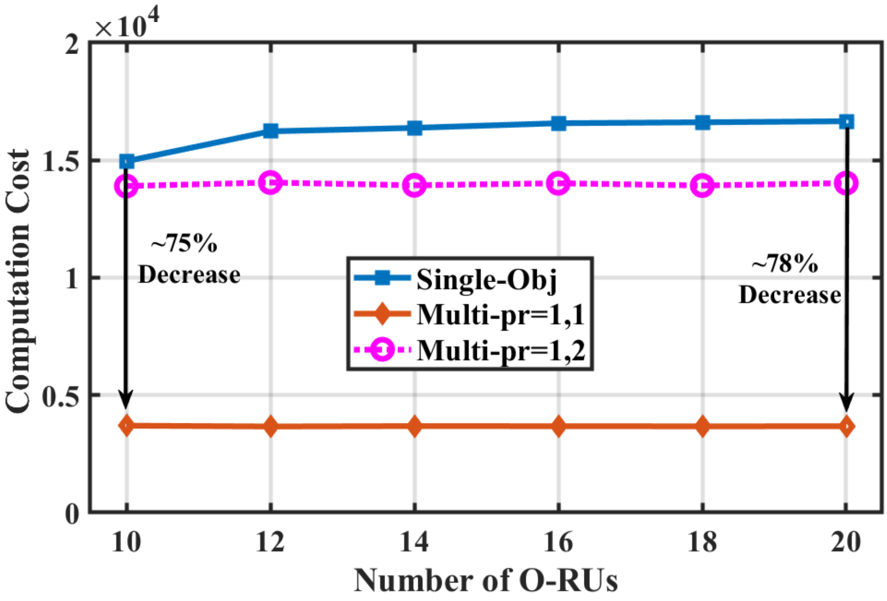


Figure 5.13: Computation cost w.r.t number of O-RUs

5.8 Conclusion

In this work, we proposed a comprehensive optimization approach for a dynamic NG RAN/MEC slicing framework to dynamically place RAN protocol stack of VNFs and MEC server per slice. This framework jointly solves the problem of operating cost-efficient edge networks and maintaining the served traffic with diverse QoS requirements by considering the bottlenecks in the capacity of O-RUs, MEC server computation capacity along with a customized FS per slice. We used an effective Benders decomposition algorithm, which reduces the complexity and meantime guarantees an exact and optimal global solution. Trace-driven simulation results have been provided to demonstrate that the proposed algorithm can effectively optimize the joint throughput and the system cost in different traffic scenarios while satisfying QoS requirements. We analyzed the spectrum efficiency and evaluated the impact of network densification on the computation load incurred in O-RUs, and observed that the network densification diminishes the mean computation load of MEC to be distributed among newly installed O-RUs. We then measure and quantify the computation capacities, and storage for different MEC services to find proper MEC settings for on-demand traffic, and adjust MEC type to meet QoS satisfaction of different UEs belonging to specific slice types. The broad implication of the present results showed a trade-off between the achieved throughput and the cost incurred to the network. Hence, we explored multi-objective optimization of maximizing throughput and minimizing cost to create slices and comparing its performance with a single objective (maximizing throughput). The results showed that up to 160% increase in throughput could be achieved with the addition of 78% in the computation cost for a single objective when compared with multi-objective without prioritization. Moreover, this increase in throughput is around 82% with the addition of 17% in the computation cost when a single objective is compared with a multi-objective with priority in throughput. Therefore, maximizing throughput alone (single objective) can result in high throughput at the expense of high cost. The amount of throughput can be reduced almost half with multi-objective with throughput prioritization, whereas costs can be reduced five-fold.

Chapter 6

Conclusions and Future Perspectives

This chapter concludes the thesis by summarizing the main contributions and presenting possible future research topics. Section 6.1, in particular, outlines the major findings from each contribution section, and Section 6.2 discusses the relevant significant open issues and potential research opportunities.

6.1 Conclusions

5G and Beyond networks (B5G) with a broad variety of capabilities have been designed to handle the current and anticipated increase in mobile data traffic demand generated by various verticals with diverse requirements. In order to facilitate this vision, B5G will be based on flexibility and reconfigurability to enable the adaptation of the network to the traffic demand. Thus, the B5G architecture shall be designed with a certain level of flexibility that comes along with the principles of softwarization. Indeed, flexibility and reconfigurability are achieved by leveraging the network architecture with the promising enablers for network softwarization, including virtualization techniques (SDN/NFV), containerization, edge computing, and network slicing to dynamically set up on the fly to fit the service requirements.

This thesis demonstrates the benefits of RAN softwarization (using network slicing and edge computing) in B5G that enable Mobile Network Operators (MNOs) to adopt cutting-edge technologies and run their network more flexibly and efficiently.

