

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Open Source Non-public 5G Networking in Licensed and Unlicensed Bands

José Pedro Nogueira Rodrigues

DISSERTATION



Master's Degree in Informatics and Computing Engineering

Supervisors: Ana Cristina Costa Aguiar & Adriano Almeida Goes

October 9, 2023

Open Source Non-public 5G Networking in Licensed and Unlicensed Bands

José Pedro Nogueira Rodrigues

Master's Degree in Informatics and Computing Engineering

October 9, 2023

Abstract

The innovative 5G technology is shaking up the current panorama of the cellular networking industry. It aims to accelerate the digitization of the many sectors that depend on it to thrive in the ever-growing digital age. This new fifth-generation standard elevates the requirements on speed, bandwidth, availability, coverage, latency, energy usage, connected devices, and battery life for operating devices. 5G also brings numerous architectural changes, which, among other things, allows for the virtualization of all the composing elements and the definition of open standard interfaces among them.

These changes 5G technology presents facilitate the development of new and easier-to-set-up non-public network solutions, networks that provide mobile services for a distinctly defined collection of users or devices part of a single organization, not subject to public telecom operators' regulation or public service commitment. Notably, the components and tools that enable establishing and operating such networks are currently under development. The emergence of 5G technology has also led to a significant development in the open-source movement, which is a direct result of the virtualization and separation of hardware-software integration that characterizes this latest generation of cellular networking technology.

This project aims to plan and deploy a non-public network on the FEUP campus using virtualized open-source components. A future stage of this project would involve deploying the network in the unlicensed spectrum, potentially leading to collisions with existing networks. A Listen-before-Talk algorithm will be evaluated and partially deployed to address this issue, incorporating untested components and incomplete functionality, including pending mock functions yet to be implemented. The network will contribute to the emerging ecosystem by verifying and refining existing features on existing or entirely new components. This network will be tested and validated using two use cases: the attachment of a device and data session to the Internet. The primary performance metric to assess the network deployment will be the efficiency and timeliness of component operations and overall system functionality.

This study and testing will help researchers and developers involved with developing, expanding, validating, and exploiting new or existing high-performance 5G components and technologies. The knowledge gained from this project will also advance the technological state-of-the-art and know-how in these components and tools by providing comprehensive documentation and facilitating the learning process for newcomers by illuminating a previously obscure field. This knowledge can be used in non-public enterprise networks with critical wireless communication requirements in public safety, infrastructure, and industry.

Keywords: *Non-public network, 5G, Open-source, Component virtualization*

ACM Classification:

CCS → *Networks* → *Network types* → *Mobile Networks*

CCS → *Networks* → *Network services* → *Network management*

CCS → *Networks* → *Network performance evaluation* → *Network measurement*

Acknowledgements

First and foremost, I want to extend my heartfelt appreciation to my mentor, **Ana Cristina Costa Aguiar**, a remarkable figure in this academic journey. Her unwavering guidance, continuous support, and profound expertise have been the cornerstones of my research experience. I am deeply thankful for her enduring patience and dedication to helping me surmount every technical hurdle and engage in illuminating discussions. Her contributions have profoundly enhanced the substance and sophistication of my work.

I also sincerely thank my co-mentor, Dr. Adriano Almeida Goes from Capgemini Engineering, for his invaluable support and assistance throughout my research. Adriano's extensive knowledge and expertise were instrumental in shaping the direction of my work and overcoming various challenges. His contributions have been truly invaluable, and I am honored to have had the opportunity to collaborate with him.

I also want to express my sincere gratitude to **Sofia Cardoso Martins**, a valuable IT team member from the Instituto de Telecomunicações. Her guidance, support, and expertise were instrumental throughout my research journey. I am genuinely grateful for her patience and willingness to assist me in overcoming technical challenges and providing insightful discussions. Their contributions have significantly enriched the quality and depth of my work.

I also express my sincere gratitude to **Cédric Roux**, a Research Engineer at **EURECOM - Communication Systems**, for his priceless assistance and support during my research. His willingness to share his knowledge, provide constructive feedback, and offer solutions to complex problems have been instrumental in completing this dissertation.

I would like to express my heartfelt gratitude to my family, significant other, and cherished friends, who have been my unwavering pillars of strength throughout my academic and personal journey. Their unshakable belief in me during the most challenging moments has been a constant source of inspiration and motivation. I am profoundly fortunate to have such an incredible support network, and I owe a significant part of my accomplishments to their presence in my life.

Finally, I acknowledge IT (UIDB/50008/2020) for hosting this research work under the scope of Project FLOYD (POCI-01-0247-FEDER-045912), funded by the European Regional Development Fund (FEDER) through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) and by National Funds (OE), through Fundação para a Ciência e Tecnologia, I.P., and funded through applicable funds (FCT/MCTES) (PIDDAC).

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation	1
1.3	Goals	2
1.4	Document Structure	3
2	Background and State of the art	4
2.1	Background	4
2.2	Introduction to 5G network architecture	5
2.2.1	Core Network (5GC)	5
2.2.2	Radio Access Network (RAN)	8
2.2.3	User Equipment (UE)	11
2.3	Frequency Bands in Wireless Networks	11
2.4	The unlicensed band	12
2.4.1	C-Band	12
2.4.2	5G New Radio	13
2.4.3	5G New Radio Unlicensed	13
2.4.4	Cognitive Radio	14
2.4.5	Listen-before-Talk	14
2.5	State of the art	15
3	Methodology	18
3.1	Problem definition	18
3.2	NPN implementation	19
3.2.1	Minimum SBA deployment	19
3.2.2	Technology Selection	19
3.3	Coexisting in the Unlicensed Bands	20
3.3.1	Challenges of Spectrum Sharing	21
3.3.2	Identifying the Deployment of the LBT Algorithm	21
3.3.3	Validation of Network	24
4	Implementation	30
4.1	Network Deployment Strategy	30
4.1.1	Radio Access Network (RAN)	30
4.1.2	5G Core	32
4.1.3	Network Initialization	34
4.1.4	Hardware configuration	39
4.2	LBT protocol implementation	40

4.2.1	Challenges and Status	43
5	Validation and results	49
5.1	Execution of the Use Cases	49
5.1.1	Device Registration Use Case	49
5.1.2	Data Transfer Sessions Use Case	64
6	Performance	69
6.1	Experimental setup	69
6.2	Throughput Analysis	69
6.3	Round-trip time analysis	72
6.3.1	Baseline	72
6.3.2	With Added Delay	72
6.4	Use Case 1: Registration of a UE in the 5G Network	73
6.4.1	Baseline	73
6.4.2	With Added Delay	73
6.5	Use Case 2: Data Transfer	74
6.5.1	Baseline	75
6.5.2	With Added Delay #1	75
6.5.3	With Added Delay #2	75
7	Conclusion	77
	References	78
A	Code Listings	81
A.1	5G Core	81
A.1.1	amf.yaml	81
A.1.2	mme.yaml	82
A.2	Control Unit (CU)	82
A.2.1	cu_gnb.conf	82
A.2.2	nr_nas_msg_sim.c	84
A.3	Distributed Unit (DU)	84
A.3.1	channel_occupancy.h	84
A.3.2	channel_occupancy.c	85
A.3.3	CMakeLists.txt	85
A.3.4	nr_prach_procedures.c	86
A.3.5	du_gnb.conf	87
A.4	User Equipment (UE)	89
A.5	Utils	89
A.5.1	readme.txt	89

List of Figures

2.1	NPN deployment categories	4
2.2	5G Service-Based Architecture	6
2.3	Portugal frequency spectrum. Source: [28]	12
2.4	Listen-before-Talk protocol	15
3.1	n46 unlicensed spectrum band	20
3.2	n102 unlicensed spectrum band	21
3.3	Contention Based RACH Procedure (CBRA)	23
3.7	Device Registration Use-Case (Generated with EventStudio System Designer) . .	28
3.8	Data Transfer Session Use-Case (Generated with EventStudio System Designer) .	29
4.1	CU/DU network configuration	31
4.2	Network Diagram	34
4.3	Open5Gs WebUI: Add subscriber data	35
4.4	Open5Gs WebUI: Additional mandatory subscriber data	36
4.5	DU Log showing LBT protocol in action	47
5.1	Wireshark capture: Registration reject (Semantically incorrect message)	53
5.2	Wireshark capture: InitialUEMessage, Registration request	55
5.3	Wireshark capture: InitialUEMessage, NAS Identity	56
5.4	Wireshark capture: DownlinkNASTransport, Authentication request	57
5.5	Wireshark capture: UplinkNASTransport, Authentication response	58
5.6	Wireshark capture: DownlinkNASTransport, Security Mode Command	58
5.7	Wireshark capture: Security Mode Command, IMEISV Request)	59
5.8	Wireshark capture: Registration reject (Security Mode Rejected, unspecified) . .	60
5.9	Wireshark capture: InitialContextSetupRequest	61
5.10	Initial Context Setup Response: Not implemented in the OAI code base	64
5.11	Wireshark capture: Establishment of GTP tunnel	65
5.12	Wireshark capture: PFCP Session Modification Request	65
5.13	Wireshark capture: PFCP Session Modification Request	66
5.14	DNS-related HTTP request failure	66
5.15	DNS-free HTTP request success	67
5.16	DNS-related HTTP request success	67
6.1	Pareto distribution delay graph Base delay: 3ms Additional variable delay: 2ms	71
6.2	Pareto distribution delay graph Base delay: 3ms Additional variable delay: 5ms	74

List of Tables

2.1	Comparison of Frameworks for 5G Core Implementation [9][18]	8
2.2	Comparison of Frameworks for 5G RAN Implementation	10
2.3	C-Band bands, frequencies, and channel bandwidths	13
4.1	OAI RAN Repository Structure	40
6.1	Throughput Analysis Results	70
6.2	Round-trip ping times - Baseline	72
6.3	Round-trip ping times - With Added Delay	72
6.4	Use Case 1 - Baseline	73
6.5	Use Case 1 - With Added Delay	73
6.6	Use Case 2 - Baseline: File download times and throughput	75
6.7	Use Case 2 - With Added Delay of 3ms with 2ms jitter: File download times and throughput	75
6.8	Use Case 2 - With Added Delay of 3ms with 5ms jitter: File download times and throughput	75

List of Abbreviations and Symbols

3GPP	The 3rd Generation Partnership Project
5GC	5G Core
5G	Fifth generation of cellular wireless technology
APN	Access Point Name
CCA	Clear Channel Assessment
COT	Channel Occupancy Time
CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
CU	Centralized Unit
CW	Contention Window
DNN	Data Network Name
DNS	Domain Name System
DU	Distributed Unit
eMBB	Enhanced Mobile Broadband
eNB/eNodeB	Upgraded radio base station for 4G LTE that connects to a 5G core network
FEUP	Faculdade de Engenharia da Universidade do Porto
gNB/gNodeB	3GPP-compliant implementation of the 5G-NR base station
GPRS	General Packet Radio Service
GTP	GPRS Tunnelling Protocol
ICMP	Internet Control Message Protocol
IMSI	International Mobile Subscriber Identifier
LAA	License Assisted Access
LBT	Listen-before-Talk
MCC	Mobile Country Code
MIMO	Multiple Input Multiple Output

MNC	Mobile Network Code
MNO	Mobile Network Operator
N3IWF	Non-3GPP Interworking Function
NAS	Non Access Stratum
NPN	Non-Public Network
NR-U	New Radio in the Unlicensed Band
PLMN	Public Land Mobile Network
PN	Public Network
RAP	Random Access Procedure
RRC	Radio Resource Control
RU	Radio Unit
SBA	Service-based architecture
SDR	Software Defined Radio
SNPN	Standalone Non-Public Network
SUCI	Subscription Concealed Identifier
SUPI	Subscription Permanent Identifier
TAC	Tracking Area Code
TEID	Tunnel Endpoint Identifier
TS	Technical Specification
URLLC	Ultra-Reliable Low Latency Communications
USRP	Universal Software Radio Peripheral

Chapter 1

Introduction

1.1 Context

Introducing 5G technology has presented both opportunities and challenges for communication networks. One key area of discussion centers around 5G non-public networks (NPNs), raising questions about their value and potential impact. This debate arises from the unique characteristics of NPNs, notably their limited regulation compared to public networks, distinguishing them from traditional public cellular networks, as 5G NPNs are independent cellular networks not accessible to the general public and restricted to authorized users or devices. This malleability allows companies from the logistics and manufacturing sectors to increase flexibility, productivity, and usability in the industrial processes executed within their factory premises by meeting very stringent requirements regarding latency, reliability, and high-accuracy positioning, which NPNs can achieve [23].

An open-source movement is also emerging among the ecosystem of NPNs, consisting of network infrastructures without exclusive ownership or control by a single entity [25], which seeks to open up the development ecosystem for network software. This would allow new small and medium players to enter the market and increase the competition, which benefits end-users.

This research work is framed within the scope of the **FLOYD** project, which aims to improve mobility services and accelerate the integration of autonomous driving in society. This research and innovation project led by Capgemini Engineering intends to advance the networking and computation technologies on which complex mobility and autonomy services and functionalities can be built. The project is subtitled *5G/SDN Intelligent Systems For LOw latencY V2X communications in cross-Domain mobility applications*.

1.2 Motivation

Several reasons motivate the research of non-public networks, particularly in the context of 5G technology. One major reason is the increased demand for high-speed, low-latency communication in various industries, such as manufacturing, transportation, and health care [27]. Non-public

networks can provide isolated, secure communication channels for these industries, allowing for more efficient operations and better protection of sensitive data. That's because they are designed to meet specific requirements and provide a high degree of control and customization. In contrast, traditional public cellular networks are operated by telecom service providers and are open to any user with a service contract. They are customized to address the prevalent requirements of the general public, a configuration that might prove inadequate for certain specialized industries. Non-public networks, particularly those based on 5G technology, can be tailored to meet the specific needs of distinctive industries [23].

Another reason for researching non-public networks is greater control and customization of network resources. Public networks, such as those provided by telecommunications companies, may not meet the specific needs of particular industries or organizations. By creating a non-public network, an organization can have more control over network resources and configure them to meet their specific requirements.

With the rapid advancement of 5G technology, there is a significant opportunity for new players in the market to innovate and develop solutions for non-public networks. The absence of comprehensive resources and complete open-source implementations, often coupled with confusing information and limited assistance for common problems in this area presents a research and innovation opportunity for companies, organizations, and researchers. This lack of information also highlights the need for further research and development to enhance the technical proficiency and practical experience in deploying and managing non-public networks.

1.3 Goals

This research project addresses some key challenges and issues associated with 5G non-public networks. The first goal of this project is to identify, document, and explore these challenges, which will provide a foundation for the following objectives. The second goal is to develop and test a proof-of-concept for a 5G non-public network using open-source software. This will provide practical experience working with these tools and technologies and demonstrate open-source software's capabilities in 5G non-public networks. Lastly, the third objective of this project is to establish a protocol for interference avoidance while operating within the unlicensed spectrum. This protocol is essential because, in Portugal, there is no dedicated spectrum allocation for Non-Private Networks (NPN), and the unlicensed spectrum necessitates the use of Listen Before Talk (LBT) protocols. This protocol will ensure regulatory adherence and provide a high-quality service to network users. To assess the network's performance comprehensively, we will analyze critical metrics, including download and upload speeds (throughput), latency, and jitter.

The accomplishment of these objectives significantly bolsters the open-source community. This project streamlines the deployment of 5G non-public networks, furnishing comprehensive guidance that empowers newcomers to navigate the complexities effortlessly, thus fostering open-source network development and advancement. Furthermore, performance evaluations will offer

critical insights into the network's operational efficiency. These findings serve as valuable benchmarks and inform potential adopters about the practicality of open-source solutions in the context of 5G non-public networks, ultimately guiding informed decision-making for network operators and service providers. It is also important to emphasize that all artifacts generated by this research project will be publicly disseminated to facilitate broader advancement.

1.4 Document Structure

Beyond the introduction, this dissertation contains six more chapters. In Chapter 2, background knowledge on the subject is presented, and similar works are studied and explained. In Chapter 3, the problem in question is defined, and the research methodology is presented. In Chapter 4, the network implementation process, as previously outlined, is presented. In Chapter 5, the findings and outcomes from the research methodology described in the previous chapter will be presented, providing insights and interpretations to the research questions addressed in this dissertation. In Chapter 6, the established 5G network infrastructure will be assessed, including performance, registration times, ping times, and user experiences in controlled scenarios. Finally, Chapter 7 will summarize key findings, reflect on the research process, and suggest future directions. To conclude this thesis, references and appendices used in its development are presented at the end.

Chapter 2

Background and State of the art

This chapter offers a perspective into the foundational background and the latest advancements in the field. It begins by providing essential context, highlighting prior research, and introducing the fundamental concepts relevant to our study. Subsequently, it transitions into an examination of the cutting-edge state of the art, where we explore the most recent technologies, methodologies, and breakthroughs that shape the current landscape of our research domain.

2.1 Background

In the context of Non-Public Networks (NPNs), several potential network deployment strategies and configurations are available[4], each tailored to specific use cases, operational requirements, and technological considerations.

Two distinct categories of deployments exist for Non-Public Networks (NPNs). The first is Standalone NPN, often referred to as SA, where all network functions are situated within the confines of the organization's premises. These networks are entirely separate from the public network (PN). The second category involves NPNs integrated into the Public Network. In this configuration, some or potentially all of the infrastructure is shared with the Public Network. This type is commonly referred to as NSA (Non-Standalone). Figure 2.1 displays several NPN deployment possibilities.

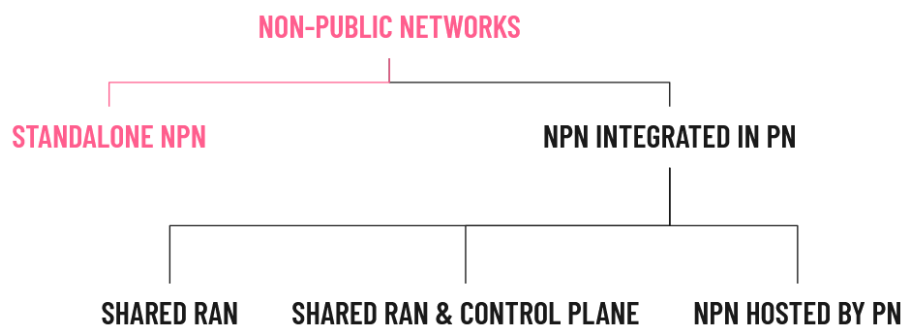


Figure 2.1: NPN deployment categories

The Standalone Non-Public Network (SNPN) is a private network built on the 3GPP 5G system architecture, separate from any Public Land Mobile Network (PLMN). This independence is evident in several ways, such as having its unique identifier, the NPN ID, which is distinct from the PLMN ID, and a fully deployed 5G system (including a Radio Access Network and Core Network) within the organization's boundaries. The fact that the NPN's Core Network is separate from the PLMN's Core Network means that all data, signaling traffic, and user plane flows from NPN devices stay within the organization's perimeter and do not cross into the PLMN. As a result, NPN devices are considered non-public network subscribers by definition.

The Public Network Integrated Non-Public Network (PNI-NPN) is a private network that uses the 3GPP 5G system architecture but is also deployed alongside a Public Land Mobile Network (PLMN). This type of NPN comprises one public sub-network and one or more private sub-networks. The public sub-network includes network functions provided by the PLMN and is under the Mobile Network Operator's (MNO) control. These functions are typically deployed outside of the organization. On the other hand, the private sub-network consists of network functions isolated from the PLMN and located within the organization. This deployment has three possible variations. In the first one, the RAN is shared between the owner and the mobile operator, while other network functions are segregated as in the case of SNPN. In the second scenario, the deployment contains a shared RAN and control plane, where the public network performs the network control tasks. In the third and last scenario, the public network hosts the NPN and can be implemented using APN or network slicing.

2.2 Introduction to 5G network architecture

The architecture of a 5G network is a complex and highly distributed system[8], consisting of several key components that work in harmony to deliver advanced wireless communication services. At a high level, a 5G network can be broadly categorized into three main components: the core network, the radio access network (RAN), and the user equipment (UE). In this section, we will provide a brief overview of each of these architectural elements.

2.2.1 Core Network (5GC)

At the heart of a 5G network lies the core network, often called the 5G Core (5GC). The 5GC serves as the central intelligence of the network and is responsible for managing various services and functions. It is designed with flexibility and scalability, allowing it to adapt to diverse communication needs.

Key functionalities of the 5GC include network slicing, enabling the creation of virtualized, dedicated networks for specific use cases, and support for emerging technologies like the Internet of Things (IoT) and edge computing.

2.2.1.1 Service-based architecture

5G Core technology introduces a new concept called Service-Based Architecture (SBA)[11]. The SBA is a modular and scalable network architecture for 5G, which enables the creation and deployment of new network services and functionalities flexibly and efficiently.

The Service-Based Architecture (SBA) is a fundamental part of the 5G network architecture, and it applies to the core network, where network functions are broken down into smaller, independent components called services.[16] These services can be combined and integrated into various ways to form new network functions. They can be deployed and managed independently, allowing for more efficient and agile network operations. Compared to previous networks, the SBA enables greater automation and orchestration of network functions and supports the development of new and innovative services and use cases.

The 5G SBA represents a major shift in how 5G networks are designed, deployed, and managed. The traditional network architecture based on monolithic network functions is replaced by a modular and flexible architecture composed of smaller, independent components. By providing a flexible and scalable network architecture, the SBA is a key element of the 5G technology roadmap that supports the growth and evolution of 5G networks to meet the diverse needs of different industries. Figure 2.2 shows what a minimal SNPN SBA could look like.

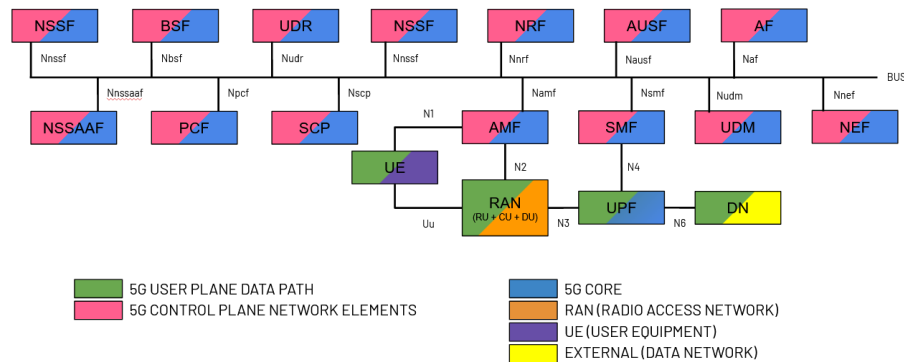


Figure 2.2: 5G Service-Based Architecture

In the 5G SBA architecture, two planes can be distinguished. The user plane is responsible for transporting data between the end-user devices and the applications or services they are accessing. It includes the functions for packet processing, forwarding, and routing of user data. On the other hand, the control plane is responsible for the signaling between the various network functions to set up, manage, and tear down connections for the user plane. It includes network resources' authentication, authorization, accounting functions, network management, and orchestration. Figure 2.2's components are explained in the following list:

- **NRF (NF Repository Function):** The NRF maintains the repository of Network Functions available in the network.

