

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Dynamic Data Radio Bearer Management for O-RAN Slicing in 5G Standalone Networks

Pedro Miguel Teixeira Ventura da Silva

MASTERS DISSERTATION



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Masters in Electrical and Computer Engineering

Supervisor: Prof. Manuel Ricardo

Co-Supervisor: André Coelho

July 24, 2024

Abstract

The rapid growth of data traffic, evolving service demands, and the need for network flexibility are prompting a paradigm shift from traditional network architectures. 5G introduced a new architecture for mobile networks, offering solutions with reduced latency and higher availability. However, these solutions still employ integrated hardware units, where a singular piece of equipment is in charge of performing a specific set of functions. This approach has significant limitations in terms of configuration procedures and interoperability issues.

The emerging Open Radio Access Networking (O-RAN) paradigm proposes a new approach for the deployment of the Radio Access Network (RAN). O-RAN offers the possibility of remote configuration and management of multiple independent virtualized components connected through standardized and open interfaces using software-based applications. For these reasons, O-RAN enables simpler and cost-effective solutions, which can interoperate with multi-vendor components.

Network slicing, which is one of the key enablers introduced in 5G and developed even deeper in O-RAN, potentiates tailoring the networks to meet heterogeneous users' requirements. The configuration and management of Data Radio Bearers (DRBs) are the building blocks for defining and adjusting the differentiated Quality of Service (QoS) levels of network slices. They enable the network to deliver the specific performance guarantees required by diverse applications running on top of a shared physical infrastructure. Dynamically adjusting network slices through DRB configuration, based on parameters monitored in real-time by an O-RAN Base Station, paves the way for meeting the user's demands.

The main contribution of this dissertation is a private Standalone 5G Network, based on the O-RAN architecture, capable of providing network slices through dynamic DRB management. The proposed solution employs a DRB Management xApp to collect metrics from the RAN and implement an algorithm that can send control actions to create or release DRBs. Moreover, it operates an algorithm to associate the traffic to the desired DRB. This solution mitigates packet loss and prioritizes specific traffic, thus ensuring optimal utilization of the network's resources. The proposed solution underwent experimental validation in an emulated environment, using OpenAirInterface, to replicate real-world use cases.

Resumo

O rápido crescimento do tráfego de dados, a evolução da procura de serviços e a necessidade de flexibilidade da rede estão a promover uma mudança de paradigma das arquiteturas de rede tradicionais. O 5G introduziu uma nova arquitetura para as redes móveis, oferecendo soluções com latência reduzida e maior disponibilidade. No entanto, essas soluções ainda empregam unidades de *hardware* integradas, onde um único equipamento é responsável pela execução de um conjunto específico de funções. Isto implica limitações significativas em termos de procedimentos de configuração e problemas de interoperabilidade.

O paradigma *Open Radio Access Networking* (O-RAN) está a emergir e propõe uma nova abordagem para a implementação da Rede de Acesso Rádio (RAN). O O-RAN oferece a possibilidade de configuração e gestão remotas de múltiplos componentes virtualizados independentes, ligados através de interfaces normalizadas e abertas, utilizando aplicações baseadas em *software*. Por estas razões, o O-RAN proporciona soluções mais simples e de menor custo, que podem interoperar com componentes de múltiplos fornecedores.

O *Network Slicing*, que é um dos principais impulsionadores introduzidos no 5G e aprofundado ainda mais no O-RAN, potencia a adaptação das redes para satisfazer os requisitos heterogêneos dos utilizadores. A configuração e a gestão dos *Data Radio Bearers* (DRBs) são a base para definir e ajustar os níveis de Qualidade de Serviço (QoS) das *slices* de rede. Estes permitem que a rede ofereça as garantias de desempenho específicas exigidas por diversas aplicações usadas sobre uma infraestrutura física partilhada. O ajuste dinâmico as *slices* de rede através da configuração e gestão de DRBs, com base em parâmetros monitorizados em tempo real por uma estação base O-RAN, abre caminho para satisfazer as exigências dos utilizadores.

A principal contribuição desta dissertação é uma rede 5G *Standalone* privada, baseada na arquitetura O-RAN, capaz de fornecer *slices* de rede através da gestão e configuração dinâmica de DRBs. A solução proposta emprega uma xApp de gestão de DRBs para recolher métricas da RAN e implementa um algoritmo capaz de enviar ações de controlo para configuração ou remoção de DRBs. Além disso, utiliza um algoritmo para associação do tráfego ao DRB desejado. Esta solução mitiga a perda de pacotes e prioriza tráfego específico, garantindo assim a utilização ótima dos recursos da rede. A solução proposta foi validada experimentalmente num ambiente emulado, utilizando o OpenAirInterface, para replicar casos de uso do mundo real.

United Nations Sustainable Development Goals

In 2015, the United Nations formulated the "2030 Agenda for Sustainable Development", which is a plan to take action in the fight for peace and prosperity among the nations [1]. The agenda outlines 17 Sustainable Development Goals and 169 Targets related to each role. These goals call for a collective effort from governments, businesses, and academic institutions to create a more sustainable and equitable future for all. Figure 1 represents the 17 Sustainable Development Goals.



Figure 1: United Nations Sustainable Development Goals [1].

The dissertation is directly aligned with several of these goals:

- **Goal 9 - Industry, Innovation and Infrastructure:** The development of efficient and sustainable industrial processes is crucial for achieving this goal. This dissertation focuses on implementing and validating a private 5G standalone network using the O-RAN architecture and tailoring its network functions to the Industry 4.0 program, directly aligning with resilient infrastructure and industry development. By leveraging O-RAN, the project aims to enhance the flexibility and efficiency of network infrastructure, promoting inclusive and sustainable industrialization and fostering innovation. O-RAN facilitates more sustainable telecommunications infrastructure by enabling interoperability among multi-vendor components. These contributions help to meet the first and fourth targets, upgrading the current industry and reducing the carbon dioxide emission per unit of added value.
- **Goal 11 - Sustainable Cities and Communities:** Smart city initiatives rely on robust and flexible communications networks to manage resources efficiently and improve urban living conditions. The dissertation contributes to the development of such networks by proposing a

flexible, efficient, and cost-effective 5G infrastructure with capacity for traffic prioritization, enabling better planning and management of urban areas.

- **Goal 12 - Responsible Consumption and Production:** The efficient use of resources, which is key to meeting this goal, achieved through O-RAN and network slicing, contributes to a more sustainable telecommunications infrastructure. In this way, it promotes responsible consumption of resources by optimizing network resource allocation and reducing redundancy.

This dissertation contributes to several UN Sustainable Development Goals through the advancement of the technological landscape in the field of telecommunications. This is achieved by the implementation of a private 5G standalone network based on the O-RAN architecture, which supports and manages network slices through DRB configuration. By promoting resilient industry infrastructure and responsible resource consumption, the dissertation contributes to broader sustainability and societal well-being objectives.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Manuel Ricardo; co-supervisor, Dr André Coelho; and Eng. Gonçalo Queirós for their guidance and support throughout the dissertation. Your insights and contributions have been essential to its success.

My heartfelt appreciation goes to my family. To my mom and dad, Manuela and Manuel, whose love, sacrifices, and unwavering belief in my abilities have been my greatest motivation for achieving my goals. To my grandparents, whose love and encouragement always kept me going on the right track. To my brother for always making my days happier and relaxed. Thank you for always being there for me in every moment. I am immensely fortunate to have you by my side.

A special and profound thank you to my girlfriend, Ana. Your patience and support have been my anchor throughout the journey. Thank you for being my better half and the person with whom I can share all my problems and celebrate every milestone.

Finally, I want to express my gratitude to all my friends for the countless moments of joy and laughter, as well as for all of your support. I will always hold a special place for each of you in my heart.

This work is co-financed by Component 5 - Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021-2026, within project Produtech-R3, with reference 60.

Pedro Silva

*“Society doesn’t have values.
People have values.”*

Milton Friedman

Contents

1	Introduction	1
1.1	Context	1
1.2	Problem Definition & Motivation	2
1.3	Objectives	3
1.4	Contributions	3
1.5	Document Structure	3
2	State of the Art	5
2.1	5G Networks	5
2.1.1	Architecture	6
2.1.2	Network Slicing	13
2.2	O-RAN	18
2.2.1	Architecture	18
2.2.2	Split Option 7.2x	21
2.2.3	Network Slicing	22
2.2.4	Software Packages	24
2.3	Related Work	25
2.4	Summary	26
3	System Specification, Design and Implementation	27
3.1	System Specification	27
3.1.1	DRB Management xApp	29
3.1.2	Traffic Mapping to DRBs and QoS	31
3.2	System Design	32
3.2.1	Hardware	32
3.2.2	Software Packages	33
3.3	System Implementation	34
3.3.1	OAI 5G Core Network	34
3.3.2	OAI 5G RAN	37
3.4	Summary	43
4	System Validation	45
4.1	Methodology	45
4.2	Component Testing	45
4.2.1	5G Core Network	46
4.2.2	FlexRIC	47
4.2.3	Disaggregated gNB	48
4.2.4	UE	49

4.2.5	DRB Management xApp	50
4.3	Interoperability Testing	51
4.4	Use Case Validation	56
4.4.1	One UE triggering RAN Reconfiguration	58
4.4.2	Two UEs triggering RAN Reconfiguration	61
4.5	Discussion	66
5	Conclusion	67
5.1	Conclusions	67
5.2	Known Limitations & Future Work	69
	References	71

List of Figures

1.1	Mobile data traffic per user (GB per month) [2].	1
2.1	4G and 5G networks comparison [3].	6
2.2	5G NR architecture. Adapted from [4].	7
2.3	5G NR user plane protocol stack. Adapted from [5].	10
2.4	5G NR control plane protocol stack [5].	10
2.5	5G Quality of Service [6].	11
2.6	Summary of 5QI [6].	13
2.7	Performance metrics and relevant use cases for 5G network slices [7].	14
2.8	Network slice concept. Adapted from [8].	15
2.9	S-NSSAI structure [9].	15
2.10	5G Network Slices working process [9].	17
2.11	Network Slice Lifecycle. Adapted from [8].	18
2.12	O-RAN architecture [10].	19
2.13	Split Option 7.2x with the functions of 5G NR [11].	22
2.14	Split Option 7.2x. Dashed lines indicate optional functions. Adapted from [12].	22
2.15	Network sliced O-RAN [13].	23
2.16	O-RAN Network Slicing Example [14].	24
3.1	High-Level specification of the system.	28
3.2	System architecture used to implement and evaluate the proposed solution.	32
3.3	Asus VivoBook F512D used for the system deployment.	32
3.4	Docker images for the CN functions.	34
3.5	Linux IP Forwarding and Iptables configuration.	34
3.6	Defined MCC, MNC, and TAC for the 5G CN.	35
3.7	Configuration of the demo-oai interface.	35
3.8	Deployed 5G Core network diagram.	36
3.9	MySQL table with pre-defined valid parameters for UEs.	36
3.10	Command to build the gNB and UE.	37
3.11	Communications between CU and DU.	37
3.12	CU configuration file for communication with DU.	38
3.13	DU configuration file for communication with CU.	38
3.14	Communication between CU and DU.	38
3.15	Defined MCC, MNC and TAC, as well as the AMF IP address, and gNB IP and interfaces in the CU configuration file	39
3.16	Defined MCC, MNC and TAC in the DU configuration file	39
3.17	Communications interfaces between CU and both AMF and UPF.	40
3.18	NGAP setup Request and response by the gNB and AMF to validate the connection.	40

3.19	Diagram of the multi UE implementation.	41
3.20	UE configuration for IMSI, key, and OPc.	42
3.21	Registration of the UE in the 5G CN.	42
3.22	FlexRIC IP address and directory to FlexRIC's service models.	43
3.23	FlexRIC service models.	43
4.1	Deploying the OAI 5G Core Network.	46
4.2	Pinging all 5G CN functions to assess connectivity.	46
4.3	5G Core Network functions exchanging information.	47
4.4	Execution of FlexRIC program.	47
4.5	Initialization of the CU.	48
4.6	Initialization of the DU.	48
4.7	Connection between the CU and DU.	48
4.8	Initialization of the UE1.	49
4.9	Interface configuration for UE1.	49
4.10	Initialization of the UE2.	50
4.11	Interface configuration for UE2.	50
4.12	Initialization of the DRB Management xApp.	51
4.13	CU successfully communicating with the AMF.	51
4.14	NearRT-RIC registering the CU and DU.	52
4.15	UE1 successfully registration with AMF.	52
4.16	UE2 successfully registration with AMF.	53
4.17	PDU session configuration and QFI assignment for UE1.	53
4.18	PDU session configuration and QFI assignment for UE2.	54
4.19	IP addressing and interface configuration of UE1 acknowledged by DU.	54
4.20	IP addressing and interface configuration of UE2 acknowledged by DU.	54
4.21	Successfully pinging both UEs from the Internet.	54
4.22	Confirmation of successful pings with wireshark.	55
4.23	E42 communication and E2 nodes registration.	55
4.24	E2 indication messages, transmitting data to and from the xApp	56
4.25	DU log confirming traffic flowing through the default DRB.	57
4.26	Triggering of the DRB Management xApp for creating DRB.	57
4.27	Triggering of the DRB Management xApp for releasing DRB.	58
4.28	CU acknowledging the creation of DRB2.	58
4.29	F1AP interface acknowledging the creation of DRB2 with 5QI 3.	59
4.30	DU acknowledging the creation of DRB2 and rerouting traffic through it.	59
4.31	Evolution of ping response time throughout the test for UE1.	60
4.32	Evolution of throughput and sojourn time in the default DRB across the testing time for UE1.	60
4.33	DU acknowledging the creation of DRB2 for UE1 and rerouting the traffic.	61
4.34	DU acknowledging the creation of DRB2 for UE2 and rerouting the traffic.	62
4.35	CU acknowledging the creation of DRB2 for UE2.	63
4.36	F1AP interface acknowledging the creation of DRB2 with 5QI 9 for UE2.	63
4.37	Evolution of ping response time throughout the test for UE1 and UE2 when both trigger the DRB Management xApp.	64
4.38	Evolution of throughput and sojourn time in the default DRB for UE1 and UE2 when both trigger the DRB Management xApp.	64
4.39	DU acknowledging the releasing of the DRB2 for UE2.	65
4.40	F1AP interface acknowledging the releasing of DRB2 for UE1.	65

List of Tables

2.1	Main 5G interfaces and associated protocols.	10
2.2	Minimum performance requirements for the 5G reference network slices [8]. . .	15
2.3	SST associated to the 5G reference slices.	15
3.1	Definition of the variables used by Algorithm 1.	29
3.2	Definition of the variables used by Algorithm 2.	31
3.3	Core Network function and respective IP addresses.	36

Abbreviations

3GPP	3 rd Generation Partnership Project
5QI	5G QoS Identifier
AF	Authentication Function
AMF	Access and Mobility Management Function
AN	Access Network
AUSF	Authentication Server Function
BBU	Baseband Units
CN	Core Network
COTS	Commercial-off-the-Shelf
CP	Control Plane
C-RAN	Cloud-RAN
DAW	Default Averaging Window
DFL	Default Priority Level
DHCP	Dynamic Host Configuration Protocol
DL	Downlink
DMDBV	Default Maximum Data Burst Volume
DRB	Data Radio Bearers
eMBB	Enhanced Mobile Broadband
eNB	eNodeB
F1AP	F1 Application Protocol
FCAPS	Fault, Configuration, Accounting, Performance and Security
GBR	Guaranteed Bit Rate
gNB	gNodeB
GPRS	General Packet Radio Service
GTP	GPRS Tunnelling Protocol
ICMP	Internet Message Control Protocol
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
ITU	International Telecommunications Industry
KPI	Key Performance Indicator
KPM	Key Performance Metrics
LTE	Long Term Evolution
MAC	Medium Access Control
MCC	Mobile Country Code
mMTC	Massive Machine Type Communications
MNC	Mobile Network Code
NAS	Non-Access Stratum
NEF	Network Exposure Function

NFV	Network Functions Virtualization
NGAP	NG Application Protocol
NR	New Radio
NRF	Network Repository Function
NSA	Non-Standalone
NSI	Network Slice Instance
NSMF	Network Slice Management Function
NSSAI	Network Slice Selection Assistance Information
NSSF	Network Slice Selection Function
NSSMF	Network Slice Subnet Management Function
Near-RT RIC	Near-Real Time RAN Intelligent Controller
Non-RT RIC	Non-Real Time RAN Intelligent Controller
OAI	OpenAirInterface
O-CU	O-RAN Central Unit
O-CU-CP	O-CU Control Plane
O-CU-UP	O-CU User Plane
O-DU	O-RAN Distributed Unit
OPc	Operator Code
O-RAN	Open-RAN
O-RU	O-RAN Radio Unit
PCF	Policy Control Function
PDB	Packet Delay Budget
PDCP	Packet Data Convergence Protocol
PDU	Processing Data Units
PER	Packet Error Rate
PFCP	Packet Forwarding Control Protocol
PHY	Physical Layer
PLMN	Public Land Mobile Network
QFI	QoS Flow Identifier
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RAB	Radio Access Bearer
RF	Radio Frequency
RIC	RAN Intelligent Controller
RLC	Radio Link Control
RRC	Radio Resource Control
SA	Standalone
SCTP	Stream Control Transmission Protocol
SDAP	Service Data Application Protocol
SD	Service Differentiator
SDN	Software-Defined Networking
SINR	Signal-to-Interference-plus-Noise Ratio
SM	Service Models
SMO	Service and Management Orchestration
srsRAN	Software Radio Systems RAN
SST	Slice/Service Type
S-NSSAI	Single Network Slice Selection Assistance Information

TAC	Tracking Area Code
TCP	Transmission Control Protocol
UDM	User Data Management Function
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink
UPF	User Plane Functions
URLLC	Ultra-Reliable Low Latency Communications
VM	Virtual Machines
V-RAN	Virtual RAN

Chapter 1

Introduction

1.1 Context

Telecommunications have been rapidly evolving, driven by an increase in users and the continuous need for improvements. Telecom operators are adopting new networking paradigms to meet these challenges, which include responding to the rise of targeted users and devices and accommodating extreme industry use cases. These demands require maintaining overall network capabilities and ensuring targeted Quality of Service (QoS) levels, especially low latency, ultra-high availability, and reliability. Figure 1.1 depicts the increase in mobile data traffic through the years. Data traffic is expected to increase significantly in the near future, imposing the scaling of the network to enhance availability and connectivity. Relying on the current hardware-based solutions will not support sustainable growth and meet guarantee the users' requirements.

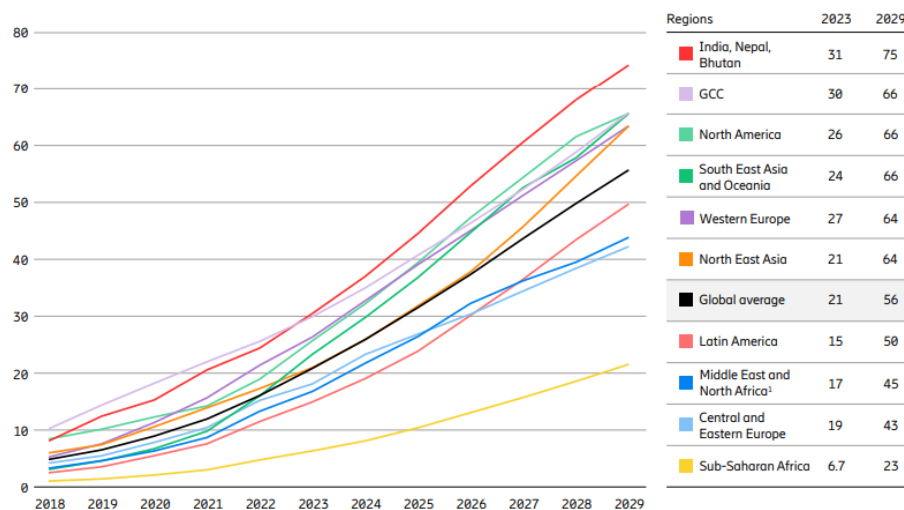


Figure 1.1: Mobile data traffic per user (GB per month) [2].

The development and deployment of 5G networks have brought significant improvements to wireless communications, enabling higher throughput and availability than ever before. This has

promoted the rise of software-based implementations in equipment and network functions, creating the possibility of developing multiple private virtual networks while using the same physical resources. However, despite all the advancements and innovation, many current commercial solutions still rely on proprietary hardware, especially on the Radio Access Network (RAN) part, which poses many challenges.

A new paradigm is emerging, predicated upon the advancements laid by 5G technology, with aspirations to serve as the foundation for the upcoming 6G networks. This paradigmatic shift is characterized by an effort to expand the scope of software solutions within network infrastructures and optimize multiple independent virtual networks capable of operating simultaneously within a shared RAN.

The emerging Open-RAN (O-RAN) paradigm aims at evolving the current deployment approach, based on all-in-one units, to modular open software-based solutions. O-RAN paves the way for faster, more resilient, and easily manageable and reconfigurable network deployments that reduce environmental impact and offer significant cost savings for operators. In addition, its architecture supports the development of software applications designed to customize networks to meet specific user requirements.

1.2 Problem Definition & Motivation

The conventional deployment models of network functions in mobile networks are based on monolithic hardware units, wherein dedicated equipment performs a specific set of functions. This conventional approach presents multiple limitations, including vendor lock-in, complex configuration procedures, and interoperability challenges. Moreover, this approach contributes to increased resource usage and poses environmental concerns, given all the materials needed for building all the hardware units. The challenges in the conventional deployment models motivate a paradigm evolution to ensure more flexible, interoperable, and environmentally sustainable solutions in the design and implementation of mobile networks.

Telecom operators should always ensure the users' requirements are met; therefore, the networks are planned and deployed to meet these goals. Nevertheless, specific scenarios can compromise the desired QoS and network availability. This enables multiple virtualized and independent networks sharing the same physical resources, called network slices, to be configured to meet the targeted requirements.

Many industry use cases have orthogonal requirements that need to be addressed by the network, including ultra-reliability and low-latency communications for some applications, while providing higher bandwidth for others. These scenarios motivate the creation and configuration of multiple virtualized and independent networks on demand to address requirements that may vary over time. In addition, the radio signal propagation conditions are not always constant - they are dynamic according to changes in environmental conditions. The ability to validate the solutions in a laboratory environment that replicates the conditions identical to those found in the targeted

use cases, especially from the network perspective, is crucial for validating novel networking solutions. This process facilitates the subsequent integration of these solutions into a production environment more effectively and reliably since they have been previously validated and evaluated.

1.3 Objectives

The main objective of the dissertation was to develop, implement and validate a solution enabling the creation of dynamic network slices within a private standalone 5G network based on the O-RAN architecture. The specific objectives of this dissertation include:

- Develop an O-RAN 5G Base Station capable of providing dynamic network slices.
- Create a software-based mechanism enabling the creation of network slices through dynamic DRB management while considering variable traffic profiles and radio channel conditions.
- Validate the developed solution in an emulated environment that replicates a real-world 5G standalone network.

1.4 Contributions

The original contributions of this dissertation are:

- An O-RAN 5G Base Station capable of offering dynamic network slices.
- A novel xApp designed for dynamic DRB management, leveraging real-time monitoring of networking parameters to enable the creation of 5G network slices.
- An emulated environment, built on OpenAirInterface, for validation and evaluation of the proposed solution.

1.5 Document Structure

Besides the introduction, this document contains three more chapters. In Chapter 2, the main background and state of the art are presented. In chapter 3, the proposed solution, as well as its design and implementation are specified. In Chapter 4, the tests drawn to validate the solution are presented along with an evaluation of the solution's overall performance. Chapter 5 ends the document with the main conclusions and future work.

Chapter 2

State of the Art

This chapter comprises the fundamental concepts and background knowledge associated with the dissertation. Section 2.1 provides an overview of the 5G architecture, including a comparison between 5G and its predecessors, as well as network slicing. Section 2.2 describes the main concepts associated with Open-Radio Access Networking (O-RAN), distinguishing it from traditional RAN approaches, presenting its architecture, and discussing how O-RAN contributes to fulfilling the 5G networking paradigm, including when it comes to network slicing, including some software packages that implement O-RAN. The related work is presented in Section 2.3. Finally, the chapter ends in Section 2.4 with a summary of the content presented.

2.1 5G Networks

The expansion of networks in terms of user demands and requirements has reached a level where the legacy standards for mobile broadband communications are unable to meet the needs of certain emerging services and applications. With the fast evolution of the networks, the use of previous communications technologies has become less compatible with the desired solutions, which motivated the development of a new standard for telecommunications [15]. The emergence of 5G technology, proposed by 3GPP, represents a noteworthy advancement in the telecommunications field. 5G introduces significant improvements over its predecessors, 4G and Long Term Evolution (LTE), with enhanced throughput, reliability, latency and support for a wide range of applications that go beyond traditional mobile broadband. Its deployment is bringing about transformative changes in various industries and enabling innovative services that were not feasible with previous generations of wireless networks. 5G improves modularity and flexibility through its service-based architecture. This enhances the integration of new services and applications. Furthermore, it allows for the computation capabilities to be closer to the end user, making it possible for real-time processing. Figure 2.1 summarizes the comparison between 4G and 5G networks.

Although 5G achieves better performance in terms of latency, throughput and reliability, it still takes advantage of several features from its predecessor, LTE. Moreover, 5G is developed in a way that makes it possible to connect the 5G NR Access Network (AN) to the Core Network (CN) of

4G, in order to smooth the transition between both and maximize cost-effectiveness. This also ensures compatibility with older UEs, preventing service interruptions and ensuring that users in diverse locations, including remote or rural areas, can still use communications services.

The principles of 5G include the separation of User Plane Function (UPF) and Control Plane (CP) functions, minimal dependencies between the Access Network (AN) and the Core Network (CN), which translates into flexibility and easy escalation, and service-oriented network functions.