We focus on the 3GPP RAN architectural aspects, applications, and issues such as finding the joint RAN slicing, optimal functional split, and MEC placements between distributed and centralized units of RAN. We also consider the huge and diverse demands of performance requirements in B5G, which entails a flexible network architecture to support the high data rates with different service level agreements (SLA).

To this end, the thesis investigates joint analysis of the functional split along with RAN slicing in the first part (**Chapter 3**). Furthermore, the service-aware 5G RAN slicing framework is explored in the second part (**Chapter 4**), and finally, dynamic RAN slicing and MEC placement in the last part (**Chapter 5**) of our contribution throughout this thesis.

In the first part (**Chapter 3**), our focus is on the C-RAN architecture, in which we analyze joint slicing and functional split optimization in the 5G RAN approach, which has not yet been studied in the literature. We first formulated this problem as a mixed integer non-linear programming framework (MINLP) which is a NP-Complete and has a high complexity to solve it. Thus, we relax the constraints and convert them into mixed integer programming (MIP) problem, which is linear and convex and reduces the solution space and complexity of the program. The contributions of the first part can be summarized into the following points:

- In this part of the thesis, we aimed to design a joint routing (from user to CU) and functional split optimization while considering different slices. The RAN slicing allowed a customized functional split deployment per slice, thus optimizing the available resources, *e.g.,* transport network capacity and RRH or CU computational capacity.
- We formulated this problem as a Mixed Integer Programming (MIP) to jointly optimize the centralization degree and throughput.
- The slice creation is extended up to the user. In general, the slicing only considers the RAN, [54, 58]. Instead, this paper aims to exploit the high density of RRHs by jointly analyzing routing in the RAN and user association.
- Our results indicated that there is a trade-off between the centralization degree and the throughput. According to the results, the computational cost in the proposed method, *i.e.,* $Multi - Splits$, is higher than the one only single functional split used. This means that the centralization degree is lower. However, this is compensated by the increase in the throughput. Furthermore, in $Multi - Splits$, the throughput increase is much higher and needs less increase in the computation cost.

The second part of the thesis presented in **Chapter 4** studied service-aware network slicing framework for 5G RAN to create isolated RAN slices based on the service requirements with customized FSs per slice on top of a network composed of a CU, a FH/BH network, and a set of RRHs. This chapter formulated the optimization framework to model the aforementioned concept where the objective function was maximizing the throughput by jointly selecting the optimal routing paths from a connected UE to CU, and FS while satisfying the QoS requirements. Furthermore, an effective heuristic method, SlicedRAN, is proposed to solve the computational complexity of the optimization framework, where SlicedRAN achieved near-optimal solutions in a short computing time compared to the optimal one. The proposed framework investigated the bottlenecks in the capacity of RRHs, FH/BH network capacity along with a minimum level of SLA for each slice imposed by the different service types. The broad implication of the present research demonstrated a strong trade-off between SLA and the FH/BH network between CU and RRHs which provide a basis for designing a virtualized network infrastructure with a cost-efficient FH/BH network whilst guaranteeing SLA of different slices.

Finally, the last research line explored in **Chapter 5** investigated dynamic RAN/MEC slicing framework in O-RAN architecture to dynamically place the RAN protocol stack of VNFs and MEC server per slice. This framework contained the bottlenecks in the capacity of O-RUs, MEC server computation capacity, together with a customized FS per slice, to jointly solve the challenge of operating cost-efficient edge networks and maintaining the served traffic with various QoS criteria. We employed a robust Benders decomposition algorithm, which lowered the complexity while ensuring an exact and optimal global solution. The proposed algorithm successfully optimized the joint throughput and system cost in various traffic scenarios while satisfying QoS criteria, as shown by trace-driven simulation results.

Then, in order to determine the right MEC settings for on-demand traffic and alter the MEC type to satisfy the QoS requirements of various UEs belonging to different slice types, we assessed the compute and storage capacity for various MEC services. The overall conclusion of the present findings demonstrated a trade-off between the throughput attained and the cost incurred to the network.

As a result, we investigated multi-objective optimization to construct slices while optimizing throughput and decreasing cost objectives, and we compared its performance to that of a single objective (maximizing throughput). The findings demonstrated that, when the priorities for the two objectives are equal, a throughput increase of up to 160% can be made possible by increasing the computation cost by 78%. Additionally, when throughput is prioritized, the increase in throughput is about 82%, while the computation cost increases by 17%.