- **SCP (Service Communication Proxy):** The SCP provides an interface between services in the SBA and the underlying network functions.
- **AMF (Access and Mobility Management Function):** The AMF is responsible for managing access to the network and mobility of devices between different network slices.
- **SMF (Session Management Function):** The SMF manages the establishment, modification, and termination of sessions between the user equipment and the network.
- **UPF (User Plane Function):** The UPF handles user data traffic in the network.
- **AUSF (Authentication Server Function):** The AUSF authenticates and authorizes user equipment and provides security information for network functions.
- **UDM (Unified Data Management):** The UDM stores user-related data, such as subscription and device information.
- **UDR (Unified Data Repository):** The UDR stores network-related data, such as network slice and policy information.
- **PCF (Policy and Charging Function):** The PCF enforces policies and charging rules for network services.
- **NSSF (Network Slice Selection Function):** The NSSF selects the appropriate network slice for the user equipment based on operator policies and user preferences.
- **BSF (Binding Support Function):** The BSF provides binding support for the authentication of user equipment.
- **NSSAAF (Network Slice and Service Authorization and Accounting Function):** The NSSAAF provides authorization and accounting support for network slices and services.
- **AF (Application Function):** The AF provides network services to applications and manages traffic flows between the applications and network functions.
- **NSSF (Network Slice Selection Function):** The NSSF selects the appropriate network slice for the user equipment based on operator policies and user preferences.
- **NEF (Network Exposure Function):** The NEF provides a standardized interface for external systems to access network functions and services.

Source: [3]

2.2.1.2 Open-source solutions

The table below compares the features and limitations of three popular 5G core implementation frameworks: Open5GS, OpenAirInterface (OpenAI), and Free5GC:

Table 2.1: Comparison of Frameworks for 5G Core Implementation [9][18]

Feature/Limitation	Open5GS	OpenAirInterface (OpenAI)	Free5GC
License	AGPLv3	Apache 2.0	Apache 2.0
Core Features	AMF, SMF, UPF, AUSF, NRF, UDM, UDR, PCF, NSSF, BSF, SCP	AMF, SMF, UPF, AUSF, NRF, UDM, UDR, NSSF	AMF, SMF, UPF, AUSF, NRF, UDM, UDR, NSSF
Programming language	C	C	Go
Deployment requirements	Docker/Container	Docker/Container	Virtual Machine
Documentation	Organized and chronological	Organized and structured	Unordered flowchart
Community Support	Large and active	Large and active	Medium-sized and growing
3GPP Release	16	15	15
N3IWF Compatability	Third-party integration	Third-party integration	Third-party integration

All three frameworks have similar core features, including AMF, SMF, UPF, AUSF, NRF, UDM, UDR, and NSSF, with OpenAI and Free5GC lacking BSF, PCF, and SCP. However, Open5GS and OpenAI are implemented in C, while Free5GC is implemented in Go. Open5GS and OpenAI require Docker/container-based deployment, while Free5GC requires virtual machine deployment. The documentation for Open5GS and OpenAI is more organized than that of Free5GC. Open5GS and OpenAI have large and active communities, while Free5GC's community is still growing. Finally, Open5GS is the most updated, compatible with 3GPP Release 16, while OpenAI and Free5GC are only compatible with Release 15. N3IWF compatibility refers to a 5G network function's ability to interwork with a third-party network function called the N3IWF (Non-3GPP Interworking Function), enabling communication between a 5G network and non-3GPP networks, such as Wi-Fi or Ethernet. All three frameworks allow third-party integration of this element.

2.2.2 Radio Access Network (RAN)

The Radio Access Network (RAN) is the segment of the 5G architecture responsible for connecting user equipment (UE) to the core network. It plays a pivotal role in ensuring robust and high-

speed wireless connectivity. 5G RAN introduces several innovations compared to its predecessors, including massive multiple-input, multiple-output (MIMO) technology and beam-forming.

Massive MIMO utilizes many antennas to improve spectral efficiency and enhance the network's overall capacity. Beam-forming allows the RAN to focus signals in specific directions, reducing interference and extending coverage. Small cells, another crucial aspect of 5G RAN, enhance network density in urban areas and provide seamless connectivity in high-demand locations.

The 5G Radio Access Network (RAN) can also be subdivided into the Radio Unit (RU), which is responsible for transmitting and receiving radio signals; the Distributed Unit (DU), which processes the radio signals; and the Centralized Unit (CU), which manages and controls the RAN.[12]

The F1 interface, standardized in TS 38.470[1] for 5G NR, separates the CU, responsible for PDCP, RRC, and SDAP functionalities:

- **PDCP (Packet Data Convergence Protocol):** PDCP is the protocol responsible for handling the packet data convergence between the radio interface and the higher layers of the network. It provides various functions, such as header compression and decompression, ciphering, integrity protection, and sequence number handling. PDCP ensures the efficient and secure transmission of user data and control signaling between the UE (User Equipment) and the RAN.
- **RRC (Radio Resource Control):** RRC is a control protocol that manages the RAN's establishment, maintenance, and release of radio resources. It handles functions related to radio bearer setup, mobility control, measurement reporting, and radio resource configuration. RRC enables efficient control and coordination between the UE and the RAN for efficient radio resource utilization and mobility management.
- **SDAP (Service Data Adaptation Protocol):** SDAP is responsible for mapping the QoS (Quality of Service) requirements of different network services to the appropriate radio bearers and scheduling mechanisms in the RAN. It ensures that the desired QoS parameters, such as latency, throughput, and reliability, are met for different types of traffic flows. SDAP enables the efficient handling of various services and their specific QoS needs within the RAN.

And the DU, responsible for RLC, MAC, and PHY functionalities:

- **RLC (Radio Link Control):** RLC is a protocol that operates between the MAC and PHY layers in the RAN. It handles tasks such as segmentation and reassembly of data packets, error correction through ARQ (Automatic Repeat Request), and congestion control. RLC ensures reliable data transmission over the radio interface by managing the flow and error recovery mechanisms.
- **MAC (Medium Access Control):** MAC controls access to the shared radio resources in the RAN. It handles tasks such as scheduling and prioritizing data transmissions, multiplexing

and demultiplexing data from different UEs, and handling random access procedures. MAC ensures efficient utilization of the available radio resources and supports various transmission schemes to meet the diverse needs of UEs.

- **PHY (Physical Layer):** PHY is the lowest layer in the RAN responsible for transmitting and receiving wireless signals. It handles tasks such as modulation and demodulation of data, channel coding and decoding, and synchronization with the network. PHY transforms digital data into analog signals for transmission over the air and vice versa. It plays a crucial role in achieving reliable and efficient wireless communication.

2.2.2.1 Open-source solutions

The table below compares the features and limitations of three popular 5G RAN implementation frameworks: OpenAirInterface (OpenAI), srsRAN, and UERANSIM:

Table 2.2: Comparison of Frameworks for 5G RAN Implementation

Feature/Limitation	OpenAirInterface (OpenAI)	srsRAN	UERANSIM
Supported Networks	4G-LTE and 5G-NR	4G-LTE and 5G-NR	5G-NR
Flexibility and Customization	Highly customizable and flexible	Highly customizable and flexible	Primarily focused on UE simulation
Implementation Language	C	C++	C++
CU/DU Split Support	Supported	Coming soon	Not supported
Known Supported RF	USRP B210, USRP X310, BladeRF, LimeSDR and EURECOM EX-PRESSMIMO2 RF	LimeSDR, BladeRF and USRP X310	Software-based simulation only
Documentation	Extensive documentation	Extensive documentation	Some documentation
Community Support	Active community	Active community	Semi-active community
Development status	Active development and constant updates	Active development and occasional updates	Development no longer active

OpenAirInterface (OpenAI) and srsRAN support 4G LTE and 5G NR networks, offering high flexibility and customization, while UERANSIM exclusively supports 5G NR and predominantly

focuses on UE simulation, offering limited customization options for the RAN configuration. OpenAI is implemented in C, while srsRAN and UERANSIM use C++. OpenAI includes support for CU/DU split, whereas srsRAN is working on its integration, and UERANSIM doesn't support it. In terms of known and tested supported RF hardware, OpenAI boasts a wider range, including USRP B210, USRP X310, BladeRF, LimeSDR, and EURECOM EXPRESSMIMO2 RF. In contrast, srsRAN supports LimeSDR, BladeRF, and USRP X310, and UERANSIM doesn't support RF hardware, functioning only as a software-based simulation. OpenAI and srsRAN offer extensive documentation and active community support, while UERANSIM provides some documentation with a semi-active community. Lastly, OpenAI and srsRAN are in active development, with OpenAI receiving regular updates and srsRAN receiving updates periodically. In contrast, development for UERANSIM has ceased.

2.2.3 User Equipment (UE)

User Equipment (UE) refers to the devices end-users use to access and utilize the 5G network's services. UEs encompass many devices, including smartphones, tablets, laptops, IoT sensors, etc. These devices are equipped with 5G-compatible hardware that enables them to communicate with the RAN and, subsequently, the core network.

5G UEs are designed to be versatile and support various applications, from high-definition video streaming to mission-critical IoT communications. They benefit from advanced features like enhanced mobile broadband (eMBB) for blazing-fast data speeds, ultra-reliable low latency communication (URLLC) for real-time applications, and massive machine-type communication (mMTC) for handling a massive number of IoT devices efficiently. The versatility of 5G UEs makes them the gateway to unlocking the full potential of the 5G network for end-users across diverse scenarios and industries.

2.3 Frequency Bands in Wireless Networks

In the realm of wireless communication, various frequency bands play a pivotal role in enabling the transmission of data and information. These frequency bands are essentially segments of the electromagnetic spectrum allocated for different types of communication. Broadly, they can be categorized into licensed and unlicensed bands.[7]

Licensed spectrum refers to the frequencies allocated to network operators by regulators, and they are subject to specific regulatory conditions. Operators must obtain licenses for using these frequencies, which can be costly and take time. The licensed spectrum is generally considered more reliable and secure due to its dedicated allocation to a single operator, ensuring a higher service quality and broader coverage. This reliability stems from the exclusive access and stringent regulatory oversight licensed bands entail, minimizing interference and guaranteeing a consistent and dependable wireless experience.

On the other hand, unlicensed spectrum is free for everyone as long as they follow specific rules to avoid interference with other users. This makes it attractive for small-scale and experimental deployments, such as private networks or IoT applications. Unlicensed spectrum also offers more flexibility and agility, allowing for dynamic spectrum sharing. However, it may suffer from interference and congestion issues, especially in densely populated areas or places with many WiFi networks.

2.4 The unlicensed band

Another crucial aspect of 5G that warrants thorough exploration in this context is 5G New Radio (5G NR) [29][13]. 5G New Radio represents a revolutionary air interface for 5G mobile networks, underpinned by advanced technologies that enable ultra-high-speed data transfer, ultra-low latency, and seamless connectivity for many devices. To appreciate the role of 5G NR within the Portuguese telecommunications landscape, it's essential to delve into the frequency spectrum allocation, as shown in 2.3. In Portugal, like in many regions, the spectrum is allocated into licensed and unlicensed bands. The licensed bands are dedicated to specific mobile operators, ensuring reliable, interference-controlled communication. Conversely, the unlicensed bands, such as the widely-used 2.4 GHz and 5 GHz frequencies, are open for shared use and have been instrumental in increasing technologies like WiFi.

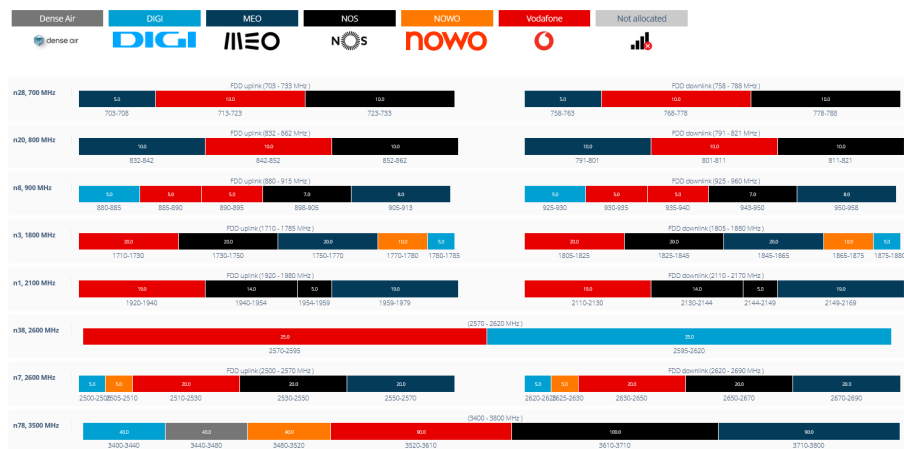


Figure 2.3: Portugal frequency spectrum. Source: [28]

2.4.1 C-Band

In the context of 5G, the C-band (shown in 2.3) represents a vital segment within the spectrum landscape strategically positioned between the low and high-frequency spectrums. Typically encompassing frequencies ranging from 3.7 to 4.7 GHz, the C-band spectrum boasts superior reliability, primarily attributed to its reduced susceptibility to interference from other radio frequency signals. This innate resilience makes it particularly well-suited for businesses and industries reliant on dependable data transmission.

Table 2.3: C-Band bands, frequencies, and channel bandwidths

Band	Frequency	Channel bandwidths
n77	3.7 GHz	10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100
n78	3.5 GHz	10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100
n79	4.7 GHz	10, 20, 30, 40, 50, 60, 70, 80, 90, 100

2.4.2 5G New Radio

5G New Radio, often referred to as 5G NR, represents a cutting-edge radio access technology (RAT) crafted by the 3rd Generation Partnership Project (3GPP) for the fifth generation (5G) mobile network. It stands as the global standard for the critical air interface component of 5G networks, aimed at revolutionizing wireless communications. The 3GPP specification 38 series [2] provides the technical details behind 5G NR, the successor of LTE, and it was conceived to meet three fundamental requirements for 5G communications [29]:

1. **Enhanced Mobile Broadband (eMBB):** 5G NR aims to deliver significantly higher data rates and lower latency than its predecessor, 4G. This means faster downloads, smoother streaming, and improved mobile device and application connectivity.
2. **Ultra-Reliable Low-Latency Communications (URLLC):** Beyond speed, 5G NR is engineered to support ultra-reliable, low-latency communication, essential for applications demanding real-time responsiveness, such as autonomous vehicles, remote surgery, and industrial automation.
3. **Massive Internet of Things (mIoT):** The NR air interface is optimized to handle the massive connectivity requirements of the Internet of Things (IoT). It can efficiently manage many IoT devices, making it feasible to connect everything from smart cities to industrial sensors seamlessly.

2.4.3 5G New Radio Unlicensed

With the increasing demand for high-speed wireless communication, 5G networks are expanding to the unlicensed spectrum. To facilitate the deployment of 5G New Radio (NR) within the unlicensed spectrum, it is imperative to introduce significant improvements at the physical layer, particularly concerning the channel access mechanism, to ensure equitable coexistence with other wireless technologies. This effort to develop NR-based access within the unlicensed spectrum, often called NR-U, commenced as part of 3GPP's Release 16 [21].

The primary challenge in implementing NR-U pertains to regulatory directives requiring a transmitter to engage in a "listen-before-talk" (LBT) protocol to ensure equitable coexistence in the shared spectrum. As a result of this LBT requirement, significant adaptations are imperative within the NR specification, primarily in two pivotal domains.

The first enhancement area revolves around formulating an efficient channel access (CA) procedure, delineating the precise protocols to be observed during transmissions while ensuring strict compliance with LBT regulations. The second critical aspect involves devising robust mechanisms for handling instances where LBT procedures fail, particularly in scenarios concerning essential signals such as those dedicated to synchronization, random access, and control channels. These mechanisms enable standalone NR-U operation, where NR-U operates independently from the pre-existing LTE infrastructure [21]. This marks the first instance where the 3GPP has established an operational mode that exclusively relies on unlicensed spectrum to facilitate control and user plane traffic.

Under NR-U's standalone mode, which removes any dependence on licensed network operators, the path is paved for implementation by a diverse range of entities, extending its appeal to private enterprises and making it an ideal choice for establishing private networks. This newfound accessibility broadens its applicability and foreshadows a new era for private 5G deployments, finely tuned to cater to the ever-increasing demands of emerging consumer and industry applications [15].

2.4.4 Cognitive Radio

Cognitive radio is a transformative technology in wireless communications that, at its core, is designed to operate intelligently and adaptively in dynamic and congested spectrum environments. Unlike traditional radio systems with fixed parameters, cognitive radios can autonomously sense, learn, and make real-time decisions about the available spectrum. They can detect unused or under-utilized frequency bands, known as spectrum holes or white spaces, and opportunistically transmit data within them. This dynamic spectrum access enhances spectrum utilization and efficiency, mitigates interference, and promotes coexistence with existing wireless systems.

Cognitive radio technology holds the potential to address the ever-growing demand for wireless bandwidth in today's interconnected world, making it a pivotal focus of research and development in wireless communications. Its implications extend to various applications, including next-generation wireless networks, wireless sensor networks, and Internet of Things (IoT) devices, offering the promise of more efficient, adaptive, and interference-resilient wireless communication systems [20].

2.4.5 Listen-before-Talk

NR-U-based systems rely on different channel access procedures to operate and coexist over these bands, requiring sensing the channel before transmission. This mechanism is called Listen-Before-Talk (LBT) [15].

The Listen-Before-Talk (LBT) protocol is a crucial mechanism that enables NR-U systems to coexist with other networks and technologies in the unlicensed band without affecting the performance of either system. The traditional LBT approach [26] used in this shared environment

functions with a fixed channel occupancy time (COT), regardless of the traffic load in each system, such as the one shown in 2.4. This could lead to one system's inefficiency if the COT is set too short or too long. In addition, the backoff mechanisms used by NR-U and other systems are different, which may cause unfair coexistence. As a result, the operational parameters of the LBT protocol could be adjusted dynamically to accommodate the system's varying traffic load and backoff window size [17].

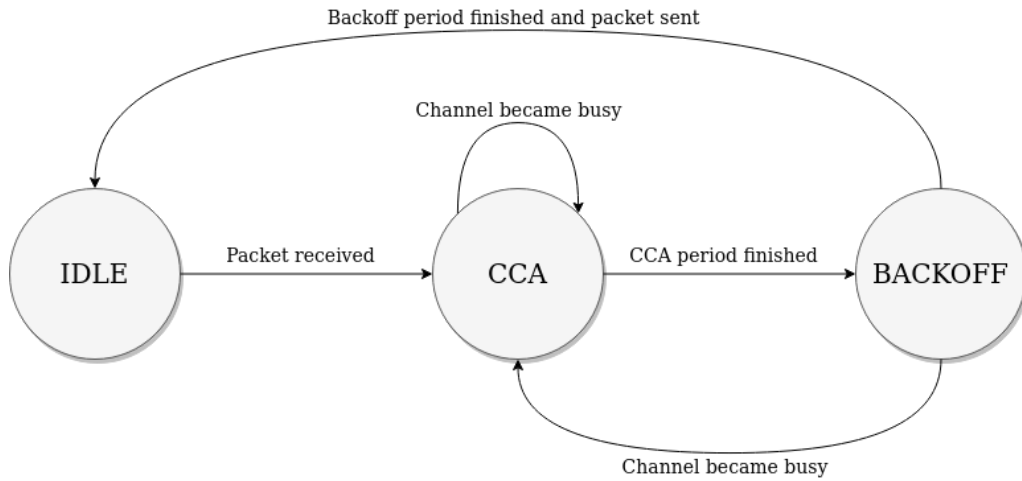


Figure 2.4: Listen-before-Talk protocol

- **IDLE:** This is the initial state where the receptor waits for a packet to arrive.
- **CCA:** If a packet is detected, the receptor switches to this state and starts monitoring the channel activity for the duration of the CCA (Clear Channel Assessment) period, equal to the channel occupancy time (COT). The receptor transitions to BACKOFF if the channel remains idle during the CCA period. Otherwise, it stays in this state and continues to monitor the channel.
- **BACKOFF :** Once the channel is idle for the CCA period, the receptor chooses a random backoff counter z in the range of $[0, CW-1]$, where CW is the contention window size. It decrements it as long as the channel remains idle. When z reaches 0, the receptor transmits the packet and transitions to the IDLE state. If the channel becomes busy during the backoff, the receptor returns to the CCA state to restart the process.

2.5 State of the art

In non-public networks within the 5G era, recent research has made significant strides in exploring multifaceted aspects, focusing on architectural design, spectrum utilization, and integrating open-source solutions. Two noteworthy contributions that have illuminated the evolving landscape of non-public 5G networks are 'Private 5G Networks: Concepts, Architectures, and Research Landscape' [30] and '5G Non-Public Networks: Standardization, Architectures, and Challenges' [24].

In the seminal work 'Private 5G Networks: Concepts, Architectures, and Research Landscape' [30], the authors provide an insightful exploration of the evolving realm of private 5G networks. These networks represent a dedicated 5G infrastructure designed to cater to specific needs within a defined geographical area. The central concept of private 5G networks revolves around their ability to amalgamate the strengths of both public and non-public 5G networks, making them a compelling choice across various sectors, including industry, business, utilities, and the public sector. The paper commences by furnishing a comprehensive overview of the core concept and architecture of private 5G networks, laying the foundation for subsequent discussions. It delves into the intricacies of implementing private 5G networks, shedding light on the pivotal enabling technologies that underpin their functionality. One of the key highlights of the paper is its extensive coverage of the diverse and appealing use cases of private 5G networks. It illuminates real-life demonstrations and showcases where these networks have made a substantial impact. This inclusive approach offers readers a holistic understanding of the practical applications and implications of private 5G networks. Furthermore, the paper diligently examines the diverse research challenges that the domain of private 5G networks presents, providing valuable insights into the areas that warrant further exploration. It outlines a roadmap for future research directions in this burgeoning field, offering a comprehensive view of the trajectory of private 5G networks in both industry and academia.

On a parallel trajectory, '5G Non-Public Networks: Standardization, Architectures, and Challenges' [24] dissects the technical intricacies of non-public 5G networks within the broader context of the 5G revolution. With the advent of 5G, numerous applications spanning various sectors have emerged, each with unique network requirements. Many of these applications demand private networks exclusively tailored to the needs of a specific enterprise. The paper starts by delineating the essential requirements and enabling solutions underpinning private 5G networks. It offers a comprehensive overview of the technical prerequisites that must be met to facilitate private network deployment efficiently. Much of the paper examines the 3rd Generation Partnership Project (3GPP) Release 16 capabilities, which are pivotal in supporting private 5G networks. The authors dissect these capabilities, providing valuable insights into the standards and protocols forming private network deployments' backbone. Moreover, the paper navigates various architectural proposals, particularly on single-site private networks. It also extends its scope to address scenarios in which the radio access network (RAN) is shared, offering readers a comprehensive understanding of network configurations and their implications. To further enrich the reader's comprehension, the paper explores mobility aspects and delves into multi-site private 5G network scenarios, dissecting the challenges and nuances associated with these deployments.

Regarding the deployment of Listen-Before-Talk (LBT) protocols to harness the untapped potential of unlicensed spectrum, it is pertinent to highlight two noteworthy works. These studies include "Adaptive Listen-Before-Talk (LBT) Scheme for Coexistence of LTE and Wi-Fi Systems in the Unlicensed Band" [17] and "Coexistence of Wi-Fi and Cellular Networks with Listen-Before-Talk in the Unlicensed Spectrum" [26].

The first work focuses on the necessity for an adaptive LBT scheme, highlighting the shortcomings of conventional fixed-channel occupancy time (COT) mechanisms in the context of varying traffic loads. It underscores that an inadequate COT setting can lead to system inefficiencies, particularly when one system becomes overloaded. Moreover, the inherent differences in back-off mechanisms between Wi-Fi and LTE systems pose challenges to fair coexistence. To address these issues, the authors propose an adaptive LBT protocol featuring two adaptation mechanisms: on-off adaptation for COT and short-long adaptation for idle time. Their simulations demonstrate substantial throughput enhancements ranging from 30% to 105%, contingent upon traffic load conditions, all while preserving Wi-Fi system performance [17].