There are two types of 5G networks: Standalone (SA) and Non-Standalone (NSA). 5G SA networks consist of both the 5G RAN and the 5G CN, while the 5G NSA networks are built with the 5G RAN connected to a legacy Core, which is the Evolved Packet Core (EPC) in the case of 4G networks [16]. This dissertation focuses on 5G SA networks.



Figure 2.1: 4G and 5G networks comparison [3].

2.1.1 Architecture

3GPP has standardized the 5G New Radio (5G NR) architecture by developing and providing technical specifications for both the 5G RAN and the 5G CN. 5G is built upon the Software-Defined Networking (SDN) paradigm, which logically separates network functions in two different planes: the control plane and the user plane. This logical separation contributes to one of the main principles of SDN, which is the centralized control of policies and management. This allows users to change the networks' infrastructure from a centralized controller, enabling the management of many parameters such as security and routing. Figure 2.2 represents a high-level overview of the 5G architecture. The 5G architecture comprises several CN functions, which exchange information between themselves and the RAN through standardized interfaces with defined roles and capabilities.

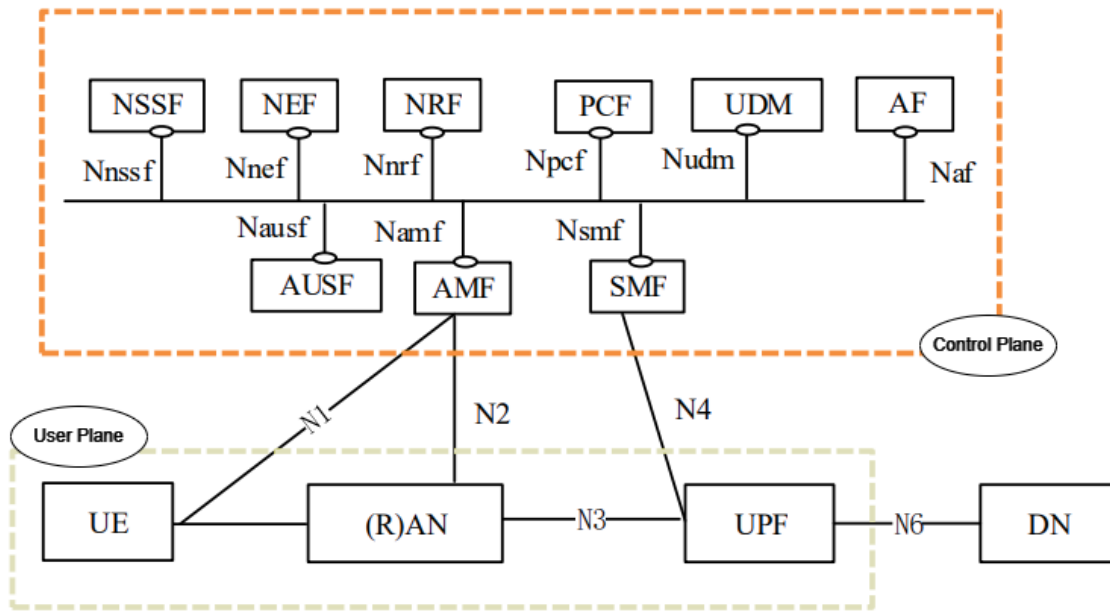


Figure 2.2: 5G NR architecture. Adapted from [4].

2.1.1.1 User Plane

The user plane is responsible for forwarding data packets to/from the User Equipment (UE). It consists of three main components: the UE, the RAN, and the UPF.

The UE is connected to the RAN, which is used to transmit and receive data. The RAN is established through a base station, the gNodeB (gNB), in the context of 5G technology. Its primary function is to ensure wireless connectivity between the UE and the 5G network while simultaneously managing network functions related to radio communications. Moreover, the gNB enables radio transmissions of user data packets and establishes a control link between the UE and the 5G CN. The separation of the gNB into multiple independent and interconnected units is central in 5G RAN. The N2 interface ensures the connectivity between RAN and AMF and uses the NG Application Protocol (NGAP) to support UE and non-UE operations, such as configuration updates, UE context transfer, PDU Session resource management, and NAS message exchange [17]. The Stream Control Transmission Protocol (SCTP) delivers the NGAP messages.

The UPF serves as the primary gateway for user data packets that are transmitted to/ from the UE. It also acts as an anchor point for data flows, ensuring they are appropriately forwarded when the UE connects to a different RAN. In addition, the UPF defines rules and handles and monitors the Quality of Service (QoS) levels provided by the user plane while keeping track of traffic usage [18]. The N3 interface ensures communications between the UPF and RAN by implementing the General Packet Radio Service (GPRS) Tunnelling Protocol (GTP), which encapsulates the Processing Data Units (PDU) session [19]. The UPF also forwards data traffic to/from external networks through the N6 interface.

2.1.1.2 Control Plane

The control plane in 5G is responsible for managing and controlling the overall operation of the network. It handles signalling, management, and coordination functions necessary for network setup, maintenance, and optimization. The 5G architecture consists of different network functions. AMF and SMF act as bridges between control and user planes.

- **Access and Mobility Management Function (AMF):** AMF establishes and manages the 5G control connection between the CN and the UEs by continuously monitoring the registration of each UE in the network and its accessibility to the network's services. Additionally, the AMF acts as the gateway for control traffic exchanged between the RAN and the UE, ensuring the control and coordination of signalling messages from other network functions. Furthermore, as a mobility management function, AMF oversees the UE's handover events triggered by the gNBs.
- **Session Management Function (SMF):** The SMF manages the data sessions between the UPF and the RAN. It is responsible for establishing, modifying, and releasing those sessions. It is also responsible for allocating and managing Internet Protocol (IP) addresses for the UE, acting as a Dynamic Host Configuration Protocol (DHCP) server, which is crucial for the dynamic configuration and assignment of IP addresses. The SMF also plays a key role in defining QoS policies and the configuration of resources in the UPF through the N4 interface. The Packet Forwarding Control Protocol (PFCP) exchanges information between the SMF and the UPF.

Another important network function is the **Network Slice Selection Function (NSSF)**. It is responsible for discovering and provisioning the most suitable network slice for each UE to meet resource demands and requirements. This enables the network to assign each UE to the appropriate network slice tailored to its needs, thereby maximizing overall performance and efficiency. The NSSF interacts with the AMF and SMF. It accesses the AMF to gather information on the UE's location and resource demands. It also interacts with the **Network Repository Function (NRF)** to retrieve the correct slice information and resources needed for the slice instantiation. The NRF stores and manages network information, such as network slice configurations and UE capabilities. In addition, it also establishes communications with the UPF to prioritize traffic based on the network slice to which the UE is assigned to.

Other network functions in the 5G CN include:

- **Policy Control Function (PCF):** The PCF enforces network policies, such as QoS and security policies. This ensures that the network is used fairly and efficiently while user data is protected.
- **User Data Management Function (UDM):** The UDM manages user data, such as subscriptions, profiles, and authentication credentials. This information is used to authenticate UEs and provide them with access to network services.

- **Authentication Function (AF):** The AF authenticates UEs and provides them with security keys. This helps to ensure that only authorized UEs can access the network.
- **Network Exposure Function (NEF):** The NEF exposes network capabilities to external entities, such as third-party service providers. This allows external entities to develop and deploy new applications and services that leverage the capabilities of the 5G network.
- **Authentication Server Function (AUSF):** The AUSF stores and manages user authentication credentials. This information is used by the AF to authenticate UEs.

2.1.1.3 5G NR protocol stacks

The 5G NR protocol stack ensures communications between the UE and the RAN through Processing Data Units (PDUs). It represents a direct evolution from Long Term Evolution (LTE) and is similar to its protocol stack. The protocol stack is divided into two operations: one for exchanging PDUs for the user plane exclusively and another for exchanging PDUs with the user plane and control plane. Both share different functions and sublayers [20]:

- **Service Data Application Protocol (SDAP):** The SDAP is responsible for mapping a QoS flow and a Data Radio Bearer (DRB). Every Processing Data Unit (PDU) session has a single SDAP protocol entity configured and assigned to it.
- **Packet Data Convergence Protocol (PDCP):** The PDCP is responsible for header compression and decompression, as well as cyphering and integrity verification operations.
- **Radio Link Control (RLC):** The RLC is in charge of segmentation, retransmission, and error correction processes.
- **Medium Access Control (MAC):** The MAC is tasked with multiplexing and scheduling operations, as well as error corrections.
- **Physical Layer (PHY):** The PHY is responsible for coding, modulation, and multi-antenna mapping.

Figure 2.3 depicts the 5G NR user plane protocol stack, with all the functions and sublayers.

The 5G NR control plane protocol stack includes the same functions and sublayers but replaces the SDAP protocol with the Radio Resource Control (RRC), which is responsible for security, QoS control, UE measurement reporting, and mobility management operations. Moreover, the control plane includes the Non-Access Stratum (NAS), which is responsible for managing the transmission and signalling of control messages between the UE and the AMF while also providing authentication and connectivity to external networks. The N1 interface is the logical interface that enables the exchange of information between these entities. Figure 2.4 depicts the 5G NR control plane protocol stack and its new additions and modifications compared to the user plane.

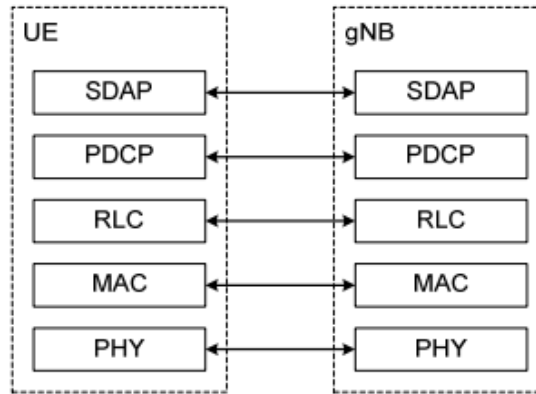


Figure 2.3: 5G NR user plane protocol stack. Adapted from [5].

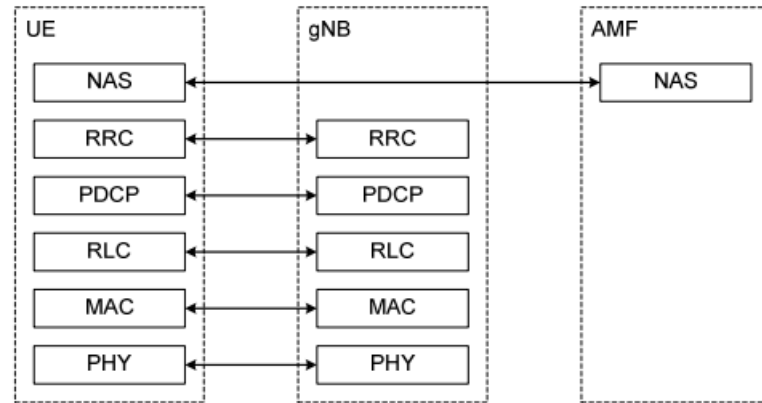


Figure 2.4: 5G NR control plane protocol stack [5].

These functions and sublayers are connected through standardized interfaces that ensure connectivity between them and enable their modes of operation. Table 2.1 summarizes the main 5G interfaces discussed in Section 2.1.1.

Table 2.1: Main 5G interfaces and associated protocols.

Interface	Main Protocol Used	Transport Layer
N1	NAS	TCP/IP
N2	NGAP	SCTP/IP
N3	GTP	UDP/IP
N4	PFCP	UDP/IP
N6	HTTP	TCP/IP

5G builds upon the concepts of SDN and Network Functions Virtualization (NFV), which lie in programmable and hardware-independent network functions that run in virtualized machines or containers. This approach aims at breaking with the current need for proprietary hardware and is becoming increasingly used.

2.1.1.4 5G Quality of Service

The Quality of Service (QoS) in wireless networks determines the overall network performance from the user's perspective. It involves prioritizing different applications, users, or data flows and ensuring a specific performance level for each data flow.

In 5G NR, network performance is managed through QoS flows, where packets within each PDU session are identified by a unique QoS Flow Identifier (QFI) within each PDU session.

5G comprises two types of Radio Access Bearers (RABs):

- **Signalling Radio Bearers (SRB):** The SRBs are utilized to transmit control plane signalling messages, such as RRC messages, which are required to establish, maintain, and terminate connections between the UE and the network.
- **Data Radio Bearers (DRB):** The DRBs, employed in the SRB-based configuration, carry user data between the UE and the network.

The UPF is responsible for classifying and mapping DL packets to appropriate QoS flows, whereas the UE performs similar functions for the UL packets. On the N3 interface between the UPF and the RAN, packets are marked with a QFI in an encapsulation header. This allows for QoS flow identification. The SDAP sublayer maps the flows to suitable DRBs. Conversely, the UE's SDAP sublayer performs the reverse mapping from DRBs to QoS flows at the receiving end.

A similar procedure is followed for UL packets. The UE labels the packet with a QFI within the SDAP header. Then, QFI marking is conducted on the N3 interface using an encapsulation header within the RAN. Following this, the UPF verifies that the QFI received matches the established QoS rules [6] [21].

Figure 2.5 depicts the QFI insertion and DRB mapping for DL packets across different traffic flows. It illustrates that multiple IP flows can be accommodated within a single QoS flow, while multiple QoS flows can be mapped by the SDAP to a single DRB, provided it meets the requirements.

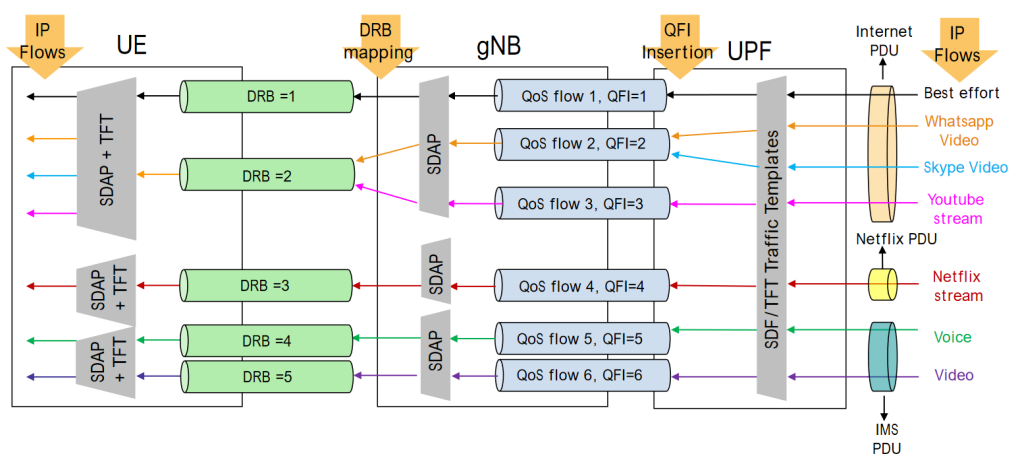


Figure 2.5: 5G Quality of Service [6].

By interacting with other CN functions, such as UDM and PCF, the SMF controls and assigns the QoS flows by creating new PDU sessions or updating and reconfiguring existing ones. There are three types of QoS Flows:

- **Non-Guaranteed Bit Rate (Non-GBR):** It does not guarantee a minimum bit rate, making it suitable for applications with less stringent QoS requirements.
- **Guaranteed Bit Rate (GBR):** Guarantees a certain minimum bit rate, which is adequate for services that require consistent and predictable data rates.
- **Delay Critical Guaranteed Bit Rate:** Provides a guaranteed minimum bit rate with stringent delay requirements, essential for mission-critical applications.

Every PDU session must be associated with a QoS flow; by default, the network maps a Non-GBR flow to facilitate intercommunication with legacy systems. Additionally, associated with these QoS flows, the 5G QoS Identifier (5QI) specifies the QoS characteristics that influence how packets are treated by the network. Several parameters are outlined for in 5QI, as specified by 3GPP:

- **Default Priority Level (DPL):** The DPL determines the order in which packets are processed and forwarded by the network. A lower numerical value indicates a higher priority.
- **Packet Delay Budget (PDB):** Specifies the maximum acceptable delay for packets, defining the longest time a packet should take to be forwarded from the source to the destination across the network.
- **Packet Error Rate (PER):** The PER defines the maximum acceptable rate of erroneous packets for a given service. A lower PER indicates a higher reliability requirement in which fewer errors are tolerated.
- **Default Maximum Data Burst Volume (DMDBV):** The DMDBV defines the largest amount of data that can be sent without violating the QoS requirements for delay-critical GBR services.
- **Default Averaging Window (DAW):** The DAW is used for GBR and Delay-critical GBR services to define the time interval over which the network should measure the guaranteed bit rate.

Figure 2.6 depicts a summarized table of 5QI outlining typical values for each parameter associated, and the main services where these values are applied.

5QI Value	Resource Type	Default Priority Level	Packet Delay Budget (NOTE 3)	Packet Error Rate	Default Maximum Data Burst Volume (NOTE 2)	Default Averaging Window	Example Services
1	GBR (NOTE 1)	20	100 ms (NOTE 11, NOTE 13)	10^{-2}	N/A	2000 ms	Conversational Voice
2		40	150 ms (NOTE 11, NOTE 13)	10^{-3}	N/A	2000 ms	Conversational Video (Live Streaming)
3		30	50 ms (NOTE 11, NOTE 13)	10^{-3}	N/A	2000 ms	Real Time Gaming, V2X messages (see TS 23.287 [121]), Electricity distribution – medium voltage, Process automation monitoring
...		...	...	...	...	...	...
5	Non-GBR (NOTE 1)	10	100 ms NOTE 10, NOTE 13)	10^{-6}	N/A	N/A	IMS Signalling
6		60	300 ms (NOTE 10, NOTE 13)	10^{-6}	N/A	N/A	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
7		70	100 ms (NOTE 10, NOTE 13)	10^{-3}	N/A	N/A	Voice, Video (Live Streaming) Interactive Gaming
...		...	...	...	...	...	...
82	Delay-critical GBR	19	10 ms (NOTE 4)	10^{-4}	255 bytes	2000 ms	Discrete Automation (see TS 22.261 [2])
83		22	10 ms (NOTE 4)	10^{-4}	1354 bytes (NOTE 3)	2000 ms	Discrete Automation (see TS 22.261 [2]); V2X messages (UE - RSU Platooning, Advanced Driving: Cooperative Lane Change with low LoA. See TS 22.186 [111], TS 23.287 [121])
84		24	30 ms (NOTE 6)	10^{-5}	1354 bytes (NOTE 3)	2000 ms	Intelligent transport systems (see TS 22.261 [2])
...		...	...	...	...	...	...

Figure 2.6: Summary of 5QI [6].

2.1.2 Network Slicing

Network Slicing is one of the major 5G enablers. Network Slicing involves dividing a shared physical network infrastructure into multiple virtual networks that enable meeting different performance requirements. This technique optimises network resources, potentially ensuring better performance and more efficient network management.

The concept of network slicing has emerged as a direct result of recent advancements in computing and Network Function Virtualization (NFV) technologies. By dividing a physical network into several logical networks, each one can provide customized services for a distinct application scenario. The 5G network slicing approach, represented by logically isolated and self-contained networks, offers a high degree of flexibility and customization to accommodate diverse business-driven use cases simultaneously in the same network infrastructure. There are three main network slices in 5G: enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications,

and Massive Machine-Type Communications (mMTC). These three main network slices serve different purposes [22]:

- **eMBB:** Focus on enhancing data transfer capabilities and improving the user Quality of Experience (QoE). It aims at delivering a seamless wireless coverage experience, even in dynamic scenarios, by providing not only high peak but also constant data rates in the covered areas, while meeting the requirements of high-speed transportation, including support for dynamic handovers.
- **URLLC:** Focus on communications services and applications that are time-critical and need ultra-reliable network connectivity, imposing stringent requirements for both latency and network availability.
- **mMTC:** Associated to services that are characterized by a massive number of connected devices transmitting non-time-critical information. These devices are typically characterized by low cost and reduced energy consumption.

Figure 2.7 depicts the main metrics and use case scenarios for the three main 5G network slices.

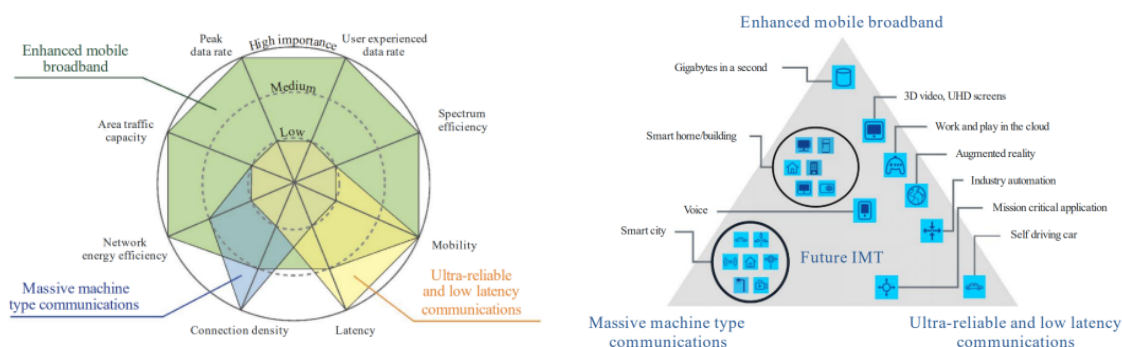


Figure 2.7: Performance metrics and relevant use cases for 5G network slices [7].

Network slices are associated with key performance indicators, including high peak data rates, user-experienced data rates, high area capacity, low latency, and high energy efficiency. Table 2.2 summarizes some of the main KPIs for each type of network slice, as well as some of the reference values associated with them.

Network slicing is a network virtualization technique that allows multiple logical networks to be created over a single shared physical infrastructure. Each slice can be customized to meet the specific needs of a particular application or service, such as latency, throughput, and security requirements. It is an end-to-end concept, covering all network segments. A network slice instance is a set of network functions and the associated resources [8]. The slices are divided into various subnets, which are virtual networks within the slice. Communication services use the network slicing technique to create these virtual networks, which can have multiple network functions. The diagram of Figure 2.8 depicts the concept.

Table 2.2: Minimum performance requirements for the 5G reference network slices [8].

KPI	Key Use Case	Values
Peak Data Rate	eMBB	DL: 20 Gbit/s, UL: 10 Gbit/s
Peak Spectral Efficiency	eMBB	DL: 20 bit/s/Hz, UL: 15 bit/s/Hz
Area Traffic Capacity	eMBB	DL: 15 Mbit/s/m ²
User Plane Latency	eMBB, URLLC	eMBB: 4 ms, URLLC: 1 ms
Control Plane Latency	eMBB, URLLC	20 ms
Connection Density	mMTC	1,000,000 devices/km ²
Mobility	eMBB	Up to 500 km/h
Reliability	URLLC	$P_{success} = 10^{-5}$ for a 32 bytes PDU

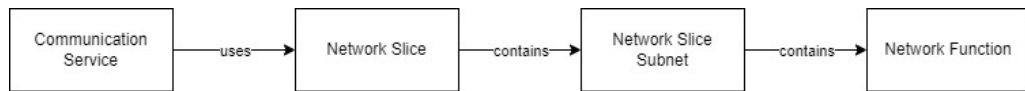


Figure 2.8: Network slice concept. Adapted from [8].

2.1.2.1 Network Slice Identification

The identification of a Network Slice is indicated by a NAS Information Element. Network Slice Selection Assistance Information (NSSAI) is made up of two fields: Slice/Service Type (SST) and Service Differentiator (SD). SST has an 8-bit field length, while SD indicates a slice ID within the SSTn, which is an optional item. This is used to identify specific slices when multiple slices are supported in the same SST. Figure 2.9 represents the S-NSSAI structure.

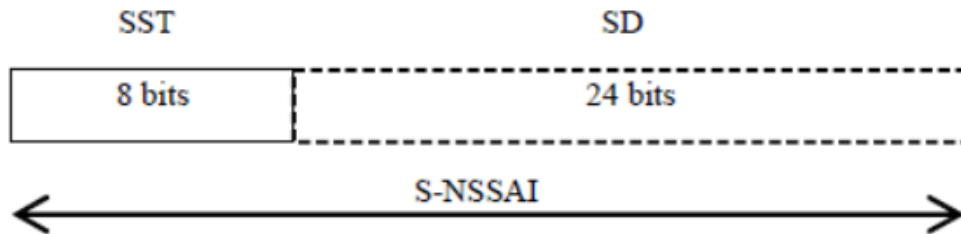


Figure 2.9: S-NSSAI structure [9].

Table 2.3 shows the SST values associated with the 5G reference network slices, as standardized by 3GPP.

Table 2.3: SST associated to the 5G reference slices.

Slice Type	SST Value
eMBB	1
URLLC	2
mMTC	5

2.1.2.2 Network Slice Selection

The slice selection considers various network functions and entities, as depicted in Figure 2.10. The decision to choose a given slice is ultimately the responsibility of the network. However, the network slice selection can be triggered by the UE, by the CN or both. When a UE sends a Registration Request, it includes a specification of NSSAI. This NSSAI is a message to the network that identifies the type of slice the UE wants to access. The requested Slice Information is then sent to UDM. UDM checks whether the requested slice is allowed for that specific UE. If it is allowed, the request is accepted by UDM; if not, it is rejected. Once the slice request is accepted, this is communicated to the UE. There are four main types of NSSAI [9]:

- **Requested NSSAI:** During the registration process, the UE requests NSSAI and expresses its preference for specific network slice parameters. The network then attempts to assign a slice that matches the requested criteria.
- **Allowed NSSAI:** During the process of registering a UE in a 5G network, the CN transmits information about the NSSAI that is allowed for a particular network slice. This information facilitates the UE awareness of the available and allowed network slice parameters.
- **Configured NSSAI:** The NSSAI is authorized based on the subscription of the UE. This information is stored both in the CN and in the UDM system, within the subscription data of the UE. Upon receiving a Registration Request, the network compares the Requested NSSAI provided by the UE with the Subscription Information.
- **Selected NSSAI:** The NSSAI selected for a specific traffic flow based on the agreement between the UE and the network.

