FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Control and Positioning of a 5G Radio Access Node Deployed in a Mobile Robotic Platform

David Miguel de Almeida Coimbra Maia

MASTERS DISSERTATION

Mestrado em Engenharia Eletrotécnica e de Computadores

Orientador: Prof. Manuel Ricardo

Coorientador: André Coelho

July 1, 2022

Abstract

5G has become increasingly popular nowadays, mainly due to its characteristics which enable high bitrates and low latencies. A promising 5G research area is the usage of Mobile Platforms, such as robots and drones, for the transportation of radio stations in order to enable the on-demand placement of 5G communications cells. The flexibility provided by an on-demand mobile 5G radio access network paves the way for its utilization in emergency scenarios, temporary events and remote locations.

At the same time, the Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) paradigms have emerged as enablers for reconfigurable mobile wireless networks. On the one hand, the NFV paradigm enables virtualizing network services so that service providers are able to offer new services and applications on-demand, without requiring additional hardware resources, which enhances the 5G network scalability. On the other hand, the SDN paradigm creates a logical separation that allows 5G networks to be more easily manageable and controllable. Both concepts are essential for the integration of independent network functions that enable the management and monitoring of virtualized networks.

The main contribution of this dissertation is a private standalone on-demand 5G network. The proposed solution consists of a 5G radio station (gNB) carried by a mobile robotic platform, which provides connectivity for 5G mobile terminals. In addition, an On-Demand Mobility Management Function, called ODMMF, was developed and integrated into the 5G Core network. ODMMF consists of a service based on Hypertext Transfer Protocol (HTTP) for monitoring the radio conditions of the served User Equipments (UEs), and remotely position and control the mobile robotic platform in real-time leveraged by its video cameras. The proposed private on-demand 5G network enables the management of radio and computing resources by a single entity while providing 5G wireless connectivity anywhere, anytime. The mobility enabled by the mobile robotic platform enables the flexible positioning of the gNB, taking into account the users' radio link quality.

Resumo

O 5G tem-se tornado cada vez mais popular, principalmente devido às suas características que permitem débitos elevados e baixas latências. Uma promissora área de investigação em 5G é o uso de Plataformas Móveis, como robôs e *drones*, para o transporte de estações rádio, por forma a permitir o posicionamento a pedido de células de comunicações 5G. A flexibilidade fornecida por uma rede de acesso rádio móvel 5G a pedido abre caminho para a sua utilização em cenários de emergência, eventos temporários e locais remotos.

Ao mesmo tempo, os paradigmas de Virtualização de Funções de Rede (NFV) e Redes Definidas por *Software* (SDN) emergiram como potenciadores de rede móveis sem fios reconfiguráveis. Por um lado, o paradigma NFV permite a virtualização de serviços de rede de modo a que operadores de telecomunicações sejam capazes de oferecer serviços e aplicações a pedido, sem a necessidade de recursos adicionais de *hardware*, o que aumenta a escalabilidade da rede 5G. Por outro lado, o paradigma SDN cria uma separação lógica que permite que as redes 5G sejam mais facilmente geridas e controladas. Ambos os conceitos são essenciais para a integração de funções de rede independentes que possibilitam a gestão e monitorização de redes virtualizadas.

A principal contribuição desta dissertação é uma rede privada 5G autónoma a pedido. A solução proposta consiste numa estação de rádio 5G (gNB) transportada por uma plataforma robótica móvel, que fornece conetividade a terminais móveis 5G. Além disso, uma função para gestão da mobilidade a pedido, denominada ODMMF, foi desenvolvida e integrada na rede *Core* 5G. ODMMF consiste num serviço, baseado em *Hypertext Transfer Protocol* (HTTP), para monitorizar as condições de rádio dos terminais dos utilizadores (UEs) servidos, e posicionar e controlar remotamente a plataforma robótica em tempo real, alavancado pelas suas câmaras de vídeo. A rede privada 5G a pedido proposta permite a gestão de recursos rádio e computação por uma única entidade, ao mesmo tempo que fornece conetividade sem fios 5G em qualquer lugar, a qualquer hora. A mobilidade possibilitada pela plataforma robótica móvel permite o posicionamento flexível do gNB, levando em consideração a qualidade da ligação rádio dos utilizadores.

Acknowledgments

I would like to thank my supervisors, Professor Manuel Ricardo and André Coelho, for all their support and concern during the development of this dissertation. In addition, I would like to thank all the collaborators from the Center for Telecommunications and Multimedia (CTM) of INESC TEC for welcoming me and providing me with the equipment and space necessary for the development of this dissertation.

I would also like to thank my colleagues for all the time we spent together working and discussing the best approach to successfully complete this dissertation.

A massive thank you to all my friends and family, especially my mother for being the greatest example anyone could follow. Finally, I would like to thank my beloved city of Vila Nova de Gaia for its support during my academic path.

David Maia

Contents

List of Figures

List of Tables

Abbreviations

3GPP	3rd Generation Partnership Project
AI	Artificial Intelligence
AMF	Access and Mobility Management Function
AN	Access Network
AUSF	Authentication Server Function
BBU	Based Band Unit
BLER	Block Error Rate
B5G	Beyond 5G
COTS	Commercial Off-The-Shelf
C-RAN	Cloud-RAN
CS	Central Station
DNS	Domain Name System
eMBB	enhanced Mobile Broadband
EPC	Evolved Packet Core
FAP	Flying Access Points
FMAP	Flying Mesh Access Points
FMN	Flying Multi-Hop Network
FPGA	Field Programmable Gate Arrays
gNB	gNodeB
GTP-U	General Packet Radio Service Tunnelling Protocol
GWP	Gateway UAV Placement
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
IMSI	International Mobile Subscriber Identity
ICMP	Internet Control Message Protocol
IP	Internet Protocol
LTE	Long Term Evolution
MAC	Medium-Access Control
MCC	Mobile Country Code
MCS	Modulation Coding Scheme
MNC	Mobile Network Code
MNO	Mobile Network Operator
mMTC	massive Machine-Type Communication
NAS	Non-Access Stratum
NAS-MM	NAS Mobility Management
NF	Network Function

NFV	Network Function Virtualization
NG-AP	NG Application Protocol
NR	New Radio
NRF	Network Function Repository Function
NSA	Non-Standalone
NUC	Next Unit of Computing
OAI	Open Air Interface
ODMMF	On-Demand Mobility Management Function
O-RAN	Open-Radio Access Network
OSI	Open Systems Interconnection
PCF	Policy Control Function
PDCP	Packet Data Convergence Protocol
PDU	Protocol Data Unit
PDR	Packet Delivery Ratio
PFCP	Packet Forwarding Control Protocol
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RB	Resource Block
REST	Representational State Transfer
RLC	Radio-Link Control
RRH	Remote Radio Head
RTT	Round Trip Time
SA	Standalone
SBA	Service-Based Architecture
SBI	Service-Based Interface
SCTP	Stream Control Transmission Protocol
SDAP	Service Data Application Protocol
SDN	Software-Defined Networking
SDR	Software Defined Radio
SIM	Subscriber Identification Module
SMF	Session Management Function
SoA	State of the Art
TCP	Transmission Control Protocol
UAV	Unmanned Aerial Vehicles
UDM	Unified Data Management
UDP	User Datagram Protocol
UE	User Equipment
UHD	USRP Hardware Driver
UPF	User Plane Function
URL	Uniform Resource Locator
URLLC	Ultra Reliable Low Latency Communication
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
VRAN	Virtual RAN
Wi-Fi	Wireless Fidelity

Chapter 1

Introduction

In recent years we observed a revolution in the mobile networks area. The increasing number of users and the rise of network operators offering extensive coverage resulted in the need for more radio resources, including base stations deployed. This motivated the research for improving the efficiency of the next generation of communications technologies for broadband cellular networks, including 5G. The first countrywide commercial 5G network was deployed in 2019 in South Korea [1] and 5G is expected to be a key enabler for a myriad of mobile communications services and applications over the next years.

A study conducted by *vXchnge* predicted that 5G networks will cover 40% of the world by 2024 and handle 25% of all mobile traffic volume, while 5G is up to 100 times faster than the 4G technology [2]. Another study, conducted in South Korea by *Opensignal*, reported that South Korea ended November of 2021 with 20 million 5G subscribers and with positive results [3].

Over the last few years, mobile robotic platforms became popular, especially due to their capability of being deployed and providing services, including communications networks. Since these platforms are easy to reprogram, dynamic, and able to move with a high degree of autonomy, they enable communications networks leveraged by recent technologies with improved performance, such as 5G, for providing wireless connectivity on-demand.

There are currently some solutions using mobile robotic platforms for providing 5G that are achieving promising results. For example, a recent solution [4], focused on video streaming applications, was developed using the concept of Open-Radio Access Network (OpenRAN). This solution takes advantage of open-source radio and virtualized functions, allowing interoperability of software and hardware from different vendors. Nevertheless, when deploying a 5G network on a mobile robotic platform important challenges need to be addressed, including meeting the Quality of Service (QoS) requirements demanded by the users and managing the communications resources available.

In a controllable 5G private network leveraged by a mobile robotic platform, the traffic-aware positioning of radio-access and core network elements, as well as the flexible deployment of new Network Functions, allowing remote network management and mobility management anywhere and in real-time, should be ensured in order to satisfy heterogeneous and dynamic user requirements. For that purpose, the placement of mobile robotic platforms aware of the QoS demanded by the network users should be an objective when designing a private on-demand 5G network.

Figure 1.1: 5G deployed in an urban environment.

1.1 Problem definition

Mobile users should be able to access the Internet anywhere with a stable connection and low latency. In order to address the challenges imposed to radio access networks in temporary events, such as emergency scenarios and crowded events, which may lead to congested networks or network failures, on-demand networks, due to the flexibility they provide, are a suitable solution. An on-demand 5G network promotes better QoS for an enterprise or entity.

This has motivated the development of 5G communications systems that combine the two concepts: a system based on a 5G private network architecture, in which the radio access network (the radio cell) can be reconfigured and repositioned on-demand. In this system, the mobile users access the Internet through the Radio Access Network (RAN) carried out by a mobile robotic platform, which then communicates with the fixed 5G Core network. This mobile RAN should be positioned and reconfigured according to the network requirements (resources, functions, traffic, and target performance indicators) by a Mobile Network Operator (MNO) that can be located remotely, using a Web browser. The design, implementation, and validation of this system are the main challenges of this dissertation.

1.2 Objectives

The goal of this dissertation is to develop a flexible 5G private on-demand network that has the radio access network deployed in a mobile robotic platform. Taking this into consideration, the following four specific objectives were defined:

- Specify and design a 5G standalone radio access and core network.
- Develop and integrate a Network Function into the 5G Core network, able to remotely position and monitor the radio access network over time.
- Develop a prototype containing the most important components of a 5G standalone network, including a gNodeB (gNB) carried by a mobile robotic platform.
- Test and validate the developed prototype.