The second work adopts a mathematical framework to analyze the coexistence performance of Wi-Fi and cellular networks employing different LBT procedures. This framework models the behavior of a cellular base station as a Markov chain, integrated with Bianchi's Markov model, which characterizes a Wi-Fi access point's behavior. The primary objective is to identify the optimal contention window size for cellular base stations, thereby maximizing the overall throughput of both networks while meeting the individual throughput requirements of each network. The numerical results validate the effectiveness of adjusting LBT parameters, emphasizing the importance of optimizing these protocols to achieve harmonious coexistence [26].

These pivotal contributions, along with the significant research on Listen-Before-Talk (LBT) protocols, provide an in-depth analysis of non-public 5G networks and serve as foundational pillars in comprehending the broader landscape of 5G technology. While the former offers comprehensive insights into architectural design, standardization efforts, and the evolving challenges of non-public networks, the latter emphasizes the importance of LBT protocols in enabling seamless coexistence within unlicensed spectrum bands. Together, these works underscore the critical role of non-public networks in catering to organizations' diverse and specialized connectivity needs, paving the way for a new generation of customized, efficient, and secure wireless communication.

Chapter 3

Methodology

This chapter outlines the chosen research methodology and the steps to be taken during the dissertation's development.

3.1 Problem definition

The deployment of 5G non-public networks on open-source software instead of dedicated equipment is a relatively new concept, enabling new entrants to participate in the market and fostering increased competition, ultimately benefiting end-users. However, this approach also presents several challenges. One of the primary issues revolves around the absence of documentation and support for integrating the necessary components required for proper network functionality. This can pose difficulties for network operators and service providers regarding implementation and maintenance, particularly if they are unfamiliar with open-source software.

The current challenge encompasses two distinct dimensions. Firstly, a concerted effort exists to address the complexities of deploying non-public networks using open-source software. This endeavor necessitates a profound comprehension of the underlying technological framework. Collaborative engagement with subject matter experts from Capgemini, coupled with rigorous research and testing, will culminate in the development of effective solutions conducive to the streamlined deployment of these networks.

Secondly, the availability of a suitable spectrum allocation stands as an indispensable prerequisite for these networks. One viable proposition entails the utilization of the non-licensed frequency band. Notably, this band remains devoid of government regulatory constraints, permitting utilization by any entity, provided interference with other users is diligently avoided. Harnessing the potential of the non-licensed band ensures both the efficacy and efficiency of network operations. Nevertheless, implementing a Listen-Before-Talk (LBT) protocol emerges as an imperative measure to mitigate the risk of interference with other coexisting systems.

3.2 NPN implementation

The project's initial phase clarified the implementation strategy for the Non-Public Network (NPN) deployment. A research endeavor was undertaken following an assessment of the chosen approach, which entails the deployment of a Standalone NPN, as mentioned earlier. This process led to the decision to commence with an initial minimum requirements SBA deployment. This strategic choice was made to align with project priorities and optimize resource allocation within the defined time frame for development.

3.2.1 Minimum SBA deployment

For this, the core functions, which have been presented before, that will be implemented are:

- **AMF** (Access and Mobility Management Function)
- **SMF** (Session Management Function)
- **UPF** (User Plane Function)
- **PCF** (Policy and Charging Function)
- **AUSF** (Authentication Server Function)
- **UDM** (Unified Data Management)

3.2.2 Technology Selection

The next step in the project involves carefully selecting the appropriate technologies for deploying the Radio Access Network (RAN) and the 5G Core.

3.2.2.1 Radio Access Network (RAN)

As part of the project's deployment of a 5G NPN, the Radio Access Network (RAN) component was a critical element to be set up correctly. After careful consideration and analysis of the available technology, shown in Table 2.2, and given the project's scope, the chosen technology for the RAN would be OpenAirInterface (OpenAI). Several advantageous factors primarily drove this choice. Its flexibility and customization capabilities align well with the project's requirements. Additionally, it explicitly mentioned compatibility with the USRP B210, the RF hardware available at the time, further enhancing its suitability. Furthermore, the framework's support for CU/DU split, a feature highly favored by the project team, influenced the decision. Lastly, extensive documentation and active community support, notably through mailing lists (which will be extensively mentioned later in this document), played a pivotal role in the selection process, ensuring essential resources for the project's success.

3.2.2.2 5G Core

The 5G Core constituted another critical component of the network infrastructure. After a comprehensive evaluation of available technologies, as detailed in Table 2.1, the chosen technology for deploying the 5G Core was ultimately Open5GS.

Upon carefully considering the features and limitations associated with these frameworks, the initial elimination process begins with Free5GC. This choice is primarily due to several factors, including its lack of containerization, comparatively weaker documentation, and community support. Furthermore, implementing Go, a language with limited prior experience, presents an additional challenge.

In the comparative evaluation between the remaining two options, Open5GS emerges as the optimal choice for this project. It offers a comprehensive set of core features aligned with our project's objectives, distinguishing it from OpenAI. Moreover, Open5GS exhibits a higher level of advancement, already demonstrating compatibility with 3GPP Release 16, whereas OpenAI lags.

3.3 Coexisting in the Unlicensed Bands

Before venturing into the realm of unlicensed bands, it is essential to make a deliberate selection. Two specific 5G frequency bands, exempt from spectrum licensing requirements, were recommended at the project's outset. These bands include the n46 band, illustrated in Figure 3.1, encompassing the frequency range of 5.935-5.995 GHz, and the n102 band, depicted in Figure 3.1, spanning the 5.160-5.905 GHz frequency range. Traditionally, these unlicensed bands find applications in global technologies such as WiFi, WiGig, and radar systems. In the European context, they are utilized for purposes such as Broadcast, fixed Point-to-Point (P2P) communication, satellite services, and cable TV relays [7].

Originally, the focus was on deploying a network in the C-band (shown in table 2.3), a licensed spectrum, but only in the lab environment and for limited periods and transmission power to ensure no interference with ongoing communications. This deployment aimed to implement the discussed use cases in a controlled setting.

However, exploring unlicensed spectrum became necessary because no C-band channels were reserved for NPNs (Non-Public Networks) in Portugal, as mentioned in the previous section. In this context, the licensed 5G MAC layer cannot be utilized since the spectrum needs to be shared with other technologies. Thus, a second part of the project would involve testing the same use cases in the unlicensed band, adhering to the principles of spectrum sharing.

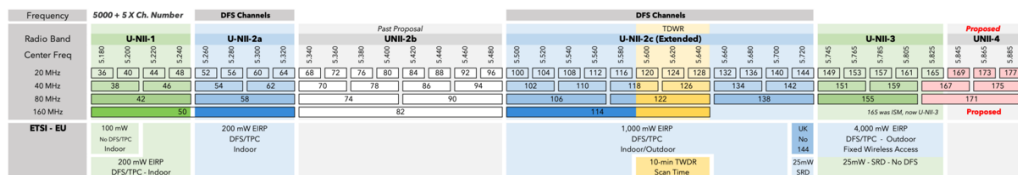


Figure 3.1: n46 unlicensed spectrum band

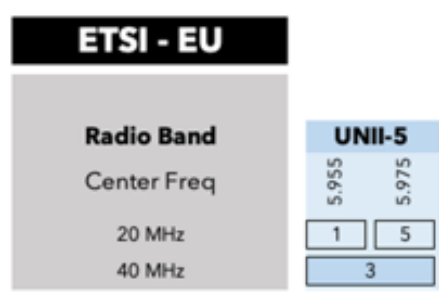


Figure 3.2: n102 unlicensed spectrum band

3.3.1 Challenges of Spectrum Sharing

The utilization of unlicensed bands in wireless communications presents unique challenges, particularly in the context of spectrum sharing. As the demand for wireless connectivity grows, efficient and fair access to the limited radio spectrum becomes increasingly critical. This section will discuss the challenges associated with spectrum sharing and identify the tasks required to address them successfully.

3.3.2 Identifying the Deployment of the LBT Algorithm

One of the primary challenges in spectrum sharing is determining the optimal deployment of the LBT (Listen-Before-Talk) algorithm. As an essential Medium Access Control (MAC) layer component, the LBT algorithm is pivotal in regulating access to the shared radio spectrum. Its purpose is to ensure fair and efficient utilization of available resources by allowing wireless devices to sense the wireless medium before transmitting data.

The LBT algorithm operates by periodically monitoring the energy levels in the channel. It must detect whether the channel is clear, indicating that no other device is transmitting or occupied, suggesting ongoing communication. By listening to the channel and avoiding transmissions when it is already in use, the LBT algorithm helps prevent collisions and ensures fair access to the shared spectrum. It's important to note that the effectiveness of an LBT algorithm in wireless access hinges on several factors, including the level of spectrum congestion and the ability to adapt to dynamic network conditions. While LBT excels at preventing collisions and promoting fair spectrum sharing, it may face challenges in highly congested environments where identifying clear channels becomes more intricate.

The LBT protocol, being a medium access control (MAC) protocol responsible for regulating access to the shared radio spectrum and ensuring fair and efficient use of the available resources, would operate, according to the protocol stack of the OSI model, in the Data link layer [31]. Here, it would implement the necessary procedures to sense the wireless medium, detect and avoid collisions, and regulate access to the shared radio resources fairly and efficiently [5].

As previously mentioned, the deployment of the OAI RAN will utilize the F1 interface, which represents the functional split between the Central Unit (CU) and the Distributed Unit (DU). Based on the architecture of the OAI RAN, it is observed that the MAC functions are primarily located

within the DU component. Consequently, considering the LBT protocol as a MAC protocol, it is deemed most suitable for integration within the DU component of the network infrastructure.

3.3.2.1 The Random Access Procedure

Deploying the Listen-Before-Talk (LBT) mechanism in the DU component aligns with its role in managing the initial access procedures, such as the one that facilitates the initial communication establishment between a User Equipment (UE) and the Radio Access Network (RAN). This procedure, called the Random Access Procedure, encompasses various intricate steps involving preamble transmission and resource allocation.

There are two types of Random Access Channel (RACH) procedures in cellular networks:

1. **Contention Based RACH Procedure (CBRA):** This is the normal procedure where the User Equipment (UE) randomly selects a preamble from the Zadoff-Chu sequence (A complex-valued sequence with low cross-correlation properties, commonly used in wireless communication) and sends a RACH request to the network.
2. **Contention Free RACH Procedure (CFRA):** In this procedure, the network itself shares the details of the cell and preamble with the UE. The UE sends the RACH request to the network, typically used in handover scenarios.

In the context of an RACH Procedure for registering a UE in the network, the adopted method is the Contention-Based RACH Procedure (CBRA). In this procedure, UEs randomly select a preamble from 64 preambles defined in each time-frequency slot in the 5G system. Due to the random selection, there is a possibility that multiple UEs may select the same preamble. When this happens, multiple UEs transmit the Physical RACH (PRACH) preamble simultaneously, resulting in PRACH collisions. This type of collision is known as "contention", and the RACH process that allows for this contention is referred to as the "Contention" RACH Process.

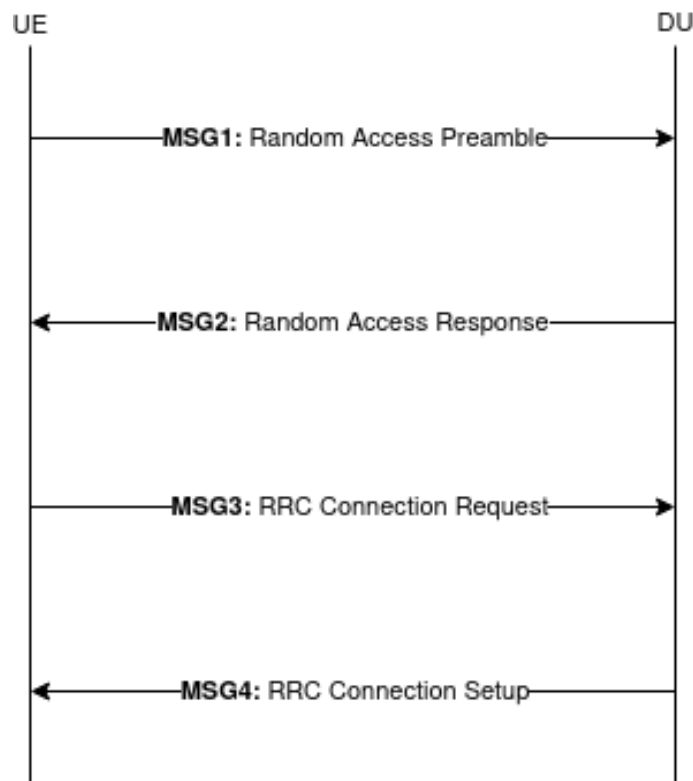


Figure 3.3: Contention Based RACH Procedure (CBRA)

As shown in Fig. 3.3, the procedure consists of several steps, typically referred to as MSG1, MSG2, MSG3, and MSG4:

1. **MSG1 (Random Access Preamble):** The UE initiates the procedure by transmitting a random access preamble, which serves as a unique identifier for the UE. This preamble is typically a Zadoff-Chu (ZC) sequence.
2. **MSG2 (Random Access Response):** Upon receiving the preamble, the DU responds with a Random Access Response message, which includes important information for the UE, such as the timing advance and an uplink grant.
3. **MSG3 (RRC Connection Request):** The UE sends an RRC Connection Request message to the DU, indicating its intention to establish a connection. This message contains essential information about the UE, such as its identity and capabilities.
4. **MSG4 (RRC Connection Setup):** The DU responds with an RRC Connection Setup message, which carries the necessary parameters and configurations for establishing the connection. This message also contains information about the security context if applicable.

Throughout this process, various control and feedback signals are exchanged between the UE and the RAN to ensure reliable communication. These signals include timing information, power control commands, and channel quality measurements.

The RA procedure allocates radio resources to the UE and establishes a dedicated connection for further communication. It plays a crucial role in the initial setup of the communication link and allows the UE to access the RAN network services.

Upon gaining a deeper comprehension of the initial connection between the UE and the RAN, it becomes evident that the implementation of the LBT protocol should occur in the reception of the MSG1 by the DU, as it marks the commencement of the Random Access Procedure.

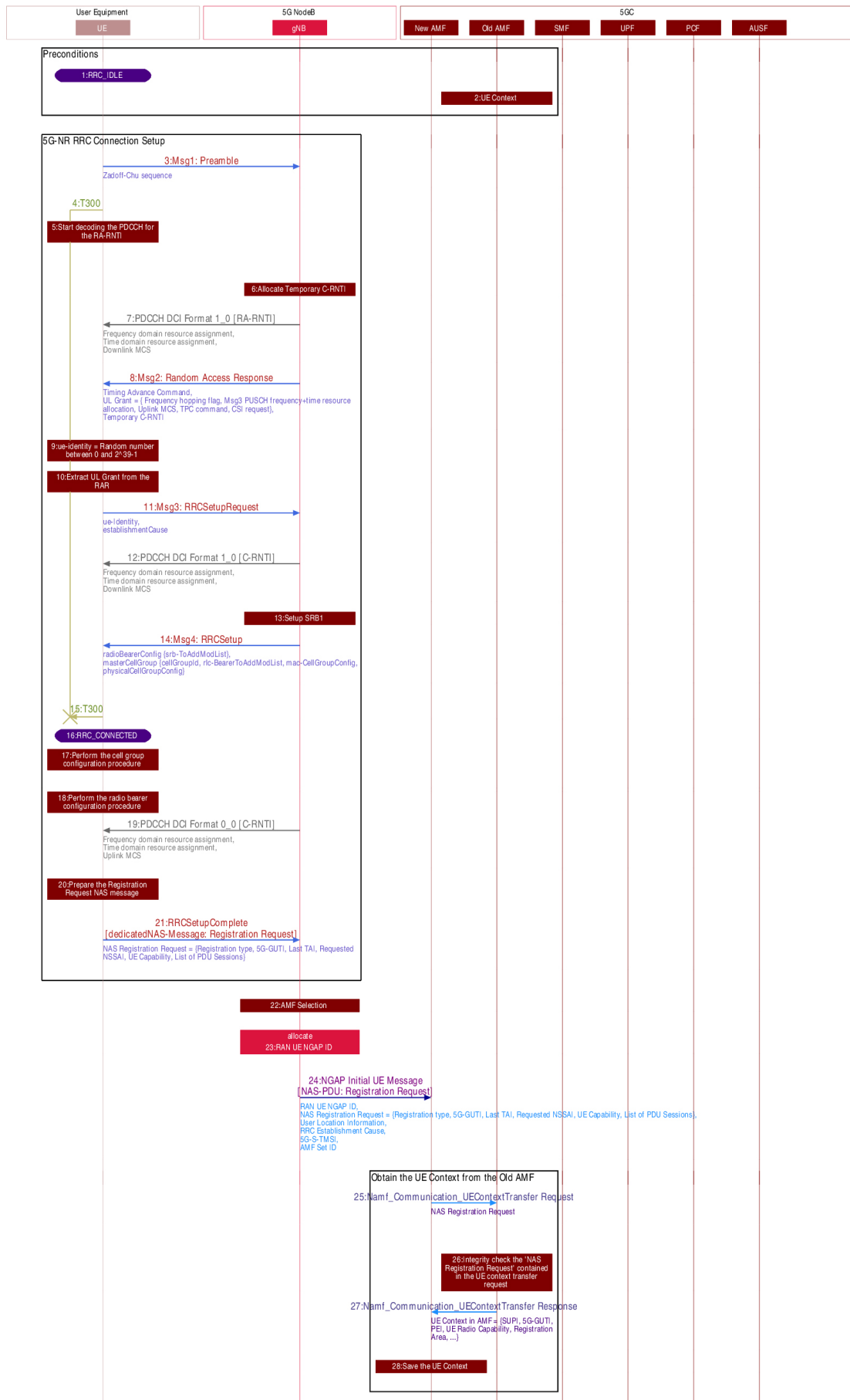
3.3.3 Validation of Network

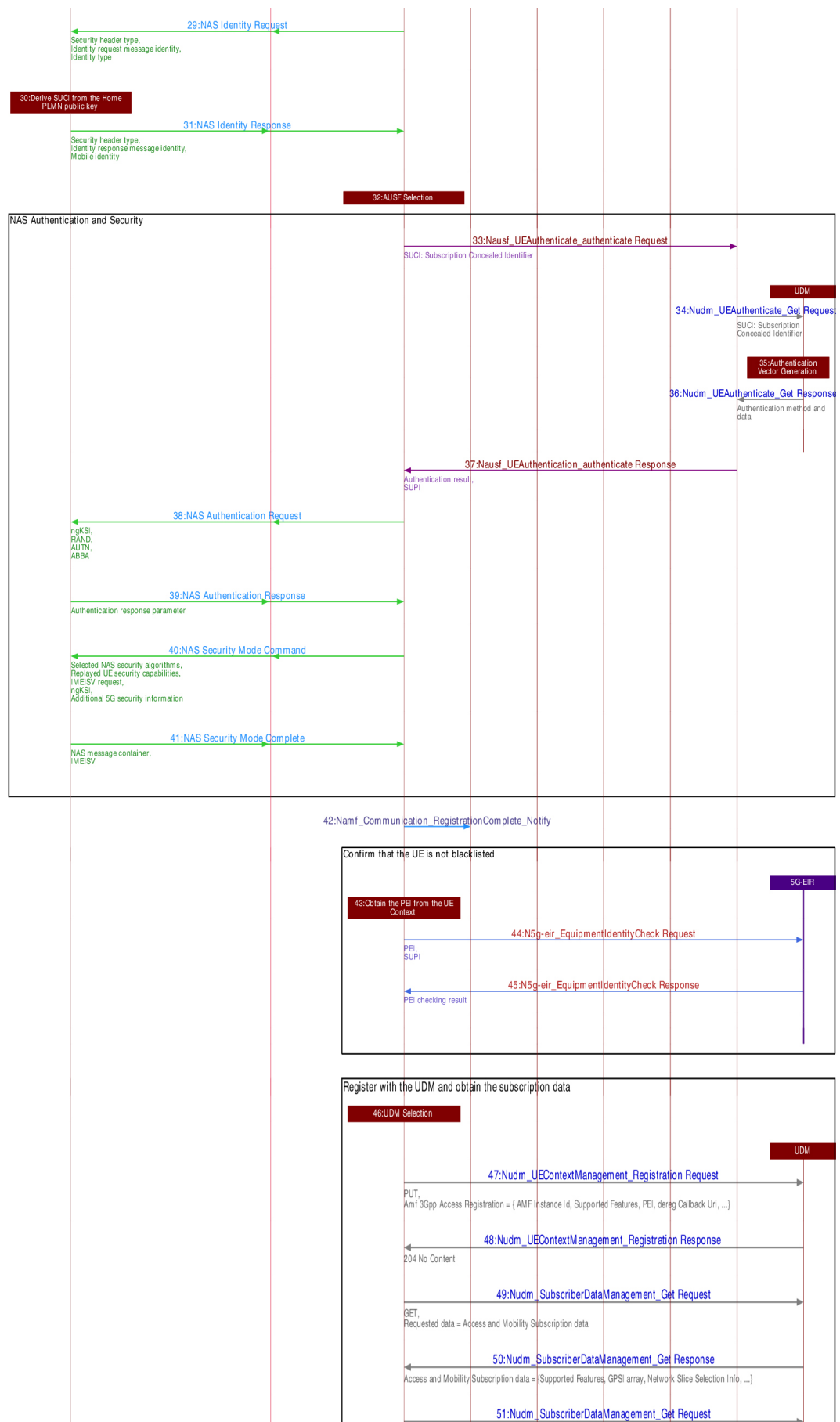
After deploying the network on the FEUP campus, the testing and validation of specific use cases to assess the network's performance will take place. The evaluated use cases will focus on device registration to the network and data transfer sessions. Extensive testing and verification will be conducted to ensure the proper functioning of all network components.

3.3.3.1 Elaboration on the Use Cases

The first use case centers around creating a reliable initial connection between the device and the network. This entails successfully transmitting essential messages among all participating components, as illustrated in the sequence diagram in [14].

Note: In a network with multiple AMFs, the registration procedure involves an "Old AMF" and a "New AMF" because the UE's context needs to be transferred when moving between AMFs (When a UE moves from one AMF's coverage area to another). However, in this single AMF network, there is no need for such a transfer as the UE's context remains with the same AMF throughout its existence. Therefore, the communications between the "Old AMF" and the "New AMF" can be disregarded since this setup only has one AMF.