Figure 2.10 represents the slice selection process from the first UE request to the network slice assignment. It also shows the role of various network functions in the slice selection.

A high-level overview of the network slice selection steps is provided in [4]. These are the main steps:

1. The UE sends a Registration Request message to the 5G CN to register with the 5G network.
2. The Registration Request message contains a list of NSSAIs that the UE wants to use for the current registration.
3. The 5G CN receives the Registration Request message and verifies if the network allows the requested NSSAIs, based on the configuration of NSSAIs for the respective network slice.
4. If the requested NSSAIs are allowed, then the 5G CN selects the network slice based on the requested NSSAIs and sends a Registration Accept message to the UE.
5. The Registration Accept message contains the allowed NSSAI and other necessary information for the UE to access the network slice.

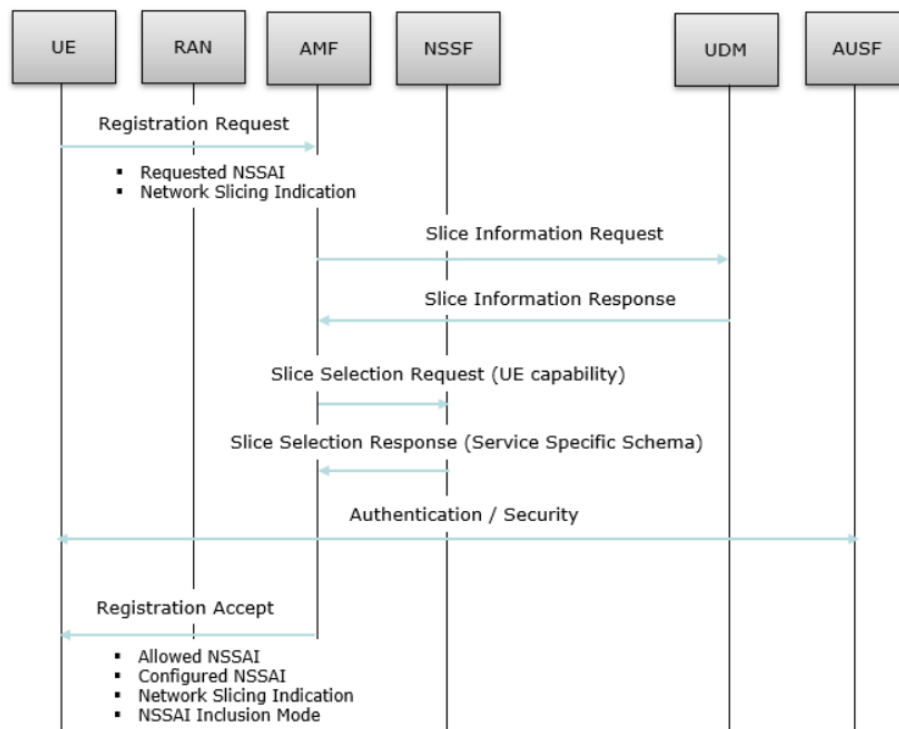


Figure 2.10: 5G Network Slices working process [9].

6. The UE receives the Registration Accept message and is registered with the 5G network.
7. The 5G CN maintains a record of the configured NSSAI for each network slice in its database, which is used to select the appropriate network slice when a UE initiates a registration.
8. When the UE wants to establish a data session, it specifies the selected NSSAI in UL NAS Transport during the PDU session establishment procedure. The network may assign a default NSSAI as the selected NSSAI, instead of the one specified by the UE.
9. If the network can use the selected NSSAI requested by the UE, it sends a PDU Session Establishment Accept message with S-NSSAI, indicating the selected NSSAI.

2.1.2.3 Network Slice Management and Lifecycle

Network Slices are self-contained and mutually isolated from one another. Building upon SDN and NFV, they are easily manageable and programmable. As such, it is possible to create and eliminate slices on demand. Network slices provide support for multi-services and multi-tenancy. As such, the common underlying infrastructure supporting the network slices enables different services without compromising the isolation of each slice. The lifecycle of a network slice instance, which consists of a set of network functions and their respective resources, is depicted in Figure 2.11.

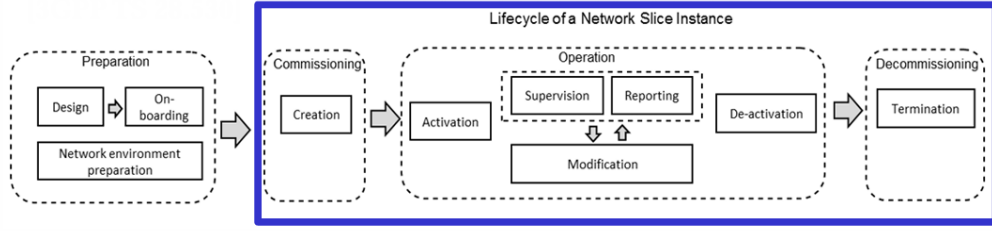


Figure 2.11: Network Slice Lifecycle. Adapted from [8].

2.2 O-RAN

In traditional networks, proprietary hardware-based solutions are responsible for implementing most network functions, which ends up in limited flexibility. O-RAN lies in the SDN to virtualize the RAN network functions (NFV principle), which can run independently in virtual machines or containers. By deploying RANs that are customized for specific use cases, new players can enter the mobile networks market. This is promoted by software-based solutions that operate on Commercial-off-the-Shelf (COTS) servers, which offer a simpler and more cost-effective approach to developing new carrier-grade products [18].

The main advantages of O-RAN compared with previous RAN deployment approaches, such as Cloud-RAN (C-RAN) or Virtual-RAN (V-RAN), are the interoperability and flexibility possible. C-RAN enables efficient network and resource sharing, flexible scheduling, and real-time management through centralized baseband pool sharing in the cloud [23]. Its evolution, V-RAN, fosters the decoupling of software and hardware functionalities, allowing multiple radio units to share the same Baseband Units (BBU) resources, thus simplifying the deployment of RAN nodes. Despite being cost-effective, these architectures still rely on proprietary software, hardware, and interfaces. The lack of openness is a significant obstacle for maximizing virtualization [24].

To address these limitations, the O-RAN Alliance [25], which is the major driving force pushing O-RAN further, proposed three main innovations for 5G RANs: desegregation of the gNB; programmable RAN Intelligent Controllers (RICs); virtualization of RAN functions.

2.2.1 Architecture

Figure 2.12 depicts the O-RAN architecture and its main functional entities and interfaces, including the disaggregation of the gNB in three main units (O-CU, O-DU, O-RU), the programmable RIC, and the virtualization of RAN functions, represented by the Service and Management Orchestrator (SMO).

2.2.1.1 Functional Entities

As shown in Figure 2.12, in O-RAN, the gNB is divided into three main independent and interconnected units:

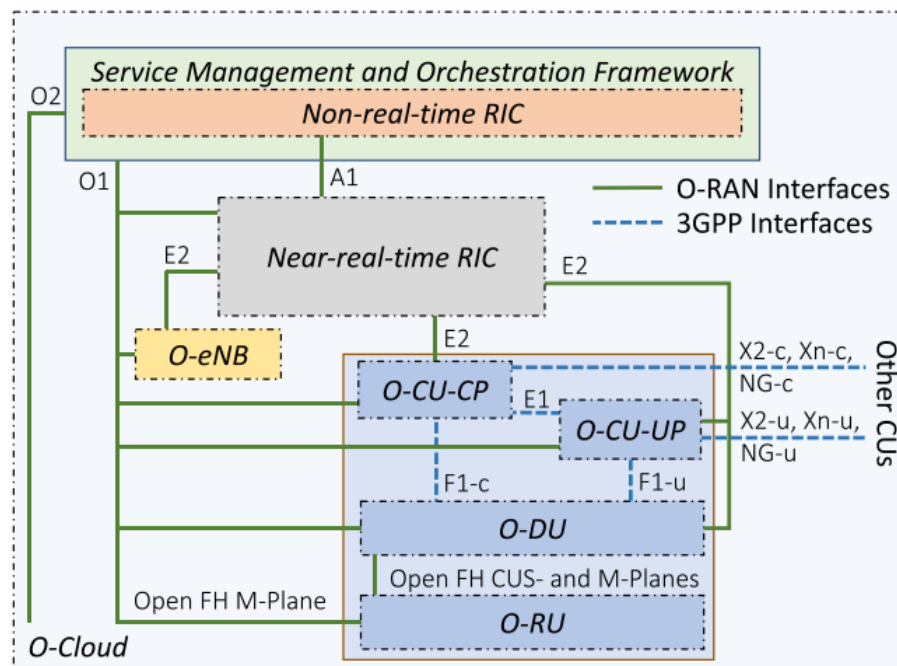


Figure 2.12: O-RAN architecture [10].

- **O-RAN Central Unit:** The O-CU has the responsibility of managing the life cycle of the connection, as well as the QoS associated with the traffic flows while ensuring the security and reliability for communications. The O-CU is split into O-CU-UP and O-CU-CP:
 - **O-RAN Central Unit Control Plane (O-CU-CP):** The O-CU-CP serves as a link between the RAN and the AMF by utilizing the N2/NG-c interface. It implements the upper layers of the 5G New Radio (NR) protocol stack for the control plane [18]. It is responsible for the signalling and management tasks of communications.
 - **O-RAN Central Unit User Plane (O-CU-UP):** The O-CU-UP serves as a link between the RAN and the UPF by utilizing the N3/NG-u interface. It implements the upper layers of the 5G NR protocol stack for the user plane [18]. It is responsible for data forwarding between the end-user devices and the core network.
- **O-RAN Distributed Unit (O-DU):** The O-DU handles the processing of user data and control signalling within the RAN. It is responsible for the coding, modulation, and layer mapping.
- **O-RAN Radio Unit (O-RU):** The O-RU is deployed at the Edge of the RAN and performs time-domain functionalities for signal processing. O-RU is a key innovation when compared to traditional RAN approaches in 5G.

The O-RAN architecture considers two types of RIC, which are responsible for the dynamic control, management, and optimization of the capabilities of the RAN. By processing the data and utilizing AI and ML algorithms, the two RICs can determine and apply control policies and

actions on the RAN [10]. They are relevant for the management and orchestration of software applications. The two RICs are:

- **Non-Real Time RIC (Non-RT RIC):** Non-RT RIC optimises capabilities on a time scale larger than 1 second. It provides additional information for the Near Real-Time RIC to manage ML models and policy settings. Through the O1 interface, it receives information regarding the status of the network and sends policy updates to define the RANs' behaviour. Software applications (rApps) support the control and management of RAN resources [12]. The Non-RT RIC can control and manage multiple RANs.
- **Near-Real Time RIC (Near-RT RIC):** Near-RT RIC is deployed at the Edge of the network. Its main focus is to control and optimize the RAN by collecting different Key Performance Metrics (KPMs), which are used by software applications (xApps) to fine-tune RAN entities to a given scenario. It is considered near-real time, given that the time scale for its optimization capabilities is lower than 1 second.

The **SMO**, where the Non-RT RIC is placed, is the component that oversees all the orchestration aspects, management, and automation of RAN elements. It possesses several configurable interfaces, which rApps use to control one or multiple RANs. SMO guarantees the intended interoperability and vendor neutrality, one of the paradigms of O-RAN. Moreover, given its centralized control, it enables network slicing within the RAN, paving the way for easier escalation and efficient resource usage.

2.2.1.2 Interfaces

O-RAN specifies two types of interfaces: one is standard and coherent with the 5G architecture specified by 3GPP; the other is introduced by O-RAN. The standardized interfaces [26] include:

- **X2-c & X2-u:** The X2-c interface is used for communications between the RAN and eNBs in the control plane, while the X2-u is used for communications between the RAN and eNBs in the user plane. eNBs are the base stations in 4G.
- **Xn-c & Xn-u:** The Xn-c interface is used for communications between the RAN and gNBs in the control plane, while the Xn-u interface is used for communications between the RAN and gNBs in the user plane.
- **F1-c & F1-u:** The F1-c interface connects the O-CU-CP to the O-DU and is used for controlling the O-DU and its resources. The F1-u interface connects the O-CU-UP to the o-DU and is used to manage the exchange of user datagrams between the two entities. The F1-c implements the F1 Application Protocol (F1AP) to handle all the signalling messages related to the interface's setup, configuration, modification and release, assuring that the O-CU-CP is able to control de O-DU. SCTP is used to deliver the messages. The F1-u applies the GTP protocol to encapsulate data from the O-CU-UP to the O-DU and uses UDP.

- **E1:** The E1 interface connects the O-CU-CP and O-CU-UP, while enabling signalling messages.

This dissertation will not consider the X2-c and X2.u interfaces, as it is focused on using a standalone network.

O-RAN introduced other interfaces to allow for the openness and disaggregation of RAN elements [26], including:

- **E2:** The E2 interface connects the Near-RT RIC with the E2 nodes: the O-CU and the O-DU. E2 is used for forwarding telemetry data from these nodes to the Near-RT RIC and its responses with control actions and configurations. This capability enables the network to have control over the ongoing operations at the base station by leveraging multiple functionalities such as monitor, suspend, control, and override messages. In addition, the network can run commands issued by xApps and gather relevant data and metrics [12].
- **A1:** The A1 interface connects the Non-RT RIC to the Near-RT RIC. A1 supports the management and orchestration of ML models between both RICs and enables the provision of policies and operation guidance from the Non-RT RIC to the Near-RT RIC for optimized RAN performance.
- **O1:** The O1 interface is used to support the management and coordination of the entities by the SMO; it is related to the Fault, Configuration, Accounting, Performance, and Security (FCAPS) function, which consists of five working levels for network management.
- **O2:** The O2 interface connects the SMO and the ORAN O-Cloud, regulating the collection of services into two distinct logical groups. The first group comprises infrastructure management services, representing a subset of O2 functions specifically designed for deploying and managing a Cloud-based infrastructure. The second group comprises deployment management services, which encompass a subset of O2 functions responsible for ensuring the proper lifecycle management of deployments in the Cloud infrastructure [12].

2.2.2 Split Option 7.2x

Securing the O-CU and O-DU components has become a major challenge. To address it, a strategic approach proposed by O-RAN involves distributing the 5G NR protocol stacks across different components, emphasizing the separation between O-RU, O-DU, and O-CU and their functions.

The functional split proposed and employed in O-RAN, Split Option 7.2x, ensures that the O-DU and O-CU functions can be deployed with flexibility either centrally or at the edge, closer to the O-RU, based on specific latency and backhaul capacity needs. This approach provides a greater level of flexibility, making it easier to consider diverse deployment scenarios. Figure 2.13 represents the separation of the various functions of the 5G NR protocol stack considered in Split Option 7.2x.

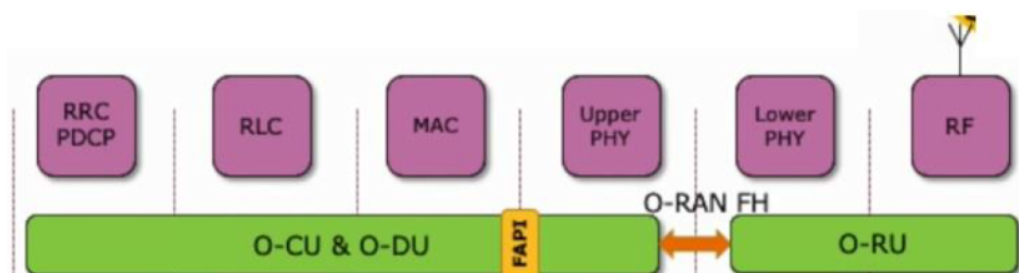


Figure 2.13: Split Option 7.2x with the functions of 5G NR [11].

Split Option 7.2x divides the physical layer functions between the O-RU and the O-DU via the Fronthaul Interface. This allows for the baseband functions to be split, reducing traffic transfers. Furthermore, this split can be configured as a packet-based interface within Ethernet, thereby taking advantage of the existing Ethernet networks. As a result, a better balance between the simplicity of the O-RU and the data rates and latency required on the Fronthaul Interface [10] can be achieved. Figure 2.14 depicts the split between the components and the main functionality of each.

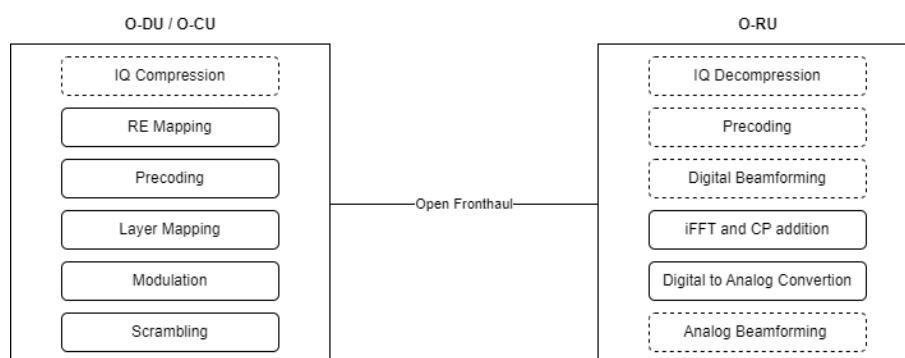


Figure 2.14: Split Option 7.2x. Dashed lines indicate optional functions. Adapted from [12].

2.2.3 Network Slicing

The O-RAN reference slicing architecture comprises slice management functions and O-RAN architectural components. As a general principle, O-RAN aims at ensuring compatibility with existing architectures. As such, the slice management functions are defined with extensions for O-RAN network functions, ensuring compatibility with the existing 3GPP architecture. Figure 2.15 depicts the various network slices over the O-RAN architecture.

The network slicing technique requires precise coordination between the O-RAN network functions and interfaces. An overview of the importance of the following O-RAN network functions and interfaces is presented in [14]. They include:

- **Non-RT RIC:** The Non-RT RIC should be aware of RAN slices and related slice properties/configurations, as well as slice KPMs to actively assist slice assurance through optimization methods, including AI/ML models.

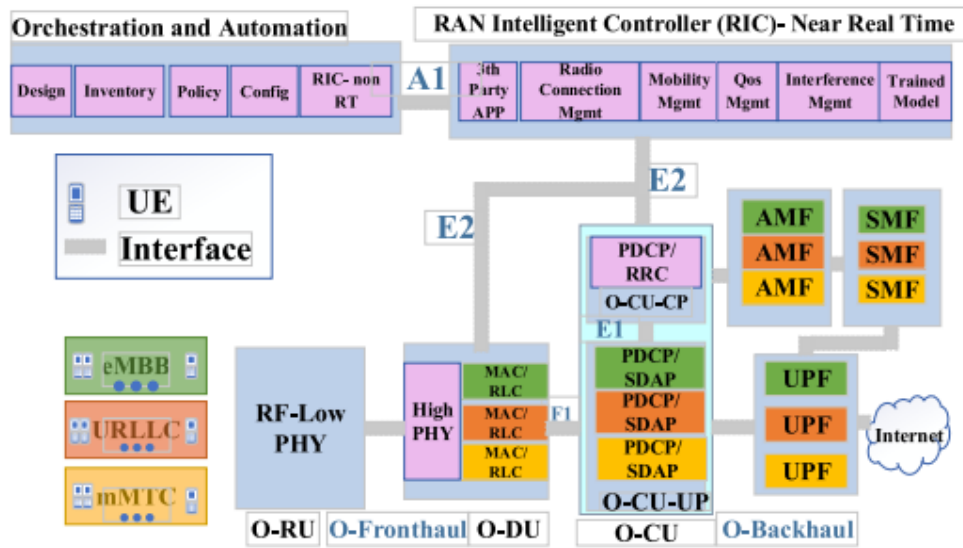


Figure 2.15: Network sliced O-RAN [13].

- **Near-RT RIC:** The Near-RT RIC enables xApps to control RAN elements and their resources through the E2 interface in near real-time. Similar to Non-RT RIC, Near-RT RIC xApps should be aware of RAN slices and can utilize slice-aware algorithms for slice assurance purposes. Slice-specific KPMs from E2 nodes are gathered and sent to Near-RT RIC and are used to apply dynamic slice optimization to ensure the desired performance.
- **O-CU:** The O-CU needs to execute slice-specific resource allocation strategies and drive the O-DU similarly. The O-CU interacts with Near-RT RIC to provide slice-specific KPMs as well as get dynamic guidance through the E2 interface. It is also responsible for the allocation of resources between network slices.
- **O-DU:** O-DU receives the slice configuration parameters from O1 interfaces and dynamic slice optimization guidance from the Near-RT RIC through the E2 interface. It provides slice-specific KPMs to the SMO through the O1 interface.
- **O-RAN interfaces:** The O-RAN architecture has several interfaces, each serving a distinct purpose for network slicing. The A1 interface allows providing policy-based guidance for slicing use cases, especially for Near-RT RIC, which can receive feedback related to deployed policies. The E2 interface is used to drive E2 nodes' slice configuration and behaviour. It enables slice-based radio resource allocations, scheduling policies, and configuration policies that may affect the performance enabled by each network slice. Near-RT RIC can receive and configure slice-specific performance data from the E2 nodes over this interface for near-RT optimization. Finally, the O1 interface's role is to enable slice-specific configuration of O-RAN nodes based on the service requirements. It is used to gather slice-specific KPMs and fault information from O-RAN Nodes. The O2 interface is used to manage the lifecycle of network slices.

According to the specifications of 3GPP, the Network Slice Management Function (NSMF) and Network Slice Subnet Management Function (NSSMF) are responsible for managing slices and slice subnets, respectively. In the context of O-RAN, the RAN slice subnet management function is the main focus, which includes interactions with the SMO [14]. O-RAN considers 4 types of deployment for the NSMF and the NSSMF:

- NSMF and NSSMF deployed within the SMO.
- NSMF and NSSMF deployed outside the SMO.
- NSMF deployed within the SMO, while NSSMF is deployed outside the SMO.
- NSSMF deployed within the SMO, while NSMF is deployed outside the SMO.

Figure 2.16 shows a representative example of network slicing within the CN (network slice subnet #1) and the RAN (network slice subnet #2)

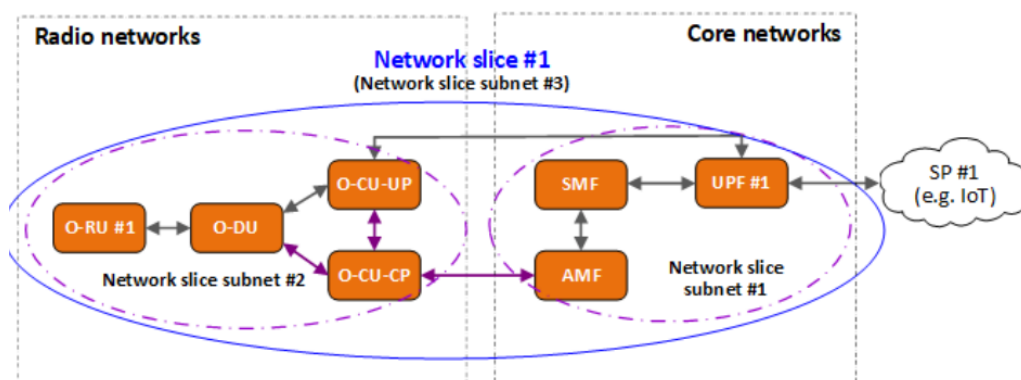


Figure 2.16: O-RAN Network Slicing Example [14].

2.2.4 Software Packages

For the deployment of 5G SA networks based on the O-RAN architecture, there are several open-source software packages. OpenAirInterface (OAI) is a representative example, widely used by the scientific community. OAI is an open-source software project that enables the development of a standalone 5G cellular network. It consists of two main components: 1) 5G NR RAN, which handles the communications between mobile devices and the Core network, including the gNB and UE software components, while being compliant with the O-RAN architecture; 2) 5G CN, which manages routing, switching, and signalling of data traffic within the network. OAI enables the management and deployment of xApps, supported by FlexRIC. An alternative software package is Software Radio Systems RAN (srsRAN). srsRAN deploys a fully functional RAN component that is compliant with the 5G and O-RAN architecture, but only implements the RAN. To consider a complete 5G SA network, the installation and configuration of different software packages are required, such as Open5GS as per their documentation [27] or the OAI 5G CN. Other software

packages were excluded from the dissertation scope due to insufficient documentation or separate implementation of RAN and CN.

2.3 Related Work

There are some works in the literature that address the implementation of private 5G SA networks and the usage of network slicing techniques in O-RAN-based 5G architectures networks.

In [18], the author implemented a 5G SA network based on the O-RAN architecture, taking advantage of the OAI software package while developing an xApp to control the mobility management function of a mobile robotic platform carrying a gNB.

During the OAI's Osaka Conference [28], the authors introduced a new solution combining OAI with FlexRIC. This solution monitors network delay and establishes a new DRB when a certain threshold is exceeded. The new DRB ensures that priority traffic is rerouted, thereby guaranteeing efficient processing of important packets within the network.

In [29], the authors implemented a 5G SA industrial testbed with an on-premises Edge design for connected Industry 4.0 applications. They detail the end-to-end testbed system design and discuss the main considerations for the industrial use cases, such as throughput, latency, and radio link budget. Additionally, they propose mobility and handover functions to enhance the desired parameters.

In [30], the authors present a detailed analysis of the effects and requirements of RAN slicing. The authors propose a framework for supporting and specifying RAN slices utilizing a set of configuration descriptors that define the characteristics, policies, and resources to be applied across the radio protocol layers of a next-generation RAN node.