1.3 Contributions

The main contributions of this dissertation are three-fold:

- Development of a 5G private on-demand network using a mobile robotic platform, prepared for monitoring and dynamic positioning of the radio access network in real-time.
- Extension to the 5G architecture to support a mobile Radio Access Network, including a new Network Function, and a new network interface.
- Experimental validation and evaluation of the developed 5G private network.

1.4 Document Structure

This document is structured as follows: in Chapter 2, State of the Art, we present the concepts and technologies relevant for understanding and solving the problem of this dissertation, including related work; in Chapter 3, System Specification, Design and Implementation, we introduce the proposed solution, discuss the alternatives to address the problem of this dissertation and describe its implementation; in Chapter 4, System Validation, we present and discuss the validation and performance evaluation of the proposed solution; finally, in Chapter 5, Conclusion, we present the main conclusions and future work.

Chapter 2

State of the Art

In this chapter, concepts and existing solutions relevant to address the problem of this dissertation are presented. In Section 2.1, an introduction to 5G is made, including 5G characterization, in which we present a brief description of 5G technology and a comparison with previous generations of communications technologies. In Section 2.2, the Open-Radio Access Network concept is presented, including its architecture based on Software-Defined Networking (SDN) and Network Functions Virtualization (NFV) techniques. In Section 2.3, the 5G network architecture is described, including the Network Functions that compose a 5G network and how they interact with each other. In section 2.4, the concept of 5G Network Functions is presented. In Section 2.5, the Standalone and Non-Standalone 5G architectures are characterized. In Section 2.6, the different open-source options for implementation are described. In Section 2.7, the different options for mobile robotic platforms are presented. Finally, in Section 2.8, existing solutions for 5G networks using mobile robotic platforms are described.

2.1 5G Characterization

Networks have been growing in size and requirements to a point where the current standard for mobile broadband communications, Long Term Evolution (LTE), is unable to meet the requirements of some emerging services and applications. Furthermore, since the LTE standard was launched more than 10 years ago, the technology development enabled more advanced technical solutions. To meet these requirements, the 3rd Generation Partnership Project (3GPP) standardized a new radio-access technology, 5G New Radio (NR), and a new 5G Core network (CN). A comparison between 4G and 5G network requirements is depicted in Figure 2.1.

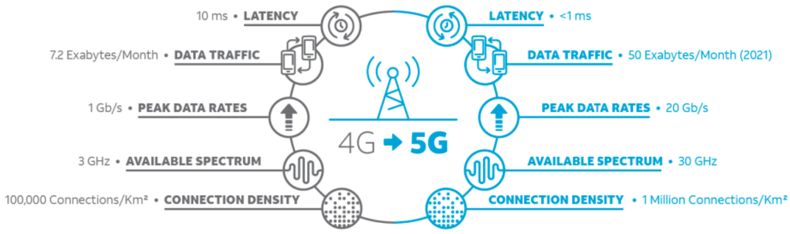


Figure 2.1: Comparison between 4G and 5G network requirements [5].

Despite being able to achieve better results in terms of latency and available bandwidth, 5G takes advantage of several features and structures of LTE. In fact, it is possible to connect the 5G New Radio (NR) access network to the legacy LTE Core network, Evolved Packet Core (EPC); this compatibility was not achieved during the transition from 3G to 4G [6]. However, there are some deployment challenges of 5G compared to 4G. For example, 5G waves have a reduced propagation range, not travelling as far as 4G waves, particularly in the 26 GHz band, leading to the need of more base stations for ensuring the same radio coverage [7].

5G networks prioritize low latency, high data rates, massive number of connected devices, and dynamic topologies that aim for always-on connectivity even when failures occur. The physical network infrastructure is suitable to be separated into different network slices, including: enhanced Mobile Broadband (eMBB), massive Machine-Type Communications (mMTC), and Ultra-Reliable and Low-Latency Communications (URLLC). eMBB corresponds to services that enable the transfer of large data volumes for further enhanced user experience. mMTC corresponds to services for a massive number of connected devices, which are characterized by low device cost and reduced device energy consumption. URLLC is envisioned for services that require ultra reliability and low latency.

Network slicing is a technique that emerged within 5G. The concept enables the multiplexing of virtualized, independent logical networks on top of the same physical network infrastructure. A network slice can support multiple mobile broadband applications with full mobility support, while another slice can be deployed to support a specific non-mobile application using shared physical resources. From the user perspective, they seem to be independent networks, despite using the same physical Core and radio networks. This is only possible due to the network functions virtualization paradigm. Representative use cases associated with three network slices are depicted in Figure 2.2.

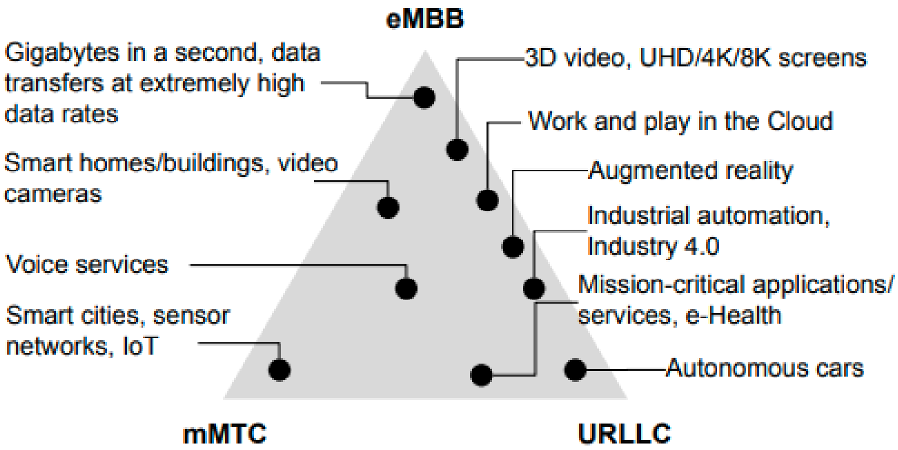


Figure 2.2: Representative 5G use cases associated with eMBB, URLLC, and mMTC network slices [8].

2.2 Radio Access Network Deployment Approaches

In the current standards, when a vendor designs a RAN, it does not have to ensure interoperability with another RAN vendor. As a result, in order to have a functional network, each Mobile Network Operator (MNO) should use the solutions from a certain vendor, creating a single vendor dependency and leading to limitations in terms of flexibility. In the traditional RAN approach, the Remote Radio Head (RRH) and the Based Band Unit (BBU) are typically provided by a single vendor and are connected using proprietary interfaces. This approach can be observed in Figure 2.3.

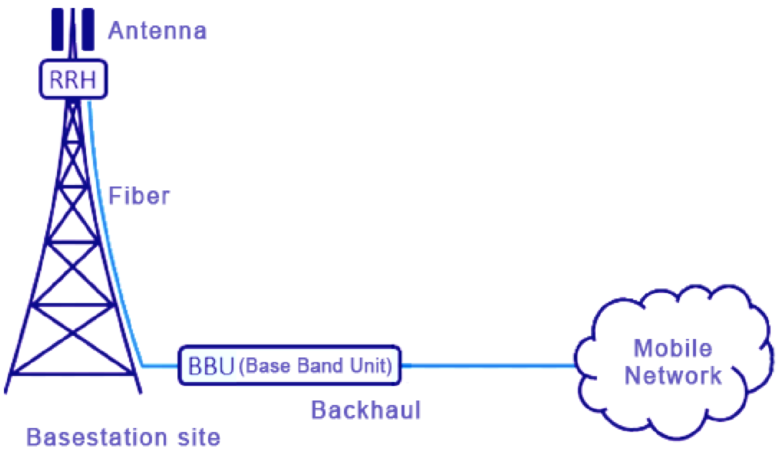


Figure 2.3: Traditional RAN approach [9].

Another approach to deploy the RAN functions is Cloud-RAN (C-RAN), also known as Centralized RAN, where different RRHs are connected to a pool of BBUs, which are aggregated in a centralized position. In this approach, the BBU does not have to be located on the site itself and simplifies radio resource management by introducing network virtualization. BBUs are virtual nodes making up a pool that can operate on a single machine, sharing resources among the connected BBUs. Despite presenting advantages, such as network virtualization, better service deployment, and the opportunity of aggregating multiple BBUs in secure data centers, the C-RAN concept is only achievable in high-density urban environments [10], and it does not address the vendor lock-in issue. The C-RAN architecture is depicted in Figure 2.4.

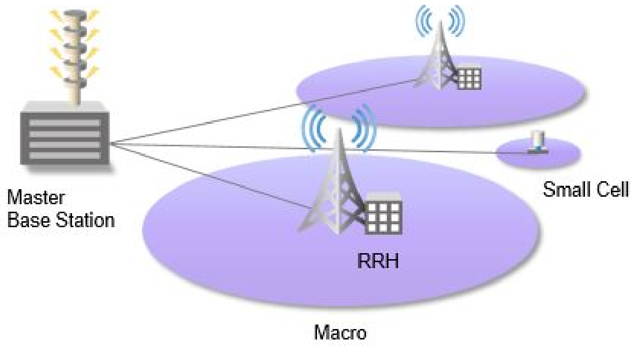


Figure 2.4: C-RAN architecture [11].

Virtual RAN (VRAN) is an alternative approach that is also possible. In VRAN, proprietary radio hardware is used, unlikely in C-RAN, the BBU hardware is no longer proprietary, being replaced with a Commercial Off-The-Shelf (COTS) server, which allows virtualizing the RAN functions. However, the interfaces between the RRH and the COTS server with virtualized functions remain closed, avoiding any vendor software working with the RRH. The VRAN approach is depicted in Figure 2.5.

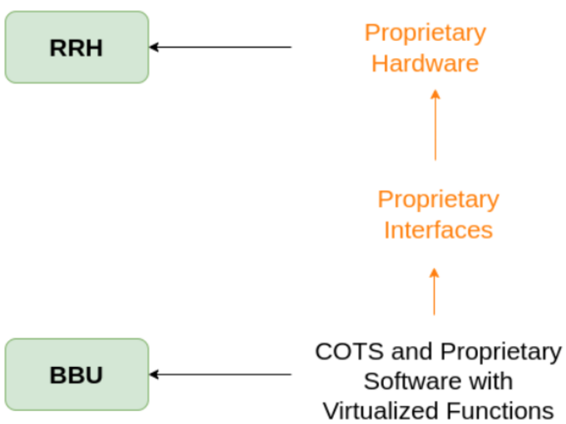


Figure 2.5: VRAN architecture.

Open-Radio Access Network (OpenRAN) [12] is a step forward for industry standards regarding the design of the radio access network in cellular networks. The concept is particularly relevant to ensure the flexible deployment of the 4G and 5G infrastructures.