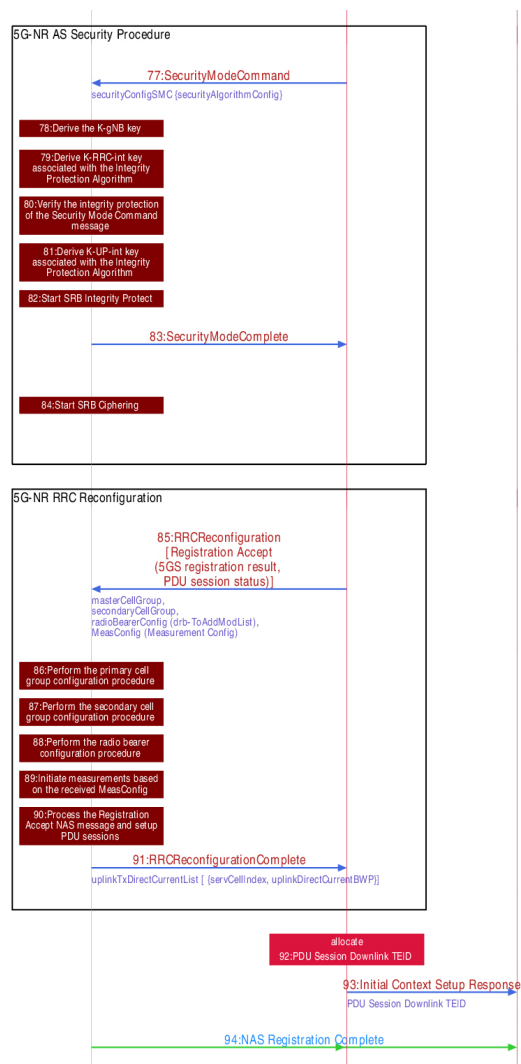


Figure 3.7: Device Registration Use-Case (Generated with EventStudio System Designer)

The second use case pertains to the effective initiation and execution of a data transfer session within the network. This particular use case emphasizes the network’s capacity to proficiently transmit data packets between the user equipment (UE) and the various network components:

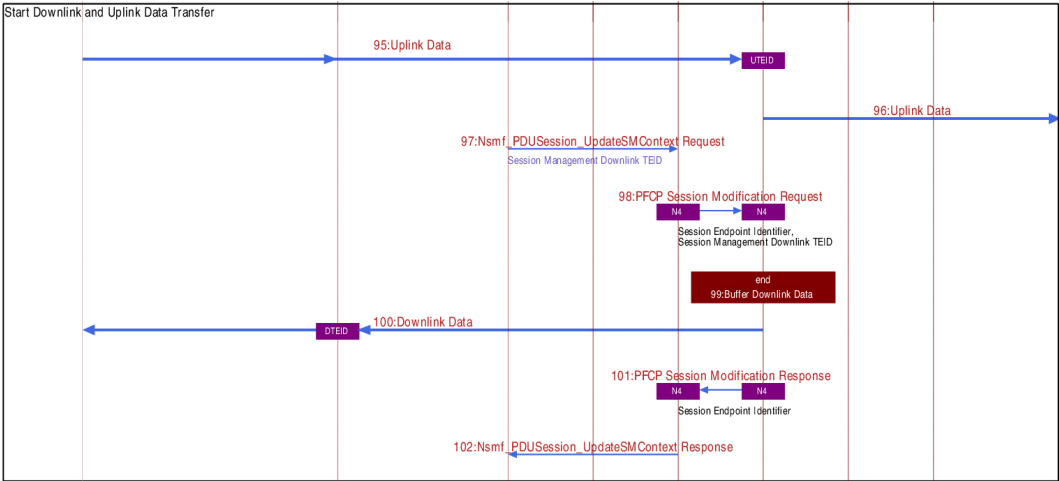


Figure 3.8: Data Transfer Session Use-Case (Generated with EventStudio System Designer)

To provide clarity regarding the sequence diagram, it is essential to comprehend the significance of the uplink TEID (UTEID) and the downlink TEID (DTEID) in the context of data transmission between the User Equipment (UE) and the core network. The uplink TEID represents the Tunnel Endpoint Identifier allocated by the UE, facilitating the flow of data traffic from the UE toward the core network. This identifier is vital in establishing and sustaining the tunnel for uplink data transmission. Conversely, the downlink TEID pertains to the Tunnel Endpoint Identifier employed for data traffic from the core network to the UE. The core network assigns the downlink TEID and serves to establish and maintain the tunnel for downlink data transmission.

Chapter 4

Implementation

This chapter explains the process of implementation of the network as previously outlined.

4.1 Network Deployment Strategy

Once the appropriate technologies for the Radio Access Network (RAN) and the 5G Core have been selected (as described in Section 3.2.2), the next step is to develop a deployment strategy that ensures a smooth and efficient rollout of the new system. This involves determining the optimal locations for deploying the RAN and Core components and their configuration and providing connectivity. This section will discuss our proposed deployment strategy and the steps involved in implementing it.

4.1.1 Radio Access Network (RAN)

After verifying that an Open5GS Core is compatible with an OpenAI RAN, the RAN was deployed using an F1 split scheme, which follows the functional split defined by 3GPP between the Centralized Unit (CU) and the Distributed Unit (DU), as mentioned in Chapter 2.

4.1.1.1 Deployment of OpenAirInterface RAN

The installation and deployment of the OpenAirInterface (OAI) open-source RAN involved several steps to set up and configure the RAN components, and these steps were as follows:

1. **Installation and System Setup:** Ubuntu 22.04 was installed on the target system using a bootable flash drive. Once the installation was complete, a dedicated folder was created for the OAI RAN.
2. **Dependency Installation:** The system packages were updated and the required dependencies for the OAI RAN were installed by executing the following command in the terminal:

```
sudo apt-get update && \  
sudo apt-get install -y build-essential \  

```

```

git \
cmake \
libssl-dev \
pkg-config \
libconfig-dev \
net-tools \
libssl-dev \
libz-dev \
iputils-ping \
libsctp1

```

3. **Cloning the OAI Repository:** The OAI repository was cloned, retrieving the OAI source code and generating a local copy on the system, by executing the following command:

```
git clone https://gitlab.eurecom.fr/oai/openairinterface5g.git
```

4. **Building and Installing Pre-Requisites:** By navigating to the OAI directory, the pre-requisite components were built using the following command:

```
./build_oai -I
```

5. **Building User-Equipment (UE) Simulator and gNB:** The compilation and building of the UE simulator and gNB components of the OAI RAN were performed using the following command:

```
./build_oai --gNB --nrUE -w SIMU
```

6. **Setting IP Addresses:** To establish communication between the machines, an Ethernet cable was connected to the enp7s0 network interface, allowing for direct connectivity. The IP addresses were manually configured as 10.42.0.1 for the CU and 10.42.0.2 for the DU, enabling communication between the two components.

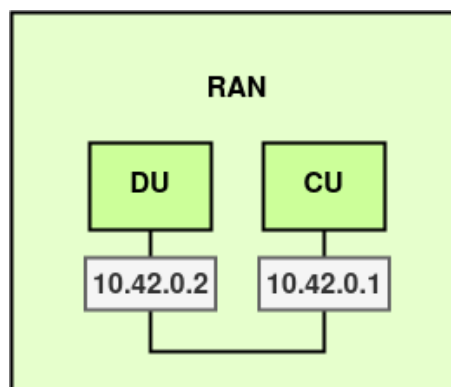


Figure 4.1: CU/DU network configuration

7. **CU/DU Configuration files:** The configuration files of the CU (`cu_gnb.conf`) and DU (`du_gnb.conf`) were updated with the necessary values for `local_n_if_name`,

`local_n_address`, and `remote_n_address`. These values specify the network interface and addresses used for communication between the components, as illustrated in Figure 4.1.

8. **Running the RAN:** To initiate the CU component, the following command was executed:

```
sudo RFSIMULATOR=server ./nr-softmodem --rfsim --sa \
-O ../../../../targets/PROJECTS/GENERIC-NR-5GC/CONF/cu_gnb.conf
```

Once the CU is running, the DU component is started by executing the following command:

```
sudo RFSIMULATOR=server ./nr-softmodem --rfsim --sa \
-O ../../../../targets/PROJECTS/GENERIC-NR-5GC/CONF/du_gnb.conf
```

Running these commands will start the RAN components, establishing communication between the CU and DU.

The execution of these steps will deploy and configure the OpenAirInterface RAN on the system successfully. The configuration files and network settings must be adjusted according to the specific requirements.

4.1.2 5G Core

After verifying that an Open5GS Core is compatible with an OpenAI RAN, the Open5GS core was deployed and sustained on the existing documentation [22].

4.1.2.1 Deployment of Open5Gs Core

A separate server machine was used for its deployment, on which the 5G SA Core was installed, minding the subsequent steps:

1. **MongoDB Installation and Execution:** Before installing the core, it was necessary to ensure MongoDB's presence and proper functioning, an open-source document-oriented database management system [19]. Following the documentation, MongoDB version 6.0 was initially installed but encountered difficulties with server startup. After investigating the issue, it became apparent that MongoDB version 6.0 was incompatible with Ubuntu 20. A fresh installation of the same version did not resolve the problem. Subsequently, MongoDB version 4.4, an older version, was chosen, resulting in a successful installation accomplished by following the steps below:

```
// Import the public key used by the package management system:
sudo apt-get update
sudo apt-get install gnupg
curl -fsSL https://pgp.mongodb.com/server-4.4.asc | \
sudo gpg -o /usr/share/keyrings/mongodb-server-4.4.gpg \
--dearmor
```

```
// Create a list file for MongoDB:
echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/
mongodb-server-4.4.gpg ] https://repo.mongodb.org/apt/ubuntu
focal/mongodb-org/4.4 multiverse" | sudo tee /etc/apt/
sources.list.d/mongodb-org-4.4.list

// Reload local package database:
sudo apt-get update

// Install the MongoDB packages:
sudo apt-get install -y mongodb-org=4.4.22 mongodb-org-server
=4.4.22 mongodb-org-shell=4.4.22 mongodb-org-mongos=4.4.22
mongodb-org-tools=4.4.22

// Start MongoDB:
sudo systemctl start mongod // (if '/usr/bin/mongod' is not
running)
sudo systemctl enable mongod // (ensure to start it on system
boot automatically)
```

2. Open5Gs Installation: Ubuntu makes it easy to install Open5GS, as shown below:

```
sudo add-apt-repository ppa:open5gs/latest
sudo apt update
sudo apt install open5gs
```

3. Open5Gs WebUI Installation: The WebUI provides an interactive platform for modifying subscriber data. Although not mandatory, it simplifies the initial stages of working with Open5GS, while advanced users can utilize the command line tool.

Note: Node.js, a runtime environment that allows executing JavaScript code outside of a browser for server-side application development, is required to install the WebUI of Open5GS.

```
// Install Node.js:
sudo apt update
sudo apt install curl
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash
-
sudo apt install nodejs

// Install WebUI of Open5GS:
curl -fsSL https://open5gs.org/open5gs/assets/webui/install |
sudo -E bash -
```

4.1.3 Network Initialization

This section provides an overview of the process and commands used to start and initialize the various elements of the network. This section aims to guide the reader through the steps involved in deploying the network components and setting them up for operation. It highlights the importance of properly starting each element and demonstrates the commands to initiate the network's functionality.

To ensure the proper initialization and functioning of the network, it is recommended to follow a specific order when turning on the network components. The suggested order is the same as the order used to present the components in this document, which is:

1. **Start the Core:** First, the Core component is initialized, allowing it to establish the necessary network services and functionalities;
2. **Start the CU:** Then, the CU component is started, which acts as the control unit responsible for managing and coordinating the network operations;
3. **Start the DU:** Once the CU is up and running, the DU component is initiated, representing the distributed unit responsible for radio-related functions and communication with the UE.
4. **Start the UE:** Lastly, the UE component is activated, representing the User Equipment that interacts with the network and establishes the necessary connections for communication.

Following this order ensures that each component has the necessary infrastructure and dependencies available for seamless communication and operation within the network.

Finally, Figure 4.2 illustrates the layout and configuration of the completed network, providing a visual representation of the interconnected components and their corresponding configurations. This diagram offers an overview of the network architecture, highlighting the relationships and connectivity between the Core, CU, and DU and the associated IP addresses utilized for communication.

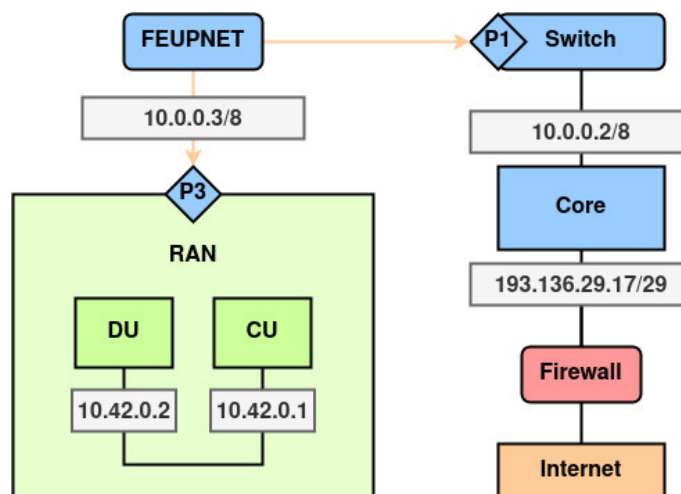


Figure 4.2: Network Diagram

4.1.3.1 The Core

To facilitate the registration process of a UE in the network, the network must possess the necessary information to authenticate and validate the device. Open5Gs offers a Web User Interface (WebUI) that enables interactive editing of subscriber data, allowing for the seamless integration of UE data into the network. Accessing the WebUI involves navigating to the localhost's port 3000, where subscribers can be conveniently added and managed, as depicted in Figure 4.3:

Create Subscriber

Subscriber Configuration

IMSI*
208930100001124

+

Subscriber Key (K)*
465B5CE8B199B49FAA5F0A2EE238A6BC

Authentication Management Field (AMF)*
8000

USIM Type
OPc

Operator Key (OPc/OP)*
E8ED289DEBA952E4283B54E88E6183CA

UE-AMBR Downlink*
1

Unit
Gbps

UE-AMBR Uplink*
1

Unit
Gbps

Slice Configurations

CANCEL SAVE

Figure 4.3: Open5Gs WebUI: Add subscriber data

In this step, it is crucial to input essential data, including the International Mobile Subscriber Identity (IMSI), Subscriber Key (K), and Operator Key (OPc/OP). These values must align with the configuration settings of the UE, as exemplified here:

```
uicc0 = {
  imsi = "208930100001124";
  key = "465B5CE8B199B49FAA5F0A2EE238A6BC";
  opc= "E8ED289DEBA952E4283B54E88E6183CA";
  dnn= "internet";
  nssai_sst=1;
  imeisv="6754567890123413"
}
```

Additionally, two vital Session Configuration parameters that require configuration are the Data Network Name (DNN)/Access Point Name (APN), as shown in Figure 4.4, which should correspond to the UE settings, and its Type, which in our specific scenario is IPv4. It is worth noting that the remaining values can be left at their default settings.

Session Configurations

DNN/APN*	Type*
<input type="text" value="internet"/>	<input type="text" value="IPv4"/>

Figure 4.4: Open5Gs WebUI: Additional mandatory subscriber data

Another critical task involves modifying the `amf.yaml` and `mme.yaml` files located within the Open5Gs directory. It is necessary to update the Public Land Mobile Network (PLMN) values (MCC/MNC), to align with the corresponding values from the UE's IMSI, and the Tracking Area Code (TAC). In our case, the MCC is 208, the MNC is 93, and the TAC is 7. Furthermore, we need to configure the NG Application Protocol (NGAP) address to match the Core's address, which, according to Figure 4.2, is 10.0.0.2.

To ensure that all the configurations take effect, it is necessary to restart the Core. This can be achieved by executing the following command:

```
sudo systemctl restart open5gs-*
```

To access the logs generated by the AMF, the following command can be used:

```
sudo tail -f /var/log/open5gs/amf.log
```

4.1.3.2 The CU

The first thing to be set up in the CU is the `cu_gnb.conf`, the configuration file that governs the definitions of the CU we're starting up. At the beginning of the file, the correct values for the TAC and the PLMN (MCC/MNC) have to be set, the `local_s_if_name` has to be set to the name of the interface that makes the connection between the CU and the DU, which in our case is the interface `enp7s0`. The values for the `local_s_address` and `remote_s_address` must also be set according to the CU's address (local) and the DU's address (remote).

In the AMF parameters, the AMF IP address has to be set to the IP address of the Core, which in our case is 10.0.0.2, and the network interface that connects the gNB and the Core also has to be set, which in our case is the interface `enp8s0` with IP 10.0.0.3.

To configure the CU, the `cu_gnb.conf` file needs to be modified, ensuring that the following parameters are correctly set:

1. **TAC and PLMN (MCC/MNC):** Set the appropriate values for the TAC and the PLMN (MCC/MNC) according to the network configuration.
2. **local_s_if_name:** Set the value of `local_s_if_name` to the interface's name that connects the CU and the DU. In this case, it should be set to `enp7s0`.
3. **local_s_address and remote_s_address:** Set the values of `local_s_address` and `remote_s_address` to the CU and DU IP addresses, respectively.

For the AMF parameters, the following must be set:

1. **AMF IP address:** Set the value of `amf_ip_address` to the IP address of the Core, which in this case is 10.0.0.2.
2. **Network interface:** Set the value of the network interface to the interface that connects the gNB and the Core. In this case, it should be set to `enp8s0` with the IP address 10.0.0.3.

By configuring these parameters correctly in the `cu_gnb.conf`, proper communication and connectivity between the CU and the DU and between the gNB and the Core is ensured.

Finally, the command used to initialize the CU consists of a combination of the following components, each serving a specific purpose in the initialization process:

1. **./nr-softmodem:** Selects the 5G gNodeB binary for execution;
2. **-O:** Tag that indicates a configuration file to be used will be supplied;
3. **../../../../targets/PROJECTS/GENERIC-NR-5GC/CONF/cu_gnb.conf:** Configuration file to be used;
4. **—sa:** Executes the gNB in standalone mode;
5. **-E:** Enables the 3/4 sampling mode, as the B210 USRP will crash with the default sampling rate of 61440000.

Forming the final command as follows:

```
sudo ./nr-softmodem -O ../../../../../../targets/PROJECTS/GENERIC-NR-5GC/CONF/
cu_gnb.conf --sa -E
```

4.1.3.3 The DU

The first thing to be set up in the DU is the `du_gnb.conf`, the configuration file that governs the definitions of the DU we're starting up. At the beginning of the file, the correct values for the TAC and the PLMN (MCC/MNC) must be set. In the MACRLCs section, the `local_s_if_name` has to be set to the interface name that makes the connection between the CU and the DU, which is the interface `enp7s0`. The values for the `local_s_address` and `remote_s_address` must also be set according to the CU's address (local) and the DU's address (remote).

In the RUs parameters, it is important to set appropriate attenuation values (`att_tx` and `att_rx`) to avoid interfering with existing networks, especially when deploying the network in the licensed spectrum. These attenuation values determine the power level of the USRP device and can be adjusted to ensure proper power reduction. In this case, the attenuation values were set to 12, corresponding to a power reduction of -12 dBs. This helps to mitigate any potential interference with neighboring networks.

Finally, the command used to initialize the DU consists of a combination of the following components, each serving a specific purpose in the initialization process:

1. **./nr-softmodem:** Selects the 5G gNodeB binary for execution;
2. **-O:** Tag that indicates a configuration file to be used will be supplied;
3. **.../.../.../targets/PROJECTS/GENERIC-NR-5GC/CONF/du_gnb.conf:** Configuration file to be used;
4. **—sa:** Executes the gNB in standalone mode;
5. **-E:** Enables the 3/4 sampling mode, as the B210 USRP will crash with the default sampling rate of 61440000.

Forming the final command as follows:

```
sudo ./nr-softmodem -O .../.../.../targets/PROJECTS/GENERIC-NR-5GC/CONF/
du_gnb.conf --sa -E
```

4.1.3.4 The UE

The command used to initialize the EU consists of a combination of the following components, each serving a specific purpose in the initialization process:

1. **./nr-uesoftmodem:** Selects the 5G User Equipment binary for execution;
2. **-O:** Tag that indicates a configuration file to be used will be supplied;
3. **.../.../.../targets/PROJECTS/GENERIC-NR-5GC/CONF/ue.conf:** Configuration file to be used;
4. **—r 106:** Bandwidth in terms of Resource Blocks;
5. **—numerology 1:** Numerology index determines sub-carrier spacing and time duration in 5G, impacting bandwidth and capacity. Setting the Numerology index to 1 (the default) in 5G indicates a sub-carrier spacing of 15 kHz and a time duration of 1 ms;
6. **—band 78:** Frequency band number the UE will operate on;
7. **—C 3619200000:** Downlink carrier frequency in Hz;
8. **—ue-fo-compensation:** Enables the frequency offset compensation at the UE. This is useful when running over the air and/or without an external clock/time source;
9. **—sa:** Executes the UE in standalone mode;
10. **-E:** Enables the 3/4 sampling mode, as the B210 USRP will crash with the default sampling rate of 61440000;

Forming the final command as follows:

```
sudo ./nr-uesoftmodem -O ../../../../targets/PROJECTS/GENERIC-NR-5GC/CONF/
ue.conf -r 106 --numerology 1 --band 78 -C 3619200000 --ue-fo-
compensation --sa -E
```

4.1.4 Hardware configuration

4.1.4.1 RAN

For the deployment of the RAN components, including the Distributed Unit (DU) and the Central Unit (CU), two distinct PC machines were configured, each equipped with the same set of components:

- **CPU:** 1x Intel Core i9 13900K 24-Core (2.2GHz-5.8GHz)
- **Storage space:** 1 TB HDD
- **Memory RAM:** 32 GB
- **Motherboard:** Asus TUF Z790-Plus Gaming D4
- **Graphics card:** Gigabyte GeForce® GTX 1650 D6 OC Rev.2 4G
- **Network:** 2x WL-PCI Express TP-Link TX401 | 10GB PCI Ethernet
- **Cooling:** 1x Cooler CPU Noctua NH-D15 chromax.black

4.1.4.2 Core

In the case of the core component, a Dell PowerEdge R710 server machine was employed.

4.1.4.3 Radio Units

The radio units providing radio capacities to the DU and the UE comprise two USRP B210 devices. These USRP (Universal Software Radio Peripheral) B210 units are versatile and software-defined radio platforms, offering the flexibility to adapt to various wireless communication standards and frequencies.

4.1.4.4 Wireless Channel

The wireless channel was established utilizing a coaxial cable, specifically an SMA(male)-to-SMA(female) coaxial cable with a length of 0.5 meters. Furthermore, this connection setup featured the integration of four such coaxial cables. In addition, signal attenuation was intentionally introduced into the system, with attenuators providing a substantial 60dB reduction in signal strength.

4.2 LBT protocol implementation

To facilitate the implementation of the Listen-Before-Talk (LBT) protocol, assistance was sought from the OpenAirInterface (OAI) community mailing lists. These mailing lists serve as vital communication platforms for members to engage with the OAI development team, seek guidance, report issues, contribute to the project, and stay updated on the latest developments and announcements related to OpenAirInterface.

Valuable insights and guidance were obtained through active participation in the mailing lists. However, it became evident that no existing functions or variables within the OAI code base directly meet the specific requirements for implementing the LBT protocol, such as power analysis. Consequently, it is necessary to develop these essential functionalities by examining the lower layers of the OAI code base.

Table 4.1: OAI RAN Repository Structure

Section	Description
ci-scripts	Meta-scripts and configuration files for the OSA CI process.
cmake_targets	Utilities for building and compiling on different platforms.
common	Common utilities for the OpenAirInterface (OAI) project.
doc	Documentation for the OAI RAN repository.
docker	Dockerfiles for building on Ubuntu and RHEL.
executables	Source files for top-level executables such as gNB and eNB.
maketags	Script to generate emacs tags.
nfapi	Code for the MAC-PHY interface.
openair1	Implementation of 3GPP LTE Rel-10/12 PHY layer and 3GPP NR Rel-15 layer.
openair2	Implementation of 3GPP LTE Rel-10 RLC/MAC/PDCP/RRC/X2AP and LTE Rel-14 M2AP, as well as 3GPP NR Rel-15 RLC/MAC/PDCP/RRC/X2AP.
openair3	Implementation of 3GPP LTE Rel-10 for S1AP and NAS GTPV1-U for both eNB and UE.
openshift	Helm charts for OpenShift deployment options of OAI.
radio	Drivers for various radios including USRP, AW2S, RFsim, etc.
targets	Configuration files of historical relevance, subject to potential deletion in the future.

Deeply analyzing the OAI repository represented at 4.1, there are 3 folders called openair1, openair2, and openair3. These folders organize the implementation of the 5G NR protocol stack into different layers.

The focus will be on the functions within Layer 1 to access the physical layer variables required for assessing the energy in the channel. Layer 1, situated at the lowest layer of the OSI model, assumes responsibility for transmitting and receiving raw data bits over the communication medium, encompassing cables and wireless signals.