In [31], the authors explain how 5G networks can facilitate the development of Industry 4.0 by ensuring precise latency requirements for typical industrial applications. Furthermore, they introduce a new latency-based RAN slice descriptor to improve the capacity of RAN slicing to meet the latency requirements of Industry 4.0 applications with deterministic periodic traffic.

The authors of [32] have proposed a new framework to enforce network slices featuring radio resource abstraction. This framework enhances the existing network-slicing solutions and is fully compliant with the 3GPP standardization. The authors have developed a fully programmable network-slicing architecture aligned with 3GPP and flexible RAN standards and introduced a two-level MAC scheduler to abstract and share the physical resources of the RAN among slices.

ProSLICE [33] is a platform designed to fully realize the complete O-RAN specification using only open-source components. It also facilitates network slicing in the RAN. A taxonomy has been devised for network slicing, considering different parameters, including key design principles, enablers, slicing resource levels, service function chaining schemes, physical infrastructures, and security aspects. The platform includes several enhancements and extensions for improved management and control of the network. ProSLICE also introduces novel xApps to allow for the creation of RAN network slices and the computation of statistics based on metrics retrieved from the RAN. It is complemented with a GUI for network slice configuration and visualization

of statistics. Despite ProSLICE being based on open-source components, the author's solution is not currently available to the public.

Currently, there are no solutions available to create private 5G SA networks and dynamically instantiate network slices that can adjust to the RAN resources and channel conditions. Although ProSLICE is the closest available solution, its limited openness motivates the development of an alternative solution. We believe there is potential in developing a private 5G SA network using solely open-source software packages while also implementing network slicing that dynamically adapts to various traffic profiles and radio channel conditions.

2.4 Summary

This chapter has delved into the foundational aspects of 5G networks and O-RAN, which are crucial within the context of this dissertation. 5G brought new capabilities that allow for lower latency and improved performance in data transfer. O-RAN promotes the interoperability of the equipment by opening the RAN components, making it more programmable and flexible. By integrating the technologies of 5G networks, such as network slicing, with the principles and architecture of the new O-RAN paradigm, it is possible to achieve a private SA network that can adjust itself to better meet the users' requirements.

Chapter 3

System Specification, Design and Implementation

This chapter outlines the system specification, design, and implementation considered in the development of the proposed solution. In Section 3.1, we discuss the 5G SA network specification and analyze the developed algorithms. Section 3.2 covers the complete design of the proposed solution, including hardware and software specifications. The chapter concludes in Section 3.3 with the proposed implementation of the 5G SA network and the xApp.

3.1 System Specification

The dissertation's main goal was to implement and validate a private 5G SA network based on the O-RAN architecture. Additionally, the dissertation aimed at developing a software-based mechanism for managing network slices using dynamic rules. To achieve this, the RAN should be able to reconfigure itself by adapting to channel and environmental conditions, ensuring an efficient connection between the UEs and the network.

Due to the current limitations of OAI, the management of network slices at the RAN level has not been fully developed and tested. As a result, network slice management will be handled through one of its enablers, the Radio Access Bearers (RABs). While RABs are different from network slices in theory, they can exhibit similar behaviour given the fact that management of RABs allows for a holistic approach to QoS, ensuring that the performance metrics are met across the entire network infrastructure. RABs act as enablers for network slicing by providing the necessary mechanisms for QoS management, resource allocation, isolation, and end-to-end performance consistency, tailoring the services to the real needs of its users.

In the proposed architecture, in Figure 3.1, the UEs need a reliable and stable connection to the external Data Network. In this way, firstly, the RAN, which is composed of a disaggregated gNB, that is, a CU and DU, must register itself in the CN. Additionally, the UEs also need to register themselves in the RAN

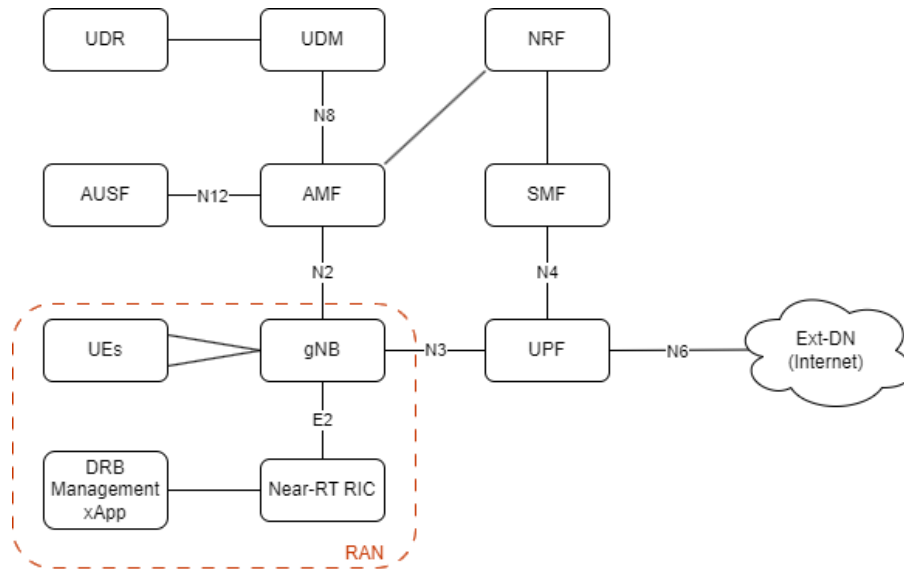


Figure 3.1: High-Level specification of the system.

After registering all the parties, the UEs communicate with the AMF via the N1 Interface to connect with the CN. Both entities exchange signalling messages for the UEs to authenticate themselves in the CN. However, the N1 Interface acts as a reference point for protocols in the 5G architecture since all communications between UEs and AMF are established, in fact, via the gNB through the N2 Interface. This interface is responsible for managing the connection between the RAN and the CN. It must be operational for the UE to initiate any connection to the 5G network, as described in Section 2.1.

To access external data networks, including the Internet, the RAN communicates with the UPF, which acts as a 5G gateway using the N3 interface, also described in Section 2.1. Communications between the two entities are crucial, given the fact that the UPF is responsible for classifying the traffic, enforcing QoS policies and inserting the IP streams in the resulting QoS flows.

Additionally, the UPF also communicates with the SMF through the N4 interface. This connection also has significant importance given the responsibility of the SMF to define the QoS policies and establish and modify the PDU sessions, setting appropriate QoS parameters for each. In this way, the configuration of resources the UPF uses for traffic mapping comes from the SMF via the N4 interface.

Our solution considers two main topics:

- **DRB Management xApp:** This original xApp should be able to continuously monitor the network's resources and trigger a reconfiguration of the RAN to create or eliminate Data Radio Bearers (DRBs).
- **Traffic Mapping to DRBs and QoS:** The RAN should be able to map the incoming traffic to a specific DRB considering multiple QoS parameters.

3.1.1 DRB Management xApp

As stated in Section 2.2, the O-RAN comprehends the use of xApps that enable the control of RAN elements and resources. The xApp uses the Near-RT RIC as specified in the O-RAN architecture to collect information about the network. This information is used by our xApp to determine whether to create a new Data Radio Bearer (DRB) or eliminate an existing one. The xApp communicates with the Near-RT RIC through the E42 interface, which is not standardized. The Near-RT RIC communicates with the gNB. In our solution, the gNB is disaggregated into O-CU and O-DU, using the E2 Interface and the E2AP.

O-RAN allows the collection of various metrics across all the layers of the 5G NR protocol stack. Additionally, it is possible to collect KPMs that transverse the entire network. Among the metrics that are fully implemented and working properly, the overall DL throughput of the connection and the delay packets suffer at Radio Link Control (RLC) - the sojourn time - are the ones most important for analysing the network's conditions.

Table 3.1 describes the notation used in the algorithm developed, which is present in Algorithm 1. Algorithm 1 was implemented as an xApp.

Table 3.1: Definition of the variables used by Algorithm 1.

Variable	Definition
$Throughput_{average}$	Running average throughput.
$Throughput_{threshold}$	Average Throughput. Level of throughput the network aims to maintain.
$SojournTime_{average}$	Running average packet sojourn time.
$SojournTime_{max-threshold}$	Checks if sojourn time is critically high Exceeding this value indicates severe network delays.
$SojournTime_{high-threshold}$	Intermediate alert level for rising network delays ($< SojournTime_{max-threshold}$).
$SojournTime_{min-threshold}$	Confirms if sojourn time is critically low, a sign of optimal network conditions with minimal delay.
$SojournTime_{low-threshold}$	Intermediate alert level for decreasing network delays ($> SojournTime_{min-threshold}$).
$control_period$	Predefined period after which network adjustments can be enforced.
t_{create_DRB}	Records the timestamp when an additional DRB is created, helping to manage the lifecycle of network resources.
t_{delete_DRB}	Records the timestamp when a DRB is deleted.
$t_{current}$	Current time. Used for determining the timing of network operations.

It is important to clarify that:

1. A default DRB is always present.
2. The algorithm can only generate one additional DRB.
3. The algorithm can only delete the newly created DRB, not the default DRB.

Algorithm 1 considers two thresholds for the maximum and minimum Sojourn Time. It is also important to note that:

$$\begin{cases} threshold_{max_sojourn_1} > threshold_{max_sojourn_2} \\ threshold_{min_sojourn_1} < threshold_{min_sojourn_2} \end{cases} \quad (3.1)$$

The pseudocode of the algorithm developed is presented in Algorithm 1. The algorithm relates the two metrics obtained from the monitoring of the network with the instructions to create or delete new DRBs. These instructions are communicated through the E2 interface.

Algorithm 1 DRB Management xApp.

```

if [SojournTimeaverage > SojournTimemax-threshold OR
    (SojournTimeaverage > SojournTimehigh-threshold AND
    Throughputaverage < Throughputthreshold)] AND
    tcurrent - tdelete_DRB > control_period then
    tcreate_DRB ← tcurrent
    Create new DRB                                ▷ Function that triggers the creation of a DRB
else if [SojournTimeaverage < SojournTimemin-threshold OR
    (SojournTimeaverage < SojournTimelow-threshold AND
    Throughputaverage > Throughputthreshold)] AND
    tcurrent - tcreate_DRB > control_period then
    tdelete_DRB ← tcurrent
    Delete DRB                                    ▷ Function that triggers the deletion of a DRB
end if

```

In Algorithm 1, several conditions are considered for creating and deleting a DRB. The first if clause checks if the average Sojourn Time of the last N measurements is greater than the first maximum threshold, or if the Sojourn Time, although not critically high, is accompanied by a low average Throughput. Both circumstances indicate network congestion or interference, which leads to QoS degradation. On the other hand, the else if clause checks if the average Sojourn Time of the last N measurements is less than the first minimum threshold and if the Sojourn Time, although not critically low, is accompanied by a high average throughput. This indicates that the network is not too congested, and QoS degradation is not severe, so an additional DRB is not required, reducing the time and overhead necessary to map the incoming traffic to the correct DRB.

Both clauses check that a certain amount of time has passed since the last operation, whether it was creating or deleting a DRB. This allows sufficient time for the RAN to reconfigure itself and redirect traffic to the proper DRB, eliminating this issue from consideration.

3.1.2 Traffic Mapping to DRBs and QoS

After the RAN is triggered to create and configure a new DRB, the RAN must map the traffic to the correct DRB. This mapping may take into consideration various criteria. Among them, only the type of traffic, whether Transmission Control Protocol (TCP), User Datagram Protocol (UDP) or Internet Control Message Protocol (ICMP) and the destination port numbers present in the packets' IP headers are used to map it to a DRB, as can be seen in Algorithm 2. The definitions of variables are presented in Table 3.2.

Table 3.2: Definition of the variables used by Algorithm 2.

Variable	Definition
<i>packet_header</i>	Header of an arrival packet
<i>destination_port_number</i>	Destination port number of a given packet
<i>wanted_port_number</i>	Port number to test
DRB	DRB to where the packet will go

Algorithm 2 Traffic Mapping to DRB.

```

if [packet_header = TCP AND destination_port_number ≠ wanted_port_number] then
    DRB ← 2                                ▷ Packet should go through DRB2
else if [packet_header = UDP AND destination_port_number ≠ wanted_port_number] then
    DRB ← 2                                ▷ Packet should go through DRB2
else if [packet_header = ICMP] then
    DRB ← 2                                ▷ Packet should go through DRB2
else
    DRB ← 1                                ▷ Packet should go through DRB1
end if

```

The QoS characteristics that are assigned to a given DRB must also be taken into account. Upon the generation of the DRB, a 5QI will be associated with it. The presence of a new DRB with different QoS characteristics must be propagated across the network and the 5G NR stack, ensuring that all network elements are aware of the new DRB configuration and can manage resources accordingly. Furthermore, the RAN must dynamically allocate and manage radio resources to meet the QoS requirements of all active DRBs. It is important to note that this algorithm does not fully comply with the 5G specifications. Although it may not be the ideal solution: 1) it does not require the reboot of the gNB; 2) it allows the RAN configuration to be adjusted according to varying conditions over time; 3) it is easily controllable, enabling changes to the type of traffic mapped to a given DRB and the associated quality of service; 4) it is implementable with OAI.

3.2 System Design

The system comprises three main units: The CN, the RAN (disaggregated gNB + UEs), and the Near-RT RIC. The architecture of the testbed developed to implement the proposed solution is illustrated in Figure 3.2.

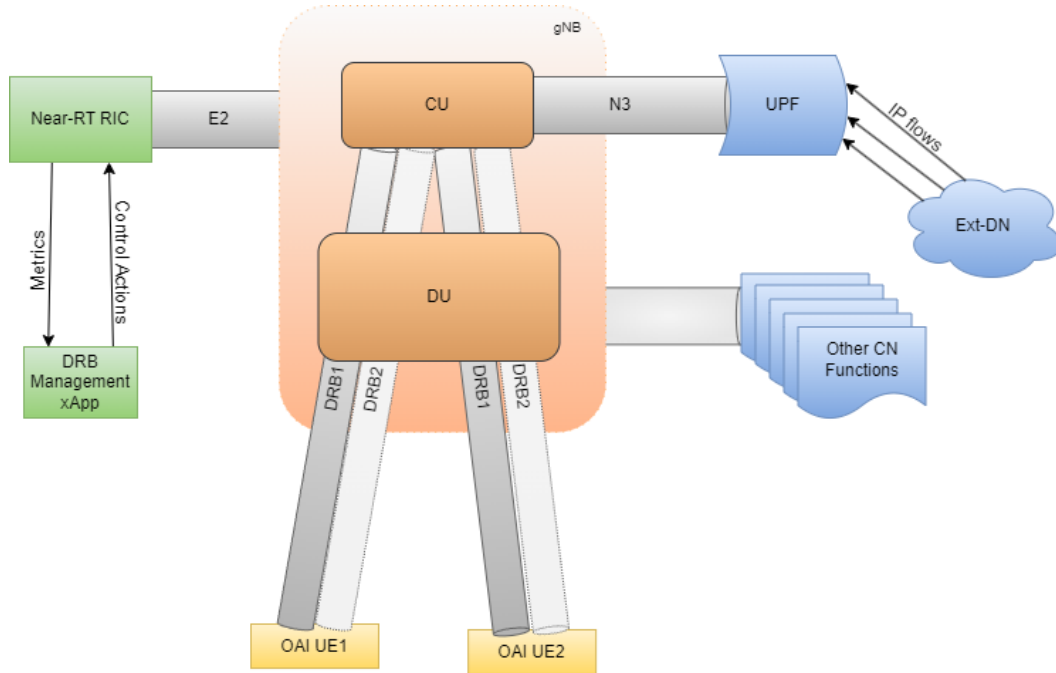


Figure 3.2: System architecture used to implement and evaluate the proposed solution.

The hardware and software packages used for implementing and deploying each unit are described in the following sections.

3.2.1 Hardware

Regarding the hardware, all the units were deployed in a single laptop, an Asus VivoBook F512D with 12 GB of RAM and 4 CPU cores. The Operating System employed was the Ubuntu 22.04 LTS.



Figure 3.3: Asus VivoBook F512D used for the system deployment.

The CN functions were deployed as Docker Containers with low computational demands, while the gNB deployment required more resources. The RAN, including the disaggregated gNB and multiple UEs, handles radio communications and signal processing, requiring significant computational power and memory, which presents challenges for evaluating the proposed solution. The Near-RT RIC, in contrast, needs relatively low computational power.

Deploying the entire network on a single laptop introduces challenges, including resource constraints and network latency due to all components running simultaneously. However, this setup simplifies the implementation and testing process, making it easier to troubleshoot and debug.

Despite not meeting the full system requirements for a 5G SA network as per [34], implementing it on a single laptop is adequate for testing and evaluating the proposed solution, offering a valuable proof of concept for a production environment.

3.2.2 Software Packages

Two different software packages were used to implement a 5G SA network, consisting of the Core Network (CN), the Radio Access Network (RAN), multiple User Equipment (UE), as well as the Near-RT RIC, which enables the operation of xApps.

3.2.2.1 5G SA Network

As discussed in Section 2.2, two primary open-source software packages enable the implementation of an O-RAN-based 5G architecture: OAI [35] and srsRAN [36]. OAI has the significant advantage of implementing both the 5G CN and the 5G RAN components required for a full 5G SA network deployment. In contrast, srsRAN necessitates an additional software package for the 5G CN, leading to potential challenges in interconnecting and interoperating these components. Furthermore, the srsRAN project does not include a UE implementation, having only been tested as a prototype with the srsRAN 4G project. Additionally, srsRAN cannot simulate multiple UE connections to the RAN.

For these reasons, the OAI software package was used in this dissertation to implement the CN [37] and the RAN [38]. OAI enables the development of a complete solution, allowing for its validation and refinement.

3.2.2.2 Near-RT RIC

Two main software packages implement the Near-RT RIC and enable the operation of xApps: the FlexRIC [39] and the O-RAN SC RIC [40]. FlexRIC introduces lower CPU usage and latency compared with O-RAN SC RIC. Moreover, FlexRIC is a software component developed side-by-side with OAI, which facilitates the integration and interoperability of both software packages. FlexRIC also implements several service models, which enable the retrieval of various metrics across all the 5G stacks, which are not present in O-RAN SC RIC.

As such, FlexRIC was used in this dissertation to implement the Near-RT RIC and the proposed xApps.

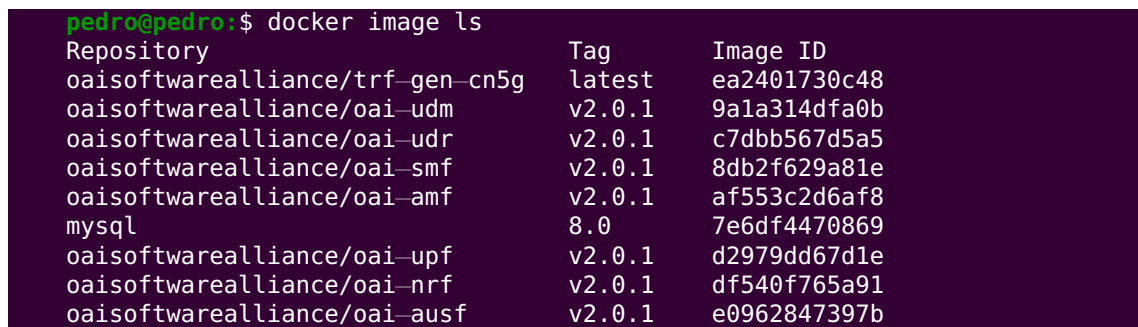
3.3 System Implementation

In this section, the implementation of the proposed solution is presented along all the three main components discussed in Section 3.2.

3.3.1 OAI 5G Core Network

For the 5G CN deployment, OAI provides some `docker-compose` files that implement the network functions as containers. We opted for a basic deployment with several Core Network functions: AMF, SMF, NRF, UPF, UDR, UDM, and AUSF. The purpose and responsibility of each function are discussed in Chapter 2. Additionally, OAI provides two different and non-standardized containers: the MySQL and the Ext-DN. The MySQL works as a server that stores information related to the UEs upon registration with the CN, such as the International Mobile Subscriber Identity (IMSI), the keys, and the Operator Code (OPc). In the context of OAI, the UEs that try to register in the CN must present and match this information with the one present on the MySQL container. Otherwise, the UEs are not able to authenticate properly. The Ext-DN is a non-standardized container deployed by the 5G CN, acting as a public server deployed in an external, such as the Internet. It is useful for generating traffic to and from the 5G network.

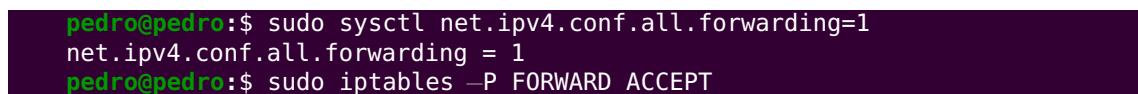
OAI provides docker images for each of the containers of the CN. The latest stable version is `V2.1.0`, which was used for pulling each image. Figure 3.4 shows that all the images for the containers that implement the CN functions were pulled correctly.



Repository	Tag	Image ID
oaisoftwarealliance/trf-gen-cn5g	latest	ea2401730c48
oaisoftwarealliance/oai-udm	v2.0.1	9a1a314dfa0b
oaisoftwarealliance/oai-udr	v2.0.1	c7dbb567d5a5
oaisoftwarealliance/oai-smf	v2.0.1	8db2f629a81e
oaisoftwarealliance/oai-amf	v2.0.1	af553c2d6af8
mysql	8.0	7e6df4470869
oaisoftwarealliance/oai-upf	v2.0.1	d2979dd67d1e
oaisoftwarealliance/oai-nrf	v2.0.1	df540f765a91
oaisoftwarealliance/oai-ausf	v2.0.1	e0962847397b

Figure 3.4: Docker images for the CN functions.

After this, it was necessary to change the IP forwarding rules and the Iptables forwarding policy, since by default, the deployed CN does not accept traffic from external networks. Figure 3.5 demonstrates the necessary changes needed.



```

pedro@pedro:~$ sudo sysctl net.ipv4.conf.all.forwarding=1
net.ipv4.conf.all.forwarding = 1
pedro@pedro:~$ sudo iptables -P FORWARD ACCEPT

```

Figure 3.5: Linux IP Forwarding and Iptables configuration.

The deployment of the 5G CN is achieved by using the `docker-compose-basic-nrf.yaml` file. This is responsible for deploying all the containers and creating all the interfaces needed for the communications between each network function, including the interface through which traffic flows to and from the gNB. For the gNB to be able to register itself in the CN properly, there are two main parameters to account for that relate to the Public Land Mobile Network (PLMN):

- **Mobile Country Code (MCC):** Identifies a country where the mobile network infrastructure operates.
- **Mobile Network Code (MNC):** Identifies a specific mobile network carrier or operator within a given country.

Another parameter is key for assuring the correct registration of the gNB in the CN, which is the **Tracking Area Code (TAC)** that identifies the tracking area where the gNB belongs to. A tracking area is a logical grouping of cells the network uses to manage and optimize mobility and signalling. It is important to note that multiple gNBs can be registered within a single CN. However, in this dissertation, a single gNB is considered. In this way, these values must match the ones the gNB presents upon registration. By default, OAI uses 208 for the MCC, 95 for the MNC, and a TAC of 0xa000. These values are defined in the AMF container, as shown in Figure 3.6.

```
amf:
  amf_name: "OAI-AMF"
  (...)
  plmn_support_list:
    - mcc: 208
    - mnc: 95
    - tac: 0xa000
```

Figure 3.6: Defined MCC, MNC, and TAC for the 5G CN.

After setting up IP forwarding, configuring Iptables, and ensuring that the MCC, MNC, and TAC values are valid, the 5G Core Network was launched correctly. The procedure involved generating all the containers representing the 5G CN functions and the `demo-oai` interface, which connects all network components and acts as the gateway for traffic to be sent and received from the gNB. Figure 3.7 represents the `demo-oai` interface configuration.

```
pedro@pedro:~$ ifconfig demo-oai
demo-oai: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.70.129 netmask 255.255.255.192 broadcast 192.168.70.191
inet6 fe80::42:57ff:fe9d:9ed1 prefixlen 64 scopeid 0x20<link>
ether 02:42:57:fd:9e:d1 txqueuelen 0 (Ethernet)
(...)
```

Figure 3.7: Configuration of the demo-oai interface.

Upon successfully launching the network, ensuring that all containers running the network functions are in a *healthy* state, and that `demo-oai` can ping each one, the OAI 5G CN was

functional. Table 3.3 includes the IP addresses of all the deployed Core Network functions. Figure 3.8 presents a diagram of the deployed 5G CN architecture.

Table 3.3: Core Network function and respective IP addresses.

Core Network Function	IP
NRF	192.168.70.130
MySQL Server	192.168.70.131
AMF	192.168.70.132
SMF	192.168.70.133
UPF	192.168.70.134
Ext-DN	192.168.70.135
UDR	192.168.70.136
UDM	192.168.70.137
AUSF	192.168.70.138

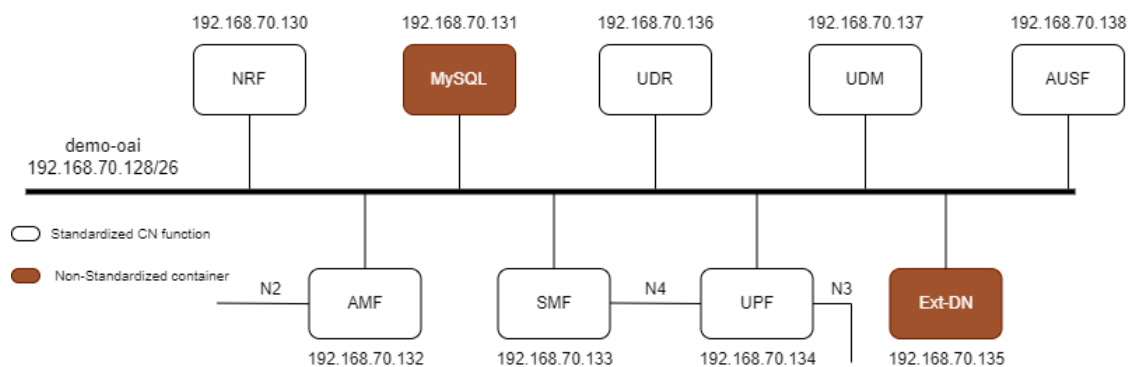


Figure 3.8: Deployed 5G Core network diagram.