In OpenRAN there is a software abstraction from the hardware platform, a concept called NFV. This is achieved by using open interfaces for the connection between the RRH and the BBU. The use of open interfaces allows any vendor's software to run on any vendor's hardware. As such, the RRH is replaceable for COTS hardware that can be purchased from any vendor, while the BBU is replaced by a COTS server and a vendor's proprietary software with virtualized functions. This approach is observed in Figure 2.6.

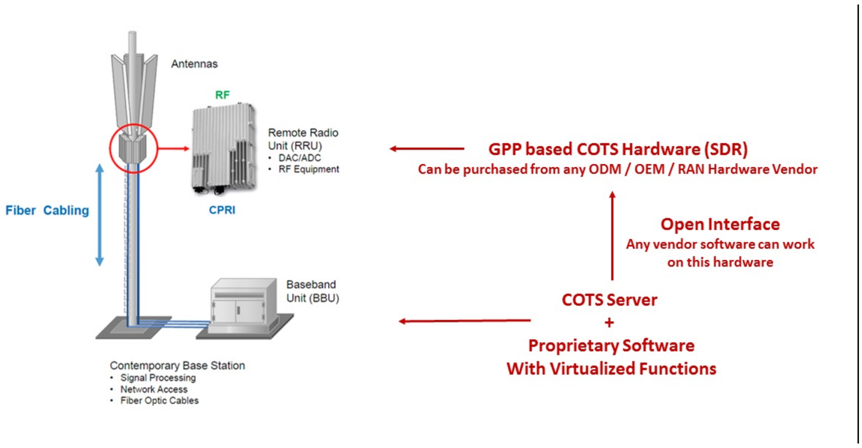


Figure 2.6: OpenRAN approach [13].

Figure 2.7 depicts the case in which an MNO deploys OpenRAN software from Vendor 2 (V2) on COTS servers. Thanks to open interfaces, the underlying hardware layer (radios from vendor A and COTS servers) can be used with software from multiple vendors (A, B, and C).

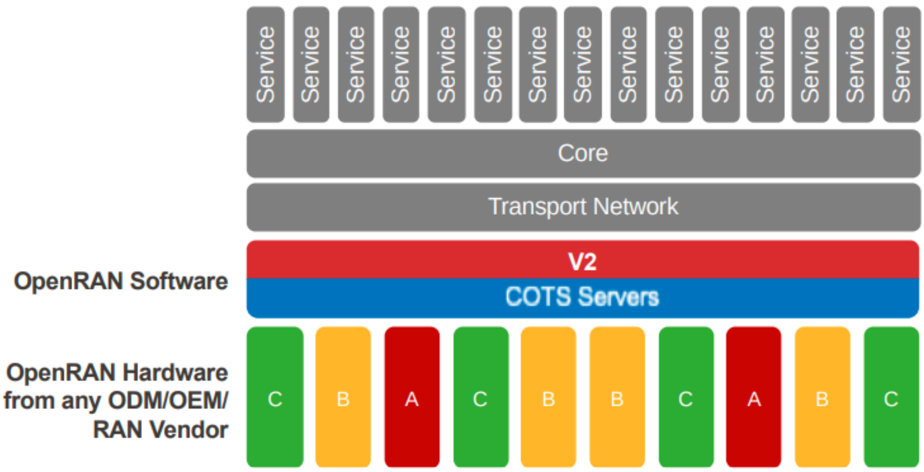


Figure 2.7: OpenRAN on top of hardware from multiple RAN vendors [13].

OpenRAN makes the access network open within all aspects and components, while leading to more market competition, promoting user choice and improving network performance. However, there are challenges associated with this paradigm. One of the main challenges is promoting its use by major players in the RAN industry; so far, only Rakuten's 4G LTE network in Japan is using OpenRAN standards [14]. The other challenge is associated with security and the possibility of vendors being susceptible to attacks, although there has been an effort by some organizations to create a security architecture and guidelines for the OpenRAN standards.

2.3 5G Network Architecture

A 5G architecture is composed of several interfaces and two functional planes: control plane and data plane. The two planes and their functions are separated in order to achieve independent scalability, evolution, and flexible deployments. This technique is named Software-Defined Networking (SDN), and consists of the logical separation of the control and data planes.

The data plane, also known as user plane, has three main components: User Equipment (UE), Radio Access Network (RAN), and UPF. The UE connects through the RAN to the UPF, which acts as a router to the Internet (cf. Figure 2.8). The data plane is the part of the 5G network through which user packets are transported. The two planes, their main functions and the separation between the access network and Core network can be observed in Figure 2.8.

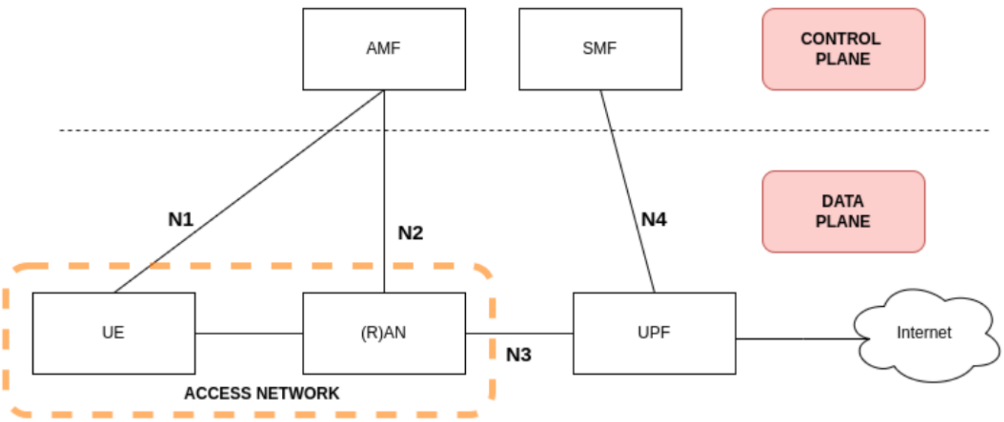


Figure 2.8: 5G network reference architecture.

The UPF acts as a gateway between the RAN and external networks. It is also in charge of packet routing and forwarding, quality of service (QoS) handling and policy enforcement functions. RAN is responsible for all radio-related network operations and is composed of a node connected to the Core network, the gNB, which ensures connectivity between the UE and the Core network.

The control plane is in charge of providing services to mobile subscribers and the most important control functions are the Access and Mobility Management Function (AMF) and the Session

Management Function (SMF). The AMF is responsible for access control and UE location management, while the SMF is in charge of managing sessions between the UE and the UPF. In order to comply with the service-oriented architecture of 5G, the control plane functions must be virtualized and run as web services in generic computing platforms.

A service-based architecture (SBA) is an area of focus for the 5G Core network when compared with the LTE Core network, EPC. This means that the architecture is focused on a highly virtualized Core network capable of integrating independent micro-services for the dynamic management of resources. It will require unprecedented levels of network automation, flexibility, and programmability in order to fulfill the requirements of new communications services and applications [15]. Figure 2.9 depicts the 5G architecture with the service-based components.

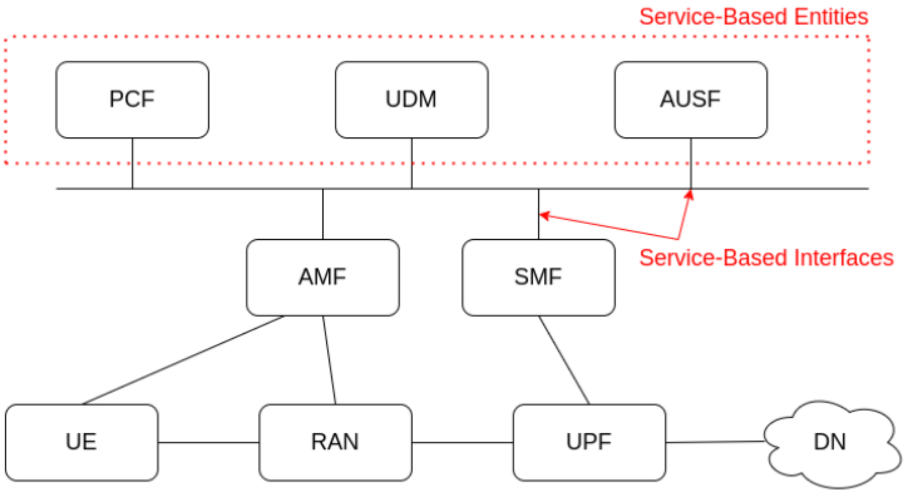


Figure 2.9: 5G system architecture with the service-based components.

In this high-level architecture, the Core network is responsible for several functions: the Policy Control Function (PCF) is in charge of policy rules; Unified Data Management (UDM) supervises authorization and authentication credentials; Authentication Server Function (AUSF) is responsible for authentication functionality [6]. With this service-based approach, the 5G Core network is able to deliver different services that operate independently from each other but share a common Service-Based Interface (SBI), enabling logical slices optimized for different services with certain characteristics.

Overall, the 5G protocol stack is similar to the 4G protocol stack [16]. The radio protocol stack consists of the following layers: 1) the Service Data Application Protocol (SDAP) is responsible for ensuring QoS radio-related requirements; 2) the Packet Data Convergence Protocol (PDCP) is in charge of Internet Protocol (IP) header compression, ciphering, and integrity protection; 3) the Radio-Link Control (RLC) is responsible for segmentation and retransmission handling; 4) Medium-Access Control (MAC) ensures multiplexing and scheduling functions; and 5) the Physical Layer handles coding and modulation functions [6]. Figure 2.10 depicts the radio protocol stack for the communications between the UE and the gNB.

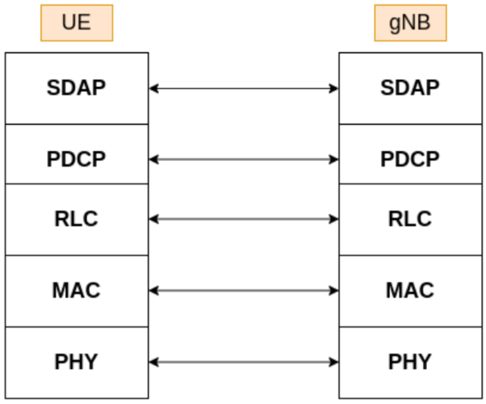


Figure 2.10: Radio protocol stack for UE and gNB.

Depending upon the direction of data, the 5G protocol stack between 5G entities is divided into two planes: control plane and data plane. Regarding the control plane stack, when information is exchanged between the Access Network (AN) and the AMF for example, a few protocols are used. While in layer 5 of the Open Systems Interconnection (OSI) model the NG Application Protocol (NG-AP) performs procedures related to UE context management and Protocol Data Unit (PDU) sessions, in layer 4 (transport layer), the Stream Control Transmission Protocol (SCTP) guarantees the delivery of messages between the AN and the AMF [6]. The protocol stack between the AN and the AMF is depicted in Figure 2.11.