6.2 Furture Perspective

The outcomes of this thesis can be used as the basis for a variety of open issues and innovative research lines in 5G and B5G networks that neither this thesis nor the state-of-the-art has addressed. To this aim, in this section, we present a number of novel directions for further research relevant to the contributions of this thesis.

We brought a new perspective to RAN functional split by introducing joint functional split and RAN slicing optimization and highlighting the importance of creating slices with a customized functional split. In addition, we showed a trade-off between the computation cost and the throughput. We next presented SlicedRAN: service-aware network slicing framework for 5G RAN, where we extended previous work by investigating the bottlenecks in the capacity of RRHs (network densification), FH/BH network capacity along with a minimum level of SLA for each slice imposed by the different service types. Finally, the last part of this thesis proposed a comprehensive optimization approach for a dynamic NG RAN/MEC slicing framework to dynamically place the RAN protocol stack of VNFs and MEC server per slice. By taking into account bottlenecks in the capacity of O-RUs, MEC server computation capacity, and a customized FS per slice, this framework jointly solves the problem of operating cost-efficient edge networks and maintaining the served traffic with diverse QoS requirements. Based on our findings during this thesis, the first potential research challenge could be **end-to-end functional split selection and slicing**. The described functional split selection is based on the RAN characteristics (architecture, resources, *etc.*). However, the creation and management of end-to-end slices across the RAN and the core network is more complex and requires further research. The end-to-end slicing will have to consider not only the functional split but also the rest of the RAN VNFs, the core VNFs, and, if the far end of the service is another UE, the far end functional split. MEC will also play a key role in slice creation. The pool of virtualized resources offered by MEC will provide each slice with differentiated capabilities, ranging from edge computing capability for URLLC to caching resources for eMBB.

Another future research issue is designing **accurate models**. The optimal selection of the functional split, the placement of VNFs, and, in general, the creation of slices rely on the accurate estimate of resources required by each slice. In the literature, there are initial analytical and experimentation-based models to estimate the computational and latency requirements of the different VNFs. However, it has been shown that results depend on a wide range of factors, such as the particular platform on which VNFs are instantiated. The reliability of the models has a huge impact on the slice provision and the efficiency of the resource usage since the over- or under-estimation of VNFs' resource requirements results in low resource usage efficiency and lack of slices' isolation, respectively. Intent-Based Network Slicing (IBNS) that slice and manage network resources efficiently using AI-driven methods is expected to fill the existing gap.

Finally, the last potential research line which is getting momentum attention is **AI-based Digital Twins** technology which creates a virtual representation of the physical objects of network components, where the concept of network slicing is a key enabler to creating isolated slices of physical network resources and make intelligent decisions using AI-based methods. This technology can be used in different emerging B5G applications domains, such as autonomous driving and connected vehicles, providing virtual environment tests with intelligent solutions to control, manage and predict harsh traffic scenarios such as sports events, traffic jams, accidents, terrorist attacks, etc.

References

[1] CISCO. Cisco Visual Networking Index: Forecast and Methodology, 2016-2021. Technical report, 2017. URL https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf.

[2] Ericsson. Ericsson Mobility Report. Mobile data traffic outlook, 2020.

[3] NGMN Alliance. NGMN 5G White Paper. Technical report, 2015.

[4] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang. What will 5g be? *IEEE Journal on Selected Areas in Communications*, 32(6):1065–1082, June 2014. ISSN 0733-8716. doi: 10.1109/JSAC.2014.2328098.

[5] ITU. ITU-R M.2083-0. IMT Vision—Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond, 2015.

[6] ITU. ITU-R M.2410-0. Minimum requirements related to technical performance for IMT-2020 radio interface(s), 2017.

[7] I Afolabi, T Taleb, K Samdanis, A Ksentini, and H Flinck. Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies and Solutions. *IEEE Communications Surveys Tutorials*, page 1, 2018. doi: 10.1109/COMST.2018.2815638.

[8] qorvo. Getting to 5G: Comparing 4G and 5G System Requirements, 2017. URL https://www.qorvo.com/design-hub/blog/getting-to-5g-comparing-4g-and-5g-system-requirements.

[9] ETSI. ETSI "Multi-Access Edge Computing", 2017.

[10] 3GPP. TS 23.501. System Architecture for the 5G System version 15.2.0 (Release 15), 2018.

[11] J. Tang, B. Shim, T. Chang, and T. Q. S. Quek. Incorporating urllc and multicast embb in sliced cloud radio access network. In *IEEE International Conference on Communications (ICC)*, pages 1–7, 2019.

[12] E. J. Kitindi, S. Fu, Y. Jia, A. Kabir, and Y. Wang. Wireless network virtualization with sdn and c-ran for 5g networks: Requirements, opportunities, and challenges. *IEEE Access*, 5: 19099–19115, 2017. doi: 10.1109/ACCESS.2017.2744672.