In order to complete the setup of the 5G Core Network, the UEs need to be able to register themselves in the 5G CN. OAI offers predefined sets of valid UE parameters that can be accessed through the MySQL container. Figure 3.9 illustrates the required steps to access the predefined valid parameters, including IMSI, key, and OPc.

```

pedro@pedro:/OAI/oai-ran5g/cmake_targets$ docker exec -it mysql bash
bash-4.4# mysql -uroot -plinux
mysql> use oai_db
mysql> SELECT * FROM AuthenticationSubscription;
| ueid | (...) | encPermanentKey | (...) | encOpcKey | (...)
| 208950000000031 | (...) | 0C0A34601D4F07677303652C0462535B | (...) |
63bfa50ee6523365ff14c1f45f88737d | (...)
(...)

```

Figure 3.9: MySQL table with pre-defined valid parameters for UEs.

3.3.2 OAI 5G RAN

The RAN consists of the gNB and the UEs. Within the OAI, the implementation and evaluation of the proposed solution can be achieved using the specific branch called `qoe-e2` from the OAI repository [41].

After cloning the repository, it was necessary to build the entire project, which generates two executable files: one for the gNB and another for the UE, named `nr-softmodem` and `nr-uesoftmodem`, respectively. Figure 3.10 demonstrates the build command used.

```

pedro@pedro:/OAI/oai-ran5g/cmake_targets$ sudo ./build_oai -w SIMU --gNB
--nrUE --build-e2 --ninja -I
Will compile gNB
Will compile nrUE
rfsimulator nr-softmodem nr-uesoftmodem rfsimulator dfts compiled
BUILD SUCCESSFUL

```

Figure 3.10: Command to build the gNB and UE.

The build command includes several necessary flags. The `-I` installs all pre-requisites, and `-w SIMU` instructs the compiler to which device will be used for Radio Frequency (RF) purposes. In this dissertation, *rfsimulator* [42] replaced radio frequency hardware. The `-gNB` and `-nrUE` build the softmodem executable files. The `-build-e2` is required to build the RAN with an E2 agent, which is essential for working with FlexRIC.

3.3.2.1 OAI gNB

Regarding the OAI gNB, we opted for an approach with a disaggregated gNB, separating its functions in CU and DU, each one with its singular configuration file.

The communications between each other occur via the F1-c and F1-u, as defined in Section 2.2. Figure 3.11 illustrates the communications between the CU and DU and the necessary variables.

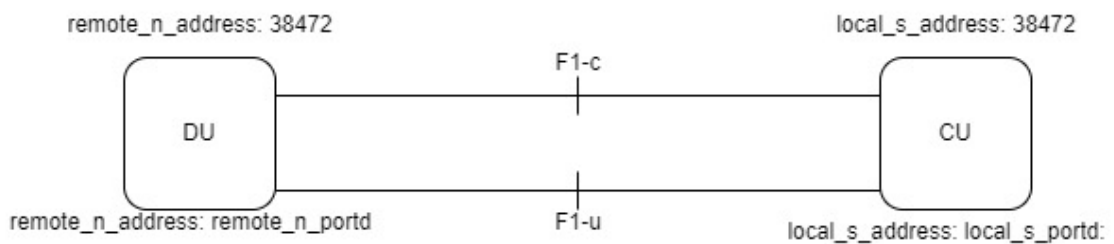


Figure 3.11: Communications between CU and DU.

To ensure communications between CU and DU using the *loopback interface*, their configuration files were adjusted to consider the right ports. Figures 3.12 and 3.13 depict both configuration files.

```
Active_gNBs = ( "gNB-Eurecom-CU");
(...)
gNB_DU_ID = 0xe00; #3584
gNB_name = "gNB-Eurecom-CU";
(...)
local_s_if_name = "lo";
local_s_address = "127.0.0.3";
```

Figure 3.12: CU configuration file for communication with DU.

```
Active_gNBs = ( "gNB-Eurecom-DU");
(...)
gNB_DU_ID = 0xe00; #3584
gNB_name = "gNB-Eurecom-DU";
(...)
local_n_if_name = "lo";
remote_n_address = "127.0.0.4";
```

Figure 3.13: DU configuration file for communication with CU.

The Wireshark log of Figure 3.14 allowed us to confirm that the CU initiates the communications with the DU by sending an F1 Setup Request. The F1 Setup Request message was sent by the CU to the DU to initiate the establishment of the F1 interface. Both components exchange information about cell configuration and setup communications parameters. The F1 setup was confirmed by the DU through an F1 Setup Response.

64	5.393665225	127.0.0.1	127.0.0.4	F1AP	282	538 F1SetupRequest, MIB, SIB1
66	5.394020942	127.0.0.4	127.0.0.1	F1AP	130	668 F1SetupResponse


```
Frame 64: 282 bytes on wire (2256 bits), 282 bytes captured (2256 bits) on interface lo, id 1
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.4
Stream Control Transmission Protocol, Src Port: 48001 (48001), Dst Port: 38472 (38472)
F1 Application Protocol
  F1AP-PDU: initiatingMessage (0)
    initiatingMessage
      procedureCode: id-F1Setup (1)
      criticality: reject (0)
      value
        F1SetupRequest
          protocolIEs: 5 items
            Item 0: id-TransactionID
            Item 1: id-gNB-DU-ID
              ProtocolIE-Field
                id: id-gNB-DU-ID (42)
                criticality: reject (0)
                value
                  GNB-DU-ID: 3584
            Item 2: id-gNB-DU-Name
              ProtocolIE-Field
                id: id-gNB-DU-Name (45)
                criticality: ignore (1)
                value
                  GNB-DU-Name: gNB-Eurecom-DU
            Item 3: id-gNB-DU-Served-Cells-List
            Item 4: id-gNB-DU-RRC-Version
```

Figure 3.14: Communication between CU and DU.

Having ascertained that CU and DU can communicate with one another properly, it was essential to guarantee that the gNB is capable of communicating with the 5G CN. To this end, the MCC, MNC, and TAC values must match the ones the AMF expects: 208 for MCC, 95 for MNC, and 0xa000 for TAC. Since the disaggregated gNB and the 5G Core Network are running on the same computer, the CU must know the IP address of the AMF and configure the IP the gNB should use to communicate with the AMF and UPF. Figure 3.15 demonstrates the IP address scheme for AMF and for the gNB to access CN functions through the defined interface. It also depicts the configuration of MCC, MNC, and TAC values.

```
gNBs =
{
    ////////// Identification parameters:
    gNB_ID = 0xe00;
    gNB_name = "gNB-Eurecom-CU";

    tracking_area_code = 0xa000;
    plmn_list = ({ mcc = 208; mnc = 95; mnc_length = 2;
    snssaiList = ({ sst = 1 }) });

    nr_cellid = 12345678L;
    ////////// AMF parameters:
    amf_ip_address = ( { ipv4 = "192.168.70.132";
    (...)})

    NETWORK_INTERFACES :
    {
        GNB_INTERFACE_NAME_FOR_NG_AMF = "demo-oai";
        GNB_IPV4_ADDRESS_FOR_NG_AMF = "192.168.70.129/26";
        GNB_INTERFACE_NAME_FOR_NGU = "demo-oai";
        GNB_IPV4_ADDRESS_FOR_NGU = "192.168.70.129/26";
        GNB_PORT_FOR_S1U = 2152;
    };
};
```

Figure 3.15: Defined MCC, MNC and TAC, as well as the AMF IP address, and gNB IP and interfaces in the CU configuration file

Although the AMF IP address, and gNB IP and interfaces should only be configured in the CU, the MCC, MNC and TAC values should also be defined in the DU, which is shown in Figure 3.16.

```
gNBs =
{
    ////////// Identification parameters:
    gNB_ID = 0xe00;
    gNB_DU_ID = 0xe00;
    gNB_name = "gNB-Eurecom-DU";

    tracking_area_code = 0xa000;
    plmn_list = ({ mcc = 208; mnc = 95; mnc_length = 2;
    snssaiList = ({ sst = 1 }) });
    nr_cellid = 12345678L;
```

Figure 3.16: Defined MCC, MNC and TAC in the DU configuration file

The gNB communicates with the AMF via the N2 interface and with the UPF via the N3 interface. In the case of disaggregated gNB, this task falls to the CU. Figure 3.17 depicts a diagram of the communications interfaces, including the variables used for IP addressing in Figure 3.15.

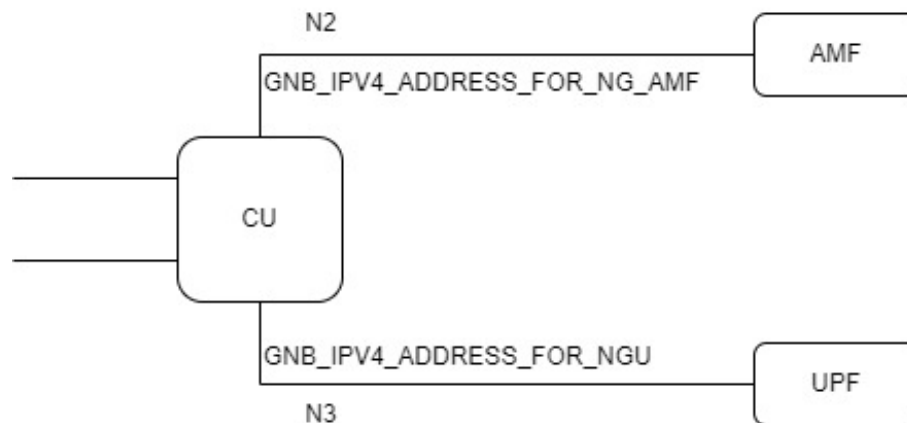


Figure 3.17: Communications interfaces between CU and both AMF and UPF.

The gNB tries to establish a connection with the AMF by sending an NGAP Setup Request, as shown in Figure 3.18. The AMF acknowledges and validates the connection by sending a NGAP Setup Response.

31	4.202851284	192.168.70.129	192.168.70.132	NGAP	126	3149 NGSetupRequest
32	4.202871261	192.168.70.132	192.168.70.129	SCTP	62	3211 SACK
33	4.203691500	192.168.70.132	192.168.70.129	NGAP	130	3341 NGSetupResponse


```

Stream Control Transmission Protocol, Src Port: 58893 (58893), Dst Port: 38412 (38412)
NG Application Protocol
  NGAP-PDU: initiatingMessage (0)
    initiatingMessage
      procedureCode: id-NGSetup (21)
      criticality: reject (0)
      value
        NGSetupRequest
          protocolIEs: 4 items
            Item 0: id-GlobalRANNodeID
              ProtocolIE-Field
                id: id-GlobalRANNodeID (27)
                criticality: reject (0)
                value
                  GlobalRANNodeID: globalGNB-ID (0)
                    globalGNB-ID
                      pLMNIdentity: 02f859
                        Mobile Country Code (MCC): France (208)
                        Mobile Network Code (MNC): Orange (95)
                      gNB-ID: gNB-ID (0)
                        gNB-ID: 0000e000 [bit length 28, 4 LSB pad bits, 0000 0000 0000 0000 1110 0000 0000 ... decimal value 3584]
            Item 1: id-RANNodeName
            Item 2: id-SupportedTAList
              ProtocolIE-Field
                id: id-SupportedTAList (102)
                criticality: reject (0)
                value
                  SupportedTAList: 1 item
                    Item 0
                      SupportedTAItem
                        tAC: 40960 (0x00a000)
  
```

Figure 3.18: NGAP setup Request and response by the gNB and AMF to validate the connection.

After assuring the CU and DU are able to communicate with each other and that the connection with the 5G CN is established and working properly, the UEs could initiate a connection to the gNB.

3.3.2.2 OAI 5G UE

To initiate a 5G UE within the OAI framework, the `nr-uesoftmodem` executable is used. Upon successfully connecting a UE to the 5G network, an IP address was assigned to the UE, enabling connectivity. However, OAI does not support multiple UEs simultaneously, as it is programmed to assign only one IP address per UE instance. To address this constraint, OAI has developed a script capable of generating multiple network namespaces, each dedicated to an individual UE. This solution enabled the assignment of distinct IP addresses to each UE, thus overcoming the limitation of single IP address allocation. The conceptual framework of this multi-UE implementation strategy within OAI is depicted in Figure 3.19.

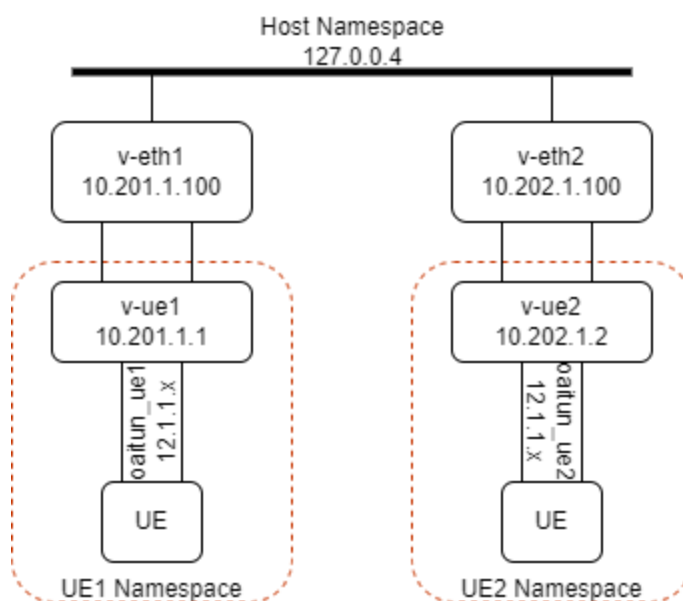


Figure 3.19: Diagram of the multi UE implementation.

In order to facilitate UEs to authenticate and register in the 5G CN, it was essential to ensure that their configurations were accurately defined. This entails verifying that their International Mobile Subscriber Identity (IMSI), key, and OPc correspond to the pre-defined parameters for the UEs stored in the MySQL container database. This verification could be accomplished by creating a new entry in the database with different, valid parameters, or by selecting known valid parameters directly from the database. The latter approach was selected due to its simplicity. For further clarity, Figure 3.20 illustrates the configurations for both UEs.

The IMSI component of the UE consists of a combination of the MCC, MNC, and Mobile Subscription Identification Number (MSIN). By default, the MCC is 208, and the MNC is 95. Therefore, the MSIN value for the first UE is 0000000031 and for the second UE is 0000000032. Figure 3.21 depicts a Wireshark log that provides conclusive evidence that the UE successfully registered and authenticated in the 5G CN.

```
#UE1
uicc0 = {
imsi = "208950000000031";
key =
"0c0a34601d4f07677303652c0462535b";
opc =
"63BFA50EE6523365FF14C1F45F88737D";
(...)
}

#UE2
uicc0 = {
imsi = "208950000000032";
key =
"0c0a34601d4f07677303652c0462535b";
opc =
"63BFA50EE6523365FF14C1F45F88737D";
(...)
}
```

Figure 3.20: UE configuration for IMSI, key, and OPc.

313 20.302681397 192.168.70.129	192.168.70.132	NGAP/N...	150	406 InitialUEMessage, Registration request
402 20.540749902 192.168.70.132	192.168.70.129	NGAP/N...	130	536 DownlinkNASTransport, Authentication request
403 20.570459188 192.168.70.129	192.168.70.132	NGAP/N...	142	678 UplinkNASTransport, Authentication response
490 20.686783677 192.168.70.132	192.168.70.129	NGAP/N...	126	804 DownlinkNASTransport, Security mode command


```

RAN-UE-NGAP-ID: 1
  Item 1: id-NAS-PDU
    ProtocolIE-Field
      id: id-NAS-PDU (38)
      criticality: reject (0)
      value
        NAS-PDU: 7e004119000d0102f859000000000000000132e0880200000000000000
          Non-Access-Stratum 5GS (NAS)PDU
            Plain NAS 5GS Message
              Extended protocol discriminator: 5G mobility management messages (126)
              0000 .... = Spare Half Octet: 0
              .... 0000 = Security header type: Plain NAS message, not security protected (0)
              Message type: Registration request (0x41)
              5GS registration type
                .... 1... = Follow-On Request bit (FOR): Follow-on request pending
                .... .001 = 5GS registration type: initial registration (1)
              NAS key set identifier
                0... .... = Type of security context flag (TSC): Native security context (for KSIAMF)
                .001 .... = NAS key set identifier: 1
              5GS mobile identity
                Length: 13
                0... .... = Spare: 0
                .000 .... = SUPI format: IMSI (0)
                .... 0... = Spare: 0
                .... .001 = Type of identity: SUCI (1)
                Mobile Country Code (MCC): France (208)
                Mobile Network Code (MNC): Orange (95)
                Routing indicator: 0000
                .... 0000 = Protection scheme Id: NULL scheme (0)
                Home network public key identifier: 0
                MSIN: 0000000031

```

Figure 3.21: Registration of the UE in the 5G CN.

3.3.2.3 FlexRIC

The FlexRIC application, selected to implement the NearRT-RIC, is an executable program in the operating system. The required version of FlexRIC for implementing and assessing the proposed solution is e2a6 as the gNB repository qoe-e2 already assumes this module for the E2 Agent. After cloning the repository, the project was built, and the E2AP and KPM versions were set to v2.0.3.

Furthermore, the IP address of the FlexRIC was defined in the CU and the DU so that they can connect to and use the E2 Agent. This IP must match the one configured directly in the FlexRIC configuration file. For simplicity, since the implementation considered only a computer, the IP was set to the computer's loopback interface, as depicted in Figure 3.22.

It was important to inform the CU and DU about the directory for the service models implemented by FlexRIC so that they could be used by the proposed solution. These service models are

```
e2_agent = {
  near_ric_ip_addr = "127.0.0.1";
  sm_dir = "/usr/local/lib/flexric/"
}
```

```
[NEAR-RIC]
(...)
NEAR_RIC_IP = 127.0.0.1
```

Figure 3.22: FlexRIC IP address and directory to FlexRIC's service models.

crucial for retrieving the required information considered in the scope of the dissertation. Figure 3.23 shows FlexRIC correctly loading one of eight service models.

```
187 11.875631512 127.0.0.1 127.0.0.1 E2AP 776 COOKIE_ECHO
Frame 187: 776 bytes on wire (6160 bits), 776 bytes captured (6160 bits) on interface lo, id 1
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Stream Control Transmission Protocol, Src Port: 45383 (45383), Dst Port: 36422 (36422)
E2 Application Protocol
  E2AP-PDU: initiatingMessage (0)
    initiatingMessage
      procedureCode: id-RICsubscriptionModification (14)
      criticality: reject (0)
      value
        RICsubscriptionModificationRequest
          protocolIEs: 1 item
          Item 0: id-RANfunctionsAdded
            ProtocolIE-Field
              id: id-RANfunctionsAdded (10)
              criticality: reject (0)
              value
                RANfunctions-List: 8 items
                Item 0: id-RANfunction-Item
                  ProtocolIE-SingleContainer
                    id: id-RANfunction-Item (8)
                    criticality: reject (0)
                    value
                      RANfunction-Item
                        ranFunctionID: 2
                        ranFunctionDefinition: 00304f52414e2d4532534d2d4b504d000018312e332e362e312e342e312e35333134382e312e322e322e3205004b504d204d6f6e69746f72
                        E2SM-KPM-RANfunction-Description
                        ranFunctionRevision: 2
                        ranFunctionOID: 1.3.6.1.4.1.53148.1.2.2.2
```

Figure 3.23: FlexRIC service models.

The metrics collected include the overall DL throughput and the sojourn time obtained from the KPM service model. OAI defines the sojourn time as the delay experienced by packets at the RLC layer, encompassing the time from their arrival through processing and transmission, but does not include the impact of retransmissions. In our proposed solution, a minor modification was necessary to ensure that the system takes retransmissions into account when calculating the sojourn time metric.

3.4 Summary

The system designed and implemented to evaluate the proposed solution is composed of three key components: the 5G Core Network, a disaggregated gNB connected to two UEs, and the Near-RT RIC. These components were integrated within a single computational environment to emulate a full 5G SA network. Ensuring that all components are able to communicate with each other and enabling connectivity between the gNB and both the 5G CN and the UEs was crucial, with configuration validations detailed in Chapter 3. Additionally, to monitor network resource use and adjust the RAN, we incorporated an original DRB Management xApp for improved performance. By creating additional DRBs in conditions of network congestion and mapping specific traffic to them, we can prioritize packets and avoid QoS degradation. thereby enhancing the user's QoE.

Chapter 4

System Validation

This chapter is dedicated to validating the solution proposed in this dissertation. Section 4.1 describes the methodology employed to validate the proposed solution. Section 4.2 focuses on testing each component. Section 4.3 assesses the performance and interoperability of the developed 5G SA network solution. Section 4.4 presents the solution validation, considering real-world use cases. Section 4.5 concludes the chapter with an analysis and discussion of the accomplished results.

4.1 Methodology

To correctly validate the system, a first set of tests was performed to ensure that each individual component was operating properly and as expected. Subsequently, additional tests were developed to assess the interoperability of the components. It was important to conduct these tests in the correct order to guarantee that all components could communicate with each other adequately. First, the 5G Core Network was deployed, followed by FlexRIC. After this, the CU and DU were also deployed, succeeded by the UEs. Finally, the DRB Management xApp was executed. The last testing batch focused on real-world use cases, where the proposed solution was evaluated for functionality and performance. Regarding the use cases, two different scenarios were considered: in the first scenario, only one UE connected triggered the DRB Management xApp, while in the second scenario, two UEs were connected, and both triggered the DRB Management xApp.

The methodology testing considers the goal and configuration implemented for each test, proceeding with functionality verification through the execution of ping commands and further analysis employing Wireshark. Ultimately, where relevant, the network was assessed for its capacity and performance using the iPerf3 [43] tool.

4.2 Component Testing

The component testing followed the same order stated in Section 4.1 to ensure that every stage was properly set.

4.2.1 5G Core Network

The 5G Core Network was tested to evaluate the state of the Docker containers that compose all the network functions and assess their connectivity. The RAN would not connect properly if all the components comprising the 5G CN were not in a *healthy* state and could not communicate with each other. The deployment of the OAI 5G Core Network was performed using containers. The command shown in Figure 4.1 ensures the containers are deployed and checks their status

```

pedro@pedro:/OAI/oai-ran5g/cmake_targets$ python3 core-network.py --type
start-basic --scenario 1
(...)
Name                                State                                Ports
MySQL                               Up (healthy)                        3306/tcp, 33060/tcp
oai-amf                              Up (healthy)                        38142/sctp, 80/tcp, 8080/tcp
oai-ausf                              Up (healthy)                        80/tcp, 8080/tcp
oai-ext-dn                           Up (healthy)
oai-nrf                              Up (healthy)                        80/tcp, 8080/tcp
oat-smf                              Up (healthy)                        80/tcp, 8080/tcp, 8805/udp
oat-udm                              Up (healthy)                        80/tcp, 8080/tcp
oat-udr                              Up (healthy)                        80/tcp, 8080/tcp
oat-upf                              Up (healthy)                        2152/udp, 8085/udp
(...)
root:DEBUG: OAI 5G Core network is configured and healthy...

```

Figure 4.1: Deploying the OAI 5G Core Network.

All the network functions, deployed as Docker containers, were in a *healthy* state. Moreover, ensuring all containers had IP-based connectivity was necessary to guarantee communications with the rest of the network. To this end, we used the *ping* command from the CN interface to every CN function, according to Figure 3.3, and with the results in Figure 4.2.

```

pedro@pedro:$ ping -I demo-oai 192.168.70.130
64 bytes from 192.168.70.130: icmp_seq=1 ttl=64 time=0.098 ms
pedro@pedro:$ ping -I demo-oai 192.168.70.131
64 bytes from 192.168.70.131: icmp_seq=1 ttl=64 time=0.157 ms
pedro@pedro:$ ping -I demo-oai 192.168.70.132
64 bytes from 192.168.70.132: icmp_seq=1 ttl=64 time=0.102 ms
pedro@pedro:$ ping -I demo-oai 192.168.70.133
64 bytes from 192.168.70.133: icmp_seq=1 ttl=64 time=0.063 ms
pedro@pedro:$ ping -I demo-oai 192.168.70.134
64 bytes from 192.168.70.134: icmp_seq=1 ttl=64 time=0.087 ms
pedro@pedro:$ ping -I demo-oai 192.168.70.135
64 bytes from 192.168.70.135: icmp_seq=1 ttl=64 time=0.090 ms
pedro@pedro:$ ping -I demo-oai 192.168.70.136
64 bytes from 192.168.70.136: icmp_seq=1 ttl=64 time=0.116 ms
pedro@pedro:$ ping -I demo-oai 192.168.70.137
64 bytes from 192.168.70.137: icmp_seq=1 ttl=64 time=0.074 ms
pedro@pedro:$ ping -I demo-oai 192.168.70.138
64 bytes from 192.168.70.138: icmp_seq=1 ttl=64 time=0.059 ms

```

Figure 4.2: Pinging all 5G CN functions to assess connectivity.

Since all the containers were able to communicate with the `demo-oai` interface, we concluded that the 5G Core Network was working as expected and fully operational. By looking at the Wireshark log in Figure 4.3, we could see 5G CN functions exchanging information with each other, including messages exchanged with the NRF to discover other network functions and update their status.