By exploring and adapting the functions within Layer 1, the required components for the LBT protocol can be developed. This will enable efficient detection of channel occupancy and effective medium access control. Such a targeted approach ensures the successful implementation of the LBT mechanism within the OpenAirInterface framework.

The file `openair1/SCHED_NR/phy_procedures_nr_gNB.c` was examined, revealing a relevant function named `phy_procedures_gNB_uespec_RX()`. This function is responsible for processing received IQ samples, which concisely represent complex-valued signals using their In-phase (I) and Quadrature (Q) components. Additionally, the function uses `gNB_I0_measurements()`, initially appearing promising as it suggests power-related measurements.

However, it was soon uncovered that these measurements, including average noise power and average sub-band noise power calculations, could not detect channel occupancy. While noise power is an essential parameter for evaluating system performance, it does not directly indicate channel occupancy or the transmitted power of the channel.

It is important to note that while these measurements were unsuitable for detecting channel occupancy, exploring these functions was a valuable learning experience for understanding the intricacies of the OpenAirInterface code base.

After extensive testing and research, the efforts yielded promising results. A significant breakthrough occurred when I discovered the following line in the file `openair1/SCHED_NR/nr_prach_procedures.c`, which I regarded as a significant finding:

```
if ((gNB->prach_energy_counter == 100) && (max_preamble_energy[0] > gNB
->measurements.prach_I0+gNB->prach_thres)) {
```

The comparison involves two conditions:

1. **gNB->prach_energy_counter == 100:** This condition checks if the value of the variable `gNB->prach_energy_counter` is equal to 100. It ensures that several PRACH (Physical Random Access Channel) energy measurements have been accumulated. Based on observations, the value of `gNB->prach_energy_counter` remains constant at 100, regardless of whether the channel is clear or occupied.
2. **max_preamble_energy[0] > gNB->measurements.prach_I0+gNB->prach_thres:** This condition compares the value of `max_preamble_energy[0]` with the sum of `gNB->measurements.prach_I0` and `gNB->prach_thres`. It aims to detect channel activity or occupancy. When the channel is clear, this comparison tends to fail since the energy value generally falls within the range of [270, 310], while the threshold (`prach_I0 + prach_thres`) is usually around [390, 395]. Contrarily, when the channel is occupied, the energy value increases to [400, 470], allowing the condition to be evaluated as true.

It is speculated that this comparison determines if the accumulated PRACH energy surpasses a certain threshold (approximately 400), potentially indicating the presence of activity or occupancy in the channel. Debug prints were added to the code to gather further evidence, and the resulting logs exhibited promising outcomes:

```
Energy 27.0 dB < Threshold 40.6
Energy 28.7 dB < Threshold 40.5
Energy 28.7 dB < Threshold 40.4
Energy 0.0 dB < Threshold 40.3

Energy 42.8 dB > Threshold 40.3

[NR_PHY] [gNB 0][RAPROC] Frame 407, slot 19 Initiating RA procedure
    with preamble 43, energy 42.8 dB (IO 283, thres 120), delay 30 start
    symbol 0 freq index 0
(...)
[NR_MAC] handle harq for rnti 79a3, in RA process

Energy 0.0 dB < Threshold 44.3
Energy 42.0 dB < Threshold 45.8
Energy 40.8 dB < Threshold 47.2
Energy 33.5 dB < Threshold 46.9

Energy 46.0 dB > Threshold 44.3

[NR_PHY] [gNB 0][RAPROC] Frame 409, slot 19 Initiating RA procedure
    with preamble 14, energy 46.0 dB (IO 323, thres 120), delay 24 start
    symbol 0 freq index 0
(...)
[NR_MAC] handle_nr_ul_harq(): unknown RNTI 0x5aab in PUSCH

Energy 0.0 dB < Threshold 45.8
Energy 28.7 dB < Threshold 45.1
Energy 28.7 dB < Threshold 44.4
Energy 25.0 dB < Threshold 43.6
```

Note: A noteworthy observation regarding the energy measurements recorded in the logs extracted from the OpenAirInterface (OAI) codebase is imperative. These logs consistently reference energy measurements in decibels (dB), which do not conventionally serve as a unit for quantifying energy or power in standard metrology. Rather, decibels are more aptly employed to denote relative ratios between two values, as they lack an absolute reference point. The logical presumption is that the intended unit of measurement for these energy values should have been decibels relative to milliwatts (dBm), a unit commonly employed in the telecommunications domain to specify absolute power levels regarding one milliwatt. However, it is worth noting that, at the present juncture, confirmation of this assumption remains pending, as comprehensive information regarding the unit of measurement used in the OAI logs has not yet been ascertained.

To gain a deeper understanding of these variables, a comprehensive study was conducted:

1. **int prach_energy_counter:** This variable calculates average PRACH energy over the first 100 PRACH opportunities. It keeps track of the number of PRACH energy measurements accumulated.
2. **uint16_t max_preamble_energy[4]=0:** This variable is an array of four elements, with all elements initialized to 0. However, in the context of this file, only the first element [0] is utilized.
3. **int prach_I0:** This variable is part of the `measurements` structure and represents the PRACH background noise level. Its value typically ranges between 270 and 330.
4. **int prach_thres:** This variable assumably represents a threshold for the PRACH, but no additional information is available. Its value consistently remains at 120.

The variable `max_preamble_energy` is particularly interesting because it represents the maximum preamble energy, which measures the power level of the Physical Random Access Channel (PRACH) signal. The maximum preamble energy can provide valuable information in sensing channel occupancy through energy output.

When the channel is unoccupied or idle, the PRACH signal's power level is expected to be relatively low. In contrast, when active transmissions occupy the channel, the PRACH signal's power level is likely higher. By monitoring and analyzing the variations in the maximum preamble energy, we can detect changes in channel occupancy.

Therefore, tracking the `max_preamble_energy` variable allows us to assess the energy output in the channel and infer the presence or absence of activity, thereby enabling the sensing of channel occupancy.

4.2.1 Challenges and Status

4.2.1.1 Challenges

During my extensive email correspondence with the OpenAirInterface (OAI) community, particularly with **Cédric Roux**, a Research Engineer at **EURECOM - Communication Systems**, valuable insights and assistance were provided. While the majority of the interactions aimed to support my project, there were also moments when discouraging statements were made, including:

- *From a naive point of view it does not seem easy, not even possible. We need to transmit according to some timing constraints that don't seem compatible with a LBT.*
- *So there are some specifications for what you want to do [...] (but) today in openair none of this exists, as far as I know.*
- *Except the basic functions needed for standard processing, no. Maybe you can adapt them to your needs. You're on your own there, good luck. :)*

- *For your needs, I don't think there is documentation, only the code.*

Despite diligent efforts, limited familiarity with the OAI architecture, inadequate documentation, and the constraints of a tight project timeline, the complete implementation of the LBT protocol proved unattainable.

4.2.1.2 Current Status

After identifying the basic functions of standard processing that allow the sensing of the channel activity, as described in 4.2, the LBT algorithm was implemented within the `nr_prach_procedures.c` file responsible for managing the RACH procedures, as illustrated in the code snippet 4.1, where the precise point where the DU receives the initial message sent by the UE as part of the Random Access (RA) procedure was found, which is where the rest of the LBT functionality was implanted. Specifically, the LBT is implemented by the gNB, following the reception of communication from a UE.

Listing 4.1: LBT implementation within `nr_prach_procedures.c`

```
// START OF THE LBT

// Random seed initialization
srand(time(NULL));

// LBT algorithm states
enum { CCA, BACKOFF } state = CCA;

int finished = 0;

while (!finished) {
    switch (state) {
        case CCA:
            LOG_W(NR_MAC, "[LBT] CCA period starting...\n");
            // Monitor channel activity for CCA period
            sleep(CCA);

            // Check if channel is still clear after CCA period
            if (is_channel_clear())
                state = BACKOFF;
            else {
                state = CCA;
            }
            break;

        case BACKOFF:
            // Generate a random backoff counter
            backoff = rand() % CONTENTION_WINDOW_SIZE-1;
```

```

LOG_W(NR_MAC, "[LBT] CCA period finished and backoff period of %i
        started..\n", backoff);

while (backoff > 0) {
    // Monitor channel activity during backoff
    if (!is_channel_clear()) {
        // Channel became busy during backoff
        state = CCA;
        break;
    }

    // This wouldn't be needed if the function was implemented
    backoff--;
}
finished = 1;
break;
}
}
LOG_A(NR_MAC, "[LBT] Channel occupancy as been evaluated as: CLEAR\n")
;

// END OF THE LBT
LOG_I(NR_PHY, "[gNB %d][RAPPROC] Frame %d, slot %d Initiating RA
        procedure with preamble %d, energy %d.%d dB (IO %d, thres %d),
        delay %d start symbol %u freq index %u\n",
        gNB->Mod_id,
        frame,
        slot,
        max_preamble[0],
        max_preamble_energy[0]/10,
        max_preamble_energy[0]%10,
        gNB->measurements.prach_IO, gNB->prach_thres,
        max_preamble_delay[0],
        prachStartSymbol,
        prach_pdu->num_ra);

/* This is where the RA procedure is kick-started */
T(T_ENB_PHY_INITIATE_RA_PROCEDURE, T_INT(gNB->Mod_id), T_INT(frame),
  T_INT(slot),
  T_INT(max_preamble[0]), T_INT(max_preamble_energy[0]), T_INT(
    max_preamble_delay[0]));

```

The successful implementation of this algorithm relies on the availability of the `is_channel_clear()` function. However, at the present moment, this function remains a

placeholder without actual implementation. Due to time constraints and limited knowledge, exploring the intricacies of the energy function proved challenging, hindering the ability to call it when required. Subsequently, the validation of the LBT component within this project remains presently unattainable.

The corresponding code files for this function, namely `channel_occupancy.h` (4.2) and `channel_occupancy.c` (4.3), were incorporated into the OAI code-base by appending them to the `CMakeLists.txt` file (4.4), which governs the compilation of C files.

Listing 4.2: `channel_occupancy.h`

```
extern int channel_occupancy;
extern int CCA;
extern int CONTENTION_WINDOW_SIZE;
extern int backoff;

#ifndef IS_CHANNEL_CLEAR_H // Include guard to prevent multiple
    definitions
#define IS_CHANNEL_CLEAR_H

int is_channel_clear();

#endif
```

Listing 4.3: `channel_occupancy.c`

```
#include "channel_occupancy.h"

int channel_occupancy = 1; // 0 = false | 1 = true
int CCA = 0.000032; // 32 microseconds = 0.000032 seconds
int CONTENTION_WINDOW_SIZE = 4096;
int backoff = 0;

int is_channel_clear() {
    // not implemented
    return 1;
}
```

Listing 4.4: `CMakeLists.txt`

```
add_library(UTIL
    ${OPENAIR_DIR}/common/utils/LOG/log.c
    ${OPENAIR_DIR}/common/utils/global/channel_occupancy.c
    ${OPENAIR_DIR}/common/utils/LOG/vcd_signal_dumper.c
    ${OPENAIR2_DIR}/UTIL/MATH/oml.c
    ${OPENAIR2_DIR}/UTIL/OPT/probe.c
    ${OPENAIR_DIR}/common/utils/threadPool/thread-pool.c
    ${OPENAIR_DIR}/common/utils/utils.c
```

```

    ${OPENAIR_DIR}/common/utis/system.c
    ${OPENAIR_DIR}/common/utis/time_meas.c
    ${OPENAIR_DIR}/common/utis/time_stat.c
)

```

The adopted LBT procedure, as described in Section 2.4.5, relies on two predefined constants: the Channel Occupancy Time (COT) and the Contention Window (CW). The value for COT is determined by the existing `WINDOW_SIZE` constant in the OAI code-base, which is specified in the file (A.5.1) and depicted in 4.5:

Listing 4.5: readme.txt

```

-----
4.2) WINDOW_SIZE
-----

This symbolic constant configures TX and RX window sizes.

Currently, this value is set to 4096 per the size of the sequence
number field (12 bits; see 6.2.3).

```

The Contention Window (CW) value for the Listen-Before-Talk (LBT) procedure in 5G NR has been a topic of ongoing discussion within the research community, and a consensus has yet to be reached. However, a specific CW value of 32 microseconds has been selected based on empirical findings detailed in a research paper dedicated to determining the optimal CW size [26]. This choice is further reinforced by alignment with the prescribed boundaries outlined in the ETSI EN 300 328 V2.0.20 standard [10], which stipulates a minimum CW duration of 18 microseconds and a maximum of 100 microseconds.

After providing a comprehensive explanation of the LBT algorithm and its deployment strategy, the culmination of these efforts yields a final implementation that exhibits the characteristics shown in Figure 4.5:

```

[NR_MAC] Frame.Slot 0.0
[NR_MAC] Frame.Slot 128.0
[NR_MAC] Frame.Slot 256.0
[NR_MAC] [LBT] CCA period starting..
[NR_MAC] [LBT] CCA period finished and backoff period of 3790 started..
[NR_MAC] [LBT] Channel occupancy as been evaluated as: CLEAR
[NR_MAC] [LBT] Backoff period finished and packet is going to be sent!
[NR_PHY] [gNB 0][RAPROC] Frame 311, slot 19 Initiating RA procedure with preamble 13, energy 55.7 dB (I0 188, thres 120), delay 8 start symbol 0 freq index 0
[MAC] UL_Info[Frame 311, Slot 19] Calling initiate_ra_proc RACH:SFN/SLOT:311/19
[NR_MAC] [gNB 0][RAPROC] CC_id 0 Frame 311 Activating Msg2 generation in frame 312, slot 7 using RA rnti 10b 55B, new rnti f37f index 0 RA index 0
[NR_MAC] [gNB 0][RAPROC] CC_id 0 Frame 312, slotP 7: Generating RA-Msg2 DCI, rnti 0x10b, state 1, CoreSetType 2

```

Figure 4.5: DU Log showing LBT protocol in action

As elucidated previously, several factors have contributed to the challenges encountered during the attempt to implement the Listen-Before-Talk (LBT) protocol fully. These factors encompass limited familiarity with the OpenAirInterface (OAI) architecture, insufficient documentation, stringent project timeline constraints, and concerns regarding the compatibility of the required timing constraints with the LBT protocol.

Due to the intricate nature of the LBT protocol and the inherent complexities of aligning it with the OAI Radio Access Network (RAN) code base, the complete implementation of the LBT protocol has proven to be an elusive goal within the current project framework. Consequently, the ongoing effort to integrate the LBT protocol into the OAI RAN code base must be temporarily suspended, acknowledging the need for further analysis and potential refinements before proceeding.

Chapter 5

Validation and results

This chapter presents the developed network's outcome while validating its correct functionality.

5.1 Execution of the Use Cases

The methodology section outlined the testing and validation approach for specific use cases to evaluate the performance of the deployed network on the FEUP campus. This validation section will demonstrate these use cases in action and present the outcomes of the testing and verification process. The focus will be on the two essential use cases: device registration to the network and data transfer sessions. The detailed analysis and presentation of results aim to provide a comprehensive understanding of the network's capabilities and performance in real-world scenarios.

5.1.1 Device Registration Use Case

The Registration Procedure of a User Equipment (UE) involves a series of steps, as depicted in the sequence diagram shown in Figure 3.7. This section will provide a comprehensive overview of each step involved in the registration process, highlighting the challenges encountered and the corresponding solutions implemented to ensure a successful registration outcome. Through a systematic analysis, the subsequent paragraphs will present a simplified account of the registration procedure, focusing on the main aspects of the communication which are represented in the logs obtained, illustrating the iterative problem identification and resolution process:

Preconditions:

1. **RRC_IDLE:** The UE is in RRC idle state, which means it is not actively connected to the network or engaged in ongoing communication or data transfer sessions. Instead, it remains in standby mode.
2. **UE Context:** As mentioned before, this system only has one AMF managing the UE's registration and connection information, so the UE's context is not yet present.

5G-NR RRC Connection Setup:

3. **Msg1-Preamble:** The UE picks a random preamble (a specific sequence of symbols or signals used to initiate a communication process). The preamble is referenced with the Random Access Preamble ID (RAPID). The preamble transmission is a Zadoff-Chu sequence. In the DU logs, we can find where this preamble is received and the RA procedure is started:

```
[NR_PHY] [gNB 0][RAPROC] Frame 823, slot 19 Initiating RA procedure
      with preamble 7, energy 55.7 dB (IO 289, thres 120), delay 7
      start symbol 0 freq index 0
[MAC] UL_info[Frame 823, Slot 19] Calling initiate_ra_proc RACH:SFN
      /SLOT:823/19
```

4. T300;
5. Start decoding the PDCCH for the RA-RNTI;
6. Allocate Temporary C_RNTI;
7. PDCCH DCI Format 1_0 [RA-RNTI];
8. **Msg2-Random Access Response:** This Msg2 contains information from the gNB, including the Timing Advance value, contention resolution identifier (C-RNTI), and other parameters required by the UE to complete the access procedure. It provides instructions to the UE regarding the next steps to take in the process. In the DU logs, we can see the msg2 is being generated:

```
[NR_MAC] [gNB 0][RAPROC] CC_id 0 Frame 823 Activating Msg2
      generation in frame 824, slot 7 using RA rnti 10b SSB, new rnti
      2aea index 0 RA index 0
[NR_MAC] [gNB 0][RAPROC] CC_id 0 Frame 824, slotP 7: Generating RA-
      Msg2 DCI, rnti 0x10b, state 1, CoreSetType 2
```

9. UE-identity = Random number between 0 and $2^{39}-1$;
10. Extract UL Grant from the RAR;
11. **Msg3-RRCSetupRequest:** The Msg3 contains important information and parameters the gNB requires to establish the RRC connection. It includes details such as the UE identity, capabilities, and other relevant network-related information. In the EU logs, we can find the msg3 being transmitted:

```
[NR_MAC] [RAPROC][824.17] RA-Msg3 transmitted
```

And in the DU logs, we can find this msg3 being received:

```
[NR_MAC] [RAPROC] Msg3 slot 17: current slot 7 Msg3 frame 824 k2 7
      Msg3_tda_id 3
[NR_MAC] [gNB 0][RAPROC] Frame 824, Subframe 7: rnti 2aea RA state
      2
```

```
[NR_MAC] Adding UE with rnti 0x2aea
[NR_MAC] [gNB 0][RAPROC] PUSCH with TC_RNTI 0x2aea received
        correctly, adding UE MAC Context RNTI 0x2aea
[NR_MAC] [RAPROC] RA-Msg3 received (sdu_lenP 7)
```

12. PDCCH DCI Format 1_0 [C-RNTI];

13. Setup SRB1;

14. **Msg4-RRCSetup:** The Msg4 message includes important information such as the RRC configuration, security parameters, and other network-specific details required for the UE to set up its RRC connection. In the DU logs, we can find the msg4 being scheduled:

```
[NR_MAC] Activating scheduling RA-Msg4 for TC_RNTI 0x2aea (state 2)
```

In the EU logs, we can find the msg4 being received, indicated by the successful conclusion of the RA procedure:

```
[MAC] [UE 0][RAPROC] Frame 825 : received contention resolution
        identity: 0x1b048d5019e6 Terminating RA procedure
[MAC] [UE 0][825.10][RAPROC] RA procedure succeeded. CB-RA:
        Contention Resolution is successful.
```

15. T300;

16. **RRC_CONNECTED:** Right after the RA procedure succeeds, in the UE logs, it can be seen how the process completes and the UE state is set to RRC_CONNECTED:

```
[NR_RRC] [UE 0] State = NR_RRC_CONNECTED (gNB 0)
```

17. Perform the cell group configuration procedure;

18. Perform the radio bearer configuration procedure;

19. PDCCH DCI Format 0_0 [C-RNTI];

20. **Prepare the Registration Request NAS message:**

21. **RRCSetupComplete[dedicatedNAS-Message: Registration Request]:** After the UE finishes configuring the cell group and the radio bearers, and finishes preparing the Registration Request message (which is used to request registration with the network), it sends a message to the gNB. This message will inform the gNB of the successful completion of the RRC setup procedure and also deliver the Registration Request message. In the logs of the CU, the receipt of this message can be witnessed:

```
[NR_RRC] Received rrcSetupComplete, 5g_s-TMSI: 0x123456789ABC,
        amf_set_id: 0x48(72), amf_pointer: 0x34(52), 5g TMSI: 0
        x56789ABC
```

```
[NR_RRC] [FRAME 00000][gNB][MOD 00][RNTI 2aea] [RAPROC] Logical
Channel UL-DCCH, processing NR_RRCSetupComplete from UE (SRB1
Active)
[NR_RRC] [FRAME 00000][gNB][MOD 00][RNTI 2aea] UE State =
NR_RRC_CONNECTED
```

As well as the acknowledgment of receipt in the DU:

```
[NR_MAC] (UE RNTI 0x2aea) Received Ack of RA-Msg4. CBRA procedure
succeeded!
```

-
22. **AMF Selection:** The process of selecting the most suitable AMF within the 5G network, in this case, is simple, as there is only one, but we can still witness its selection in the logs from the CU:

```
[NGAP] [gNB 0] Chose AMF 'open5gs-amf0' (assoc_id 151) through
selected PLMN Identity index 0 MCC 208 MNC 93
```

23. **Allocate RAN UE NGAP ID:** The RAN UE NGAP ID is a numerical value that uniquely identifies the UE within the RAN and is used for communication and management purposes, and we can see it being set both in the CU:

```
[F1AP] Setting GNB_CU_UE_F1AP_ID 10986 associated with UE RNTI 2aea
(instance 0)
[F1AP] GNB_DU_UE_F1AP_ID 10986 associated with UE RNTI 2aea
```

As well as in the DU:

```
[F1AP] cu_ue_flap_id 10986
[F1AP] du_ue_flap_id 10986 associated with UE RNTI 2aea
[F1AP] Adding cu_ue_flap_id 10986 for UE with RNTI 2aea
```

24. **NGAP Initial UE Message[NAS-PDU: Registration Request]:** This message initiates the registration procedure and starts the process of the UE becoming an active subscriber within the 5G network.

A significant obstacle was encountered during the project when a Registration Request message triggered a Registration Reject response from the Core due to a semantic error. This problem happened as one of the expected IEs (an Information Element (IE) refers to a structured data unit used for conveying specific information within a protocol message), the 5GMM IE (5G Mobility Management Information Element is an IE used for conveying mobility-related information) was expected to be in non-clear-text, which means, encrypted, but the OAI RAN was sending it in clear-text, violating the protocol. After a lot of back-and-forth research, it was uncovered through the mailing lists that this is a bug in the OAI RAN, which was promptly fixed by applying the following patch:

At this stage, a significant obstacle was encountered, shown in Figure 5.1, as the transmission of this message to the Core resulted in a Registration reject (Semantically incorrect message). The issue arose due to the absence of an expected IE (an Information Element (IE) refers to a structured data unit used for conveying specific information within a protocol message) called the 5GMM IE (5G Mobility Management Information Element), responsible for conveying mobility-related information. Specifically, the IE was expected to be transmitted in an encrypted format, but the OAI RAN erroneously sent it in clear text, violating the protocol specification.