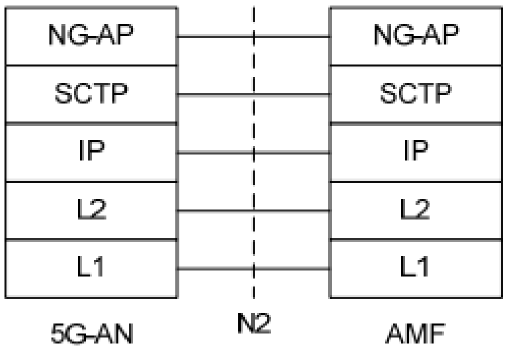


Figure 2.11: Control plane protocol stack between the AN and the AMF [17].

When information is exchanged between the UE and the AMF, various protocols are deployed. Non-Access Stratum (NAS) is responsible for managing and establishing communication between the mobile terminals and the radio Access network, while NAS Mobility Management (NAS-MM) is associated with the following functions: UE location, UE authentication, and integrity protection [17]. This communication is represented by the N1 interface (cf. Figure 2.8) and it acts as a logical interface, with the N2 interface acting as a relay agent, sending information between the two entities. The protocol stack between the UE and the AMF is depicted in Figure 2.12.

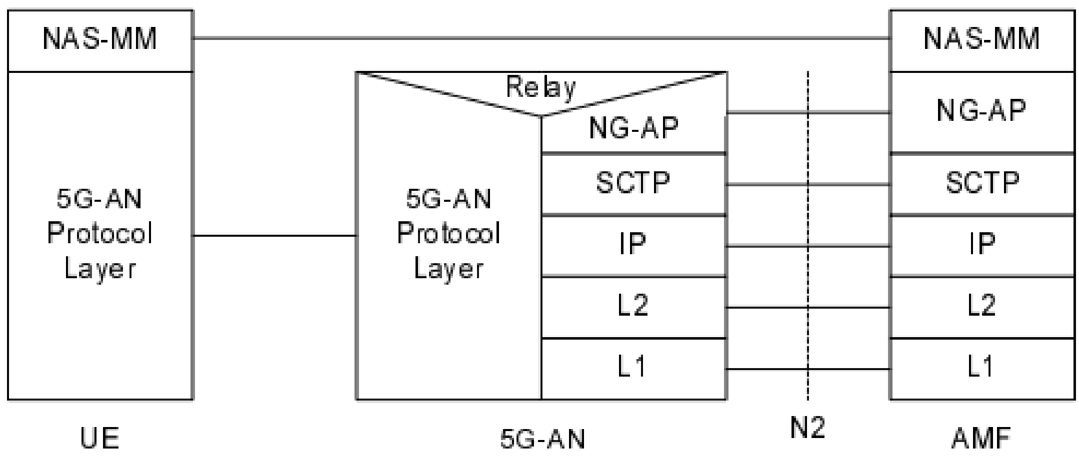


Figure 2.12: Control plane protocol stack between the UE and the AMF [17].

Regarding the data plane stack, when information related to a PDU session is exchanged, the PDU layer is responsible for defining the structure of packets carried between entities: if the session type is IPv4, IPv4 packets are considered; if it is Ethernet, Ethernet packets are used [17]. The General Packet Radio Service Tunnelling Protocol (GTP-U) allows forwarding user data (tunnelling) from the Access network to the UPF (interface N3), while also encapsulating all end-user PDUs [17]. The data plane stack is depicted in Figure 2.13.

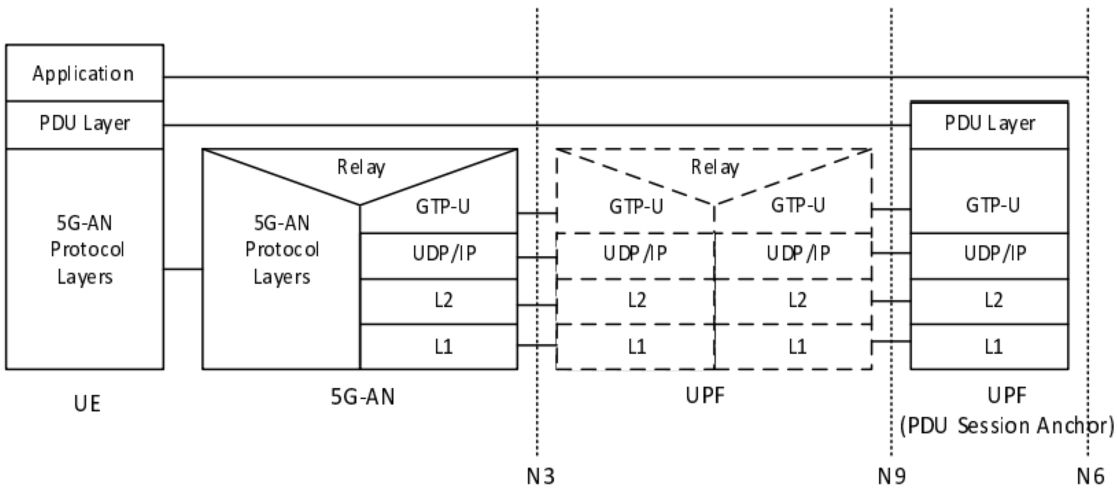


Figure 2.13: Data plane protocol stack [17].

2.4 Network Functions Concept

Within the 5G standard, Network Functions (NFs) offer different functionalities and different services. A 5G NF is independent from other NFs and should be managed independently. The service-based architecture deploys a centralized framework named Network Repository Function (NRF), which is responsible for maintaining a profile of available NF instances and their supported services. The different NFs must use service-based interfaces in order to communicate with each other or other network components.

Instead of the traditional approach, these SBIs are deployed using a REST-based architecture. Representational State Transfer (REST) is an architectural method for providing standards of communications between computer systems on the Web, making communication easier. It is based on a set of principles that seek to increase performance and scalability.

In order to deploy the REST principles in these SBIs, the HTTP protocol, more specifically the HTTP/2 version, is used, since it allows much better performance, in particular for handling RESTful operations and general Web development. As such, the 5G Core network can be seen as a set of independent Web services that communicate with each other using HTTP and applying REST principles, while operating over the standard TCP/IP protocol stack.

The HTTP protocol is also composed of a set of methods that are responsible for several functions. There are six main HTTP methods: 1) *GET*, used to read a resource; 2) *POST*, applied to create a resource; 3) *PATCH*, used to modify part of a resource; 4) *DELETE*, employed to delete a resource; 5) *PUT*, applied to update the entire resource; 6) *OPTIONS*, used to request information regarding a resource.

Since proposing an original Network Function is an objective of this dissertation, it should be compliant with the HTTP status codes defined by the 5G standard. The HTTP status codes are issued by a server in response to a client's request made to the server. The first digit of an HTTP status code specifies one of five standard classes of responses: 1) *1XX* - informational response, the request was received, continuing process; 2) *2XX* - successful, the request was successfully received, and accepted; 3) *3XX* - redirection, further action needs to be taken in order to complete the request; 4) *4XX* - client error, the request contains bad syntax or cannot be fulfilled; and finally, 5) *5XX* - server error, the server failed to fulfil an apparently valid request [18].

In Figure 2.14, the list of HTTP status codes per HTTP method, which shall be supported on SBIs, can be observed. In this list, **"M"** means mandatory, implying that all 3GPP Network Functions must support the processing of the specific HTTP status code for the specific HTTP method; **"SS"** indicates service specific, which means that processing the HTTP status code depends on the definition of the specific Network Function (NF); finally, **"N/A"** stands for not applicable, which means that the HTTP status code shall not be used for the specific HTTP method within the 3GPP NFs [19].

HTTP status code	HTTP method					
	DELETE	GET	PATCH	POST	PUT	OPTIONS
100 Continue	N/A	N/A	N/A	N/A	N/A	N/A
200 OK (NOTE 1, NOTE 2)	SS	M	SS	SS	SS	M
201 Created	N/A	N/A	N/A	SS	SS	N/A
202 Accepted	SS	N/A	SS	SS	SS	N/A
204 No Content (NOTE 2)	M	N/A	SS	SS	SS	SS
300 Multiple Choices	N/A	N/A	N/A	N/A	N/A	N/A
303 See Other	SS	SS	N/A	SS	SS	N/A
307 Temporary Redirect	SS	SS	SS	SS	SS	SS
308 Permanent Redirect	SS	SS	SS	SS	SS	SS
400 Bad Request	M	M	M	M	M	M
401 Unauthorized	M	M	M	M	M	M
403 Forbidden	M	M	M	M	M	M
404 Not Found	M	M	M	M	M	M
405 Method Not Allowed	SS	SS	SS	SS	SS	SS
406 Not Acceptable	N/A	M	N/A	N/A	N/A	SS
408 Request Timeout	SS	SS	SS	SS	SS	SS
409 Conflict	N/A	SS	SS	SS	SS	N/A
410 Gone	SS	SS	SS	SS	SS	SS
411 Length Required	N/A	N/A	M	M	M	SS
412 Precondition Failed	SS	SS	SS	SS	SS	N/A
413 Payload Too Large	N/A	N/A	M	M	M	SS
414 URI Too Long	N/A	SS (NOTE 3)	N/A	N/A	SS	N/A
415 Unsupported Media Type	N/A	N/A	M	M	M	SS
429 Too Many Requests	M	M	M	M	M	M
500 Internal Server Error	M	M	M	M	M	M
501 Not Implemented	SS	SS	SS	SS	SS	SS
502 Bad Gateway	M	M	M	M	M	M
503 Service Unavailable	M	M	M	M	M	M
504 Gateway Timeout	SS	SS	SS	SS	SS	SS

Figure 2.14: HTTP status code supported on SBIs [19].

For illustrative purposes, let us consider an NF with a role similar to the one played by the AMF, being responsible for managing and authenticating client subscriptions. In case of incorrect client details, which leads to unsuccessful subscription, or network congestion, which results in the request not being processed by the NF, HTTP status codes are issued. In Figure 2.15, the resulting HTTP status codes and descriptions for these use cases can be observed.

Protocol or application Error	HTTP status code	Description
SUBSCRIPTION_NOT_FOUND	404 Not Found	The request for modification or deletion of subscription is rejected because the subscription is not found in the NF.
NF_CONGESTION	503 Service Unavailable	The NF instance experiences congestion and performs overload control, which does not allow the request to be processed. (NOTE 4) (NOTE 7)

Figure 2.15: Protocol and application errors associated with subscription and network congestion on 5G SBIs [19].

2.5 5G Standalone & Non-Standalone

The 5G Standalone (SA) architecture refers to using network components and functions based on 5G only, meaning that there is no dependency regarding the LTE radio technology, 4G Core network, or any other previous generation of communications technologies. The replacement of the

4G Core network by the virtualized 5G service-based Core network constitutes an important step forward in the development of the SA architecture. This approach takes advantage of the concept of cloud computing, where an application-based software infrastructure is able to offer different services through the Internet, in order to deploy the concept of a service-based Core network, in which the 5G ecosystem is divided into smaller, independent components. This complements the previously referred concept of separating the 5G network into network slices (Section 2.1), simplifying the process of updating each individual independent micro-service, while improving the overall scalability of the network. A simple high-level diagram with the components of the SA architecture can be observed in Figure 2.16.