[13] 3GPP. TR 38.801. Technical Specification Group Radio Access Network; Study on new radio access technology: Radio access architecture and interfaces release 14, 2017.

[14] D Harutyunyan and R Riggio. Flexible functional split in 5G networks. In *2017 13th International Conference on Network and Service Management (CNSM)*, pages 1–9, nov 2017. doi: 10.23919/CNSM.2017.8255992.

[15] C Y Chang, R Schiavi, N Nikaein, T Spyropoulos, and C Bonnet. Impact of packetization and functional split on C-RAN fronthaul performance. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–7, 2016. doi: 10.1109/ICC.2016.7511579.

[16] M Jaber, M A Imran, R Tafazolli, and A Tukmanov. 5G Backhaul Challenges and Emerging Research Directions: A Survey. *IEEE Access*, 4:1743–1766, 2016. doi: 10.1109/ACCESS.2016.2556011.

[17] Small Cell Forum. Small cell virtualization functional splits and use cases, 2016.

[18] 3GPP. TR 36.872. Small cell enhancements for E-UTRA and E-UTRAN - Physical layer aspects, December 2013.

[19] 3GPP. TR 36.842. Study on Small Cell enhancements for E-UTRA and E-UTRAN; Higher layer aspects, January 2014.

[20] 3GPP. Overview of 3gpp release 10 v0.1.8, March 2013.

[21] Available online:. https://www.tutorialspoint.com/lte/lte-network-architecture.htm/. Technical report, .

[22] 3GPP. TS 36.300. Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 release 11 v11.4.0, March 2013.

[23] 3GPP. Overview of 3gpp release 11 v0.1.4, March 2013.

[24] 3GPP. Overview of 3gpp release 12 v0.0.8, March 2013.

[25] China Mobile. C-RAN: The Road Green RAN White Paper V3. 0, 2011.

[26] Yonghua Lin, Ling Shao, Zhenbo Zhu, Qing Wang, and Ravie K Sabhikhi. Wireless network cloud: Architecture and system requirements. *IBM Journal of Research and Development*, 54 (1):4–1, 2010.

[27] Krishna C Garikipati, Kassem Fawaz, and Kang G Shin. Rt-opex: Flexible scheduling for cloud-ran processing. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, pages 267–280. ACM, 2016.

[28] Vinay Suryaprakash, Peter Rost, and Gerhard Fettweis. Are heterogeneous cloud-based radio access networks cost effective? *IEEE Journal on Selected Areas in Communications*, 33(10): 2239–2251, 2015.

[29] Hongwu Liu, Sang-Jo Yoo, and Kyung Sup Kwak. Opportunistic relaying for low-altitude uav swarm secure communications with multiple eavesdroppers. *Journal of Communications and Networks*, 20(5):496–508, 2018. doi: 10.1109/JCN.2018.000074.

[30] O-RAN. O-RAN Working Group 1, "O-RAN Architecture Description - v2.00," Technical Specification, 2020.

[31] L. M. P. Larsen, A. Checko, and H. L. Christiansen. A survey of the functional splits proposed for 5g mobile crosshaul networks. *IEEE Communications Surveys Tutorials*, 21(1):146–172, 2019. doi: 10.1109/COMST.2018.2868805.

[32] Jianhua Tang, Ruihan Wen, Tony QS Quek, and Mugen Peng. Fully exploiting cloud computing to achieve a green and flexible c-ran. *IEEE Communications Magazine*, 55(11):40–46, 2017.

[33] ONF TR-521. SDN Architecture, 2016.

[34] ETSI NFV. Network Functions Virtualisation, An introduction, benefits, enablers, challenges and call for action, 2012.

[35] Ivan Farris, Tarik Taleb, Yacine Khettab, and Jaeseung Song. A survey on emerging sdn and nfv security mechanisms for iot systems. *IEEE Communications Surveys & Tutorials*, 21(1): 812–837, 2019. doi: 10.1109/COMST.2018.2862350.

[36] Jose Ordonez-Lucena, Pablo Ameigeiras, Diego Lopez, Juan J Ramos-Munoz, Javier Lorca, and Jesus Folgueira. Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges. *IEEE Communications Magazine*, 55(5):80–87, 2017.

[37] ONF TR-526. Applying SDN Architecture to 5G Slicing, 2016.

[38] X Wang, C Cavdar, L Wang, M Tornatore, H S Chung, H H Lee, S M Park, and B Mukherjee. Virtualized Cloud Radio Access Network for 5G Transport. *IEEE Communications Magazine*, 55(9):202–209, 2017. ISSN 0163-6804. doi: 10.1109/MCOM.2017.1600866.