3	0.000054473	192.168.70.136	192.168.70.130	TCP	66	0	214	51120 → 8080 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2203366037 TSecr=2198506678
4	0.000129133	192.168.70.136	192.168.70.130	HTTP2	98	24	304	Magic
5	0.000145383	192.168.70.130	192.168.70.136	TCP	66	0	370	8080 → 51120 [ACK] Seq=1 Ack=25 Win=65536 Len=0 TSval=2198506678 TSecr=2203366037
6	0.000166623	192.168.70.136	192.168.70.130	HTTP2	93	27	463	SETTINGS[0]
7	0.000175780	192.168.70.130	192.168.70.136	HTTP2	81	15	544	SETTINGS[0]
8	0.000181351	192.168.70.136	192.168.70.130	HTTP2	79	13	623	WINDOW_UPDATE[0]
9	0.000193533	192.168.70.136	192.168.70.130	TCP	66	0	689	51120 → 8080 [ACK] Seq=65 Ack=16 Win=65536 Len=0 TSval=2203366037 TSecr=2198506678
10	0.000203131	192.168.70.130	192.168.70.136	TCP	66	0	755	8080 → 51120 [ACK] Seq=16 Ack=65 Win=65536 Len=0 TSval=2198506678 TSecr=2203366037
11	0.000236985	192.168.70.130	192.168.70.136	HTTP2	75	9	830	SETTINGS[0]
12	0.000239309	192.168.70.136	192.168.70.130	HTTP2	164	98	994	HEADERS[1]: PATCH /nnrf-nfm/v1/nf-instances/c594b16e-ddd2-43ea-ba40-7852d2fe7bbb
13	0.000286137	192.168.70.136	192.168.70.130	HTTP2	75	9	1069	SETTINGS[0]
14	0.000315703	192.168.70.130	192.168.70.136	HTTP2/_	133	67	1202	DATA[1], JavaScript Object Notation (application/json)
15	0.000322014	192.168.70.130	192.168.70.136	TCP	66	0	1268	8080 → 51120 [ACK] Seq=25 Ack=172 Win=65536 Len=0 TSval=2198506679 TSecr=2203366037
16	0.000479189	192.168.70.130	192.168.70.136	HTTP2	170	104	1438	HEADERS[1]: 204 No Content
17	0.000594466	192.168.70.136	192.168.70.130	TCP	66	0	1504	51120 → 8080 [FIN, ACK] Seq=239 Ack=129 Win=65536 Len=0 TSval=2203366038 TSecr=2198506678
18	0.000638348	192.168.70.130	192.168.70.136	TCP	66	0	1570	8080 → 51120 [FIN, ACK] Seq=129 Ack=240 Win=65536 Len=0 TSval=2198506679 TSecr=2203366037
19	0.000664096	192.168.70.136	192.168.70.130	TCP	66	0	1636	51120 → 8080 [ACK] Seq=240 Ack=130 Win=65536 Len=0 TSval=2203366038 TSecr=2198506679
20	0.228031466	192.168.70.133	192.168.70.134	PFPC	58		1694	PFPC Heartbeat Request
21	0.228230419	192.168.70.134	192.168.70.133	PFPC	58		1752	PFPC Heartbeat Response
22	1.592229822	192.168.70.133	192.168.70.130	TCP	132	66	1884	48120 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=501 Len=66 TSval=741925168 TSecr=1444439771
23	1.592227312	192.168.70.130	192.168.70.133	TCP	133	67	2017	48120 → 8080 [PSH, ACK] Seq=67 Ack=1 Win=501 Len=67 TSval=741925168 TSecr=1444439771

Frame 15: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface demo-oai, id 0
 Ethernet II, Src: 02:42:c0:a8:46:82 (02:42:c0:a8:46:82), Dst: 02:42:c0:a8:46:88 (02:42:c0:a8:46:88)
 Internet Protocol Version 4, Src: 192.168.70.130, Dst: 192.168.70.136
 Transmission Control Protocol, Src Port: 8080, Dst Port: 51120, Seq: 25, Ack: 172, Len: 0

Figure 4.3: 5G Core Network functions exchanging information.

4.2.2 FlexRIC

The deployment of FLeXRIC was performed by running the application generated by the project's build process. Upon the program's execution, it first confirmed and acknowledged the configurations set in the configuration file and loaded the service models. After that, it remained on standby, awaiting E2 messages from the gNB. Figure 4.4 shows the execution of the program and the loading of the service models.

```

pedro@pedro:/OAI/flexric/build/$ sudo ./examples/ric/nearRT-RIC
Setting the config -c file to /usr/local/etc/flexric/flexric.conf
Setting path -p for the shared libraries to /usr/local/lib/flexric/
[NEAR-RIC]: nearRT-RIC IP Address = 127.0.0.1, PORT = 36421
[NEAR-RIC]: Loading SM ID = 2 with def = ORAN-E2SM-KPM
[NEAR-RIC]: Loading SM ID = 3 with def = ORAN-E2SM-RC
[NEAR-RIC]: Loading SM ID = 142 with def = MAC_STATS_V0
[NEAR-RIC]: Loading SM ID = 143 with def = RLC_STATS_V0
[NEAR-RIC]: Loading SM ID = 144 with def = PDCP_STATS_V0
[NEAR-RIC]: Loading SM ID = 145 with def = SLICE_STATS_V0
[NEAR-RIC]: Loading SM ID = 146 with def = TC_STATS_V0
[NEAR-RIC]: Loading SM ID = 148 with def = GTP_STATS_V0
[NEAR-RIC]: Initializing

```

Figure 4.4: Execution of FlexRIC program.

The correct execution of the program allowed us to conclude that the FlexRIC application was fully operational.

4.2.3 Disaggregated gNB

As stated in Section 3.3, the proposed solution considers a case of a disaggregated gNB in CU and DU. In this way, both components were executed in a different way.

4.2.3.1 CU

The CU was initialized using the `nr-softmodem` program with the CU configuration file. The `-sa` ensured the execution was in standalone mode. Figure 4.5 depicts the command for execution.

```
pedro@pedro:/0AI/oai-ran5g/cmake_target/ran_build/build/$ sudo ./nr-softmodem -0 ../../../../targets/PROJECTS/GENERIC-NR-5GC/CONF/cu_gnb.conf --sa
```

Figure 4.5: Initialization of the CU.

After successfully launching the CU, the CU waits for the connection with the DU.

4.2.3.2 DU

The DU was initialized using the `nr-softmodem` program with the DU configuration file. The `-sa` option ensured the execution was in standalone mode, and the `-rfsim` option provided information that the program should run in emulation mode with *rfsimulator*. Figure 4.6 depicts the command for execution.

```
pedro@pedro:/0AI/oai-ran5g/cmake_target/ran_build/build/$ sudo ./nr-softmodem -0 ../../../../targets/PROJECTS/GENERIC-NR-5GC/CONF/du_gnb.conf --rfsim --sa
```

Figure 4.6: Initialization of the DU.

After successfully launching the DU, the CU and DU connected to each other, configuring the F1 interface. Figure 4.7 shows the CU log file. It confirmed that CU and DU were connected, given the fact that the CU was aware of the DU name and ID and configured the F1AP accordingly, sending the F1 Setup Request and receiving the F1 Setup Response.

```
[F1AP] CU_handle_F1_SETUP_REQUEST
[F1AP] req->transaction_id 1
[F1AP] req->gNB_DU_id 3584
[F1AP] req->gNB_DU_name gNB-Eurecom-DU
[F1AP] req->num_cells_available 1
[NR_RRC] Received F1 Setup Request from gNB_DU 3584 (gNB-Eurecom-DU) on
assoc_id 6
[RRR] Accepting DU 3584 (gNB-Eurecom-DU), sending F1 Setup Response
```

Figure 4.7: Connection between the CU and DU.

4.2.4 UE

As stated in Section 3.3, since we were employing multiple UEs, we created different namespaces, one for each UE.

4.2.4.1 UE1

Figure 4.8 presents the commands used to launch UE1. First, we ran the script that created the namespace and the links between the Core Network interface and the namespace itself. After that, we executed the `nr-uesoftmodem` program that launched the UE. Since we were using the *rfsimulator* to emulate radio link connections between components, it was necessary to inform the IP of the *rfsimulator* in the namespace. Additionally, the provision of the UE configuration file was necessary, as well as indicating the frequency and band used. The flags to inform about the standalone and emulation through *rfsimulator* also had to be present to avoid misinterpretation by the software.

```

pedro@pedro:/0AI/oai-ran5g/cmake_target/ran_build/build/$ sudo ./multi-ue.
sh -cl -e
creating namespace for UE ID 1 name ue1
opening shell in namespace ue1
root@pedro:/home/pedro/0AI/oai-ran5g/cmake_targets/ran_build/build# sudo -E
LD_LIBRARY_PATH=. RFSIMULATOR=10.201.1.100 ./nr-uesoftmodem -O ../../../../
targets/PROJECTS/GENERIC-NR-5GC/CONF/ue1.conf --band 78 -C 3619200000 --
rfsim --sa

```

Figure 4.8: Initialization of the UE1.

To ensure that all connections were properly set and the UE were able to communicate with all components, the configurations of the interfaces used are presented in Figure 4.9. The interfaces were properly defined and configured, therefore UE1 had been successfully launched.

```

root@pedro:/home/pedro/0AI/oai-ran5g/cmake_targets/ran_build/build#
ifconfig
oaitun_ue1: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 12.1.1.130 netmask 255.255.255.0 destination 12.1.1.130
    inet6 fe80::d8f4:deba:30fd:9b92 prefixlen 64 scopeid 0x20<link>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500
    (UNSPEC)
    (...)

v-ue1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.201.1.1 netmask 255.255.255.0 broadcast 0.0.0.0
    inet6 fe80::bce4:ffff:fe72:9e75 prefixlen 64 scopeid 0x20<link>
    ether be:e4:ff:72:9e:75 txqueuelen 1000 (Ethernet)
    (...)

```

Figure 4.9: Interface configuration for UE1.

4.2.4.2 UE2

Similarly, for UE1 to launch UE2, we followed the same approach, with the same set of commands and flags for creating the namespace and launching the UE. Figure 4.10 represents the commands for creating the namespace and executing the UE, with the `nr-uesoftmodem` considering the UE2 configuration file. The `rfsimulator` IP was also updated to ensure connectivity with other components.

```

pedro@pedro:/OAI/oai-ran5g/cmake_target/ran_build/build/$ sudo ./multi-ue.
sh -c2 -e
creating namespace for UE ID 2 name ue2
opening shell in namespace ue2
root@pedro:/home/pedro/OAI/oai-ran5g/cmake_targets/ran_build/build# sudo -E
LD_LIBRARY_PATH=. RFSIMULATOR=10.202.1.100 ./nr-uesoftmodem -O ../../../../
targets/PROJECTS/GENERIC-NR-5GC/CONF/ue2.conf --band 78 -C 3619200000 --
rfsim --sa

```

Figure 4.10: Initialization of the UE2.

In the same way as UE1, to ensure that all connections were properly established and the UE was able to communicate with all components, the configurations considered for the interfaces are presented in Figure 4.11.

```

root@pedro:/home/pedro/OAI/oai-ran5g/cmake_targets/ran_build/build#
ifconfig
oaitun_ue2: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 12.1.1.131 netmask 255.255.255.0 destination 12.1.1.131
    inet6 fe80::aa24:2b83:e475:40f5 prefixlen 64 scopeid 0x20<link>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500
    (UNSPEC)
    (...)

v-ue2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.202.1.2 netmask 255.255.255.0 broadcast 0.0.0.0
    inet6 fe80::c06d:77ff:fe26:545c prefixlen 64 scopeid 0x20<link>
    ether c2:6d:77:26:54:5c txqueuelen 1000 (Ethernet)
    (...)

```

Figure 4.11: Interface configuration for UE2.

Based on the absence of error messages and the proper definition and configuration of all interfaces, we concluded that UE2 had been successfully launched.

4.2.5 DRB Management xApp

The last component requiring testing was the xApp itself. It was crucial to verify that the xApp launched correctly, loaded the service models, and remained idle, awaiting connection with the Near-RT RIC through the E42 interface and then with E2 nodes such as the CU and DU. Figure 4.12 shows the execution of the xApp, loading the service models and entering an idle state while waiting for connections. As such, the xApp was successfully launched.

```

pedro@pedro:/OAI/flexric/build/$ sudo ./examples/xApp/c/ran_reconf/
xapp_ran_reconf
[xApp]: Initializing ...
[xApp]: nearRT-RIC IP Address = 127.0.0.1, PORT = 36422
[E2 AGENT]: Opening plugin from path = /usr/local/lib/flexric/libkpm_sm.so
(...)
[NEAR-RIC]: Loading SM ID = 2 with def = ORAN-E2SM-KPM
(...)
[xApp]: DB filename = /tmp/xapp_db_1718622478255211

```

Figure 4.12: Initialization of the DRB Management xApp.

4.3 Interoperability Testing

Upon testing each individual component, it was critical to verify that all components were able to function and communicate with each other effectively. This process entailed deploying every component and evaluating every connection to ensure the proper operation of the system, especially in the context of the full operational status of the 5G SA network.

The connection of the 5G Core Network with the disaggregated gNB is depicted in Figure 4.13.

31	4.202851284	192.168.70.129	192.168.70.132	NGAP	126	126 NGSetupRequest
33	4.203691500	192.168.70.132	192.168.70.129	NGAP	130	256 NGSetupResponse


```

Frame 31: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface demo-oai, id 0
Ethernet II, Src: 02:42:ac:43:05:14 (02:42:ac:43:05:14), Dst: 02:42:c0:a8:46:84 (02:42:c0:a8:46:84)
Internet Protocol Version 4, Src: 192.168.70.129, Dst: 192.168.70.132
Stream Control Transmission Protocol, Src Port: 58893 (58893), Dst Port: 38412 (38412)
NG Application Protocol
  NGAP-PDU: initiatingMessage (0)
    initiatingMessage
      procedureCode: id-NGSetup (21)
      criticality: reject (0)
      value
        NGSetupRequest
          protocolIEs: 4 items
            Item 0: id-GlobalRANNodeID
            Item 1: id-RANNodeName
              ProtocolIE-Field
                id: id-RANNodeName (82)
                criticality: ignore (1)
                value
                  RANNodeName: gNB-Eurecom-CU
            Item 2: id-SupportedTAList
              ProtocolIE-Field
                id: id-SupportedTAList (102)
                criticality: reject (0)
                value
                  SupportedTAList: 1 item
                    Item 0
                      SupportedTAItem
                        tAC: 40960 (0x00a000)

```

Figure 4.13: CU successfully communicating with the AMF.

Given the exchange of packets between the CU and the 5G Core Network (CN), we concluded that the disaggregated gNB has been successfully registered in the 5G Core Network. This registration facilitates communication via the AMF through the N2 interface. Moreover, it was critical for the gNB to maintain communication with FlexRIC through the E2 interface, as this capability

is a cornerstone of the proposed solution. Through this interface, metrics are conveyed to the DRB Management xApp, enabling the feed of valuable data for network adjustments. In return, control actions are dispatched from the xApp to the gNB, managing its reconfiguration. Figure 4.14 shows the Near-RT RIC, implemented with FlexRIC, registering the CU and DU.

```
(...)
[iApp]: Initializing ...
[iApp]: nearRT-RIC IP Address 127.0.0.1, PORT = 36422
[NEAR-RIC]: Initializing Task Manager with 2 threads
[E2AP]: E2 SETUP-REQUEST rx from PLMN 208.95 Node ID 3584 RAN type
ngran_gNB_CU [NEAR-RIC]:
(...)
[E2AP]: E2 SETUP-REQUEST rx from PLMN 208.95 Node ID 3584 RAN type
ngran_gNB_DU ID 3584 [NEAR-RIC]:
(...)
```

Figure 4.14: NearRT-RIC registering the CU and DU.

After ensuring proper communication between the separated gNB and the 5G Core network and FlexRIC, it was essential to confirm that the UEs were able to register with the 5G CN. This can be verified by examining the logs provided by Wireshark in Figure 4.15 for UE1 and in Figure 4.16 for UE2. These logs show that both UEs successfully registered with the AMF after their launch.

500	20.723099276	192.168.70.132	192.168.70.129	NGAP/NL...	242	1224 InitialContextSetupRequest
507	20.997663479	192.168.70.129	192.168.70.132	NGAP	82	1306 InitialContextSetupResponse

Frame 507: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface demo-oai, id 0						
Ethernet II, Src: 02:42:ac:43:05:14 (02:42:ac:43:05:14), Dst: 02:42:c0:a8:46:84 (02:42:c0:a8:46:84)						
Internet Protocol Version 4, Src: 192.168.70.129, Dst: 192.168.70.132						
Stream Control Transmission Protocol, Src Port: 58893 (58893), Dst Port: 38412 (38412)						
NG Application Protocol						
NGAP-PDU: successfulOutcome (1)						
successfulOutcome						
procedureCode: id-InitialContextSetup (14)						
criticality: reject (0)						
value						
InitialContextSetupResponse						
protocolIEs: 2 items						
Item 0: id-AMF-UE-NGAP-ID						
ProtocolIE-Field						
id: id-AMF-UE-NGAP-ID (10)						
criticality: ignore (1)						
value						
AMF-UE-NGAP-ID: 35						
Item 1: id-RAN-UE-NGAP-ID						
ProtocolIE-Field						
id: id-RAN-UE-NGAP-ID (85)						
criticality: ignore (1)						
value						
RAN-UE-NGAP-ID: 1						

Figure 4.15: UE1 successfully registration with AMF.

After completing registration with the AMF, the UEs were assigned an IP address for information exchange. Prior to this, the 5G Core Network configured a PDU session to establish the tunnels from the UPF to gNB. Subsequently, a QFI was assigned, as shown in Figure 4.17 for UE1 and Figure 4.18 for UE2.

1517	33.898953303	192.168.70.132	192.168.70.129	NGAP/N...	242	4210 InitialContextSetupRequest
1544	34.199351320	192.168.70.129	192.168.70.132	NGAP	82	4292 InitialContextSetupResponse
1568	34.407552423	192.168.70.129	192.168.70.132	NGAP	102	4394 UERadioCapabilityInfoIndication


```

Frame 1544: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface demo-oai, id 0
Ethernet II, Src: 02:42:ac:43:05:14 (02:42:ac:43:05:14), Dst: 02:42:c0:a8:46:84 (02:42:c0:a8:46:84)
Internet Protocol Version 4, Src: 192.168.70.129, Dst: 192.168.70.132
Stream Control Transmission Protocol, Src Port: 58893 (58893), Dst Port: 38412 (38412)
NG Application Protocol
  NGAP-PDU: successfulOutcome (1)
    successfulOutcome
      procedureCode: id-InitialContextSetup (14)
      criticality: reject (0)
      value
        InitialContextSetupResponse
          protocolIEs: 2 items
            Item 0: id-AMF-UE-NGAP-ID
              ProtocolIE-Field
                id: id-AMF-UE-NGAP-ID (10)
                criticality: ignore (1)
                value
                  AMF-UE-NGAP-ID: 36
            Item 1: id-RAN-UE-NGAP-ID
              ProtocolIE-Field
                id: id-RAN-UE-NGAP-ID (85)
                criticality: ignore (1)
                value
                  RAN-UE-NGAP-ID: 2

```

Figure 4.16: UE2 successfully registration with AMF.

588	22.243866952	192.168.70.132	192.168.70.129	NGAP/N...	270	26019 PDUSessionResourceSetupRequest
590	22.308418604	192.168.70.129	192.168.70.132	NGAP	118	26137 PDUSessionResourceSetupResponse


```

NG Application Protocol
  NGAP-PDU: successfulOutcome (1)
    successfulOutcome
      procedureCode: id-PDUSessionResourceSetup (29)
      criticality: reject (0)
      value
        PDUSessionResourceSetupResponse
          protocolIEs: 3 items
            Item 0: id-AMF-UE-NGAP-ID
            Item 1: id-RAN-UE-NGAP-ID
              ProtocolIE-Field
                id: id-RAN-UE-NGAP-ID (85)
                criticality: ignore (1)
                value
                  RAN-UE-NGAP-ID: 1
            Item 2: id-PDUSessionResourceSetupListSURes
              ProtocolIE-Field
                id: id-PDUSessionResourceSetupListSURes (75)
                criticality: ignore (1)
                value
                  PDUSessionResourceSetupListSURes: 1 item
                    Item 0
                      PDUSessionResourceSetupItemSURes
                        pDUSessionID: 10
                        PDUSessionResourceSetupResponseTransfer: 0003e0c0a8468158de38860009
                          PDUSessionResourceSetupResponseTransfer
                            dLQoSFlowPerTNLInformation
                              uPTransportLayerInformation: gTPTunnel (0)
                                gTPTunnel
                                  transportLayerAddress: c0a84681 [bit length 32, 1100 0000 1010 1000 0100 0110 1000 0001]
                                  gTP-TEID: 58de3886
                              associatedQoSFlowList: 1 item
                                Item 0
                                  AssociatedQoSFlowItem
                                    qosFlowIdentifier: 9

```

Figure 4.17: PDU session configuration and QFI assignment for UE1.

Upon the completion of PDU session establishment, IP addresses were assigned as illustrated in Figure 4.19 for UE1 and Figure 4.20 for UE2. This enabled the UEs to establish communications, allowing them to transmit and receive traffic from both the Internet through the disaggregated gNB and UPF. This marked the successful deployment and full operational status of the last required elements for the implementation of a private 5G Standalone SA network.

1701	35.429269131	192.168.70.132	192.168.70.129	NGAP/NL	270	94471	PDU Session Resource Setup Request
1702	35.524451968	192.168.70.129	192.168.70.132	NGAP	118	94589	PDU Session Resource Setup Response
NG Application Protocol							
NGAP-PDU: successfulOutcome (1)							
successfulOutcome							
procedureCode: id-PDU Session Resource Setup (29)							
criticality: reject (0)							
value							
PDU Session Resource Setup Response							
protocolIEs: 3 items							
Item 0: id-AMF-UE-NGAP-ID							
Item 1: id-RAN-UE-NGAP-ID							
ProtocolIE-Field							
id: id-RAN-UE-NGAP-ID (85)							
criticality: ignore (1)							
value							
RAN-UE-NGAP-ID: 2							
Item 2: id-PDU Session Resource Setup List S-U Res							
ProtocolIE-Field							
id: id-PDU Session Resource Setup List S-U Res (75)							
criticality: ignore (1)							
value							
PDU Session Resource Setup List S-U Res: 1 item							
Item 0							
PDU Session Resource Setup Item S-U Res							
pDU Session ID: 10							
PDU Session Resource Setup Response Transfer: 0003e0c0a84681c48c84ae0009							
PDU Session Resource Setup Response Transfer							
dL QoS Flow Per TNL Information							
uP Transport Layer Information: gTP Tunnel (0)							
gTP Tunnel							
transport Layer Address: c0a84681 [bit length 32, 1100 0000 1010 1000 0100 0110 1000 0001]							
gTP-TEID: c48c84ae							
associated QoS Flow List: 1 item							
Item 0							
Associated QoS Flow Item							
qosFlowIdentifier: 9							

Figure 4.18: PDU session configuration and QFI assignment for UE2.

```
[LIBCONFIG] nas.noS1: 3/3 parameters successfully set, (3 to default value)
[OIP]   Interface oaitun_ue1 successfully configured, ip address
12.1.1.136, mask 255.255.255.0 broadcast address 12.1.1.255
```

Figure 4.19: IP addressing and interface configuration of UE1 acknowledged by DU.

```
[LIBCONFIG] nas.noS1: 3/3 parameters successfully set, (3 to default value)
[OIP]   Interface oaitun_ue2 successfully configured, ip address
12.1.1.137, mask 255.255.255.0 broadcast address 12.1.1.255
```

Figure 4.20: IP addressing and interface configuration of UE2 acknowledged by DU.

To confirm that the 5G SA network was fully operational, we logged into the container representing the Internet, the Ext-DN, and pinged both UEs. The results can be seen in Figure 4.21 and in the Wireshark log in Figure 4.22.

```
root@15b6e65f7743:/tmp# ping 12.1.1.136
64 bytes from 12.1.1.136: icmp_seq=1 ttl=63 time=23.1 ms
64 bytes from 12.1.1.136: icmp_seq=1 ttl=63 time=26.7 ms
root@15b6e65f7743:/tmp# ping 12.1.1.137
64 bytes from 12.1.1.137: icmp_seq=2 ttl=63 time=34.7 ms
64 bytes from 12.1.1.137: icmp_seq=2 ttl=63 time=28.9 ms
```

Figure 4.21: Successfully pinging both UEs from the Internet.

207108	68.989393755	192.168.70.135	12.1.1.137	ICMP	98	134096	Echo (ping) request
207125	68.995483228	192.168.70.135	12.1.1.137	GTP <I...	142	134238	Echo (ping) request
207462	69.053494223	192.168.70.135	12.1.1.136	ICMP	98	134336	Echo (ping) request
207463	69.053743160	192.168.70.135	12.1.1.136	GTP <I...	142	134478	Echo (ping) request
207464	69.070910785	12.1.1.137	192.168.70.135	GTP <I...	142	134620	Echo (ping) reply
207465	69.071062059	12.1.1.137	192.168.70.135	ICMP	98	134718	Echo (ping) reply
207686	69.135352864	12.1.1.136	192.168.70.135	GTP <I...	142	134860	Echo (ping) reply
207874	69.142483421	12.1.1.136	192.168.70.135	ICMP	98	134958	Echo (ping) reply

Frame 1149: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface demo-oai, id 0
 Ethernet II, Src: 02:42:c0:a8:46:87 (02:42:c0:a8:46:87), Dst: 02:42:c0:a8:46:86 (02:42:c0:a8:46:86)
 Internet Protocol Version 4, Src: 192.168.70.135, Dst: 12.1.1.136
 Internet Control Message Protocol

Figure 4.22: Confirmation of successful pings with wireshark.