Figure 2.16: 5G SA high-level architecture.

In [20], both emulation-driven and practical experiments of a 5G SA network were carried out. Regarding the practical experiment, the following equipment was used: a laptop with Ubuntu 20.04, an *Alpha Network Inc* gNB and an *APAL MiFi Device* as the UE.

5G Non-Standalone (NSA) refers to the architecture in which the 5G NR-based RAN is connected to the 4G Evolved Packet Core (EPC) network. This architecture enables operators to provide higher throughput values in the RAN and gradually migrate to 5G without requiring a complete end-to-end system. Moreover, this allows capitalizing the existing 4G infrastructure. Although the most usual implementation is the combination of 5G NR with EPC, 4G LTE RAN can also be used with the 5G Core network, especially in open-source scenarios [21]. However, the NSA architecture brings up several disadvantages: 5G NR can only be used where LTE infrastructure is already deployed and the Core network functionalities are limited by EPC. As such, functions and techniques introduced by the 5G Core network, including network slicing, network automation and independent micro-services, can not be implemented.

In [8], two network scenarios were considered: an SA network based on software only, and an NSA network implemented in software and hardware, both using a 5G Core-based open-source SBA model. In addition, the latter scenario was implemented using 4G RAN in hardware (Software-Defined Radio) and the 5G Core network running on *Docker* containers. The main goal of this demonstration was to present in a practical deployment the main functionalities of the 5G Core when implemented in an NSA environment. The architecture employed for this experiment is depicted in Figure 2.17.

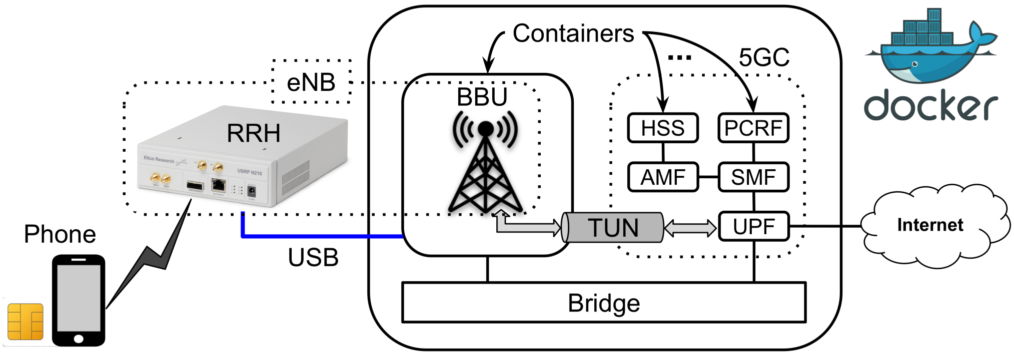


Figure 2.17: 5G NSA experiment [8].

2.6 5G Open-Source Implementations

2.6.1 Open-Source 5G Core Software Packages

In order to implement or emulate a 5G SA or NSA network, several closed and open source Core projects can be considered.

Open Air Interface (OAI) [22] is an open-source project, currently developed by OpenAirInterface Software Alliance (OSA), capable of implementing a fully functional 5G network. Although fairly recent when it comes to the access network, it provides all the necessary components for the Core network development.

Open5GCore [23] is a practical implementation of the 3GPP 5G Core network developed by the Fraunhofer Institute, capable of implementing a standalone 5G network. Although it enables fast and targeted 5G innovation and a reliable testbed, the software is licensed.

Free5GC [20] is an open-source project for 5G Core networks. The goal of this project is to implement the 5G Core network. Out of all those experimented during the development of this dissertation, it was revealed to be the most user-friendly and stable. However, it does not have a radio access network, unlike OAI, relying on UERANSIM – a UE/RAN simulator for testing purposes. Using this simulator, we can use a tool called WebConsole, accessed through the localhost address on the browser, which allows the user to add one or more UEs.

Open5GS [24] is a C-language open source implementation of 5G Core and the 4G Core networks. Similar to Free5GC, it uses the UERANSIM simulator for the RAN. It also has a WebUI application, similar to WebConsole, implemented in *Node.JS* and React.

Magma [25] is an open-source platform developed by *Facebook* and supported by many other entities, such as the Linux Foundation and OAI. The main objective is to ensure open and cloud-native 5G functions, in order to guarantee interoperability between software, hardware and RAN vendors across the ecosystem.

These solutions are often used in 5G demonstrations in order to evaluate the main functionalities of 5G and the characteristics of its components. The comparison between the five 5G software packages for the Core network is presented in Table 2.1.

	OAI	Open5GCore	Free5GC	Open5GS	Magma
Programming Language	C, C++	Various	Go	C	Various
Open-Sourced	Yes	No	Yes	Yes	Yes
Interface	CLI	CLI, WebUI	CLI, WebUI	CLI, WebUI	CLI
Platform Support	Linux	Linux	Linux	Linux, MacOS	Linux, MacOS
Documentation	Very Good	Very Good	Good	Good	Good
Implementation(s)	4G, 5G	5G	5G	4G, 5G	4G, 5G

Table 2.1: 5G Core software packages comparison.

2.6.2 Open-Source 5G RAN software packages

In order to implement or emulate a 5G SA or NSA network, several closed and open source RAN software packages can also be considered.

OAI implements a 4G LTE and 5G Radio Access Network, including UE and gNB. It provides one of the best radio access networks by means of open-source RAN software, but most of its applications and experiments presented in the literature are based on 4G networks yet. However, due to the many restrictions associated with Software Defined Radio (SDR) and real-word evaluations, OAI also offers a 4G and 5G RAN emulator.

UERANSIM [26] is a C++ open source project that emulates a UE and a gNB. Due to its simplicity and easiness to use, it is often used for testing software packages that only provide the 5G Core network, since it allows to emulate the RAN transparently to the Core network.

Free5GRAN [27] is a C/C++ open-source 5G RAN stack. Although under active developments, it presents an approach focused on a seamless integration with SDR. It has already been tested with three different SDR models: USRP X310, USRP N210 and USRP B210. Due to being in an earlier stage of development, it is not as popular as the others.

srsRAN [28] has been focused on developments considering 4G. The full-stack 5G SA for gNB is expected to be launched during 2022. This project is also known as srsLTE due to its free and open-source SDR component library for 4G LTE; srsRAN is considered the evolution towards 5G NR.

The comparison between the five projects is presented in Table 2.2.

	OAI-RAN	UERANSIM	Free5GRAN	srsRAN	OAI-Emulator
Programming Language	C, C++	Various	Go	C	C, C++
Open-Source	Yes	Yes	Yes	Yes	Yes
SDR Compatibility	Yes	No	Yes	Yes	No
Platform Support	Linux	Linux	Linux	Linux	Linux
Documentation	Very Good	Good	Good	Good	Very Good
Implementation(s)	4G, 5G	5G	5G	4G, 5G	4G, 5G

Table 2.2: 5G RAN software packages comparison.

2.6.3 Hardware

In order to meet OpenRAN objective of substantially reducing vendor lock-in, it is necessary to develop frameworks based on SDRs that make it possible to use the same hardware platform for different software implementations, reduce development time, and achieve high levels of performance.

The open-source software described in Sections 2.6.1 and 2.6.2 can be mostly executed on commodity hardware but, in order to implement functionalities related to the physical layer, these are usually run on the Field Programmable Gate Arrays (FPGAs) of the SDRs. These platforms allow researchers to deploy and experiment end-to-end networks, even though they may not have access to the hardware equipment deployed by most service providers. In Table 2.3 is depicted a summary of the capabilities of each SDR and RAN software packages supported.

SDR	Frequency Range	Bandwidth (MHz)	RAN Software
bladeRF	[300 MHz; 3.8 GHz]	28	OAI, srsLTE
bladeRF 2.0 micro	[47 MHz; 6 GHz]	56	OAI, srsLTE
LimeSDR	[100 kHz; 3.8 GHz]	61.44	OAI, srsLTE
USRP B205mini-i	[70 MHz; 6 GHz]	56	srsLTE
USRP B210	[70 MHz; 6 GHz]	56	OAI, srsLTE
USRP X310	[10 MHz; 6 GHz]	100	OAI

Table 2.3: Capabilities of SDRs and their integration with RAN software.

The different SDRs are used according to the type of deployment, with Universal Software Radio Peripheral (USRP) solutions such as USRP B210 and USRP N310 being typically used as rooftop base stations and small towers, respectively. Other SDRs such as LimeSDR and bladeRF are commonly used to operate as small cells, given the fact that they are less powerful when compared to USRP SDR solutions.

2.7 Mobile Robotic Platforms

In order to create a 5G private on-demand network, a mobile robotic platform able to carry a gNB and capable of providing mobility in a safe and reliable manner is worthy of being considered. Since the remote control and positioning of the mobile robotic platform is an objective of this dissertation, the capability of being controlled through a Web browser is a desired feature.

Spot, developed by *Boston Dynamics* [29], is an example of a mobile robotic platform that is able to carry a considerable payload weight, and offers the possibility of being controlled remotely.

Unitree's [30] solutions are also suitable mobile robotic platforms, including the models Go1 Edu Robot, *Aliengo*, and *A1*.

The *Phantom 2* [31], developed by *Dji*, is a representative Unmanned Aerial Vehicle (UAV) that can also be considered for providing mobility to a gNB. Although the *Phantom 2's* payload is carried in a less stable way, it is considerable cheaper than the other mobile robotic platforms.

A comparison between off-the-shelf mobile robotic platforms is presented in Table 2.4.

Mobile Robotic Platform	**Developer**	**Price (€)**	**Remote Web Control**
Go1 Edu	Unitree	2.600	Yes
Spot	Boston Dynamic	71.000	Yes
Aliengo	Unitree	33.000	Yes
A1	Unitree	14.000	Yes
Phantom 2	Dji	559	No

Table 2.4: Mobile robotic platforms comparison.

2.8 Related Work

In the literature, there are some works addressing the combination of 5G networks, mobile robotic platforms and network functions.

With the increasing need of providing broadband Internet access in temporary crowded events, remote locations, and disaster scenarios, UAV-based networks are emerging. These networks have characteristics that make them suitable for these scenarios as they provide flexibility to be deployed anywhere, anytime. Moreover, their mobility and ability to carry on-board payload, make UAVs valuable platforms to carry communications nodes, including gNBs and Wi-Fi Access Points (APs).