[39] R Ferrus, O Sallent, J Perez-Romero, and R Agusti. On 5G Radio Access Network Slicing: Radio Interface Protocol Features and Configuration. *IEEE Communications Magazine*, 56(5): 184–192, 2018. ISSN 0163-6804. doi: 10.1109/MCOM.2017.1700268.

[40] Mehrdad Moradi, Wenfei Wu, Li Erran Li, and Zhuoqing Morley Mao. Softmow: Recursive and reconfigurable cellular wan architecture. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pages 377–390. ACM, 2014.

[41] Xin Jin, Li Erran Li, Laurent Vanbever, and Jennifer Rexford. Softcell: Scalable and flexible cellular core network architecture. In *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, pages 163–174. ACM, 2013.

[42] ETSI GS NFV 002. Network Functions Virtualization (NFV); Architectural Framework, v. 1.1.1, 2014.

[43] ETSI GS NFV-MAN 001. Network Functions Virtualisation (NFV); Management and Orchestration, v. 1.1.1, 2014.

[44] Chathurika Ranaweera, Elaine Wong, Ampalavanapillai Nirmalathas, Chamil Jayasundara, and Christina Lim. 5g c-ran architecture: A comparison of multiple optical fronthaul networks. In *2017 International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE, 2017.

[45] UPC Thesis. Validation and Extension of Kubernetes-based Network Functions (KNFs) in OSM for Cloud Native (CN) applications in 5G and beyond, January 2021.

[46] Samsung. Virtualized Radio Access Network: Architecture, Key technologies and Benefits. Technical report, 2019.

[47] Mung Chiang and Tao Zhang. Fog and iot: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6):854–864, 2016.

[48] ETSI. Mobile edge computing Introductory technical white paper. Available online: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge-computing-introductory-technical-white-paper-v1, 2018-09-14.pdf. Technical report, Sep 2014.

[49] Gerhard P Fettweis. The tactile internet: Applications and challenges. *IEEE Vehicular Technology Magazine*, 9(1):64–70, 2014.

[50] ETSI/MEC. Technical white paper. Available online: http://www.etsi.org/technologies-clusters/technologies/ multi-access-edge-computing. Technical report, 2017.

[51] Meryem Simsek, Adnan Aijaz, Mischa Dohler, Joachim Sachs, and Gerhard Fettweis. 5g-enabled tactile internet. *IEEE Journal on Selected Areas in Communications*, 34(3):460–473, 2016.

[52] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials*, 17(4):2347–2376, 2015.

[53] ETSI. ETSI “Cloud RAN and MEC: A Perfect Pairing”, 2018.

[54] A Garcia-Saavedra, X Costa-Perez, D.J Leith, and G Iosifidis. FluidRAN: Optimized vRAN/MEC Orchestration. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 1–9, 2018.

[55] Mohammad Asif Habibi, Faqir Zarrar Yousaf, and Hans D. Schotten. Mapping the vnfs and vls of a ran slice onto intelligent pops in beyond 5g mobile networks. *IEEE Open Journal of the Communications Society*, 3:670–704, 2022. doi: 10.1109/OJCOMS.2022.3165000.

[56] Sustainability. What is 5G? Emerging 5G Mobile Services and Network Requirements. Technical report, 2017.

[57] Available online:. https://www.avl.com/-/system-engineering-for-assisted-autonomous-driving. Technical report, .

[58] A Garcia-Saavedra, J X Salvat, X Li, and X Costa-Perez. WizHaul: On the Centralization Degree of Cloud RAN Next Generation Fronthaul. *IEEE Transactions on Mobile Computing*, page 1, 2018. ISSN 1536-1233. doi: 10.1109/TMC.2018.2793859.

[59] Xenofon Foukas, Navid Nikaein, Mohamed M Kassem, Mahesh K Marina, and Kimon Kontovasilis. FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks. In *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '16, pages 427–441, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4292-6. doi: 10.1145/2999572.2999599. URL http://doi.acm.org/10.1145/2999572.2999599.

[60] X Foukas, M K Marina, and K Kontovasilis. Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, MobiCom '17, pages 127–140, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4916-1. doi: 10.1145/3117811.3117831. URL http://doi.acm.org/10.1145/3117811.3117831.

[61] A Maeder, M Lalam, A De Domenico, E Pateromichelakis, D Webben, J Bartelt, R Fritzsche, and P Rost. Towards a flexible functional split for cloud-RAN networks. In *2014 European Conference on Networks and Communications (EuCNC)*, pages 1–5, 2014. doi: 10.1109/EuCNC.2014.6882691.