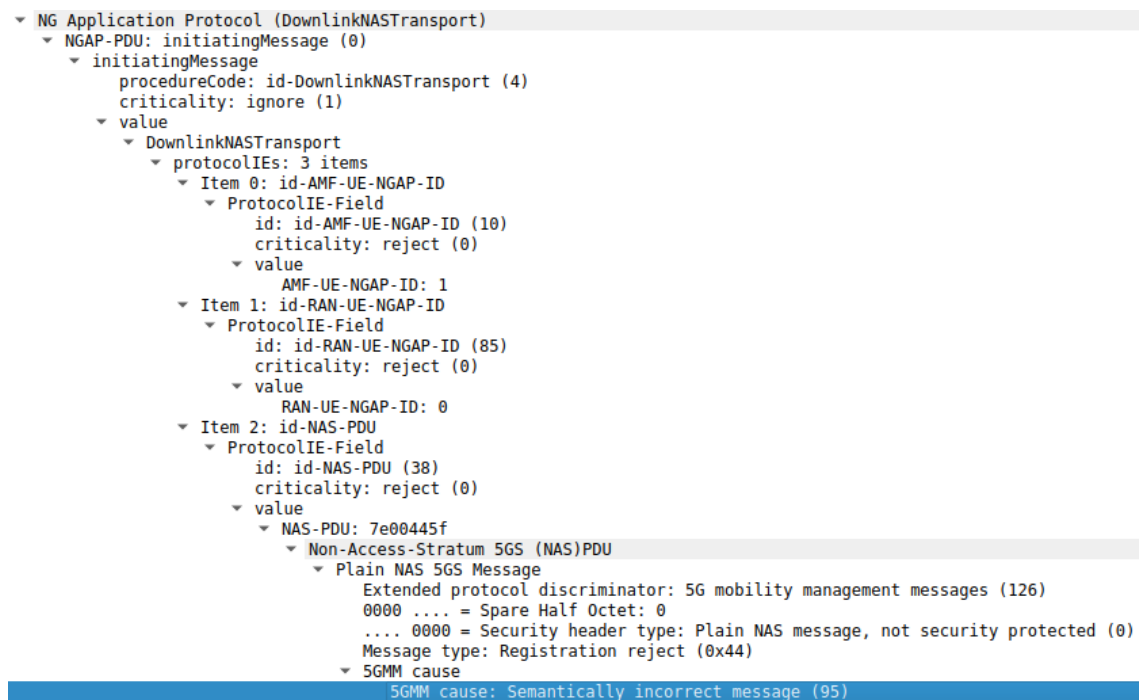


Figure 5.1: Wireshark capture: Registration reject (Semantically incorrect message)

Extensive investigation and consultation revealed that this issue was a bug in the OAI RAN implementation. Swift action was taken to rectify the problem by applying a patch provided by Cédric Roux from the OAI project. This resolved the encryption-related discrepancy and ensured compliance with the protocol requirements.

```
diff --git a/openair3/NAS/NR_UE/nr_nas_msg_sim.c b/openair3/NAS/
NR_UE/nr_nas_msg_sim.c
index 607e3e20d1..6bad21c037 100644
--- a/openair3/NAS/NR_UE/nr_nas_msg_sim.c
+++ b/openair3/NAS/NR_UE/nr_nas_msg_sim.c
@@ -448,11 +448,13 @@ void generateRegistrationRequest(
    as_nas_info_t *initialNasMsg, nr_ue_nas_t *nas)
    size += fill_suci(&mm_msg->registration_request.
        fgsmobileidentity, nas->uicc);
}
```



```

+#if 0
    mm_msg->registration_request.presencemask |=
        REGISTRATION_REQUEST_5GMM_CAPABILITY_PRESENT;
    mm_msg->registration_request.fgmmcapability.iei =
        REGISTRATION_REQUEST_5GMM_CAPABILITY_IEI;
    mm_msg->registration_request.fgmmcapability.length = 1;
    mm_msg->registration_request.fgmmcapability.value = 0x7;
    size += 3;
+#endif

    mm_msg->registration_request.presencemask |=
        REGISTRATION_REQUEST_UE_SECURITY_CAPABILITY_PRESENT;
    mm_msg->registration_request.nruesecuritycapability.iei =
        REGISTRATION_REQUEST_UE_SECURITY_CAPABILITY_IEI;

```

The provided patch modifies the file `nr_nas_msg_sim.c` in the `openair3/NAS/NR_UE` directory. The patch includes changes to the code within the `generateRegistrationRequest` function.

In the patch, the lines starting with `-` indicate the code being removed, while the lines starting with `+` indicate the code being added. In this case, the patch comments out the block of code related to the 5GMM capability information element. Using the `#if 0` preprocessor directive, the code block is effectively disabled, preventing the inclusion of the 5GMM capability IE in the generated Registration Request message.

This patch addresses the earlier bug, where the OAI RAN incorrectly sent the 5GMM IE in clear text instead of encrypted. By commenting out this code block, the patch ensures that the 5GMM capability IE is not included in the Registration Request message, resolving the protocol's encryption requirements violation.

Following the application of the mentioned patch, the subsequent exchange of messages can be observed in the logs of the AMF:

```

[amf] INFO: InitialUEMessage (../src/amf/ngap-handler.c:372)
[amf] INFO: [Added] Number of gNB-UEs is now 1 (../src/amf/context.
c:2502)
[amf] INFO: RAN_UE_NGAP_ID[0] AMF_UE_NGAP_ID[1] TAC[7] CellID[0
xe0000] (../src/amf/ngap-handler.c:533)
[amf] INFO: [suci-0-208-93-0000-0-0-0100001124] Unknown UE by SUCI
(../src/amf/context.c:1776)
[amf] INFO: [Added] Number of AMF-UEs is now 1 (../src/amf/context.
c:1563)
[gmm] INFO: Registration request (../src/amf/gmm-sm.c:985)
[gmm] INFO: [suci-0-208-93-0000-0-0-0100001124] SUCI (../src/amf/
gmm-handler.c:152)

```

As well as in the Wireshark capture conducted in the CU:

```
> Frame 11: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface enp8s0, id 0
> Ethernet II, Src: ASUSTekC_3f:af:8f (c8:7f:54:3f:af:8f), Dst: Dell_7c:34:c8 (00:26:b9:7c:34:c8)
> Internet Protocol Version 4, Src: 10.0.0.3, Dst: 10.0.0.2
> Stream Control Transmission Protocol, Src Port: 52838 (52838), Dst Port: 38412 (38412)
✓ NG Application Protocol (InitialUEMessage)
  ✓ NGAP-PDU: initiatingMessage (0)
    ✓ initiatingMessage
      procedureCode: id-InitialUEMessage (15)
      criticality: ignore (1)
    ✓ value
      ✓ InitialUEMessage
        ✓ protocolIEs: 5 items
          > Item 0: id-RAN-UE-NGAP-ID
          > Item 1: id-NAS-PDU
          > Item 2: id-UserLocationInformation
          > Item 3: id-RRCEstablishmentCause
          > Item 4: id-UEContextRequest
```

Figure 5.2: Wireshark capture: InitialUEMessage, Registration request

Obtain the UE Context from the Old AMF:

(In this single AMF network, this section does not occur)

25. Namf_Communication_UEContextTransfer Request;
26. Integrity check the 'NAS Registration Request' contained in the UE context transfer request;
27. Namf_Communication_UEContextTransfer Response;
28. Save the UE Context;

-
29. **NAS Identity Request:** The way the OAI RAN is set up, the identity is already sent to the AMF in the Initial UE Message, as shown below, so this request and its response do not occur, so it moves on to step 32:

```

  ▾ NG Application Protocol (InitialUEMessage)
    ▾ NGAP-PDU: initiatingMessage (0)
      ▾ initiatingMessage
        procedureCode: id-InitialUEMessage (15)
        criticality: ignore (1)
      ▾ value
        ▾ InitialUEMessage
          ▾ protocolIEs: 5 items
            ▾ Item 0: id-RAN-UE-NGAP-ID
              ▾ ProtocolIE-Field
                id: id-RAN-UE-NGAP-ID (85)
                criticality: reject (0)
              ▾ value
                RAN-UE-NGAP-ID: 0
            ▾ Item 1: id-NAS-PDU
              ▾ ProtocolIE-Field
                id: id-NAS-PDU (38)
                criticality: reject (0)
              ▾ value
                ▾ NAS-PDU: 7e004119000d0102f8390000000010000011422e088020000000000000
                  ▾ 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
                    > NAS key set identifier
                    ▾ 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): Thales communications & Security (93)
                      Routing indicator: 0000
                      .... 0000 = Protection scheme Id: NULL scheme (0)
                      Home network public key identifier: 0
                      MSIN: 0100001124
                    > UE security capability
                  > Item 2: id-UserLocationInformation
                  > Item 3: id-RRCEstablishmentCause
                  > Item 4: id-UEContextRequest

```

Figure 5.3: Wireshark capture: InitialUEMessage, NAS Identity

30. Derive SUCI from the Home PLMN public key;
31. NAS Identity Response;
32. AUSF Selection;

NAS Authentication and Security:

33. Nausf_UEAuthenticate_authenticate Request;
34. Nudm_UEAuthenticate_Get Request;
35. Authentication Vector Generation;

- 36. Nudm_UEAuthenticate_Get Response;
- 37. Nausf_UEAuthentication_authenticate Response;
- 38. **NAS Authentication Request:** The NAS Authentication Request message prompts the UE to provide its authentication credentials for verification. These credentials can include authentication tokens such as a RAND and an AUTN, generated by the core network:

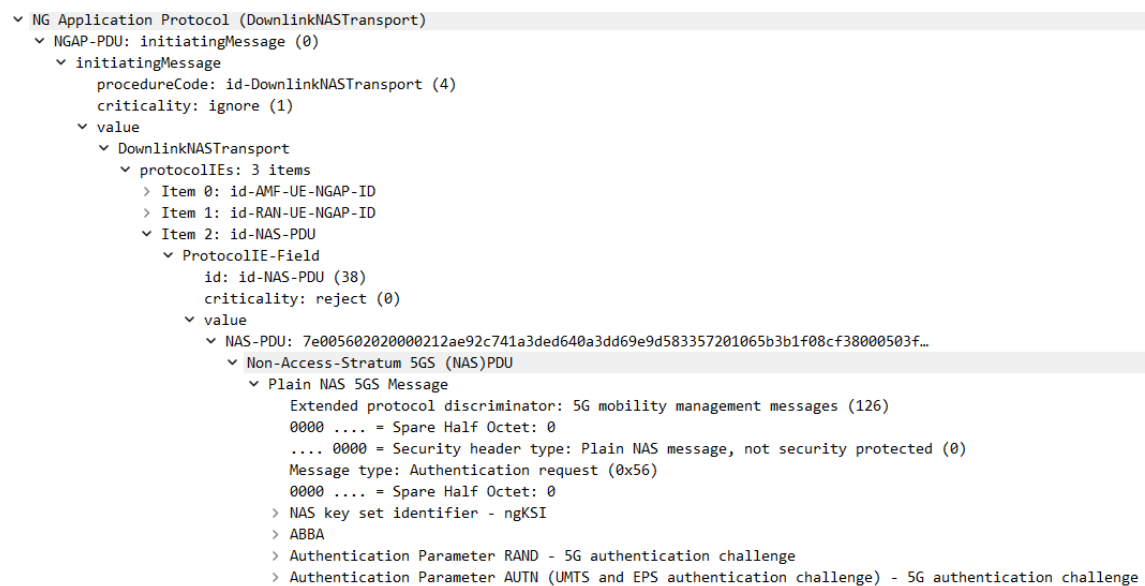


Figure 5.4: Wireshark capture: DownlinkNASTransport, Authentication request

- 39. **NAS Authentication Response:** The UE must respond to the request by sending the NAS Authentication Response, which includes the response value (AUTN) generated using the UE's authentication key (K) and the RAND received from the network:

```

▼ NG Application Protocol (UplinkNASTransport)
  ▼ NGAP-PDU: initiatingMessage (0)
    ▼ initiatingMessage
      procedureCode: id-UplinkNASTransport (46)
      criticality: ignore (1)
      ▼ value
        ▼ UplinkNASTransport
          ▼ protocolIEs: 4 items
            > Item 0: id-AMF-UE-NGAP-ID
            > Item 1: id-RAN-UE-NGAP-ID
            ▼ Item 2: id-NAS-PDU
              ▼ ProtocolIE-Field
                id: id-NAS-PDU (38)
                criticality: reject (0)
                ▼ value
                  ▼ NAS-PDU: 7e00572d10d0ebe6950d8d1b2a40fe7c53c41b6fc6
                    ▼ 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: Authentication response (0x57)
                        > Authentication response parameter
                      > Item 3: id-UserLocationInformation
                    
```

Figure 5.5: Wireshark capture: UplinkNASTransport, Authentication response

40. **NAS Security Mode Command:** The NAS Security Mode Command message instructs the UE to activate security mechanisms for the subsequent communication with the network. It includes parameters and configuration information related to the security mode, such as the type of encryption and integrity protection algorithms to be used:

```

▼ NG Application Protocol (DownlinkNASTransport)
  ▼ NGAP-PDU: initiatingMessage (0)
    ▼ initiatingMessage
      procedureCode: id-DownlinkNASTransport (4)
      criticality: ignore (1)
      ▼ value
        ▼ DownlinkNASTransport
          ▼ protocolIEs: 3 items
            > Item 0: id-AMF-UE-NGAP-ID
            > Item 1: id-RAN-UE-NGAP-ID
            ▼ Item 2: id-NAS-PDU
              ▼ ProtocolIE-Field
                id: id-NAS-PDU (38)
                criticality: reject (0)
                ▼ value
                  ▼ NAS-PDU: 7e033438d853007e005d0202028020e1360102
                    ▼ Non-Access-Stratum 5GS (NAS)PDU
                      > Security protected NAS 5GS message
                      ▼ 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: Security mode command (0x5d)
                        > NAS security algorithms
                        0000 .... = Spare Half Octet: 0
                        > NAS key set identifier - ngKSI
                        > UE security capability - Replayed UE security capabilities
                        > IMEISV request
                        > Additional 5G security information
                    
```

Figure 5.6: Wireshark capture: DownlinkNASTransport, Security Mode Command

41. **NAS Security Mode Complete:** During the project, another challenge arose when a Security Mode Command request necessitated the inclusion of the International Mobile Equipment Identity and Software Version (IMEISV) from the UE, as shown in Figure 5.7. However, the IMEISV was stored in the `ue.conf` file being given to the UE, as depicted below:

```
uicc0 = {
    imsi = "208930100001124";
    key = "465B5CE8B199B49FAA5F0A2EE238A6BC";
    opc = "E8ED289DEBA952E4283B54E88E6183CA";
    dnn= "internet";
    nssai_sst=1;
    imeisv="6754567890123413"
}
```

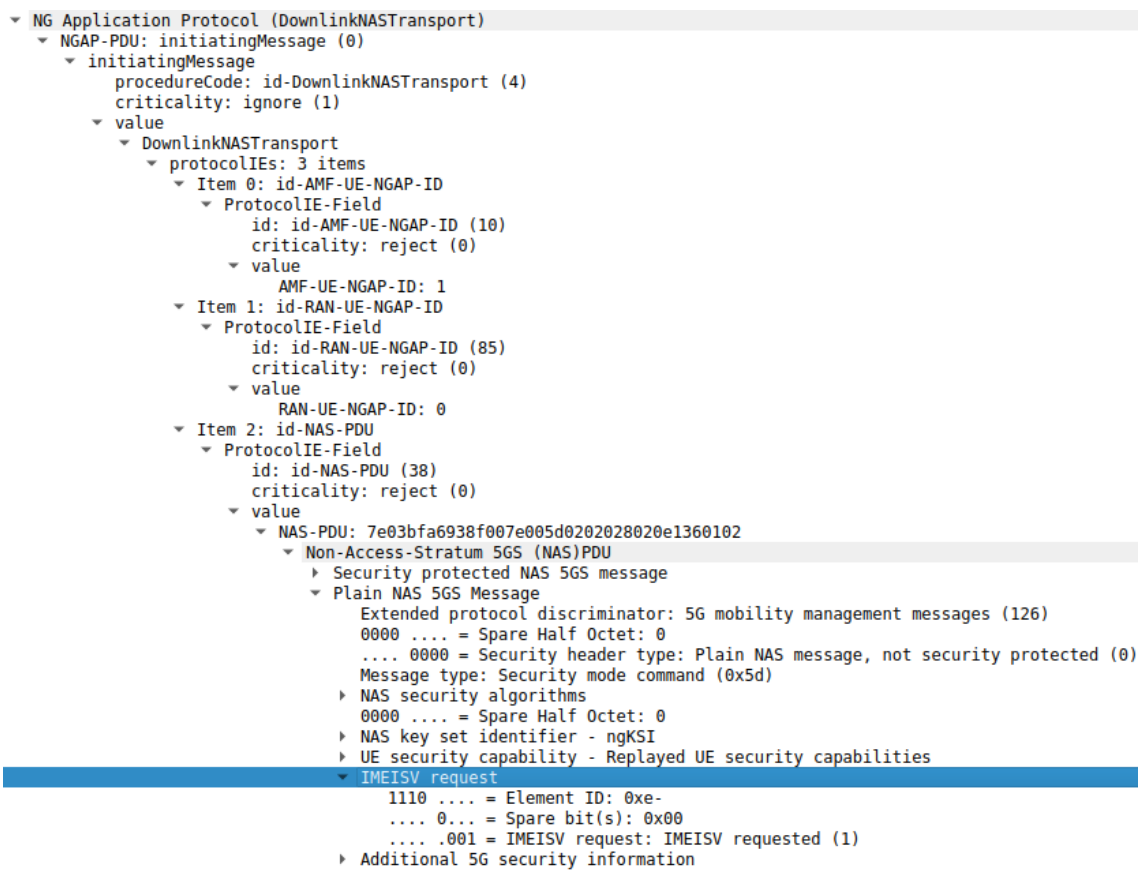


Figure 5.7: Wireshark capture: Security Mode Command, IMEISV Request)

Regardless, the UE failed to retrieve the IMEISV from the configuration file during the Security Mode Command request. This resulted in the gNB making three unsuccessful attempts to obtain the IMEISV, ultimately leading to a Registration reject message (Security Mode Rejected, unspecified), as depicted in Figure 5.8.

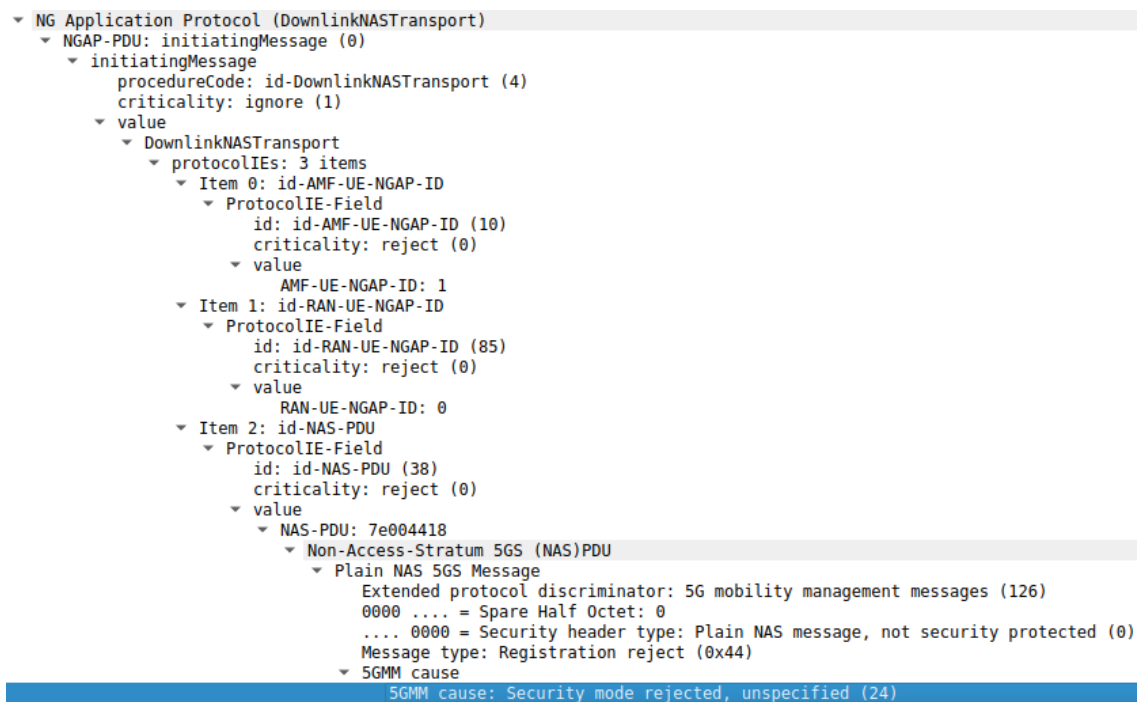


Figure 5.8: Wireshark capture: Registration reject (Security Mode Rejected, unspecified)

Further investigation revealed that a commit had been made on the OAI repository by [Vladimir Dorovskikh from Capgemini on the 9th of February, of 2023](#). This commit addressed a similar issue related to the retrieval of the IMEISV and appeared to be a potential solution for the problem encountered. Upon updating my local repository, as this commit had already been accepted and merged with the main develop branch, I retested the setup. It confirmed that the IMEISV request was now fulfilled, and as a result, the security mode was successfully established, indicating that the commit had resolved the issue.

42. Namf_Communication_RegistrationComplete_Notify;

! Steps 43 to 74 have been omitted from the analysis, as there is no available log evidence or captured messages about these steps within the deployed Open5Gs Core. It can be inferred that these steps are either bypassed or functioned as intended, considering that the UE registration process proceeded smoothly without encountering any issues at this stage.