The dynamic placement, routing, and resource management in UAV-based networks, which may be implemented by means of centralized Network Functions, are research challenges considered in the literature. In [32], a centralized routing solution for a Flying Multi-Hop Network (FMN) was presented, called RedeFINE. RedeFINE takes advantage of the possibility of using a central node that has holistic information about the state of the network, including the future positions defined for the UAVs, to improve network performance. This centralized routing solution mitigates disruptions and interference that may occur, due to the high mobility of UAVs and frequent changes in the quality of the wireless links in these networks. For that purpose, it defines in advance the forwarding tables and the instants they shall be updated in the UAVs. RedeFINE aims at complementing a previous algorithm, which defines the positions for Flying Mesh Access Points (FMAPs) so that they meet the traffic demand of the ground users [33]. In Figure 2.18, an FMN used for providing wireless connectivity in a crowded event (music festival) is depicted.

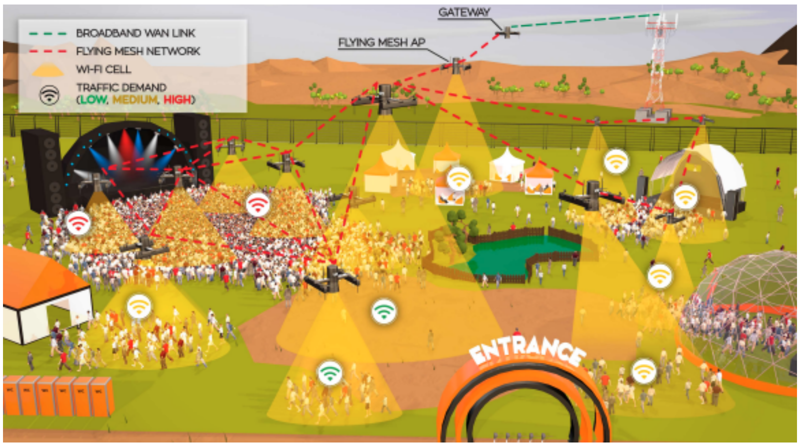


Figure 2.18: The FMN deployed in a music festival for providing wireless connectivity to the terminals on the ground [32].

The RedeFINE algorithm showed improvements in the performance of FMNs, especially regarding throughput and Packet Delivery Ratio (PDR), while slightly increasing the end-to-end delay, due to the high Packet Delivery Ratio, which increases network congestion. Since it is a centralized routing solution capable of enabling high-capacity and uninterruptible communications by considering holistic information to predict the future state of the network, RedeFINE can be considered an interesting solution to be used as a Network Function into a 5G private on-demand network.

In [34], a traffic-aware Gateway UAV Placement (GWP) algorithm for UAV-based networks is proposed. A multi-tier network architecture composed of two types of UAVs is considered: Flying Access Points (FAPs), responsible for providing Internet access to ground users, and a gateway UAV connecting the network to the Internet. The GWP algorithm runs in a Central Station (CS), deployed in the Cloud or at the Edge of the network. The CS is in charge of three important functions: defining the updated positions of the FAPs so that the traffic demand of the ground users is met, calculating the updated forwarding tables by running RedeFINE or another state of the art routing algorithm, and determining the updated position of the gateway UAV, in order to enable wireless links with high enough capacity to carry the traffic between the FMAPs and the gateway UAV. An illustrative design of the network architecture can be observed in Figure 2.19.

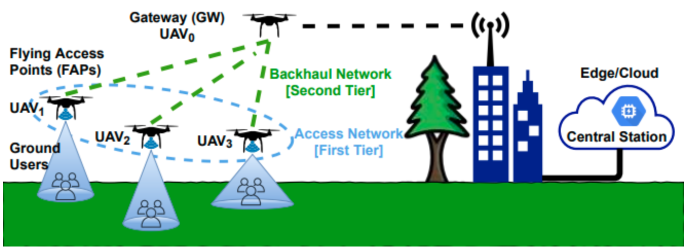


Figure 2.19: Multi-tier UAV-based network architecture [34].

The performance evaluation of GWP was carried out using the *ns-3* simulator, considering two networking scenarios: one with four FAPs equidistant from each other (cf. Figure 2.20), and a second scenario, with ten FMAPs randomly positioned in order to form two zones with different traffic demands (cf. Figure 2.21). In both scenarios, the baseline corresponds to the gateway UAV placed in the FMAPs center.

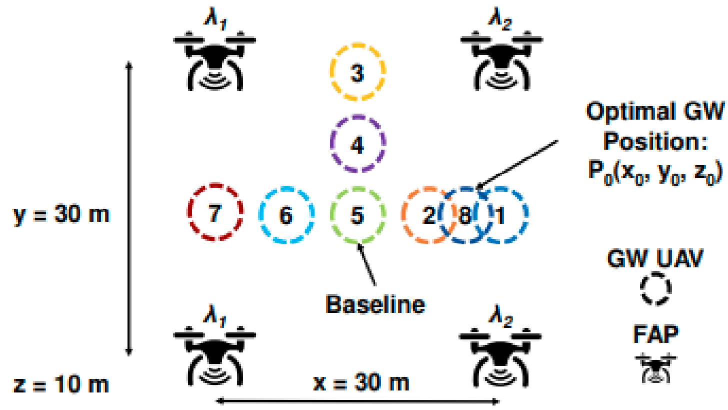


Figure 2.20: First GWP evaluation scenario, in which Position 8 corresponds to the gateway UAV position defined by the GWP algorithm [34].

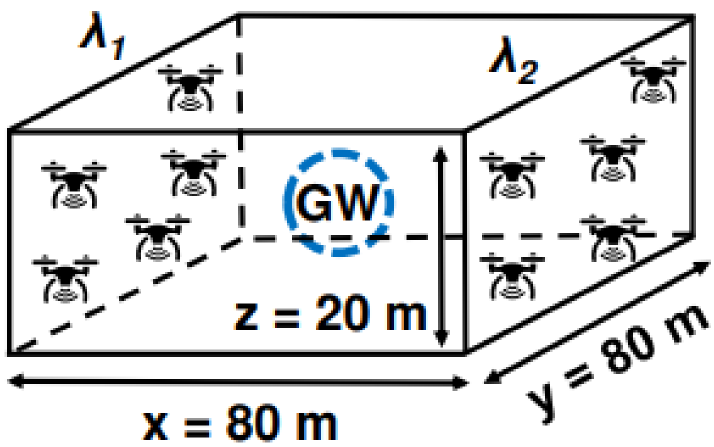


Figure 2.21: Second GWP evaluation scenario, in which the dashed circle represents the gateway UAV placed in the FAPs center and λ represents the traffic-demand associated with each zone [34].

The performance evaluation of GWP demonstrated improvements in terms of throughput and delay values. Considering that GWP is able to handle highly dynamic network topologies, a desirable characteristic for 5G networks, this makes GWP a suitable solution to be used in 5G private on-demand networks.

Lorenzo Bertizzolo [35] created a 5G network, based on the OpenRAN architecture, focused on video streaming applications using a UAV. The experiment topology consisted of a low-complexity, closed-loop control solution to assist UAV-based video streaming applications on 5G networks. The network topology consisted of a multi-cell RAN testbed where three base stations were deployed to serve three UAV-based UEs. Three scenarios were considered: one where the UAV acts as a traditional ground user; another where the UAV-based UE is an aerial user while using an omnidirectional antenna; and a third similar to the second one but with a wide-angle directional antenna. This study allowed us to understand the concept of OpenRAN and how it can be used to maximize network performance and meet the QoS requirements.

Muhammad Khan [36] developed a solution considering a Beyond 5G (B5G) and 6G network perspective regarding Edge computing on the Internet of Things (IoT), where due to the record-breaking increase in traffic volumes, the use of Multi-access Edge Computing (MEC), a cloud-based concept that enables ultra-low latency, is considered as a promising solution to provide cloud-computing capabilities within the RAN. Unlike previous solutions, it takes into account a theoretical approach for defining the Edge computing concept and what it represents for 6G networks. This study envisions that beyond 5G and 6G networks will lie on Artificial Intelligence (AI) and Machine Learning techniques and claims that Edge computing can be used to meet strict QoS requirements.

In [37] the unique characteristics of UAVs and how they should coexist in the aerial space in a safe and reliable manner are explored. Metrics such as time efficiency and power consumption were studied, in order to illustrate the potential advantages of UAV-based networks over conventional networks. The authors argue that, when using Wi-Fi to provide wireless connectivity by means of UAVs, the communications requirements can be too demanding for the capabilities of that technology. The study showed the potential benefits of using 5G in a private on-demand network.

To the best of our knowledge, the solutions presented in the State of the Art (SoA) do not address the objectives of this dissertation, bringing up an opportunity to propose novel potential contributions for 5G private on-demand networks.

Chapter 3

System Specification, Design and Implementation

This chapter presents the specification, design and implementation of the proposed solution, including an original Network Function to control a mobile robotic platform carrying a gNB. In Section 3.1, the specification of the on-demand 5G network developed is presented. In Section 3.2, the design of the proposed solution for solving the problem of this dissertation is presented. Finally, in Section 3.3, the proposed implementation is presented.

3.1 System Specification

The main objective of this dissertation is to design a private on-demand 5G network able to provide wireless connectivity to a UE that demands access to the Internet. The private on-demand 5G network should employ an end-to-end 5G standalone architecture. The proposed solution should be able to be deployed by an MNO, which should be capable of remotely reconfiguring the network dynamically, while providing wireless connectivity to a UE in a given area without coverage.

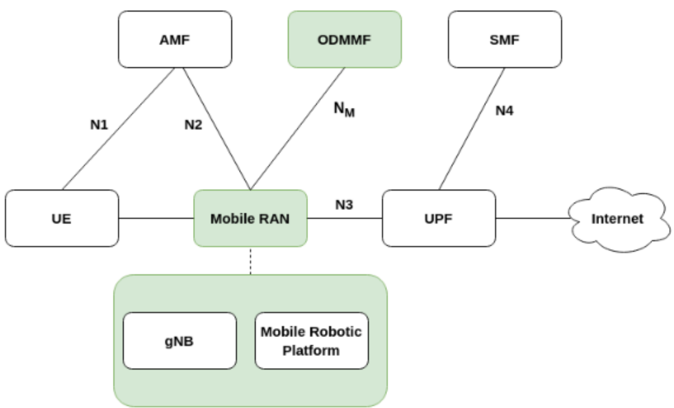


Figure 3.1: 5G system architecture with the ODMMF Network Function.

In our proposed architecture, the User Equipment demands 5G wireless connectivity, characterized by its low latencies and high bitrates. For that purpose, the UE establishes a connection with the mobile RAN, composed of the gNB carried by the mobile robotic platform, using radio antennas that allow for the propagation of 5G radio waves. The N1 interface in Figure 3.1 is a reference point, since the UE does not communicate physically with the AMF; instead, the gNB is responsible for forwarding all UE details to the AMF (via Wi-Fi or Ethernet) through the N2 interface. The N2 interface was described in Section 2 (cf. Figure 2.11).

The mobile RAN should also communicate with the UPF, through the N3 interface, when the UE wants to access the Internet, in which the UPF acts as a 5G gateway. The N3 interface was also characterized in Section 2 (cf. Figure 2.13). This mobile RAN consists of two elements: the gNB and the mobile robotic platform carrying the gNB.