[62] Behnam Ojaghi, Ferran Adelantado, Angelos Antonopoulos, and Christos Verikoukis. Impact of network densification on joint slicing and functional splitting in 5g. *IEEE Communications Magazine*, 60(7):30–35, 2022. doi: 10.1109/MCOM.001.2100680.

[63] Behnam Ojaghi, Ferran Adelantado, and Christos Verikoukis. On the benefits of vdu standardization in softwarized ng-ran: Enabling technologies, challenges, and opportunities. *IEEE Communications Magazine*, pages 1–7, 2022. doi: 10.1109/MCOM.001.2200390.

[64] Paul B. Menage and Google Inc. Adding generic process containers to the linux kernel. In *Proceedings of the Ottawa Linux Symposium*, 2007.

[65] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, 2014(239):2, 2014.

[66] Young Han Kim et al. Slicing the next mobile packet core network. In *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*, pages 901–904. IEEE, 2014.

[67] Akihiro Nakao, Ping Du, Yoshiaki Kiriha, Fabrizio Granelli, Anteneh Atumo Gebremariam, Tarik Taleb, and Miloud Bagaa. End-to-end network slicing for 5g mobile networks. *Journal of Information Processing*, 25:153–163, 2017.

[68] Navid Nikaein, Eryk Schiller, Romain Favraud, Kostas Katsalis, Donatos Stavropoulos, Islam Alyafawi, Zhongliang Zhao, Torsten Braun, and Thanasis Korakis. Network store: Exploring slicing in future 5g networks. In *Proceedings of the 10th International Workshop on Mobility in the Evolving Internet Architecture*, pages 8–13. ACM, 2015.

[69] Arijit Banerjee, Rajesh Mahindra, Karthik Sundaresan, Sneha Kasera, Kobus Van der Merwe, and Sampath Rangarajan. Scaling the lte control-plane for future mobile access. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT

'15, pages 19:1–19:13, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3412-9. doi: 10.1145/2716281.2836104. URL http://doi.acm.org/10.1145/2716281.2836104.

[70] Tarik Taleb, Adlen Ksentini, and Abdellatif Kobbane. Lightweight mobile core networks for machine type communications. *IEEE Access*, 2:1128–1137, 2014.

[71] Zafar Ayyub Qazi, Phani Krishna Penumarthi, Vyas Sekar, Vijay Gopalakrishnan, Kaustubh Joshi, and Samir R Das. Klein: A minimally disruptive design for an elastic cellular core. In *Proceedings of the Symposium on SDN Research*, page 2. ACM, 2016.

[72] Tarik Taleb, Marius Corici, Carlos Parada, Almerima Jamakovic, Simone Ruffino, Georgios Karagiannis, and Thomas Magedanz. Ease: Epc as a service to ease mobile core network deployment over cloud. *IEEE Network*, 29(2):78–88, 2015.

[73] Tarik Taleb. Toward carrier cloud: Potential, challenges, and solutions. *IEEE Wireless Communications*, 21(3):80–91, 2014.

[74] Tarik Taleb, Adlen Ksentini, and Riku Jantti. " anything as a service" for 5g mobile systems. *IEEE Network*, 30(6):84–91, 2016.

[75] Ibrahim Afolabi, Tarik Taleb, Konstantinos Samdanis, Adlen Ksentini, and Hannu Flinck. Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. *IEEE Communications Surveys & Tutorials*, 20(3):2429–2453.

[76] A de la Oliva, J A Hernandez, D Larrabeiti, and A Azcorra. An overview of the CPRI specification and its application to C-RAN-based LTE scenarios. *IEEE Communications Magazine*, 54(2):152–159, 2016. ISSN 0163-6804. doi: 10.1109/MCOM.2016.7402275.

[77] H. Chien, Y. Lin, C. Lai, and C. Wang. End-to-end slicing with optimized communication and computing resource allocation in multi-tenant 5g systems. *IEEE Transactions on Vehicular Technology*, 69(2):2079–2091, 2020.

[78] G. Tseliou, F. Adelantado, and C. Verikoukis. Netslic: Base station agnostic framework for network slicing. *IEEE Transactions on Vehicular Technology*, 68(4):3820–3832, 2019.

[79] V. G. Nguyen and Y. H. Kim. Slicing the next mobile packet core network. In *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*, pages 901–904, 2014.

[80] Akihiro Nakao, Ping Du, Yoshiaki Kiriha, Fabrizio Granelli, Anteneh Atumo Gebremariam, Tarik Taleb, and Miloud Bagaa. End-to-end network slicing for 5g mobile networks. *Journal of Information Processing*, 25:153–163, 2017. doi: 10.2197/ipsjjip.25.153.

[81] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014.