75. Allocate AMF UE NGAP ID;

76. **Initial Context Setup Request[NAS-PDU: Registration Accept]:** The core network sends this message to the UE to request the setup of the necessary context and resources, such as

the UE security capabilities and the security key, for the UE's connection to the network as the network has accepted the UE's registration request:

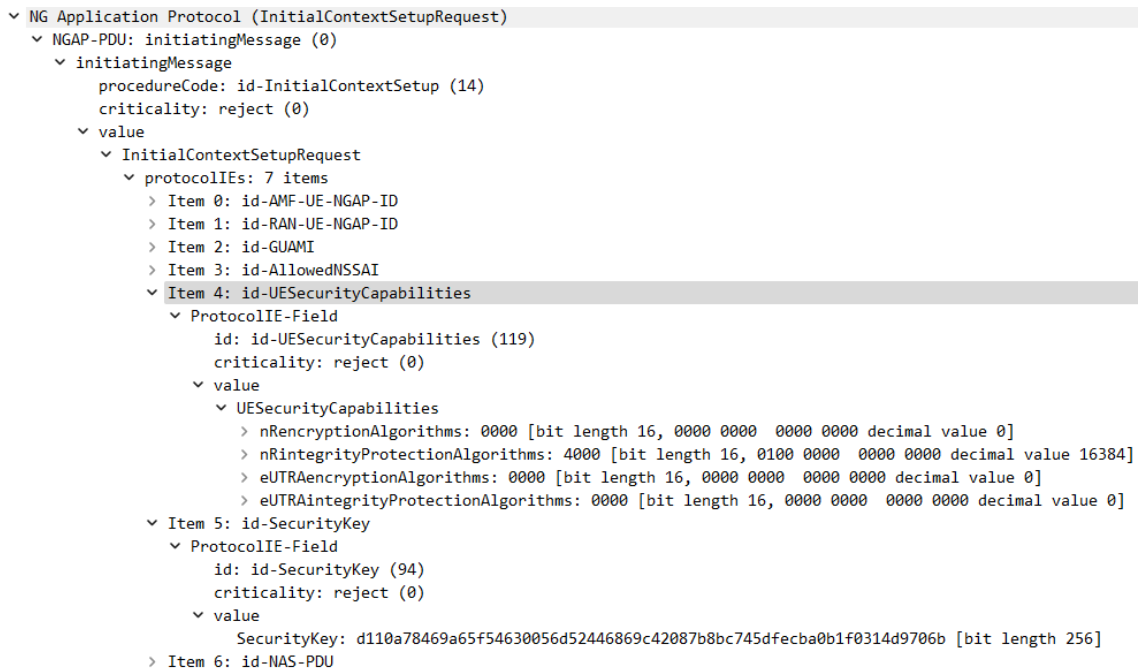


Figure 5.9: Wireshark capture: InitialContextSetupRequest

5G-NR AS Security Procedure:

77. **SecurityModeCommand:** The SecurityModeCommand message is used to command the activation of Access Stratum (AS) security, and these exchanges can be seen in the logs of the UE:

```

[NR_RRC] [UE 0] Received securityModeCommand (gNB 0)
[NR_RRC] [UE 0] SFN/SF 0/0: Receiving from SRB1 (DL-DCCH),
    Processing securityModeCommand (eNB 0)
[NR_RRC] [UE 0] Security algorithm is set to nea0
[NR_RRC] [UE 0] Integrity protection algorithm is set to nia2

```

As well as in the logs of the CU:

```

[NR_RRC] UE 2aea Logical Channel DL-DCCH, Generate
    SecurityModeCommand (bytes 3)

```

78. Derive the K-gNB key;
79. Derive K-RRC-int key associated with the Integrity Protection Algorithm;
80. Verify the integrity protection of the Security Mode Command message;

81. Derive K-UP-int key associated with the Integrity Protection Algorithm;
82. Start SRB Integrity Protect;
83. **SecurityModeComplete:** This message indicates to the network that the UE is ready to proceed with secure communication, as evidenced in the logs of the UE:

```
[NR_RRC] rrc_ue_process_securityModeCommand, security mode complete
case
[NR_RRC] driving kRRCenc, kRRCint and kUPenc from KgNB=
d110a78469a65f54630056d52446869c42087b8bc745dfecba0b1f0314d9706b

[NR_RRC] [UE 0] SFN/SF 0/0: Receiving from SRB1 (DL-DCCH), encoding
securityModeComplete (gNB 0), rrc_TransactionIdentifier: 0
[NR_RRC] securityModeComplete payload: 28 00 00 00 00 00 00 00 c0
b5 02 bc cc 7f 00 00
```

As well as in the logs of the CU:

```
[NR_RRC] [FRAME 00000][gNB][MOD 00][RNTI 2aea] received
securityModeComplete on UL-DCCH 1 from UE
```

84. Start SRB Ciphering;

5G-NR RRC Reconfiguration:

85. **RRCReconfiguration[Registration Accept(5GS registration result, PDU session status)]:** The RRCReconfiguration message is sent by the network to the UE as part of the registration acceptance procedure. It informs the UE about the successful registration and provides necessary updates or modifications to the RRC connection, such as establishing or modifying PDU sessions. This procedure can be seen in the logs of the UE:

```
[NR_RRC] [UE 0] Frame 0: Receiving from SRB1 (DL-DCCH), Processing
RRCReconfiguration (gNB 0)
[NR_RRC] Measurement Configuration is present
[RRC] Received mac_CellGroupConfig from gNB
[RRC] UE RRC instance already contains mac CellGroupConfig which
will be overwritten
[MAC] Applying CellGroupConfig from gNodeB
```

As well as in the CU:

```
[NR_RRC] do_RRCReconfiguration(): size 213
[NR_RRC] [gNB 0] Frame 0, Logical Channel DL-DCCH, Generate
NR_RRCReconfiguration (bytes 213, UE id 2aea)
```

86. **Perform the primary cell group configuration procedure:** While establishing the cell group configuration, the UE encountered a segmentation fault, leading to a crash:

```
[RRC] UE context modification response contains new CellGroupConfig
      for UE 5071, triggering reconfiguration
[RRC] UE 5071 replacing existing CellGroupConfig with new one
      received from DU
Segmentation fault
```

This unexpected problem prompted further study to determine the cause and find a resolution. The subsequent investigation revealed that the issue encountered during the cell group configuration was already being addressed in the OAI repository. Commits in a specific branch targeting the cell group configuration had been merged into the "to-be-integrated" branch of the 27th week of 2023. This branch was subsequently merged into the main develop branch on July 10th, 2023. To fix the problem, I updated my local version of the OAI repository, which successfully resolved the issue and allowed the registration of the UE to proceed.

87. Perform the secondary cell group configuration procedure;
88. Perform the radio bearer configuration procedure;
89. Initiate measurements based on the received MeasConfig;
90. Process the Registration Accept NAS message and setup PDU sessions;
91. **RRCReconfigurationComplete:** Once the UE has completed the reconfiguration tasks, it sends an RRCReconfigurationComplete message back to the network. This message acknowledges that the UE has successfully applied the requested changes and is ready to continue communicating with the updated RRC configuration. The completion of this task can be witnessed in the logs of the UE:

```
[NR_RRC] rrcReconfigurationComplete Encoded 10 bits (2 bytes)
[NR_RRC] [FRAME 00000][ UE][MOD 00][RNTI 2aea] Logical Channel UL-
      DCCH (SRB1), Generating RRCReconfigurationComplete (bytes 2,
      gNB_index 0)
```

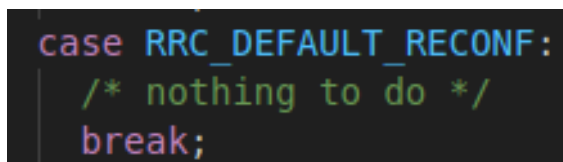
As well as in the CU:

```
[NR_RRC] Receive RRC Reconfiguration Complete message UE 2aea
```

92. PDU Session Downlink TEID;
93. **Initial Context Setup Response:** Upon extensive investigation into the absence of this specific message, I sought assistance from the OAI mailing lists to shed light on the matter. Subsequently, it was revealed to me that:

- *For the lack of initial context setup response, it's a bug, but it doesn't seem to affect connectivity.*

As depicted in Figure 5.10, it is apparent that the implementation of the aforementioned functionality is absent or incomplete:



```
case RRC_DEFAULT_RECONF:
    /* nothing to do */
    break;
```

Figure 5.10: Initial Context Setup Response: Not implemented in the OAI code base

94. **NAS Registration Complete:** Lastly, we arrive at the concluding message of the User Equipment (UE) registration process within the network, which demonstrated a successful outcome as anticipated, corroborated by the accompanying AMF logs presented below:

```
[gmm] INFO: [imsi-208930100001124] Registration complete (../src/
amf/gmm-sm.c:1917)
[amf] INFO: [Added] Number of AMF-Sessions is now 1 (../src/amf/
context.c:2523)
```

5.1.2 Data Transfer Sessions Use Case

The Data Transfer Session use case involves establishing reliable and efficient communication between User Equipment (UE) and the network. This section aims to provide a detailed account of the steps involved in the procedure, as depicted in Figure 3.8. It will highlight the challenges encountered during the session setup and the solutions implemented to ensure a smooth data transfer experience. Through a systematic analysis of the logs obtained, this paragraph will present a concise overview of the critical aspects of the communication, emphasizing the iterative process of problem identification and resolution undertaken to achieve a successful data transfer session:

Start Downlink and Uplink Data Transfer:

95. **Uplink Data:** To facilitate the seamless communication between the UE and the UPF, the establishment of a GTP (GPRS Tunneling Protocol) tunnel is a foundational step. The GTP tunnel, established upon successfully registering the UE in the network, functions as a secure and efficient pathway for data transmission. In this step, the data request from the UE is sent to the Uplink TEID, which refers to the Tunnel Endpoint Identifier assigned to the uplink direction of the GTP tunnel:

```

47 61.699205604 127.0.0.4 127.0.0.7 PFCP 713 PFCP Session Establishment Request
48 61.699521377 127.0.0.7 127.0.0.4 PFCP 160 PFCP Session Establishment Response

Frame 48: 160 bytes on wire (1280 bits), 160 bytes captured (1280 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 127.0.0.7, Dst: 127.0.0.4
User Datagram Protocol, Src Port: 8805, Dst Port: 8805
Packet Forwarding Control Protocol
  Flags: 0x21, SEID (5)
  Message Type: PFCP Session Establishment Response (51)
  Length: 112
  SEID: 0x000000000000008fc
  Sequence Number: 150240
  Spare: 0
  Node ID : IPv4 address: 127.0.0.7
  Cause : Request accepted(success)
    IE Type: Cause (19)
    IE Length: 1
    Cause: Request accepted(success) (1)
  F-SEID : SEID: 0x000000000000006cc, IPv4 127.0.0.7
  Created PDR : [Grouped IE]: PDR ID: 2
  Created PDR : [Grouped IE]: PDR ID: 3
  Created PDR : [Grouped IE]: PDR ID: 4
  [Response To: 47]
  [Response Time: 0.000315773 seconds]

```

Figure 5.11: Wireshark capture: Establishment of GTP tunnel

96. **Uplink Data:** Subsequently, the User Plane Function (UPF) transmits the data to the Internet.
97. **Nsmf_PDUSession_UpdateSMContext Request:** The AMF undertakes the modification of the Session Management Context, incorporating updates from the gNB. The Downlink TEIDs associated with each PDU session are conveyed to the SMF as part of this process.
98. **PFCP Session Modification Request:** Within the control plane, the SMF initiates session updates, triggering corresponding adjustments within the UPF's data plane:

```

49 61.720849801 127.0.0.4 127.0.0.7 PFCP 114 PFCP Session Modification Request
50 61.720953761 127.0.0.7 127.0.0.4 PFCP 65 PFCP Session Modification Response

Frame 50: 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 127.0.0.7, Dst: 127.0.0.4
User Datagram Protocol, Src Port: 8805, Dst Port: 8805
Packet Forwarding Control Protocol
  Flags: 0x21, SEID (5)
  Message Type: PFCP Session Modification Response (53)
  Length: 17
  SEID: 0x000000000000008fc
  Sequence Number: 150241
  Spare: 0
  Cause : Request accepted(success)
    IE Type: Cause (19)
    IE Length: 1
    Cause: Request accepted(success) (1)
  [Response To: 49]
  [Response Time: 0.000103960 seconds]

```

Figure 5.12: Wireshark capture: PFCP Session Modification Request

99. **Buffer Downlink Data:** Upon establishing a dedicated downlink path, the UPF discontinues the data buffering mechanism.
100. **Downlink Data:** The UPF forwards the buffered data to the gNB, utilizing the Downlink TEID specific to the relevant PDU session. All subsequent downlink data also traverses this established path.

101. **PFCP Session Modification Response:** In response to the control plane's directives, the UPF's data plane provides a corresponding response, signaling the completion of session modifications:

49	61.720849801	127.0.0.4	127.0.0.7	PFCP	114 PFCP Session Modification Request
50	61.720953761	127.0.0.7	127.0.0.4	PFCP	65 PFCP Session Modification Response


```

Frame 50: 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 127.0.0.7, Dst: 127.0.0.4
User Datagram Protocol, Src Port: 8805, Dst Port: 8805
Packet Forwarding Control Protocol
  Flags: 0x21, SEID (S)
  Message Type: PFCP Session Modification Response (53)
  Length: 17
  SEID: 0x000000000000008fc
  Sequence Number: 150241
  Spare: 0
  Cause : Request accepted(success)
    IE Type: Cause (19)
    IE Length: 1
    Cause: Request accepted(success) (1)
    [Response To: 49]
    [Response Time: 0.000103960 seconds]

```

Figure 5.13: Wireshark capture: PFCP Session Modification Request

102. **Nsmf_PDUSession_UpdateSMContext Response:** The SMF, following successful context update execution, signals the AMF about finalizing the session management context update process.

5.1.2.1 Data Transfer Sessions Use Case: Challenges

Initially, the data session between the UE and UPF was successful, though limited to specific scenarios. The interactions involving Session Establishment and Session Modification between the SMF and the UPF via the N4 interface and PFCP protocol seemed to work fine. However, looking at ICMP traffic on the GTP tunnel interface, it can be noticed that while the UPF received incoming packets, it didn't send any responses back. This hinted at a possible involvement of the FEUP firewall, causing this one-sided communication behavior by possibly dropping all ICMP packets. It was then decided to test with HTTP WGET requests, but an issue arose with DNS-related HTTP requests failing to work as expected. Oddly enough, a request without DNS involvement succeeded. This pointed to the fact that packet exchange with the UE was functional, even though specific configuration aspects required further investigation:

```

cu@cu:~/Desktop/CU/openairinterface5g/cmake_targets/ran_build/build$ wget www.google.com --bind-address=1
0.45.0.66
--2023-08-14 17:10:50-- http://www.google.com/
Resolving www.google.com (www.google.com)... failed: Temporary failure in name resolution.
wget: unable to resolve host address 'www.google.com'

```

Figure 5.14: DNS-related HTTP request failure

```

cu@cu:~/Desktop/CU/openairinterface5g/cmake_targets/ran_build/build$ wget http://198.38.82.5/ --bind-address=10.45.0.66
--2023-08-14 17:10:29-- http://198.38.82.5/
Connecting to 198.38.82.5:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html.22'

index.html.22          [ <=>          ]  1,03K  --.-KB/s   in 0,03s

2023-08-14 17:10:29 (29,5 KB/s) - 'index.html.22' saved [1058]

```

Figure 5.15: DNS-free HTTP request success

In summary, the investigation revealed that DNS queries weren't making their way to the Core, sparking suspicions of misconfigurations on the UE's end. Simply configuring Google's DNS server (8.8.8.8) proved insufficient in rectifying the issue. Notably, DNS queries weren't traversing the GTP tunnel. The solution came by adding a default IP route, where the gateway is directed to the AMF's address within that tunnel. This measure successfully addressed the problem. The following command showcases the route addition:

```
sudo route add default gw 10.45.0.1 oaitun_ue1
```

This solution successfully resolved the problem, as depicted below:

```

cu@cu:~/Desktop/CU/openairinterface5g/cmake_targets/ran_build/build$ wget www.google.com --bind-address=10.45.0.67
--2023-08-14 18:19:11-- http://www.google.com/
Resolving www.google.com (www.google.com)... 142.250.185.4, 2a00:1450:4003:803::2004
Connecting to www.google.com (www.google.com)|142.250.185.4|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html              [ <=>          ] 19,24K  --.-KB/s   in 0,04s

2023-08-14 18:19:12 (470 KB/s) - 'index.html' saved [19702]

```

Figure 5.16: DNS-related HTTP request success

The remaining challenges involved tackling a persistent `ERROR_CODE_OVERFLOW` issue with the USRP, which would lead to the UE crashing, and addressing the occurrences of `Invalid Packet (IPv6)` errors that occurred in the UPF but seemed not to affect the data transfer session itself.

Tackling the persistent `ERROR_CODE_OVERFLOW`, it seemed that the UE host had CPU frequency scaling activated. Based on further investigation, disabling this configuration is crucial to ensure the real-time operation of the testbed. To achieve this, the governor settings (CPU frequency scaling policy) were adjusted to *performance*, enabling each core to operate at its maximum frequency (5.5 GHz for the UE host, in this instance).

The previous adjustments made, specifically the deactivation of frequency scaling, did not lead to a definitive resolution of the recurring `ERROR_CODE_OVERFLOW` error. Further investigation led to the realization that the DU and CU/UE hosts, interconnected with the USRP, were equipped with the generic kernel instead of the low-latency variant. Subsequently, installing the low-latency kernel on both machines transpired, coupled with the deactivation of frequency scaling on the DU host. After a series of tests, the `ERROR_CODE_OVERFLOW` issue has not resurfaced. The

underlying proposition is that in the absence of the low-latency kernel, the processing rate of the samples relayed by the USRP proves inadequate relative to the pace at which these samples are transmitted. Consequently, the host reports a buffer overflow, as the buffers it manages cannot accommodate the received samples.

Lastly, concerning the `Invalid Packet (IPv6)` errors, indications suggest that these issues may not originate within the User Plane. Instead, it's conceivable that they pertain to a packet interaction between the UPF and an additional Core component. Notably, the specific packet in question was absent from the captures performed on the `ogstun` interface (the interface used to transmit data between the UE and the network), thus implying a divergence from typical GTP traffic. Further investigation is warranted to ascertain these errors' root causes and underlying context within the network architecture.

Chapter 6

Performance

This section comprehensively explores the operational dynamics of the established 5G network infrastructure. This chapter scrutinizes the network's performance in controlled scenarios through meticulous measurements and systematic comparisons. By examining registration times, round-trip ping times, and the temporal characteristics of distinct phases, the study sheds light on user-centric experiences and potential variations.

6.1 Experimental setup

The experimental setup employed for this study is characterized by specific parameters that define its operational framework. The frequency band is 78, with a subcarrier spacing of 30 KHz. The carrier bandwidth spans across 106 Resource Blocks (RBs), equivalent to 38.16 MHz, facilitating a broad spectrum for communication. Furthermore, the carrier frequency selected for this setup is 3619.2 MHz, strategically positioned to optimize wireless connectivity within the designated frequency band.

6.2 Throughput Analysis

The network's capabilities were evaluated through essential metrics chosen to offer a holistic view of its performance. Throughput, a vital performance indicator, was assessed using various evaluation methods, including Ookla, Fast, Nperf, and Google Fiber speed tests, with each measurement being conducted three times. These methods collectively capture the network's ability to deliver data efficiently.

To provide a comprehensive understanding of how these evaluation tools operate, it is essential to clarify their characteristics:

- **Ookla:** Ookla is a web-based platform widely used for measuring internet speed. It utilizes a web application that sends and receives data packets to assess the network's download and upload speeds. It is essential to note that Ookla primarily focuses on consumer-level internet connections and uses a variety of server locations worldwide to conduct its tests.

- **Fast:** Fast is another web-based service that measures internet performance. It relies on a web browser to run its tests and evaluates key metrics such as download and upload speeds and latency. Fast also provides insights into network performance variations over time.
- **Nperf:** Nperf is a versatile tool for assessing network performance. It conducts tests through web-based and dedicated mobile applications, allowing for a more extensive evaluation of network capabilities, measuring download bitrate, upload bitrate, latency, and the jitter of the connection, offering a well-rounded view of user experience.
- **Google Fiber:** While initially associated with Google's high-speed internet service, these tests have evolved to be more general-purpose tools that can evaluate the performance of various internet connections. They typically provide detailed information on download and upload speeds, ping times, and connection jitter, which are crucial for real-time applications. As such, they can assess the performance of a wide range of internet service providers and network types.

Each evaluation method may differ in terms of the type of traffic it generates, the use of web-based applications, and its target audience. However, collectively, they allow for a comprehensive analysis of the network's efficiency in delivering data.

The analysis revealed the following throughput results:

Evaluation Method	Download (Mbps)	Upload (Mbps)	Latency (ms)	Jitter (ms)
Ookla	20.59	7.36	-	-
Fast	15	9.5	30	-
Nperf	21.10	5.87	19.38	1.3
Fiber	15.4	6.95	-	2.33
Global Average	18.02	7.42	24.69	1.82

Table 6.1: Throughput Analysis Results

It is noteworthy that the observed download and upload speeds do not meet the minimum user-experienced data rate standards prescribed by the International Mobile Telecommunications-2020 standard, which specifies approximately 100 Mbps for download and 50 Mbps for upload speeds [8]. It is important to highlight that, leveraging the OAI (OpenAirInterface) infrastructure, the maximum throughput achieved within the scientific community currently registers at approximately 200 Mbps [6].

The subsequent experiments were executed in two distinct conditions:

- **Baseline:** First, the experiments were conducted using the established controlled environment, and the baseline configuration was used for testing. This means the Core, Distributed Unit (DU), and Central Unit (CU) were next to each other;

- **With Added Delay:** Next, to simulate a real-world application scenario, the network configuration was adjusted to reflect a more centralized deployment, where the Core was centralized. At the same time, the Distributed Unit (DU) and Central Unit (CU) could remain in the network location. A Pareto distribution delay comprising a fixed delay of 3ms and an additional variable delay of 2ms, as shown in Figure 6.1, was applied to the network to replicate real-world conditions. The utilization of a Pareto distribution in this context is motivated by its characteristic high tail, which allows for the modeling of occasional but extreme events, such as very high latency spikes. These rare latency spikes, while infrequent, can have a significant impact on network performance and can give a comprehensive and realistic performance analysis. The results are presented below, shedding light on how the configuration and distance of network elements influence the network's performance and registration times. To apply this delay to the network, the following commands were used:

```
# Apply delay to network
sudo tc qdisc add dev enp8s0 root netem delay 3ms 2ms distribution
    pareto

# Show delay set to network
sudo tc qdisc show dev enp8s0

# Remove delay added to the network
sudo tc qdisc del dev enp8s0 root
```

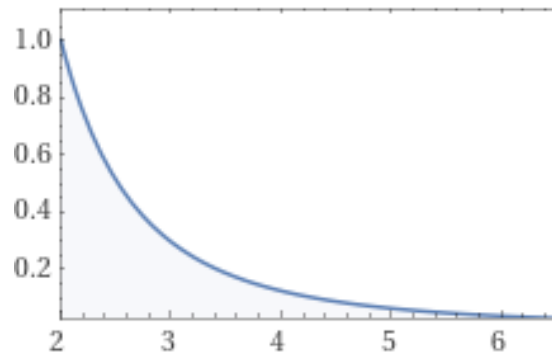


Figure 6.1: Pareto distribution delay graph | Base delay: 3ms | Additional variable delay: 2ms

Furthermore, it is imperative to note that each value presented in this analysis represents the mean (μ) and corresponding standard deviation (σ). Each experimental scenario was replicated three times, the resultant mean value was recorded, the standard deviation was calculated, and both were reported within this analysis. This stringent approach ensures a reliable representation of the network's performance metrics and offers a robust basis for drawing meaningful conclusions.

6.3 Round-trip time analysis

To assess network responsiveness, round-trip times between key network elements were measured. ICMP packets were sent and received to measure the round-trip times between the key network elements. The communication between the Core and the RAN and between the CU and the DU is executed directly between the respective machines. Conversely, the UE-to-Core communication is conducted via the established GTP tunnel between the UE and the network. The recorded round-trip times were:

6.3.1 Baseline

Measurement	μ (milliseconds)	σ (milliseconds)
Core to RAN	0.74	0.06
CU to DU	0.15	0.033
UE to Core	19.40	2.39

Table 6.2: Round-trip ping times - Baseline

6.3.2 With Added Delay

Measurement	μ (milliseconds)	σ (milliseconds)
Core to RAN	3.95	1.07
CU to DU	0.14	0.04
UE to Core	22.19	2.59

Table 6.3: Round-trip ping times - With Added Delay

These round-trip times offer insights into the delays introduced during communication within the network, such as the significant discrepancy in latency between a direct ping from the Core network to the Radio Access Network (RAN) and a ping initiated from a User Equipment (UE) within the same Core network environment, which experiences a substantially higher latency, which has captured our attention.

While it is clear that the UE's communication path involves additional network elements and complexity compared to the direct core-to-RAN communication, it is plausible that the GPRS Tunneling Protocol (GTP) tunnels, commonly utilized in 5G networks, may contribute to this latency differential. GTP tunnels, while designed to facilitate UE-core network communication efficiently, can introduce latency, especially under conditions of network congestion or sub-optimal configurations.

A comprehensive investigation is indispensable to pinpoint the specific reasons for this latency difference. Future research endeavors will entail conducting a detailed network analysis and diagnostics, considering network architecture, routing, traffic patterns, and the performance of GTP tunnels, to elucidate the root causes of this observed latency variation and potentially optimize network performance for improved user experiences.

6.4 Use Case 1: Registration of a UE in the 5G Network

As elucidated within the validation chapter (Chapter 5), it is imperative to acknowledge that not all phases of the initial use case can be meticulously delineated and temporally assessed due to the unavailability of requisite logs. Consequently, our analysis focuses solely on the examination of the four distinct phases, as presented in the ensuing tables (Tables 6.4 and 6.5). The temporal metrics associated with each phase were verified by subtracting the timing details extracted from the packet traces. These traces meticulously delineate the start and completion of each phase, a process explained in Section 5.1.1.