The UPF must also communicate with the SMF through the N4 interface and is responsible for a number of key session management procedures. Considering the layer 5 of the TCP/IP model, the Packet Forwarding Control Protocol (PFCP) is in charge of handling signaling procedures and packet forwarding. In layer 4, the N4 interface takes advantage of the low latency and simplicity enabled by UDP. The protocol stack that characterizes the N4 interface is depicted in Figure 3.2

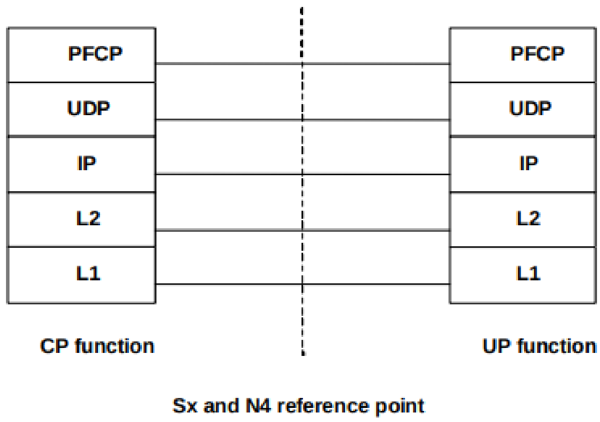


Figure 3.2: N4 interface protocol stack.

Since we propose an original Network Function, implemented in the Core Network, we needed to name it just like it happens with all other 5G's NFs. **On-Demand Mobility Management Function** (**ODMMF**) was the name chosen for our Network Function, since the aim is that an MNO is able to control the mobile robotic platform movements and receive information regarding the quality of the wireless connection established between a UE and the gNB.

The ODMMF NF is connected to the mobile RAN through the new proposed N_M interface, which characterizes the network interface used to connect a NF with a mobile RAN. The easy deployment and scalable creation of new NFs are a key point of the 5G Core. However, this should be done taking into account the technical specification of the service-based architecture defined on the 5G standard [19]. The Network Function proposed by this dissertation should be as compliant as possible with the 5G standard.

For illustrative purposes, considering the case where an MNO controls the mobile robotic platform remotely using a Web browser to restore the connection lost by a UE, it is vital that the connection with the mobile robotic platform has the least latency possible; otherwise, there is a considerable delay between the commands given to move and the actual movement, which may lead to poor QoE and possible damages to the mobile robotic platform. With this, the ODMMF Network Function should be developed using technologies and practices that enables the MNO to receive information as soon as it is acquired in real-time.

ODMMF is designed for being deployed at a Core level. Due to security concerns and the possibility of worldwide remote access, it should be made available online. The objective is that the ODMMF makes available two services: one for displaying information regarding the quality of the radio link between the gNB and the UEs, and the other for the management of the mobility of the robotic platform. From an MNO perspective, these services can be accessed on-demand through their respective Uniform Resource Locators (URLs). The ODMMF integrated into the 5G system architecture is depicted in Figure 3.1.

Since the proposed NF is part of the Core network, a different option, where the ODMMF is connected to the AMF, and the AMF forwards the information to the mobile RAN, is also a possibility. However, since most of the commercial mobile robotic platform available already have Web services that enable their control, we proposed a new dedicated interface in the proposed architecture. Moreover, since these mobile robotic platforms and NFs are usually very time-sensitive, the extra time for the AMF to forward the information is a disadvantage when considering such alternative option.

The 5G protocol stack proposed for the N_M consists of the following layers: 1) HTTP/2, an evolution of the first version of HTTP (HTTP1.1), which is responsible for accessing the data by means of a Web service, and enables full request and response multiplexing; 2) Transport Layer Security (TLS), which is used for security purposes at the transport layer by offering encryption and privacy. These two protocols operate over the standard TCP/IP protocol stack. The service-based protocol stack is depicted in Figure 3.3.

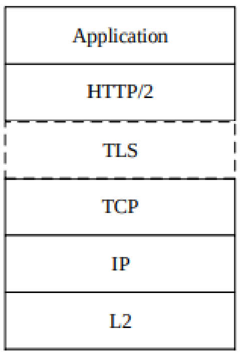


Figure 3.3: Proposed N_M protocol stack, based on the 5G SBI stack [19].

An important aspect of the mobile robotic platform is the vision capability it must have. For safely access and control, it must have multiple video cameras that allow an MNO to understand the environment surrounding the mobile robotic platform. These cameras should also be controlled using HTTP commands. Finally, it should possess the ability to be connected to a computer implementing the 5G Core Network.

3.2 System Design

The testbed developed to implement and evaluate the proposed solution is depicted in Figure 3.4. It is composed of three main components: Core network, Mobile RAN and User Equipment.

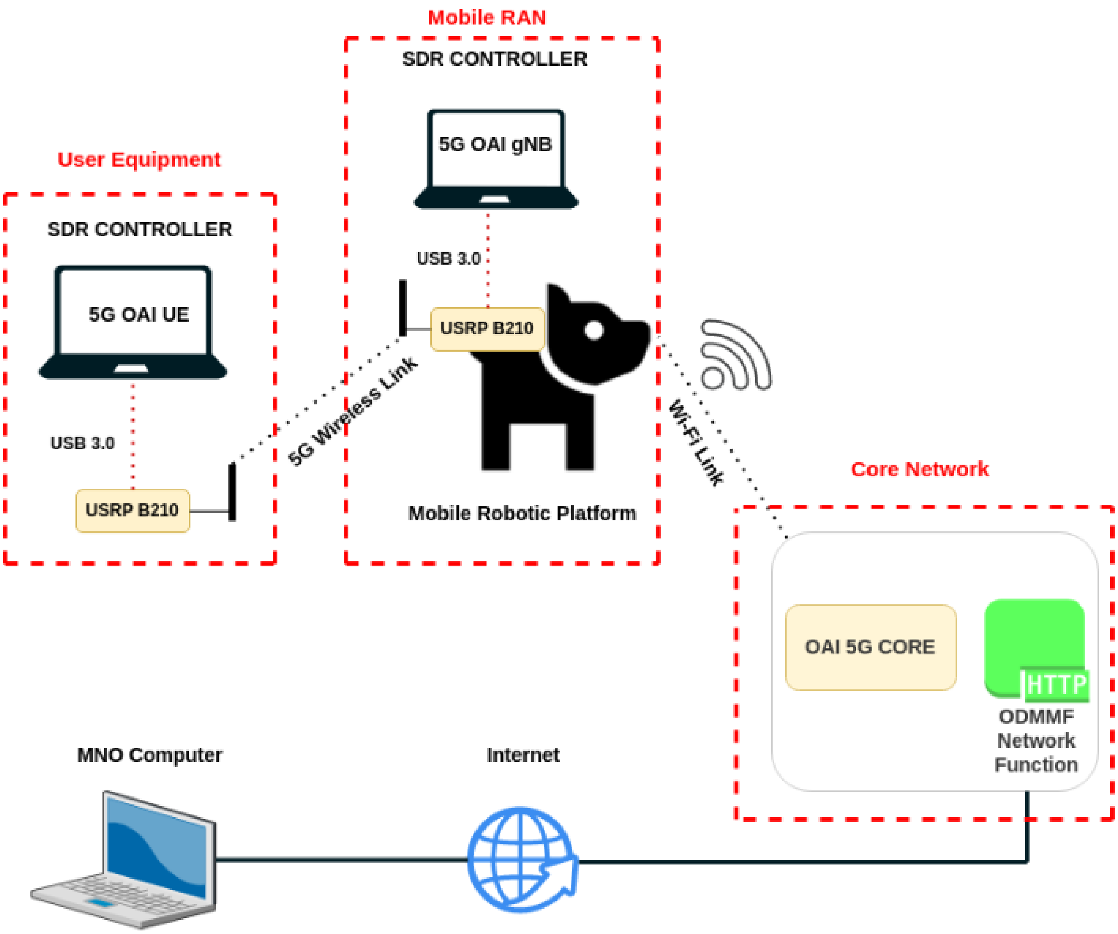


Figure 3.4: Testbed architecture used to test and evaluate the proposed solution.

After some research and experimentation of existing software packages to deploy a 5G network, two more suitable options were identified: Open Air Interface and Free5GC. Since Free5GC only provides the Core network and does not offer any RAN-related implementation [38], the software chosen to implement the proposed 5G network was Open Air Interface, since it provides a

5G SA network implementation, including RAN and Core network [39]. It is one of the most used 5G software packages by the community for cellular networks leveraged on SDRs.

3.2.1 Core Network

The Core network was implemented using *Docker* containers, which use fewer resources when compared to VMs, allowing the use of less powerful computers. In our case, an *Intel NUC5i5MYBE* was chosen as the hardware where the 5G NR Core was deployed. In Figure 3.5, a picture of this equipment can be observed.

Figure 3.5: *Intel NUC Board NUC5i5MYBE*.

The connection between the mobile RAN and the Core network components is ensured by a Wi-Fi control link. This technology was selected by a matter of testbed convenience, but other communications technologies can be used as well since this interconnection can be made over any standard IP network, as shown in Fig. 2.13 (cf. interface N3). The communications between the UE and the mobile RAN are ensured using 5G technology, where OAI has mechanisms that assure that. Within a feasible distance, the two SDRs are synchronized and prepared for a stable exchange of information. The Core Network is composed two main elements: the OAI 5G Core and the ODMMF Network Function.

In this dissertation, the protocol used to ensure communications between the multiple components of the 5G network was NG-AP, responsible for creating a communications channel between the gNB and the Core network. It is also in charge of establishing the initial UE context. NG-AP relies on a reliable transport mechanism and is designed to run on top of SCTP (cf. Figure 2.11).

3.2.2 Mobile RAN and User Equipment

The mobile RAN is composed of the mobile robotic platform and the gNB. The gNB is responsible for all radio transmissions in the 5G network and ensures connectivity with the UPF and AMF, which are part of the Core network. The ability for the gNB to be carried by the mobile robotic platform gives the RAN an enhancement in terms of flexibility for reconfiguring the network on-demand.

To implement the UE and gNB, an *HP EliteDesk 800 G2* desktop and an *ASUS Vivobook S15* laptop were used, respectively. Since the RAN is implemented on separate hosts, two SDRs were used to allow communications between the UE and gNB. From the alternatives presented in Section 2.6.3, the current SDRs best suited for providing radio communications in 5G NR implementations are the USRP B210 and USRP X310 [40]. The chosen SDR was the USRP B210 [41], due to its cost-effectiveness and popularity across the community. The SDR, which can be observed in Figure 3.6, is connected to a computer through a Universal Serial Bus (USB) 3.0 interface, in order to ensure a faster and more reliable connection.

Figure 3.6: USRP B210 and the respective USB 3.0 connection.