After the NearRT-RIC was properly implemented, the DRB Management xApp, upon receiving a notification through the E42 interface, proceeded to establish a connection with E2 nodes. This transition enabled the xApp to move out of the idle state and begin the process of acquiring metrics related to network resource usage and the overall quality of the connection. The connection established between the NearRT-RIC, the xApp, and the E2 nodes is detailed in Figure 4.23. The successful reception and decoding of these metrics by the FlexRIC, as evidenced in the Wireshark log depicted in Figure 4.24, indicated that the system was functioning according to the intended operational parameters.

```
[xApp]: E42 SETUP-REQUEST tx
[xApp]: E42 SETUP-RESPONSE rx
[xApp]: Registered E2 Nodes = 2
Connected E2 nodes = 2
```

Figure 4.23: E42 communication and E2 nodes registration.

Following a thorough assessment of each singular network component and conducting individual and interoperable tests, we confirmed that all elements were operational. The comprehensive assessment showed that the network functions correctly as an integrated system. With these technical verifications complete, we performed the use case validation. This stage ensured that the network met all practical application requirements and performed effectively under real-world conditions.

```

796 26.776742663 127.0.0.1 127.0.0.1 E2AP 206 RICIndication
Frame 796: 206 bytes on wire (1648 bits), 206 bytes captured (1648 bits) on interface lo, id 1
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Stream Control Transmission Protocol, Src Port: 36422 (36422), Dst Port: 45303 (45303)
E2 Application Protocol
  E2AP-PDU: initiatingMessage (0)
    initiatingMessage
      procedureCode: id-RICIndication (5)
      criticality: ignore (1)
      value
        RICIndication
          protocolIEs: 6 items
            Item 0: id-RICrequestID
            Item 1: id-RANfunctionID
            Item 2: id-RICactionID
            Item 3: id-RICindicationType
            Item 4: id-RICindicationHeader
            Item 5: id-RICindicationMessage
          ProtocolIE-Field
            id: id-RICindicationMessage (26)
            criticality: reject (0)
            value
              RICIndicationMessage: 400046000000000000140000000002200480010089200980cc0be76c8b439581000101004452422e526c63:
              [RANfunction setup frame: 187 (2 -> KPM)]
              [Version (frame): KPM v2]
              [Version (dissector): KPM v2]
          E2SM-KPM-IndicationMessage
            indicationMessage-formats: indicationMessage-Format3 (2)
              indicationMessage-Format3
                ueMeasReportList: 1 item
                  Item 0
                    UEMeasurementReportItem
                      ueID: gNB-DU-UEID (1)
                      gNB-DU-UEID
                        measReport
                          measData: 1 item
                          measInfoList: 2 items
                            Item 0
                              MeasurementInfoItem
                                measType: measName (0)
                                measName: DRB.RlcSduDelayDL
                                labelInfoList: 1 item
                              Item 1
                                MeasurementInfoItem
                                  measType: measName (0)
                                  measName: DRB.UETHpDL

```

Figure 4.24: E2 indication messages, transmitting data to and from the xApp

4.4 Use Case Validation

As documented in Section 4.1, which outlines the methodology, we examined two use cases to validate the proposed solution, each representative of real-world scenarios. Ping and iPerf3 tools were employed to simulate UEs' usage of the network's resources in each test.

The initial phases of each test remained consistent. Initially, we deployed the 5G CN and executed both the CU and DU. Subsequently, we initialized one or two UEs and waited for them to connect to the disaggregated gNB and register in the 5G CN. It is important to note that the IP addresses assigned to the UEs by the network may vary from those discussed and illustrated in the document, owing to the timing differences related to the events outlined in Section 3.3.

Once all components were ready, we accessed the Ext-DN container, acting as the Internet, and conducted a ping to the UE. Following this, we initiated an iPerf3 session, configuring the UE as the server and the Ext-DN as the client.

By monitoring the logs provided by the DU, we confirmed that the traffic was flowing correctly. As only the default DRB was configured (corresponding to LCID 4), all traffic was being exchanged through it, as shown in the DU log in Figure 4.25.

During the concurrent ping and iPerf3 sessions, both the throughput and sojourn time began to increase. Upon reaching the predefined thresholds, the xApp initiated the reconfiguration of the

```

[NR_MAC]   Frame.Slot 0.0
UE RNTI 8962 CU-UE-ID 1, average RSRP -44 (16 meas)
UE 8962: MAC:    DL          24752 bytes
UE 8962: LCID 1: DL          961 bytes
UE 8962: LCID 2: DL           0 bytes
UE 8962: LCID 4: DL         1956 bytes

[NR_MAC]   Frame.Slot 128.0
UE RNTI 8962 CU-UE-ID 1, average RSRP -44 (16 meas)
UE 8962: MAC:    DL         27335 bytes
UE 8962: LCID 1: DL          961 bytes
UE 8962: LCID 2: DL           0 bytes
UE 8962: LCID 4: DL         2331 bytes

[NR_MAC]   Frame.Slot 256.0
UE RNTI 8962 CU-UE-ID 1, average RSRP -44 (16 meas)
UE 8962: MAC:    DL        1036917 bytes
UE 8962: LCID 1: DL          961 bytes
UE 8962: LCID 2: DL           0 bytes
UE 8962: LCID 4: DL       1004019 bytes

[NR_MAC]   Frame.Slot 384.0
UE RNTI 8962 CU-UE-ID 1, average RSRP -44 (16 meas)
UE 8962: MAC:    DL        9209992 bytes
UE 8962: LCID 1: DL          961 bytes
UE 8962: LCID 2: DL           0 bytes
UE 8962: LCID 4: DL       9153503 bytes

```

Figure 4.25: DU log confirming traffic flowing through the default DRB.

RAN to introduce a new DRB, as illustrated in Figure 4.26. The new DRB (LCID 5) was created with a 5QI value of 3, prioritizing the traffic passing through it over the traffic utilizing the default DRB (LCID 4), which had a 5QI of 9. The ICMP packets generated by the ping were designated to pass through the new DRB, while the iPerf3 traffic was consistently directed through the default DRB.

```

[xApp]: CONTROL-REQUEST tx
[xApp]: CONTROL ACK rx
[xApp]: Successfully received CONTROL-ACK
Creating second bearer

```

Figure 4.26: Triggering of the DRB Management xApp for creating DRB.

In a similar manner, once the network congestion was reduced, the DRB Management xApp was activated once more to eliminate the previously added DRB. This process is illustrated in Figure 4.27.

```
[xApp]: CONTROL-REQUEST tx
[xApp]: CONTROL ACK rx
[xApp]: Successfully received CONTROL-ACK
Releasing second bearer
```

Figure 4.27: Triggering of the DRB Management xApp for releasing DRB.

4.4.1 One UE triggering RAN Reconfiguration

In the first test, only one UE was considered. The UE was assigned the IP address 12.1.1.130. After triggering the DRB Management xApp to create a new DRB, the RAN successfully reconfigured itself and added the second DRB, as proved by the CU log in Figure 4.28 and Wireshark log of the F1AP interface in Figure 4.29.

```
[O-RAN RC]: Creating a new bearer
[NR_RRC] UE 1: configure DRB ID 2 for PDU session ID 10
[PDCP] added drb 2 to UE ID 1
[RRC] trigger new bearer 2 for UE 8962 xid 0
[F1AP] CU Task Received F1AP_UE_CONTEXT_MODIFICATION_REQ for instance 0:
sending Sctp message via assoc_id 6
[E2-AGENT]: CONTROL ACKNOWLEDGE tx
[F1AP] CU Task Received Sctp_DATA_IND for instance 0: sending Sctp message
via assoc_id 0
[F1AP] flap_ue_context_modif_resp->gNB_CU_ue_id is: 1
[F1AP] flap_ue_context_modification_resp->gNB_DU_ue_id is: 35170
[GTPU] Tunnel Outgoing TEID updated to b5cbcd89 and address to 300007f
[RRC] UE 8962 replacing existing CellGroupConfig with new one received from
DU
```

Figure 4.28: CU acknowledging the creation of DRB2.

Based on both figures, it was confirmed that the RAN established a new DRB and assigned it a 5QI of 3. Moreover, upon examining the DU log shown in Figure 4.30, it was demonstrated that the network was redirecting the desired ICMP traffic through the new DRB while leaving the traffic generated by iPerf3 to the default DRB. The ICMP traffic, characterized by a smaller number of transmitted bytes than the iPerf3 session, was evidently exchanged through DRB2 (LCID 5) as intended in the downlink. The traffic from the iPerf3 session was still being exchanged through the default DRB (LCID 4).

The analysis involved plotting the evolution of the ping response time to assess the triggering behaviour of the DRB Management xApp. Additionally, we plotted the evolution of the sojourn time and throughput, highlighting the specific moment of the DRB Management xApp triggering in both graphics.

The results, as observed in Figure 4.31 and Figure 4.32, validate the test's success. When the network experienced excessive congestion, the QoS parameters were no longer met, leading to the trigger of the xApp. Following the addition of the new DRB, the ping response time, whose traffic was exchanged through the new DRB, significantly improved. Simultaneously, the throughput of the default DRB remained consistently high while the delay at the RLC layer even increased.

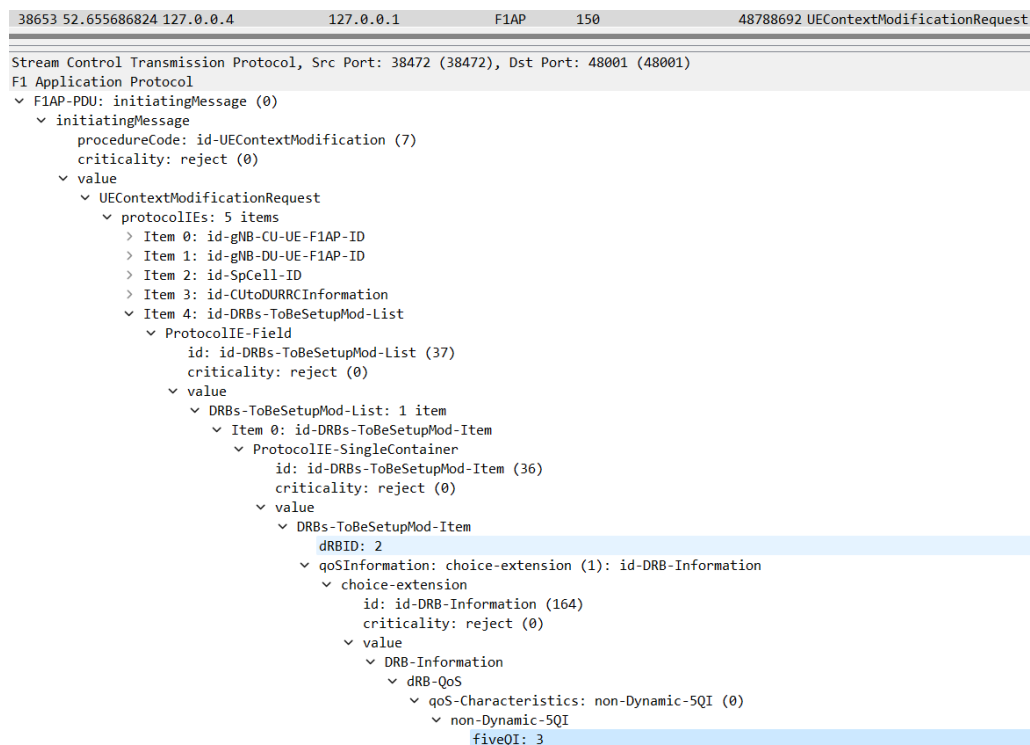


Figure 4.29: F1AP interface acknowledging the creation of DRB2 with 5QI 3.

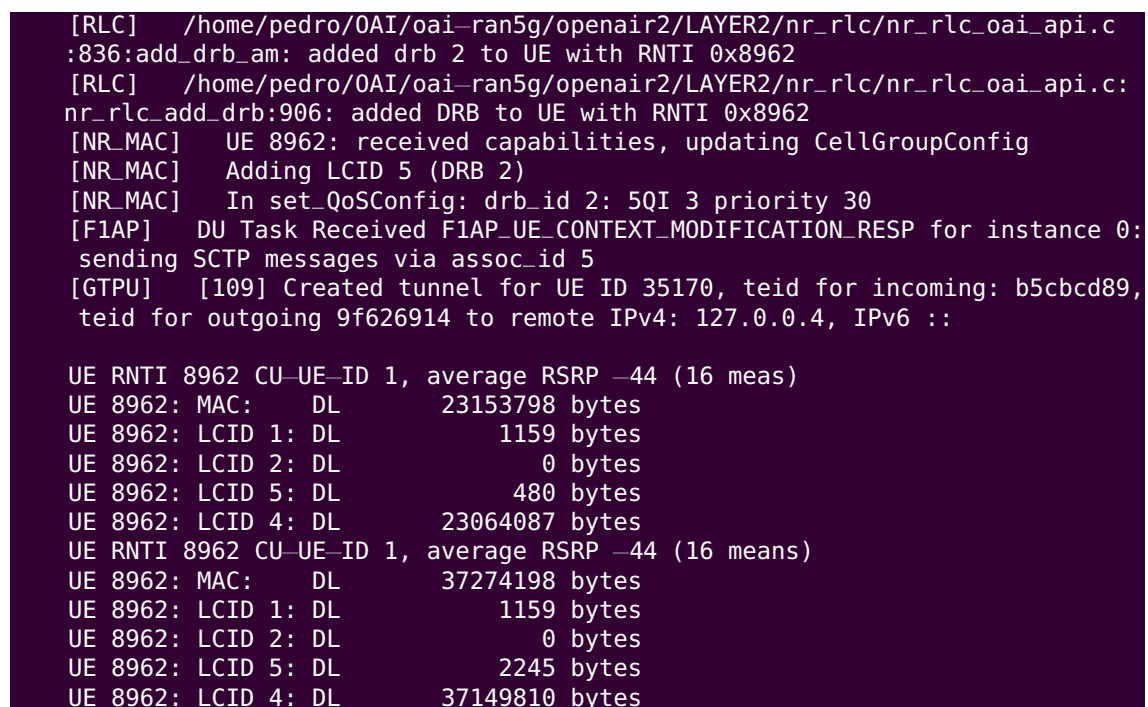


Figure 4.30: DU acknowledging the creation of DRB2 and rerouting traffic through it.

When the sojourn time and throughput were insufficient to sustain the additional DRB, the xApp triggered again to initiate its release.

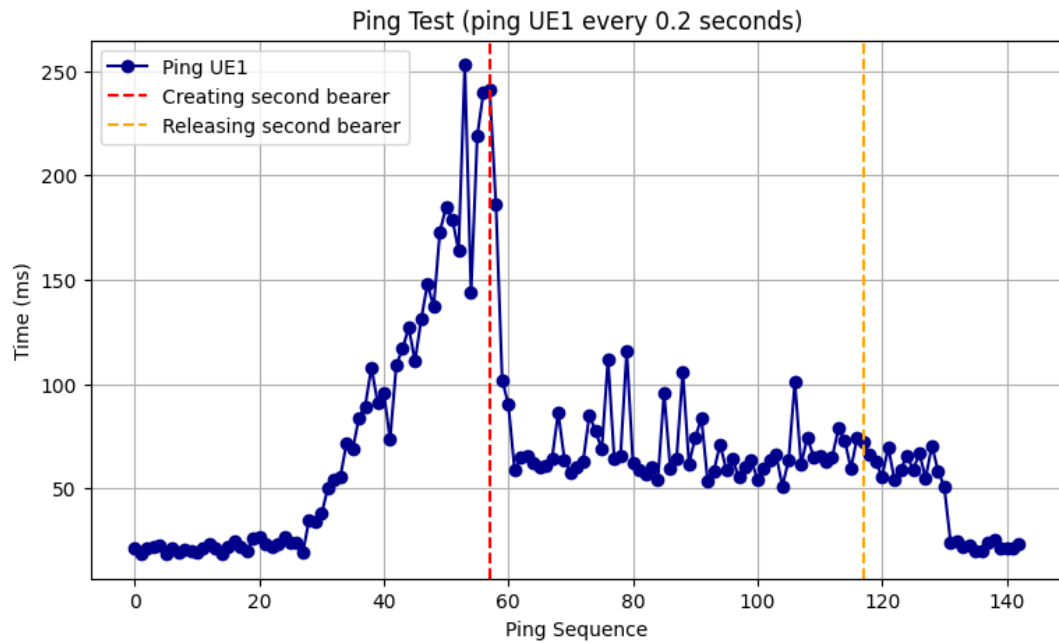


Figure 4.31: Evolution of ping response time throughout the test for UE1.

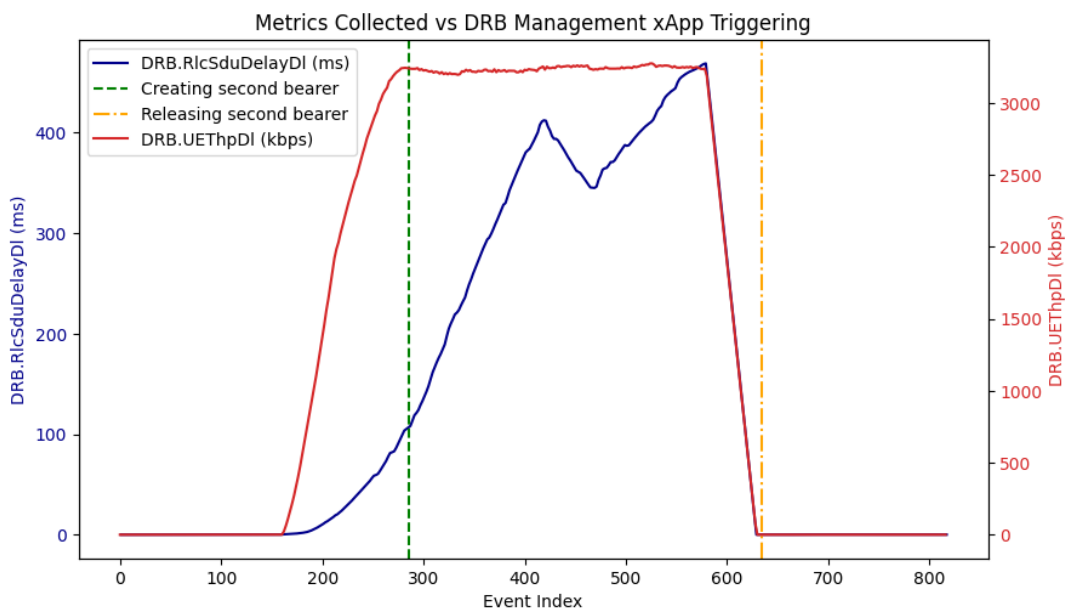


Figure 4.32: Evolution of throughput and sojourn time in the default DRB across the testing time for UE1.

This test only considered one UE connected to the disaggregated gNB. Since our solution is designed to work with multiple UEs connected at once, we tested with an additional UE connected to the disaggregated gNB, as this is representative of the intended behaviour.

4.4.2 Two UEs triggering RAN Reconfiguration

In this test scenario, the focus was on two UEs connected to the disaggregated gNB. The network assigned IP addresses 12.1.1.136 for UE1 and 12.1.1.137 for UE2. Similar to the initial test, both UEs were responsive to ping requests from the Internet, and iPerf3 sessions were initiated with both UEs serving as servers. This resulted in a gradual increase in throughput until reaching maximum capacity, accompanied by a corresponding rise in sojourn time. Upon surpassing the threshold, in the case of UE1, the xApp prompted the RAN to reconfigure and introduce a new DRB, as illustrated in the DU log represented in Figure 4.33.

```
[RLC] /home/pedro/OAI/oai-ran5g/openair2/LAYER2/nr_rlc/nr_rlc_oai_api.c
:836:add_drb_am: added drb 2 to UE with RNTI 0x6d1f
[RLC] /home/pedro/OAI/oai-ran5g/openair2/LAYER2/nr_rlc/nr_rlc_oai_api.c:
nr_rlc_add_drb:906: added DRB to UE with RNTI 0x6d1f
[NR_MAC] UE 6d1f: received capabilities, updating CellGroupConfig
[NR_MAC] Adding LCID 5 (DRB 2)
[NR_MAC] In set_QoSConfig: drb_id 2: 5QI 3 priority 30
[F1AP] DU Task Received F1AP_UE_CONTEXT_MODIFICATION_RESP for instance 0:
sending SCTP messages via assoc_id 35
[GTPU] [109] Created tunnel for UE ID 27935, teid for incoming: 6c6af9f7,
teid for outgoing 43926d9d to remote IPv4: 127.0.0.4, IPv6 ::

UE RNTI 6d1f CU-UE-ID 1 average RSRP -44 (15 meas)
UE 6d1f: MAC: DL 31993104 bytes
UE 6d1f: LCID 1: DL 1154 bytes
UE 6d1f: LCID 2: DL 0 bytes
UE 6d1f: LCID 5: DL 1257 bytes
UE 6d1f: LCID 4: DL 31888703 bytes
UE RNTI 88b0 CU-UE-ID 2 average RSRP -44 (16 meas)
UE 88b0: MAC: DL 30048 bytes
UE 88b0: LCID 1: DL 958 bytes
UE 88b0: LCID 2: DL 0 bytes
UE 88b0: LCID 4: DL 5121 bytes

UE RNTI 6d1f CU-UE-ID 1 average RSRP -44 (16 meas)
UE 6d1f: MAC: DL 45203845 bytes
UE 6d1f: LCID 1: DL 1154 bytes
UE 6d1f: LCID 2: DL 0 bytes
UE 6d1f: LCID 5: DL 2893 bytes
UE 6d1f: LCID 4: DL 45067012 bytes
UE RNTI 88b0 CU-UE-ID 2 average RSRP -44 (16 meas)
UE 88b0: MAC: DL 496481 bytes
UE 88b0: LCID 1: DL 958 bytes
UE 88b0: LCID 2: DL 0 bytes
UE 88b0: LCID 4: DL 464784 bytes
```

Figure 4.33: DU acknowledging the creation of DRB2 for UE1 and rerouting the traffic.

Right after the creation of DRB2 for UE1, the threshold for acceptable sojourn time was exceeded for UE2. This development prompted the DRB Management xApp to trigger a RAN reconfiguration for UE2, issuing instructions to establish a new DRB, as detailed in the DU log depicted in Figure 4.34. For this newly created DRB, 5QI was set to 9. Further information can

be collected from the CU log, as shown in Figure 4.35, which confirms the successful creation of the DRB with the desired attributes for UE2.

```
[RLC] /home/pedro/OAI/oai-ran5g/openair2/LAYER2/nr_rlc/nr_rlc_oai_api.c
:836:add_drb_am: added drb 2 to UE with RNTI 0x88b0
[RLC] /home/pedro/OAI/oai-ran5g/openair2/LAYER2/nr_rlc/nr_rlc_oai_api.c:
nr_rlc_add_drb:906: added DRB to UE with RNTI 0x88b0
[NR_MAC] UE 88b0: received capabilities, updating CellGroupConfig
[NR_MAC] Adding LCID 5 (DRB 2)
[NR_MAC] In set_QoSConfig: drb_id 2: 5QI 9 priority 90
[F1AP] DU Task Received F1AP_UE_CONTEXT_MODIFICATION_RESP for instance 0:
sending SCTP messages via assoc_id 35
[GTPU] [109] Created tunnel for UE ID 34992, teid for incoming: 8048f7cd,
teid for outgoing 6e712794 to remote IPv4: 127.0.0.4, IPv6 ::

UE RNTI 6d1f CU-UE-ID 1 average RSRP -44 (15 meas)
UE 6d1f: MAC: DL 64700801 bytes
UE 6d1f: LCID 1: DL 1154 bytes
UE 6d1f: LCID 2: DL 0 bytes
UE 6d1f: LCID 5: DL 6592 bytes
UE 6d1f: LCID 4: DL 64514793 bytes
UE RNTI 88b0 CU-UE-ID 2 average RSRP -44 (16 meas)
UE 88b0: MAC: DL 15017074 bytes
UE 88b0: LCID 1: DL 1156 bytes
UE 88b0: LCID 2: DL 0 bytes
UE 88b0: LCID 5: DL 378 bytes
UE 88b0: LCID 4: DL 14941494 bytes

UE RNTI 6d1f CU-UE-ID 1 average RSRP -44 (16 meas)
UE 6d1f: MAC: DL 74503521 bytes
UE 6d1f: LCID 1: DL 1154 bytes
UE 6d1f: LCID 2: DL 0 bytes
UE 6d1f: LCID 5: DL 8607 bytes
UE 6d1f: LCID 4: DL 74292663 bytes
UE RNTI 88b0 CU-UE-ID 2 average RSRP -44 (16 meas)
UE 88b0: MAC: DL 24819794 bytes
UE 88b0: LCID 1: DL 1156 bytes
UE 88b0: LCID 2: DL 0 bytes
UE 88b0: LCID 5: DL 2378 bytes
UE 88b0: LCID 4: DL 24719395 bytes
```

Figure 4.34: DU acknowledging the creation of DRB2 for UE2 and rerouting the traffic.

Once more, by reviewing the log from Wireshark, depicted in Figure 4.36, we verified that the DRB2 for UE2 was created successfully as intended.

After ensuring the proper establishment of the second DRB for both UEs following the activation of the xApp, it was evident from the DU logs depicted in Figure 4.33 and Figure 4.34 that the new DRB was functioning correctly. The network successfully redirected the ICMP packets to the new DRB while maintaining the traffic from the iPerf3 session through the default DRB for both UEs in the downlink direction.