[82] T. Taleb, A. Ksentini, and A. Kobbane. Lightweight mobile core networks for machine type communications. *IEEE Access*, 2:1128–1137, 2014.

[83] Zafar Ayyub Qazi, Phani Krishna Penumarthi, Vyas Sekar, Vijay Gopalakrishnan, Kaustubh Joshi, and Samir Ranjan Das. KLEIN: A minimally disruptive design for an elastic cellular core. In Brighten Godfrey and Martín Casado, editors, *Proceedings of the Symposium on SDN Research, SOSR 2016, Santa Clara, CA, USA, March 14 - 15, 2016*, page 2. ACM, 2016. doi: 10.1145/2890955.2890961. URL https://doi.org/10.1145/2890955.2890961.

[84] Chia-Yu Chang, Navid Nikaein, Raymond Knopp, Thrasyvoulos Spyropoulos, and S Sandeep Kumar. Flexcran: A flexible functional split framework over ethernet fronthaul in cloud-ran. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2017.

[85] Sergio Gonzalez-Diaz, Andres Garcia-Saavedra, Antonio De La Oliva, Xavier Costa-Perez, Robert Gazda, Alain Mourad, Thomas Deiss, Josep Mangues-Bafalluy, Paola Iovanna, Stefano Stracca, et al. Integrating fronthaul and backhaul networks: Transport challenges and feasibility results. *IEEE Transactions on Mobile Computing*, 2019.

[86] Andres Garcia-Saavedra, George Iosifidis, Xavier Costa-Perez, and Douglas J Leith. Joint optimization of edge computing architectures and radio access networks. *IEEE Journal on Selected Areas in Communications*, 36(11):2433–2443, 2018.

[87] P. L. Vo, M. N. H. Nguyen, T. A. Le, and N. H. Tran. Slicing the edge: Resource allocation for ran network slicing. *IEEE Wireless Communications Letters*, 7(6):970–973, 2018.

[88] H. Zhang and V. W. S. Wong. A two-timescale approach for network slicing in c-ran. *IEEE Transactions on Vehicular Technology*, 69(6):6656–6669, 2020.

[89] B. Ojaghi, F. Adelantado, E. Kartsakli, A. Antonopoulos, and C. Verikoukis. Sliced-ran: Joint slicing and functional split in future 5g radio access networks. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2019. doi: 10.1109/ICC.2019.8761081.

[90] 5G-Crosshaul. H2020 5G-Crosshaul project Grant No. 671598. Detailed analysis of the technologies to be integrated in the XFE based on previous internal reports from WP2/3, 2018.

[91] Xavier Costa-Perez, Andres Garcia-Saavedra, Xi Li, Thomas Deiss, Antonio De La Oliva, Andrea Di Giglio, Paola Iovanna, and Alain Moored. 5g-crosshaul: An sdn/nfv integrated fronthaul/backhaul transport network architecture. *IEEE Wireless Communications*, 24(1):38–45, 2017.

[92] Armen S. Asratian, Tristan M. J. Denley, and Roland Häggkvist. *Bipartite Graphs and their Applications*. Cambridge Tracts in Mathematics. Cambridge University Press, 1998. doi: 10.1017/CBO9780511984068.

[93] *Coordinated Multi-Point in Mobile Communications: From Theory to Practice*. Cambridge University Press, 2011. doi: 10.1017/CBO9780511783029.

[94] Agapi Mesodiakaki, Ferran Adelantado, Luis Alonso, and Christos Verikoukis. Energy-efficient context-aware user association for outdoor small cell heterogeneous networks. In *2014 IEEE International Conference on Communications (ICC)*, pages 1614–1619. IEEE, 2014.

[95] NGMN. 5G Extreme Requirements: Radio Access Network Solutions. Technical report, 2018.

[96] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman Co., USA, 1990. ISBN 0716710455.

[97] IBM ILOG Cplex. V12. 1: User's manual for cplex. *International Business Machines Corporation*, 46(53):157, 2009.

[98] R. G. Jeroslow. Trivial integer programs unsolvable by branch-and-bound. *Math. Program.*, 6 (1):105–109, December 1974. ISSN 0025-5610. doi: 10.1007/BF01580225. URL https://doi.org/10.1007/BF01580225.

[99] Emilie Danna, Edward Rothberg, and Claude Le Pape. Exploring relaxation induced neighborhoods to improve mip solutions. *Mathematical Programming*, 102(1):71–90, 2005.

[100] Ilias Tsompanidis, Ahmed H Zahran, and Cormac J Sreenan. Mobile network traffic: A user behaviour model. In *2014 7th IFIP Wireless and Mobile Networking Conference (WMNC)*, pages 1–8. IEEE, 2014.