6.4.1 Baseline

Registration phase	μ (milliseconds)	σ (milliseconds)
5G-NR RRC Connection Setup	34.44	1.11
NAS Authentication and Security	46.25	1.73
5G-NR AS Security Procedure	235.25	10.69
5G-NR RRC Reconfiguration	15.40	0.72

Table 6.4: Use Case 1 - Baseline

6.4.2 With Added Delay

Registration phase	μ (milliseconds)	σ (milliseconds)
5G-NR RRC Connection Setup	34.26	0.98
NAS Authentication and Security	52.54	2.18
5G-NR AS Security Procedure	245.14	12.93
5G-NR RRC Reconfiguration	13.52	0.84

Table 6.5: Use Case 1 - With Added Delay

In the comparative analysis of our baseline and with-added-delay simulation experiments for Use Case 1, which involves the registration of a User Equipment (UE) in the 5G network, it is noteworthy that the two environments exhibited similarity across various phases of the registration process.

However, a notable and expected difference was detected during the NAS Authentication and Security registration phase. Notably, during this phase, communication occurs between the Core network and the Radio Access Network (RAN), representing a critical juncture in the registration of the UE.

To simulate a more realistic scenario where physical distance can introduce latency between the Core and the RAN, the previously mentioned controlled delay was strategically inserted into the simulated environment, mimicking the expected time delay in this phase. This divergence allowed us to capture and assess the potential effects of latency on the registration process, shedding light on the performance and responsiveness of the 5G network under conditions that more closely mirror real-world operational distances.

What proved to be unexpected when comparing the duration's of these registration phases with the previously mentioned ping round-trip times is the substantial disparity observed. This distinction remains pronounced even when considering the numerous messages exchanged during each phase. While a conclusive explanation is currently lacking regarding whether this discrepancy stems from hardware-related issues, shortcomings in the 5G implementation, or other factors, it is noteworthy that the synthetic delay introduced does not appear to be the primary contributing factor. This observation is reinforced by the fact that the difference between the baseline and the configuration with added delay remains relatively insignificant.

6.5 Use Case 2: Data Transfer

This section explores data transfer times by downloading controlled files from a recognized source repository, [TestFileDownload](#), and registering the average duration required for the download to reach completion. Each measured value represents the mean result obtained from five separate attempts. This investigation provides essential insights into data transfer efficiency under various network and environmental conditions. The aim is to enhance understanding of the network's ability to handle real-world data-intensive scenarios through systematic experimentation and analysis.

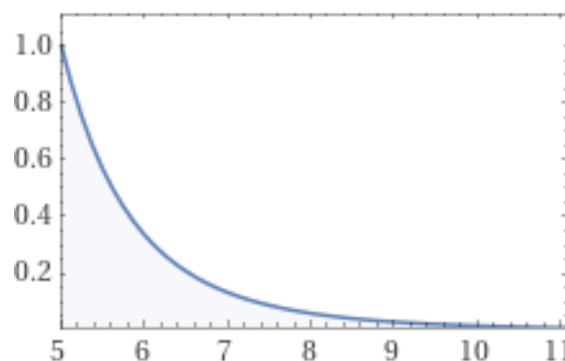


Figure 6.2: Pareto distribution delay graph | Base delay: 3ms | Additional variable delay: 5ms

A second, more severe Pareto distribution delay comprising a fixed delay of 3ms and an additional variable delay of 5ms, as shown in Figure 6.2, was also applied to the network to analyze

the impact of a variable network delay. The results are presented below, shedding light on how the configuration and distance of network elements influence the network's performance:

6.5.1 Baseline

File size (MB)	μ (milliseconds)	σ (milliseconds)	Throughput (Mbit/s)
1	512	16.06	15.63
10	4574	58.31	17.49
20	8680	148.58	18.43

Table 6.6: Use Case 2 - Baseline: File download times and throughput

6.5.2 With Added Delay #1

File size (MB)	μ (milliseconds)	σ (milliseconds)	Throughput (Mbit/s)
1	676	97.74	11,83
10	5841	316.39	13,70
20	10269	897.37	15,58

Table 6.7: Use Case 2 - With Added Delay of 3ms with 2ms jitter: File download times and throughput

6.5.3 With Added Delay #2

File size (MB)	μ (milliseconds)	σ (milliseconds)	Throughput (Mbit/s)
1	745	216.62	10,74
10	6063	1632.56	13,19
20	11223	2416.12	14,26

Table 6.8: Use Case 2 - With Added Delay of 3ms with 5ms jitter: File download times and throughput

These results provide a comprehensive understanding of the network's performance in three contrasting scenarios, allowing for a deeper analysis of user-centric experiences and potential performance variations.

The predominant observation, notwithstanding the observed variability among the results, a characteristic that may be attributed to a range of factors, including potential imperfections in the physical infrastructure, is the notably lower throughput achieved, as mentioned before in 6.2.

It is imperative to acknowledge that several other researchers and investigators have also encountered analogous challenges during their network deployment endeavors, as highlighted by additional [sources](#)¹.

Another noteworthy observation, is that the observed significant decrease in throughput resulting from the deliberate introduction of a delay between RAN and Core components, simulating physical distance, emphasizes the network's sensitivity to timing variations.

¹<https://www.linkedin.com/feed/update/urn:li:activity:7101310848068923392/>

Chapter 7

Conclusion

In conclusion, this project has marked a significant milestone in the development and deployment of a 5G Non-Public Network (NPN) proof-of-concept, utilizing open-source software successfully within the FEUP campus environment. The journey was characterized by diligent research, robust problem-solving, and the documentation of valuable solutions to challenges that arose during deployment. These insights contribute substantially to the evolving landscape of NPN development.

While substantial progress was achieved in implementing the Listen-Before-Talk (LBT) protocol, it is essential to acknowledge the complexity inherent in the OpenAirInterface Radio Access Network (OAI RAN) codebase. The intricate nature of the codebase presented challenges that, within the scope of this project, prevented the full realization of the LBT protocol's deployment.

Moreover, the observed network performance fell notably short of initial expectations. It is noteworthy, however, that this performance gap was not unique to this project, as other researchers have encountered similar difficulties. This aspect, together with the insights gained through this endeavor, presents an opportunity for future analysis and investigation.

In terms of future work, it is imperative to explore avenues for further improvement, including a more comprehensive examination of the intricate OAI RAN codebase and the refinement of network configurations to address the observed performance gaps. Additionally, the motivation still exists to consider the integration of the LBT protocol, enabling the network's utilization within the unlicensed spectrum. This expansion, though, warrants a thorough analysis of the potential implementation in forthcoming research efforts.

References

- [1] 3GPP. Technical specification group radio access network; nr; ng-ran; general aspects and principles for functional splits; stage 2 (release 15). ETSI TS 138 470, ETSI, 2020.
- [2] 3rd Generation Partnership Project (3GPP). 3GPP technical specification group radio access network; 38 series. <https://www.3gpp.org/dynareport?code=38-series.htm>. Accessed: 2023-05-20.
- [3] 3rd Generation Partnership Project (3GPP). System architecture for the 5g system (5gs). Technical report, 3GPP, Year of publication, e.g., 2021.
- [4] 5G-PPP Non-Public Networks Working Group. 5g-ppp white paper on non-public networks. https://5g-ppp.eu/wp-content/uploads/2022/11/WhitePaperNPN_MasterCopy_V1.pdf, 2022. Accessed: March 30, 2023.
- [5] R. Ali, B. Kim, S.W. Kim, H.S. Kim, and F. Ishmanov. (relbt): A reinforcement learning-enabled listen-before-talk mechanism for lte-laa and wi-fi coexistence in iot. *Computer Communications*, 150:498–505, 2020.
- [6] OpenAirInterface Software Alliance. Openairinterface new throughput record: 200 mbps. <https://www.youtube.com/watch?app=desktop&v=iGvN4XIcEJE>, 2022. Accessed in 2023.
- [7] Rojeena Bajracharya, Rakesh Shrestha, and Haejoon Jung. Future is unlicensed: Private 5g unlicensed network for connecting industries of future. *Sensors*, 20(10), 2020.
- [8] Erik Dahlman, Stefan Parkvall, and Johan Skold. *5G NR: The Next Generation Wireless Access Technology*. Academic Press, Inc., USA, 1st edition, 2018.
- [9] Francisco Joaquim De Souza Neto, Edson Amatucci, Nadia Adel Nassif, and Pedro Augusto Marques Farias. Analysis for comparison of framework for 5g core implementation. In *2021 International Conference on Information Science and Communications Technologies (ICISCT)*, pages 1–5, 2021.
- [10] ETSI. Wideband transmission systems; data transmission equipment operating in the 2,4 ghz band; harmonised standard for access to radio spectrum. ETSI TS 300 328, ETSI, 2019.
- [11] ETSI. ETSI TS 129 500 V17.9.0: 5G System; Non-Public Networks (NPN); Stage 2 (Release 17). Technical Specification 129500, ETSI, 2021.
- [12] ETSI Technical Committee Network Technologies. 5g; nr; architecture description. Technical Specification 138 473, ETSI, January 2021.

- [13] ETSI Technical Specification. Dect-2020 new radio (nr); part 1: Overview; release 1. ETSI TS 103 636-1 V1.4.1, ETSI, 2023.
- [14] EventHelix. 5g-nr standalone access registration. <https://www.eventhelix.com/5G/standalone-access-registration/5g-standalone-access-registration.pdf>, 2023. [Accessed: June 30, 2023].
- [15] Mohammed Hirzallah, Marwan Krunz, Balkan Kecicioglu, and Belal Hamzeh. 5g new radio unlicensed: Challenges and evaluation. *IEEE Transactions on Cognitive Communications and Networking*, 7(3):689–701, 2021.
- [16] Marin Ivezic. 5g security - 5g core sba components architecture. <https://5g.security/5g-edge-miot-technology/5g-core-sba-components-architecture/>, 2022. Accessed: March 30, 2023.
- [17] Chung K. Kim, Chan S. Yang, and Chung G. Kang. Adaptive listen-before-talk (lbt) scheme for lte and wi-fi systems coexisting in unlicensed band. In *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 589–594, 2016.
- [18] Minjae Kim, Kyungmin Park, Jonggeun Park, Youngsoo Kim, Lee Jonghoon, and Daesung Moon. Analysis of current 5g open-source projects. *Electronics and Telecommunications Trends*, 36(2):83–92, 2021.
- [19] Hema Krishnan, M.Sudheep Elayidom, and T. Santhanakrishnan. MongoDB – a comparison with nosql databases. *International Journal of Scientific and Engineering Research*, 7:1035–1037, 05 2016.
- [20] Harit Mehta. Recent advances in cognitive radios. cse.wustl.edu, 2014. Retrieved in 2023.
- [21] Jinyoung Oh, Younsun Kim, Yingzhe Li, Jonghyun Bang, and Juho Lee. Expanding 5g new radio technology to unlicensed spectrum. In *2019 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2019.
- [22] Open5GS. Open5gs documentation. <https://open5gs.org/open5gs/docs/guide/01-quickstart/>, 2023. [Accessed: March 30, 2023].
- [23] Jose Ordonez-Lucena, Jesús Folgueira Chavarria, Luis M. Contreras, and Antonio Pastor. The use of 5g non-public networks to support industry 4.0 scenarios. In *2019 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–7, 2019.
- [24] Jonathan Prados-Garzon, Pablo Ameigeiras, Jose Ordonez-Lucena, Pablo Muñoz, Oscar Adamuz-Hinojosa, and Daniel Camps-Mur. 5g non-public networks: Standardization, architectures and challenges. *IEEE Access*, 9:153893–153908, 2021.
- [25] Ahmad Rostami, Dhruvin Patel, Madhusudan Giyyarpuram, and Finn Pedersen. 5g non-public network for industrial iot: Operation models, 2023.
- [26] Yujae Song, Ki Won Sung, and Youngnam Han. Coexistence of wi-fi and cellular with listen-before-talk in unlicensed spectrum. *IEEE Communications Letters*, 20(1):161–164, 2016.
- [27] Gábor Soós, Dániel Ficzer, Tamás Seres, Sándor Veress, and István Németh. Business opportunities and evaluation of non-public 5g cellular networks—a survey. *Infocommunications Journal*, 12(3):31–38, 2020.

- [28] Spectrum Tracker. Frequency spectrum data for portugal. <https://www.spectrum-tracker.com/Portugal>, 2023. Source of data: Frequency spectrum regulator Portugal. Date of last update: 11.08.2023.
- [29] Frederick W. Vook, Amitava Ghosh, Emilio Diarte, and Michael Murphy. 5g new radio: Overview and performance. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 1247–1251, 2018.
- [30] Miaowen Wen, Qiang Li, Kyeong Jin Kim, David López-Pérez, Octavia A. Dobre, H. Vincent Poor, Petar Popovski, and Theodoros A. Tsiftsis. Private 5g networks: Concepts, architectures, and research landscape. *IEEE Journal of Selected Topics in Signal Processing*, 16(1):7–25, 2022.
- [31] Wikipedia contributors. Data link layer — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Data_link_layer&oldid=1161524078, 2023. [Online; accessed 30-June-2023].

Appendix A

Code Listings

A.1 5G Core

A.1.1 amf.yaml

```
(...)  
  
amf:  
  sbi:  
    - addr: 127.0.0.5  
      port: 7777  
  ngap:  
    - addr: 10.0.0.2 # <--- ID of the Core machine  
  metrics:  
    - addr: 127.0.0.5  
      port: 9090  
  guami:  
    - plmn_id:  
        mcc: 208 # <--- Chosen MCC  
        mnc: 93 # <--- Chosen MNC  
      amf_id:  
        region: 2  
        set: 1  
  tai:  
    - plmn_id:  
        mcc: 208 # <--- Chosen MCC  
        mnc: 93 # <--- Chosen MNC  
      tac: 7 # <--- Chosen TAC  
  plmn_support:  
    - plmn_id:  
        mcc: 208 # <--- Chosen MCC  
        mnc: 93 # <--- Chosen MNC  
      s_nssai:
```

```

        - sst: 1
security:
  integrity_order : [ NIA2, NIA1, NIA0 ]
  ciphering_order : [ NEA0, NEA1, NEA2 ]
network_name:
  full: Open5GS
amf_name: open5gs-amf0

(...)

```

A.1.2 mme.yaml

```

(...)

mme:
  freeDiameter: /etc/freeDiameter/mme.conf
  slap:
    - addr: 127.0.0.2
  gtpc:
    - addr: 127.0.0.2
  metrics:
    - addr: 127.0.0.2
    port: 9090
  gummei:
    plmn_id:
      mcc: 208 # <--- Chosen MCC
      mnc: 93  # <--- Chosen MNC
    mme_gid: 2
    mme_code: 1
  tai:
    plmn_id:
      mcc: 208 # <--- Chosen MCC
      mnc: 93  # <--- Chosen MNC
    tac: 7     # <--- Chosen TAC
  security:
    integrity_order : [ EIA2, EIA1, EIA0 ]
    ciphering_order : [ EEA0, EEA1, EEA2 ]
  network_name:
    full: Open5GS
  mme_name: open5gs-mme0

(...)

```

A.2 Control Unit (CU)

A.2.1 cu_gnb.conf

(...)

```
// Tracking area code, 0x0000 and 0xfffe are reserved values
tracking_area_code = 7;      # <--- Chosen TAC
plmn_list = ({
    mcc = 208;                # <--- Chosen MCC
    mnc = 93;                 # <--- Chosen MNC
    mnc_length = 2;
    snssaiList = ({ sst = 1 })
});

nr_cellid = 12345678L;
force_256qam_off = 1;

tr_s_preference = "f1";      # <--- F1 interface, which represents the
                             functional split between the CU and DU

local_s_if_name = "enp7s0"; # <--- Network interface which connects the
                             CU to the DU machine
local_s_address = "10.42.0.1"; # <--- CU machine IP on that interface
remote_s_address = "10.42.0.2"; # <--- DU machine IP on that interface
local_s_portc = 501;
local_s_portd = 2152;
remote_s_portc = 501;
remote_s_portd = 2152;
min_rxtxtime                                = 6;
```

(...)

```
////////// AMF parameters:
amf_ip_address = ( { ipv4   = "10.0.0.2"; # <--- Network interface
                    which connects the CU to the DU machine
                    ipv6    = "192:168:30::17";
                    active  = "yes";
                    preference = "ipv4";
                    }
                );

NETWORK_INTERFACES :
{

    GNB_INTERFACE_NAME_FOR_NG_AMF = "enp8s0"; # <--- Network interface
        which connects the CU (RAN) to the Core machine
```

```

        GNB_IPV4_ADDRESS_FOR_NG_AMF      = "10.0.0.3"; # <--- CU machine IP on
            that interface
        GNB_INTERFACE_NAME_FOR_NGU       = "enp8s0"; # <--- Network interface
            which connects the CU (RAN) to the Core machine
        GNB_IPV4_ADDRESS_FOR_NGU        = "10.0.0.3"; # <--- CU machine IP on
            that interface
        GNB_PORT_FOR_S1U                  = 2152; # Spec 2152
    };
}
);

(...)

```

A.2.2 nr_nas_msg_sim.c

```

void generateRegistrationRequest(as_nas_info_t *initialNasMsg, nr_ue_nas_t
    *nas)
{
    // (...)

    #if 0
        mm_msg->registration_request.presencemask |=
            REGISTRATION_REQUEST_5GMM_CAPABILITY_PRESENT;
        mm_msg->registration_request.fgmmcapability.iei =
            REGISTRATION_REQUEST_5GMM_CAPABILITY_IEI;
        mm_msg->registration_request.fgmmcapability.length = 1;
        mm_msg->registration_request.fgmmcapability.value = 0x7;
        size += 3;
    #endif

    // (...)
}

```

A.3 Distributed Unit (DU)

A.3.1 channel_occupancy.h

```

extern int channel_occupancy;
extern int CCA;
extern int CONTENTION_WINDOW_SIZE;
extern int backoff;

#ifndef IS_CHANNEL_CLEAR_H // Include guard to prevent multiple
    definitions

```

```
#define IS_CHANNEL_CLEAR_H
```

```
int is_channel_clear();
```

```
#endif
```

A.3.2 channel_occupancy.c

```
#include "channel_occupancy.h"
```

```
int channel_occupancy = 1; // 0 = false | 1 = true
```

```
int CCA = 0.000032; // 32 microseconds = 0.000032 seconds
```

```
int CONTENTION_WINDOW_SIZE = 4096;
```

```
int backoff = 0;
```

```
int is_channel_clear() {  
    // not implemented  
    return 1;  
}
```

A.3.3 CMakeLists.txt

```
# (...)
```

```
#####
```

```
# Utilities
```

```
#####
```

```
add_library(HASHTABLE
```

```
    ${OPENAIR_DIR}/common/utls/hashtable/hashtable.c
```

```
    ${OPENAIR_DIR}/common/utls/hashtable/obj_hashtable.c
```

```
)
```

```
include_directories(${OPENAIR_DIR}/common/utls/hashtable)
```

```
add_library(UTIL
```

```
    ${OPENAIR_DIR}/common/utls/LOG/log.c
```

```
    ${OPENAIR_DIR}/common/utls/global/channel_occupancy.c # <--- Add my LBT  
    file for compilation
```

```
    ${OPENAIR_DIR}/common/utls/LOG/vcd_signal_dumper.c
```

```
    ${OPENAIR2_DIR}/UTIL/MATH/oml.c
```

```
    ${OPENAIR2_DIR}/UTIL/OPT/probe.c
```

```
    ${OPENAIR_DIR}/common/utls/threadPool/thread-pool.c
```

```
    ${OPENAIR_DIR}/common/utls/utls.c
```

```
    ${OPENAIR_DIR}/common/utls/system.c
```

```
    ${OPENAIR_DIR}/common/utls/time_meas.c
```

```
    ${OPENAIR_DIR}/common/utls/time_stat.c
```

```
)
```



```
# (...)
```

A.3.4 nr_prach_procedures.c

```
// (...)
```

```
#include "common/utils/global/channel_occupancy.h"
```

```
bool custom_logs = true;
```

```
// (...)
```

```
void Ll_nr_prach_procedures(PHY_VARS_gNB *gNB,int frame,int slot) {
```

```
    // (...)
```

```
    prachStartSymbol = prach_pdu->prach_start_symbol+prach_oc*N_dur;
```

```
    /* Here I can regularly check the channel occupancy and update the
       global variable */
```

```
    channel_occupancy = (max_preamble_energy[0] > gNB->measurements.
```

```
        prach_I0 + gNB->prach_thres ? 0 : 1);
```

```
    if(backoff > 0) backoff--;
```

```
    // (...)
```

```
    /* This is detecting an incoming communication by an UE */
```

```
    if ((gNB->prach_energy_counter == 100) && (max_preamble_energy[0] >
        gNB->measurements.prach_I0+gNB->prach_thres)) {
```

```
        // START OF THE LBT
```

```
        // Random seed initialization
```

```
        srand(time(NULL));
```

```
        // LBT algorithm states
```

```
        enum { CCA, BACKOFF } state = CCA;
```

```
        int finished = 0;
```

```
        while (!finished) {
```

```
            switch (state) {
```

```
                case CCA:
```

```
                    LOG_W(NR_MAC, "[LBT] CCA period starting...\n");
```

```
                    // Monitor channel activity for CCA period
```

```

        sleep(CCA);

        // Check if channel is still clear after CCA period
        if (is_channel_clear())
            state = BACKOFF;
        else {
            state = CCA;
        }
        break;

    case BACKOFF:
        // Generate a random backoff counter
        backoff = rand() % CONTENTION_WINDOW_SIZE-1;
        LOG_W(NR_MAC, "[LBT] CCA period finished and backoff period of %i
            started..\n", backoff);

        while (backoff > 0) {
            // Monitor channel activity during backoff
            if (!is_channel_clear()) {
                // Channel became busy during backoff
                state = CCA;
                break;
            }

            // This wouldn't be needed if the function was implemented
            backoff--;
        }
        finished = 1;
        break;
    }
}
LOG_A(NR_MAC, "[LBT] Channel occupancy as been evaluated as: CLEAR\n")
;

// END OF THE LBT

// (...)

```

A.3.5 du_gnb.conf

```

(...)

// Tracking area code, 0x0000 and 0xfffe are reserved values
tracking_area_code = 7;    # <--- Chosen TAC
plmn_list = ({
mcc = 208;                  # <--- Chosen MCC

```

```

mnc = 93;                                # <--- Chosen MNC
mnc_length = 2;
snssaiList = ({ sst = 1 }) );

(...)

MACRLCs = (
{
    num_cc          = 1;
    tr_s_preference = "local_L1";
    tr_n_preference = "f1";  # <--- F1 interface, which represents the
                             functional split between the CU and DU
    local_n_if_name = "enp7s0"; # <--- Network interface which connects the
                                CU to the DU machine
    local_n_address = "10.42.0.2"; # <--- DU machine IP on that interface
    remote_n_address = "10.42.0.1"; # <--- CU machine IP on that interface
    local_n_portc = 501;
    local_n_portd = 2152;
    remote_n_portc = 501;
    remote_n_portd = 2152;

}
);

(...)

RUs = (
{
    local_rf      = "yes";
    nb_tx         = 1;
    nb_rx         = 1;
    att_tx        = 0;
    att_rx        = 0;
    bands         = [78];  # <--- Chosen frequency band
    max_pdschReferenceSignalPower = -27;
    max_rxgain    = 114;
    eNB_instances = [0];
    #beamforming 1x4 matrix:
    bf_weights = [0x00007fff, 0x0000, 0x0000, 0x0000];
    clock_src = "internal";

}
);

(...)

```

A.4 User Equipment (UE)

A.5 Utils

A.5.1 readme.txt

(...)

4.2) WINDOW_SIZE

This symbolic constant configures TX and RX window size.

Currently this value is set to 4096 in accordance with the size of
sequence
number field (12 bits, see 6.2.3).

(...)