In order to validate the implemented functionalities and confirm the system's proper operation, the change in ping response times and the evolution of the throughput and sojourn time in the

```

[0-RAN RC]: Creating a new bearer
[NR_RRC]   UE 2: configure DRB ID 2 for PDU session ID 10
[PDCP]    added drb 2 to UE ID 2
[RRC]     trigger new bearer 2 for UE 88b0 xid 1
[F1AP]    CU Task Received F1AP_UE_CONTEXT_MODIFICATION_REQ for instance 0:
sending SCTP message via assoc_id 36
[E2-AGENT]: CONTROL ACKNOWLEDGE tx
[F1AP]    flap_ue_context_modif_resp->gNB_CU_ue_id is: 2
[F1AP]    flap_ue_context_modification_resp->gNB_DU_ue_id is: 34992
[GTPU]    Tunnel Outgoing TEID updated to 8048f7cd and address to 300007f
[RRC]    UE 88b0 replacing existing CellGroupConfig with new one received from
DU

```

Figure 4.35: CU acknowledging the creation of DRB2 for UE2.

132576	54.509596099	127.0.0.4	127.0.0.1	F1AP	150	590 UEContextModificationRequest
<pre> criticality: reject (0) value UEContextModificationRequest protocolIEs: 5 items Item 0: id-gNB-CU-UE-F1AP-ID ProtocolIE-Field id: id-gNB-CU-UE-F1AP-ID (40) criticality: reject (0) value GNB-CU-UE-F1AP-ID: 2 Item 1: id-gNB-DU-UE-F1AP-ID Item 2: id-SpCell-ID Item 3: id-CUtoDURRCInformation Item 4: id-DRBs-ToBeSetupMod-List ProtocolIE-Field id: id-DRBs-ToBeSetupMod-List (37) criticality: reject (0) value DRBs-ToBeSetupMod-List: 1 item Item 0: id-DRBs-ToBeSetupMod-Item ProtocolIE-SingleContainer id: id-DRBs-ToBeSetupMod-Item (36) criticality: reject (0) value DRBs-ToBeSetupMod-Item drbID: 2 qosInformation: choice-extension (1): id-DRB-Information choice-extension id: id-DRB-Information (164) criticality: reject (0) value DRB-Information drb-QoS qos-Characteristics: non-Dynamic-5QI (0) non-Dynamic-5QI fiveQI: 9 </pre>						

Figure 4.36: F1AP interface acknowledging the creation of DRB2 with 5QI 9 for UE2.

default DRB were plotted for both UEs. The results are presented in Figure 4.37 and Figure 4.38, respectively.

Upon the initialization of an iPerf3 session for UE1, the data started flowing through the default DRB, increasing throughput. The throughput reached a maximum value while the sojourn time gradually increased, resulting in an extended delay for UE1 to respond to ping requests. However, the creation of an additional DRB for UE1 instantaneously reduced the ping response time. Furthermore, as the iPerf3 session continued to generate traffic, the initiation of traffic reception by UE2, generated by another iPerf3 session, rose the throughput of UE2, although to a level smaller

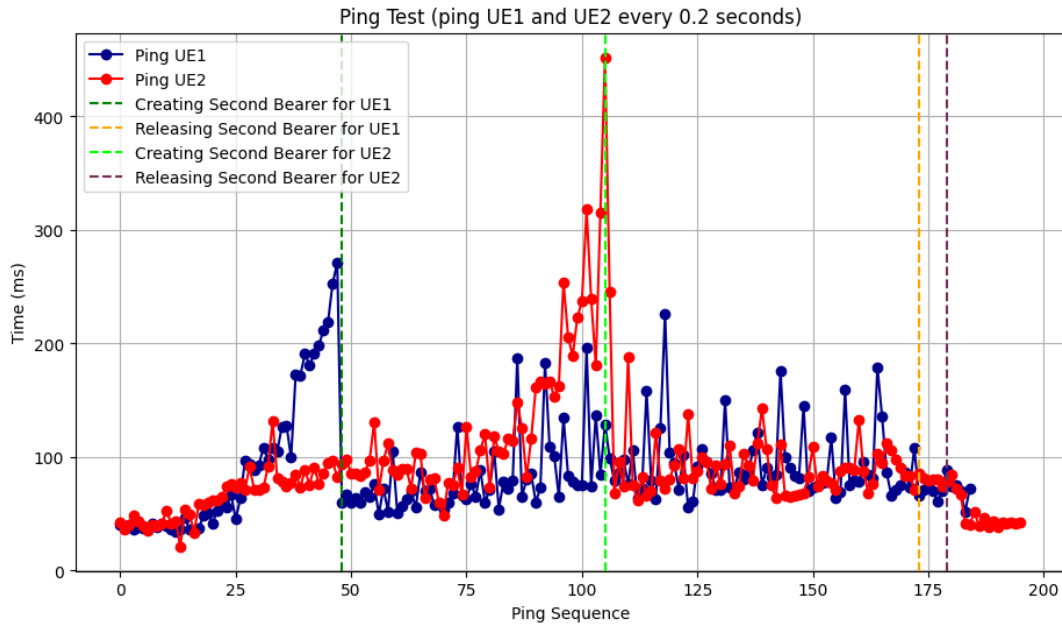


Figure 4.37: Evolution of ping response time throughout the test for UE1 and UE2 when both trigger the DRB Management xApp.

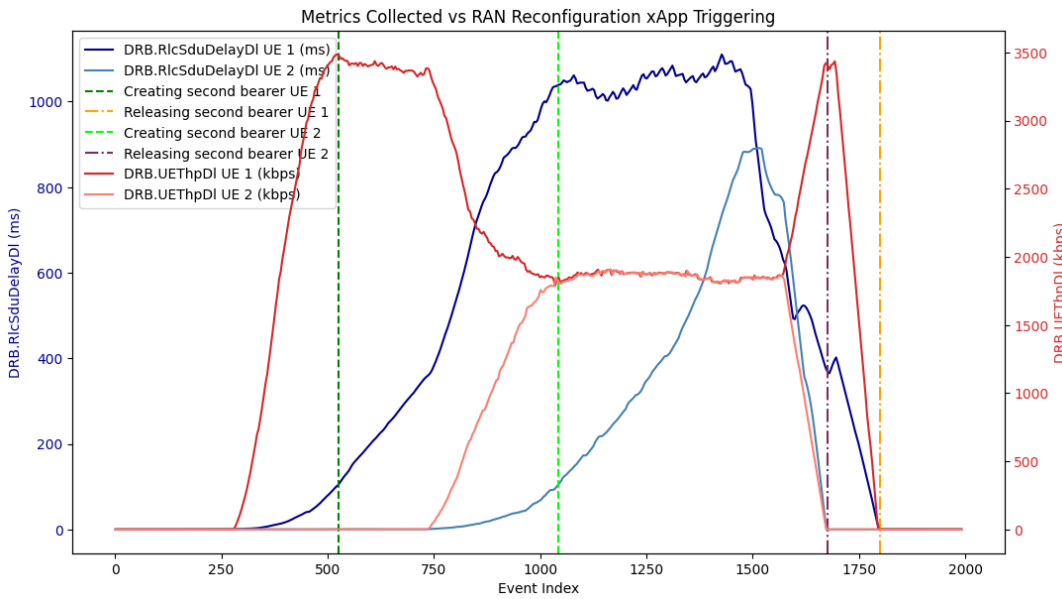


Figure 4.38: Evolution of throughput and sojourn time in the default DRB for UE1 and UE2 when both trigger the DRB Management xApp.

than the maximum achievable when only one UE was connected. Conversely, the throughput of UE1 decreased to approximately half of the maximum throughput achieved when it was the only UE connected. Moreover, as the sojourn time of UE1 continued to increase, the sojourn time of UE2 also increased rapidly, exceeding the acceptable bounds for responding to ping requests. This triggered the xApp to create an additional DRB for UE2. Similarly to UE1, the establishment of

an additional DRB significantly reduced the ping response time to acceptable levels. At a later stage, when the network no longer required the use of an additional DRB for both UEs, the xApp prompted the RAN to reconfigure and release the additional DRB, according to the expectations. With the completion of the iPerf3 session, only the ping requests remained active. As the DRB was released, the reconfiguration was required to redirect the traffic from the ICMP packets resulting from the ping requests back to the default DRB. The DU log outlines the procedure for releasing DRB2 of UE1, as depicted in Figure 4.39. The confirmation of the successful release is evident in the Wireshark log depicted in Figure 4.40. This confirms the successful remapping of ICMP traffic to the default DRB, demonstrating the success of the test.

```
(...)
[GTPU] Deleted tunnel TEID -2142701619 (RB 2) for ue id 34992, remaining
bearers: 1
[NR_MAC] remove LCID 5 (DRB 2)
[F1AP] DU Task Received F1AP_UE_CONTEXT_MODIFICATION_RESP for instance 0:
sending SCTP messages via assoc_id 35

UE RNTI 6d1f CU-UE-ID 1 average RSRP -44 (16 meas)
UE 6d1f: MAC: DL 106205561 bytes
UE 6d1f: LCID 1: DL 1336 bytes
UE 6d1f: LCID 2: DL 0 bytes
UE 6d1f: LCID 4: DL 105898193 bytes
UE RNTI 88b0 CU-UE-ID 2 average RSRP -44 (16 meas)
UE 88b0: MAC: DL 43423287 bytes
UE 88b0: LCID 1: DL 1335 bytes
UE 88b0: LCID 2: DL 0 bytes
UE 88b0: LCID 4: DL 43239196 bytes
```

Figure 4.39: DU acknowledging the releasing of the DRB2 for UE2.

```
213996 70.309942647 127.0.0.4 127.0.0.1 F1AP 106 986 UEContextModificationRequest
Internet Protocol Version 4, Src: 127.0.0.4, Dst: 127.0.0.1
Stream Control Transmission Protocol, Src Port: 38472 (38472), Dst Port: 58032 (58032)
F1 Application Protocol
  F1AP-PDU: initiatingMessage (0)
    initiatingMessage
      procedureCode: id-UEContextModification (7)
      criticality: reject (0)
      value
        UEContextModificationRequest
          protocolIEs: 4 items
            Item 0: id-gNB-CU-UE-F1AP-ID
              ProtocolIE-Field
                id: id-gNB-CU-UE-F1AP-ID (40)
                criticality: reject (0)
                value
                  GNB-CU-UE-F1AP-ID: 1
            Item 1: id-gNB-DU-UE-F1AP-ID
            Item 2: id-SpCell-ID
            Item 3: id-DRBs-ToBeReleased-List
              ProtocolIE-Field
                id: id-DRBs-ToBeReleased-List (33)
                criticality: reject (0)
                value
                  DRBs-ToBeReleased-List: 1 item
                    Item 0: id-DRBs-ToBeReleased-Item
                      ProtocolIE-SingleContainer
                        id: id-DRBs-ToBeReleased-Item (32)
                        criticality: reject (0)
                        value
                          DRBs-ToBeReleased-Item
                            dRBID: 2
```

Figure 4.40: F1AP interface acknowledging the releasing of DRB2 for UE1.

4.5 Discussion

The tests that were conducted allowed us to validate the deployed private 5G SA network and the functionalities that were implemented. The objectives of the dissertation were achieved by developing an xApp that is able to reconfigure the RAN and create an additional DRB when the acceptable values for throughput and sojourn time are exceeded. Moreover, it was able to handle properly differentiated traffic flows and ensure that latency-sensitive traffic was prioritized appropriately, improving the overall QoE for different users.

We utilized the O-RAN architecture, enabling open interfaces such as the E2. This allowed us to obtain the necessary RAN metrics and incorporate them into the algorithm used in the proposed DRB Management xApp. As a result, we were able to employ an additional DRB and route specific traffic through it.

In the initial testing phase, we assessed and confirmed the individual performance of each component within our testbed. Upon successful validation, the subsequent test scenario was focused on validating the proper functioning of a private 5G network, as well as ensuring that all previously tested components are able to communicate with one another. Finally, the third set of tests encompassed two experimental scenarios designed to validate the developed solution in an emulated environment that emulates real-world use cases. The first use case scenario involved a single UE connected to the disaggregated gNB, while triggering the xApp. The second use case scenario involved two UEs connected to the disaggregated gNB, both triggering the xApp. Both use case scenarios involved the creation of additional DRBs and the rerouting of traffic according to the desired pattern.

The experimental evaluation conducted led to the conclusion that the proposed solution represents a valuable contribution to the implementation of network slicing in 5G network through xApps, which are capable of reconfiguring the RAN to create additional DRBs and reroute traffic in order to meet the targeted QoS levels. The successful outcome of the tests indicates that the developed solution adeptly manages differentiated traffic flows, ensuring appropriate prioritization of latency-sensitive traffic. Moreover, the results demonstrate that the DRB management ensured by the proposed xApp performs as intended, thereby contributing to enhanced network performance.

Chapter 5

Conlusion

5.1 Conclusions

The main objective of this dissertation was to develop, implement and validate a solution enabling the creation of dynamic network slices within a private standalone 5G network based on the O-RAN architecture. The proposed solution allowed us to overcome some limitations related to availability and QoS degradation in 5G network deployments.

On the one hand, 5G plays a pivotal role in enhancing the QoS and QoE for mobile users by providing increased data rates, reducing latency to ultra-low levels, enhancing reliability, and optimizing the management of communications resources. In addition to networking architecture improvements, 5G also offers significant advancements in network capacity and spectrum efficiency, facilitating a more interconnected environment where a massive number of devices can communicate seamlessly. The introduction of network slicing in 5G enables the customization of the network to effectively meet the diverse requirements of different users. This method allows for multiple virtual networks on the same physical infrastructure, with each network slice dedicated to a specific service type or user group. Network slicing improves flexibility, scalability, and security, as each network slice is isolated.

On the other hand, the emerging O-RAN paradigm proposes a novel approach to deploying RANs without relying on closed proprietary solutions. O-RAN offers the potential for faster, more resilient, easily manageable, and reconfigurable network deployments, which can reduce environmental impact and provide significant cost savings for operators. Moreover, the O-RAN architecture allows operators to control and extract metrics on the network's resource usage through open interfaces and develop software applications, called rApps and xApps, tailored to customize networks to meet specific user requirements.

The integration of the 5G network slicing technique with the O-RAN architecture introduces significant advancements in the realm of mobile communications. This convergence aims at improving network infrastructure management by providing heightened levels of flexibility and availability. By leveraging the inherent capabilities of network slicing within 5G deployments alongside the open architecture principles of O-RAN, this approach enhances the performance of mobile

networks by continuously ensuring the QoS levels are met.

The proposed solution combines the 5G technology and O-RAN, leveraging both network slicing and xApps. It consists of a private 5G SA network composed of a 5G Core Network and a disaggregated gNB, comprising a Central Unit and a Distributed Unit. Additionally, it includes multiple User Equipments. These components were implemented using the OpenAirInterface open-source software package. Moreover, the work included the deployment of the Near-RT RIC and the development of a novel xApp capable of controlling network slices by dynamically managing Data Radio Bearers using metrics collected from the Radio Access Network. The FlexRIC open-source software package implements the Near-RT RIC, which runs the xApp. All components described were deployed on a single computer.

The solution comprises two main functionalities. The first functionality is related to the DRB Management xApp. This software application is able to collect metrics, through the E2 interface about the network's resource usage to analyse when the network is too congested and the assigned QoS levels are not being met, generating control actions to create and release a Data Radio Bearer. The second functionality is related to traffic mapping. When the conditions of the network cause the xApp to trigger and the RAN to reconfigure itself, traffic is mapped to a suitable Data Radio Bearer, ensuring latency-sensitive applications can still operate through periods of bandwidth-intensive communications, such as data downloads or iPerf3 sessions.

Throughout the development of the solution, several challenges were encountered. The primary challenge revolved around the time taken to establish a functional 5G SA network, which constituted the initial phase of the solution's development. The complexity of the code and insufficient documentation of OAI delayed the development process. Moreover, the existing OAI documentation outlined features that were either incompletely implemented or not implemented in an anticipated manner, resulting in wasted time in experimentation. Additionally, as the proposed solution necessitated the utilization of a specific OAI branch, which had features yet to be integrated, numerous stability issues and errors limited the software execution. Another significant challenge was related to the demanding hardware requirements of both software packages. While the implementation on a singular computer with lower specifications than indicated in the documentation proved sufficient for the task, it still led to intermittent crashes.

The mentioned challenges significantly limited the testing and validation of the entire system. The deployment of different components and the seamless interoperability among them posed difficulties. A single faulty component or malfunction could lead to errors requiring the entire testing process to be stopped and relaunched, increasing the proposed solution's validation time.

Despite all the challenges that occurred and all the improvements that can be made to the proposed solution, the main objectives of the dissertation were fully achieved, resulting in three main contributions: 1) an O-RAN 5G Base Station capable of offering dynamic network slices, and 2) a novel xApp designed for dynamic DRB management, leveraging real-time monitoring of networking parameters to enable the creation of 5G network slices and 3) an emulated environment built on OpenAirInterface to validate and evaluate the proposed solution.

5.2 Known Limitations & Future Work

The present dissertation has successfully achieved all proposed objectives, but there are certain system limitations and areas where improvements can be made.

The main limitation pertains to the incapacity of the 5G Core Network to create multiple QoS flows through the N3 interface. Currently, the UPF can only create and configure one QoS flow. As per the 5G specification, the UPF should be able to support multiple QoS flows, each associated with different IP streams. In the proposed solution, an alternative approach was employed where the SDAP layer examines packet headers from IP streams originating from the QoS flow and maps the traffic to DRBs according to specific criteria. This also limits the mapping of traffic to DRBs in the uplink channels, since having only one QoS flow makes it impossible to predict the behaviour of the UEs when putting the data in DRBs for the uplink. The primary improvement may involve creating and configuring more than one QoS flow, enabling the SDAP layer to map QoS flows to DRBs rather than the IP streams themselves and allowing UEs to put traffic accordingly in DRBs in the uplink channels.

A minor limitation is associated with the 5QI parameters. Among the parameters linked to 5QI, only the Default Priority Level is appropriately configured for most of the potential 5QI indicators, including 5QI 3 and 5QI 9, which were considered in testing the proposed solution. The remaining indicators are only correctly implemented for the 5QI 9, which is the default 5QI. In future works, more 5QI parameters may be implemented, which will facilitate more effective performance testing of the solution and significantly enhance communications employing different DRBs associated with different 5QI values.

One additional minor limitation is related to the thresholds utilized to trigger the DRB Management xApp, which leads to RAN reconfiguration and the creation or release of a new DRB. The proposed solution considers the 5QI values associated with the default DRB, combined with a priori observation of the collected metrics to ascertain the optimal threshold. Since our solution uses only one computer for deploying the entire 5G network, which limits some of the results by the CPU usage of the computer used for the testing, the threshold could not be solely defined using the associated 5QI. An enhancement may involve the use of additional physical equipment, such as SDRs and more computers, to implement the different network components, thereby mitigating potential limitations that may arise when relying solely on a single computer.

References

- [1] “United Nations Sustainable Development Goals.” Available from: <https://sdgs.un.org/goals>.
- [2] S. Davis and et al., “Ericsson Mobility Report Letter,” 2023.
- [3] T. Sivalingam, K. B. Shashika Manosha, N. Rajatheva, M. Latva-aho, and M. B. Disanayake, “Positioning of Multiple Unmanned Aerial Vehicle Base Stations in Future Wireless Network,” in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–6, 2020.
- [4] 3GPP, “System architecture for the 5G System (5GS), 3GPP TS 23.501 V18.3.0,” 2023. Available: <http://www.3gpp.org>.
- [5] 3GPP, “NR; NR and NG-RAN Overall description; Stage-2, 3GPP TS 38.300 version 16.10,” 2022. Available: <http://www.3gpp.org>.
- [6] Devopedia, “5g quality of service,” 2021. Available from: <https://devopedia.org/5g-quality-of-service>.
- [7] ITU, “IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond,” *Recommendation ITU-R M.2083-0*, September 2015.
- [8] M. Ricardo, “The 5th Generation of Cellular Networks.” PowerPoint presentation, 2022.
- [9] ShareTechNote, “5G/NR - Network Slicing,” 2023. Available from: https://www.sharetechnote.com/html/5G/5G_NetworkSlicing.html.
- [10] M. Polese, L. Bonati, S. D’Oro, S. Basagni, and T. Melodia, “Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges,” *IEEE Communications Surveys and Tutorials*, vol. 25, 2023.
- [11] E. Jordan, “Open RAN functional splits, explained,” 2021. Available from: <https://www.5gtechnologyworld.com/open-ran-functional-splits-explained/>.
- [12] D. Wypiór, M. Klinkowski, and I. Michalski, “Open RAN;Radio Access Network Evolution, Benefits and Market Trends,” *Applied Sciences*, vol. 12, 2022.
- [13] M. Karbalaee Motalleb, V. Shah-Mansouri, S. Parsaeefard, and O. L. Alcaraz López, “Resource Allocation in an Open RAN System Using Network Slicing,” *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 471–485, 2023.
- [14] O.-R. W. G. 1, “O-RAN Slicing Architecture,” tech. rep., O-RAN Alliance, 2023.

- [15] D. M. de Almeida Coimbra Maia and M. Ricardo, “Control and Positioning of a 5G Radio Access Node Deployed in a Mobile Robotic Platform,” tech. rep., Faculdade de Engenharia da Universidade do Porto, 2022.
- [16] G. Liu, Y. Huang, Z. Chen, L. Liu, Q. Wang, and N. Li, “5G Deployment: Standalone vs. Non-Standalone from the Operator Perspective,” *IEEE Communications Magazine*, vol. 58, no. 11, pp. 83–89, 2020.
- [17] 3GPP, “NG-RAN; NG Application Protocol (NGAP),” Technical Specification 38.413, , 2022.
- [18] G. M. T. F. Queirós and M. Ricardo, “Autonomous Control and Positioning of a Mobile 5G Radio Access Node Employing the O-RAN Architecture,” tech. rep., Faculdade de Engenharia da Universidade do Porto, 2023.
- [19] 3GPP, “General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface,” Technical Specification 29.060, , 2022.
- [20] ShareTechNote, “5G/NR - Radio Protocol Stack Architecture,” 2023. Available from: https://www.sharetechnote.com/html/5G/5G_RadioProtocolStackArchitecture.html.
- [21] ShareTechNote, “5G/NR - QoS,” 2023. Available from: https://www.sharetechnote.com/html/5G/5G_QoS.html.
- [22] S. Zhang, “An Overview of Network Slicing for 5G,” *IEEE Wireless Communications*, vol. 26, no. 3, pp. 111–117, 2019.
- [23] L. Gavrilovska, V. Rakovic, and D. Denkovski, “From Cloud RAN to Open RAN,” *Wireless Personal Communications*, vol. 113, 08 2020.
- [24] S. Niknam, A. Roy, H. S. Dhillon, S. Singh, R. Banerji, J. H. Reed, N. Saxena, and S. Yoon, “Intelligent O-RAN for Beyond 5G and 6G Wireless Networks,” in *2022 IEEE Globecom Workshops (GC Wkshps)*, pp. 215–220, 2022.
- [25] O-RAN ALLIANCE. Available from: <https://www.o-ran.org/>.
- [26] O-RAN Work Group 1, “O-RAN Architecture Description,” tech. rep., O-RAN Alliance, 2023.
- [27] R. P. Alves, J. Guilherme, A. Da, S. Alves, M. R. Camelo, W. O. D. Feitosa, V. F. Monteiro, F. Rodrigo, and P. Cavalcanti, “Experimental Comparison of 5G SDR platforms: srsRAN x OpenAirInterface,” 2023.
- [28] R. Schmidt, “2023-06 O-RAN F2F Osaka.” Available from: <https://wiki.o-ran-sc.org/display/EV/2023-06+O-RAN+F2F+Osaka>.
- [29] Y. R. Wei, A. S. Keshavamurthy, R. Wittmann, and A. R. Zahonero, “A Standalone 5G Industrial Testbed Design Considerations for Industry 4.0,” in *2022 52nd European Microwave Conference (EuMC)*, pp. 884–887, 2022.
- [30] 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*, vol. 56, no. 5, pp. 184–192, 2018.

- [31] J. García-Morales, M. C. Lucas-Estañ, and J. Gozalvez, “Latency-Based 5G RAN Slicing Descriptor to Support Deterministic Industry 4.0 Applications,” in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1359–1362, 2019.
- [32] A. Ksentini and N. Nikaein, “Toward Enforcing Network Slicing on RAN: Flexibility and Resources Abstraction,” *IEEE Communications Magazine*, vol. 55, no. 6, pp. 102–108, 2017.
- [33] A. Kak, V.-Q. Pham, H.-T. Thieu, and N. Choi, “ProSLICE: An Open RAN-based approach to Programmable RAN Slicing,” in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 197–202, 2022.
- [34] P. Gorla, “OAI 5G SA Setup Guide,” 2023. Available from: <https://hackmd.io/@praveeng/6GMLAB-OAI-5G-SETUP-GUIDE>.
- [35] “OpenAirInterface (OAI).” Available from: <https://openairinterface.org>.
- [36] “srsRAN.” Available from: <https://www.srsran.com/>.
- [37] “OAI Core Network.” Available from: <https://openairinterface.org/oai-5g-core-network-project/>.
- [38] “OAI Radio Access Network.” Available from: <https://openairinterface.org/oai-5g-ran-project/>.
- [39] “FlexRIC.” Available from: <https://gitlab.eurecom.fr/mosaic5g/flexric>.
- [40] “O-RAN SC Near-RT RIC.” Available from: <https://wiki.o-ran-sc.org/display/RICP/Introduction+and+guides>.
- [41] “OAI Repository - Branch qoe-e2.” Available from: https://gitlab.eurecom.fr/oai/openairinterface5g/-/tree/qoe-e2?ref_type=heads.
- [42] “Rf Simulator Wiki.” Available from: <https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/radio/rfsimulator/README.md>.
- [43] “iPerf - The ultimate speed test tool for TCP, UDP and SCTP.” Available from: <https://iperf.fr/>.