[101] M. Skorin-Kapov, L.; Matijasevic. Analysis of qos requirements for e-health services and mapping to evolved packet system qos classes. In *Int. J. Telemed. Appl.*, pages 1–18, 2010.

[102] Yinan Qi, M. Hunukumbure, M. Nekovee, J. Lorca, and V. Sgardoni. Quantifying data rate and bandwidth requirements for immersive 5g experience. In *2016 IEEE International Conference on Communications Workshops (ICC)*, pages 455–461, May 2016. doi: 10.1109/ICCW.2016.7503829.

[103] O-RAN Working Group 1. O-RAN Architecture Description v2.00, Technical Specification, 2020.

[104] Behnam Ojaghi, Ferran Adelantado, Angelos Antonopoulos, and Christos Verikoukis. Slicedran: Service-aware network slicing framework for 5g radio access networks. *IEEE Systems Journal*, pages 1–12, 2021. doi: 10.1109/JSYST.2021.3064398.

[105] 3GPP. TR 28.801. Study on Management and Orchestration of Network Slicing for Next Generation Network (Release 15), 2017.

[106] Xenofon Foukas and Bozidar Radunovic. Concordia: Teaching the 5g vran to share compute. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, SIGCOMM '21, page 580–596, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383837. doi: 10.1145/3452296.3472894. URL https://doi.org/10.1145/3452296.3472894.

[107] Peter Rost, Christian Mannweiler, Diomidis S Michalopoulos, Cinzia Sartori, Vincenzo Sciancalepore, Nishanth Sastry, Oliver Holland, Shreya Tayade, Bin Han, Dario Bega, et al. Network slicing to enable scalability and flexibility in 5g mobile networks. *IEEE Communications magazine*, 55(5):72–79, 2017.

[108] S. E. Elayoubi, S. B. Jemaa, Z. Altman, and A. Galindo-Serrano. 5g ran slicing for verticals: Enablers and challenges. *IEEE Communications Magazine*, 57(1):28–34, 2019. doi: 10.1109/MCOM.2018.1701319.

[109] D. Chitimalla, K. Kondepu, L. Valcarenghi, M. Tornatore, and B. Mukherjee. 5g fronthaul-latency and jitter studies of cpri over ethernet. *IEEE/OSA Journal of Optical Communications and Networking*, 9(2):172–182, 2017. doi: 10.1364/JOCN.9.000172.

[110] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511804441.

[111] J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerisch Mathematik*, 4, 1962.

[112] Alysson M Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers & operations research*, 32, 2005.

[113] Ahmed Ibrahim, Octavia A Dobre, Telex MN Ngatched, and Ana Garcia Armada. Bender's decomposition for optimization design problems in communication networks. *IEEE Network*, 34, 2019.

[114] Hui Lin and Halit Üster. Exact and heuristic algorithms for data-gathering cluster-based wireless sensor network design problem. *IEEE/ACM Transactions on Networking*, 22(3):903–916, 2014. doi: 10.1109/TNET.2013.2262153.

[115] Behnam Behdani, J. Cole Smith, and Ye Xia. The lifetime maximization problem in wireless sensor networks with a mobile sink: Mixed-integer programming formulations and algorithms. *IIE Transactions (Institute of Industrial Engineers)*, 45(10):1094–1113, October 2013. ISSN 2472-5854. doi: 10.1080/0740817X.2013.770189. Funding Information: This work has been supported by the National Science Foundation through grant CMMI-1100765 and the Defense Threat Reduction Agency through grant HDTRA1-10-1-0050. The authors are very grateful for the insightful remarks contributed by two referees, which led to an improved version of this article.

[116] Peter Kall and János Mayer. Building and solving stochastic linear programming models with slp-ior. In *Applications of stochastic programming*, pages 79–93. SIAM, 2005.

[117] Zhen Liu, Jiawei Zhang, Yanan Li, and Yuefeng Ji. Hierarchical mec servers deployment and user-mec server association in c-rans over wdm ring networks. *Sensors*, 20(5), 2020. ISSN 1424-8220. doi: 10.3390/s20051282. URL https://www.mdpi.com/1424-8220/20/5/1282.

[118] Vinay Suryaprakash, Peter Rost, and Gerhard Fettweis. Are heterogeneous cloud-based radio access networks cost effective? *IEEE Journal on Selected Areas in Communications*, 33(10): 2239–2251, 2015. doi: 10.1109/JSAC.2015.2435275.

[119] Forbes. Classifying the modern edge computing platforms, [online], available: https://www.forbes.com/sites/janakirammsv/2020/07/14/classifying-the-modern-edge-computing-platforms/?sh=2fc722144543.