Since the carrier frequency selected was 3.6 GHz, the dipole antennas chosen to attach to the USRP B210 were tailored to this frequency band; their model was *TG.30.8113* [42]. The use of these antennas enabled an optimized wireless link between the two SDRs. In Figure 3.7, the two SDRs with the used antennas can be observed.

Figure 3.7: USRP B210 SDRs and the respective antennas.

3.2.3 Mobile Robotic Platform

Due to being a capable mobile robotic platform with multiple video cameras that provide vision ability, the Go1 Edu Robot [30], developed by *Unitree*, was chosen to carry the gNB in our testbed. In Figure 3.8 the Go1 Edu Robot can be observed.

Figure 3.8: Go1 Edu Robot.

One of the main advantages provided by the Go1 Edu Robot is the possibility of being controlled using a Web browser. In Figure 3.9, the Web control page can be observed.

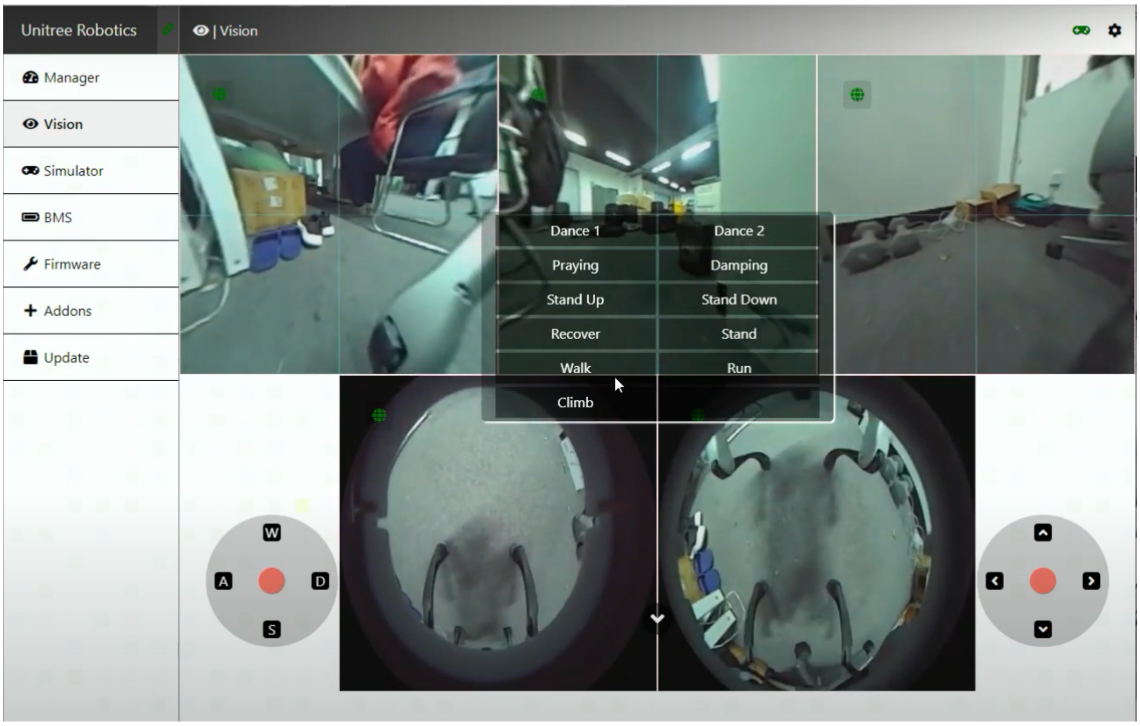


Figure 3.9: Web interface provided by the Go1 Edu Robot.

The Go1 Edu Robot plays a key role in the development of the on-demand private 5G network, being responsible for giving mobility to the gNB and establishing communication with ODMMF.

This mobile robotic platform has two ways to be controlled: 1) using a radio remote controller, which implies that the person controlling the platform is located within a relatively short distance of the robotic platform, in line-of-sight; and 2) using the Web-based application of *Unitree*, while using a computer or smartphone connected to the Wi-Fi network created by the Access Point of the mobile robotic platform.

In our testbed, the Next Unit of Computing (NUC) running the Core Network connects to the Wi-Fi Access Point of Go1 Edu, which creates a network named *Unitree_Go134025A*, so that it is able to reach the Web page provided by the HTTP server running in the mobile robotic platform.

The Go1 Edu Web control page includes the video feed captured by multiple video cameras, which give a real-time perception of the surrounding environment of the mobile gNB carried on-board. To make the platform move, the MNO accessing the Web page must select the *Vision* option, followed by the type of movement (in this case it would be *Walk*), and press the button with the icon of a remote controller. Following this, two joysticks appear and the AWSD keys are used to move the mobile robotic platform left, front, backwards and right, respectively. In Figure 3.10 a screenshot of the visual information provided by the video cameras of the mobile robotic platform moving in the INESC TEC building is depicted.

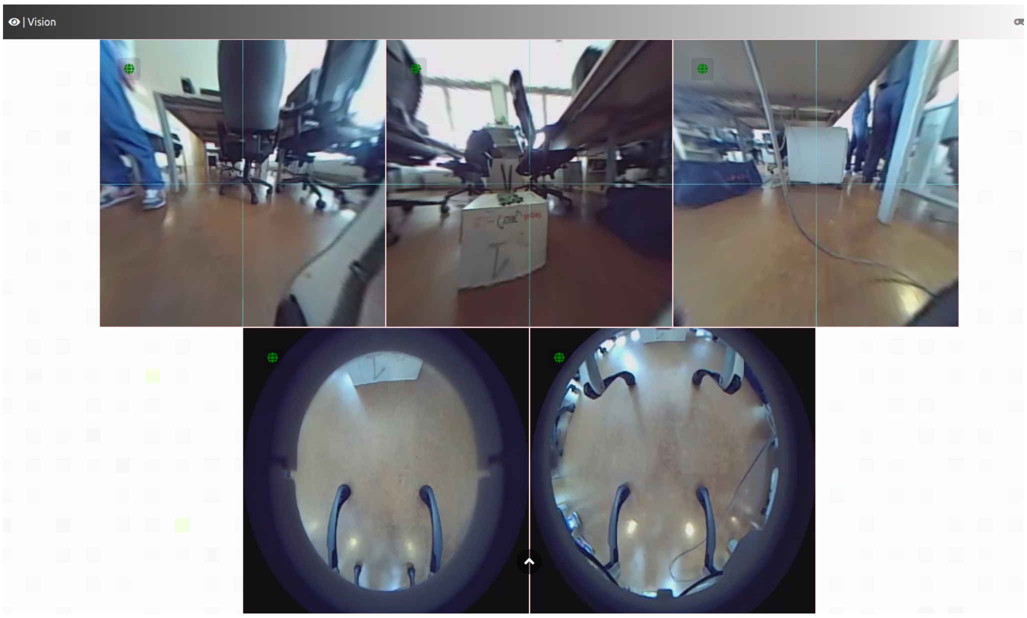


Figure 3.10: View provided by *Unitree* robotic mobile platform video cameras of the surrounding environment.

The Go1 Edu can also be controlled using a mobile application developed by *Unitree*, which is similar to the Web control page. Within the app, it also provides a simulation environment that allows to control a virtual Go1 Edu robot for training purposes. In Figure 3.11, the view provided by the mobile application is depicted. It illustrates the visual information obtained by a mobile robotic platform in simulation mode, which allows a clear definition of the surrounding environment for positioning a gNB on-demand.

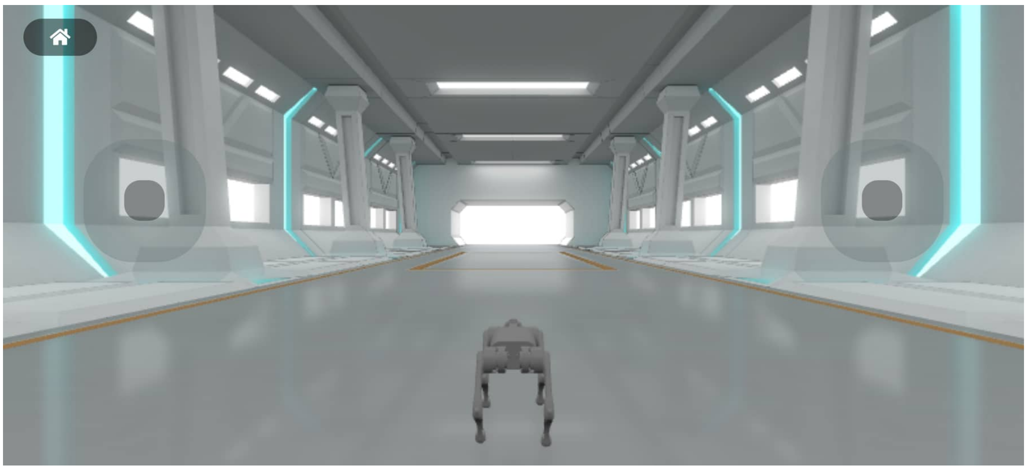


Figure 3.11: View provided by *Unitree* mobile robotic platform control app.

The Go1 Edu Robot was directly connected to the Core Network using a Wi-Fi connection. This is aligned with the architecture defined for the 5G Core network, in which the MNO, in charge of managing the proposed 5G network, only needs to known the URL associated with the ODMMF service to control the position of the gNB on-demand.

3.3 System Implementation

3.3.1 OAI Core Configuration

Regarding the Core network, OAI provides the image of each Core network component in a *Docker* configuration [43]. After pulling each image, the *docker-compose.yaml* file must be changed in order to achieve a successful and stable communication between all components. Since the traffic from *Docker* containers is not forwarded to the outside world by default, it is necessary to enable IP forwarding using two *Linux* commands. The command used to configure the Linux kernel for allowing IP forwarding is depicted in Figure 3.12, whereas Figure 3.13 depicts the command used to change the iptables FORWARD policy from DROP to ACCEPT.

```
david@david-PC:~$ sudo sysctl net.ipv4.conf.all.forwarding=1
net.ipv4.conf.all.forwarding = 1
```

Figure 3.12: Linux kernel IP forwarding configuration.

```
david@david-PC:~$ sudo iptables -P FORWARD ACCEPT
```

Figure 3.13: Iptables forwarding configuration.

After the packet forwarding rules were defined, the Subscriber Identification Module (SIM) details for OAI UE were added the *oai_db.sql* file. The information added includes the International Mobile Subscriber Identity (IMSI) and its related key so that the Core network can identify and authenticate subscribers, such as UEs. When the Core network is launched, an interface called *demo-oai* is created, which contains all network components and static IP address configuration. An IP address scheme of the Core Network components is presented in Table 3.1 and in Figure 3.14.

Core network component	IP
AMF	192.168.70.132
SMF	192.168.70.133
UPF	192.168.70.134
NRF	192.168.70.130
MYSQL Server	192.168.70.131
EXT-DN	192.168.70.135

Table 3.1: Core network components and the associated IP addresses.

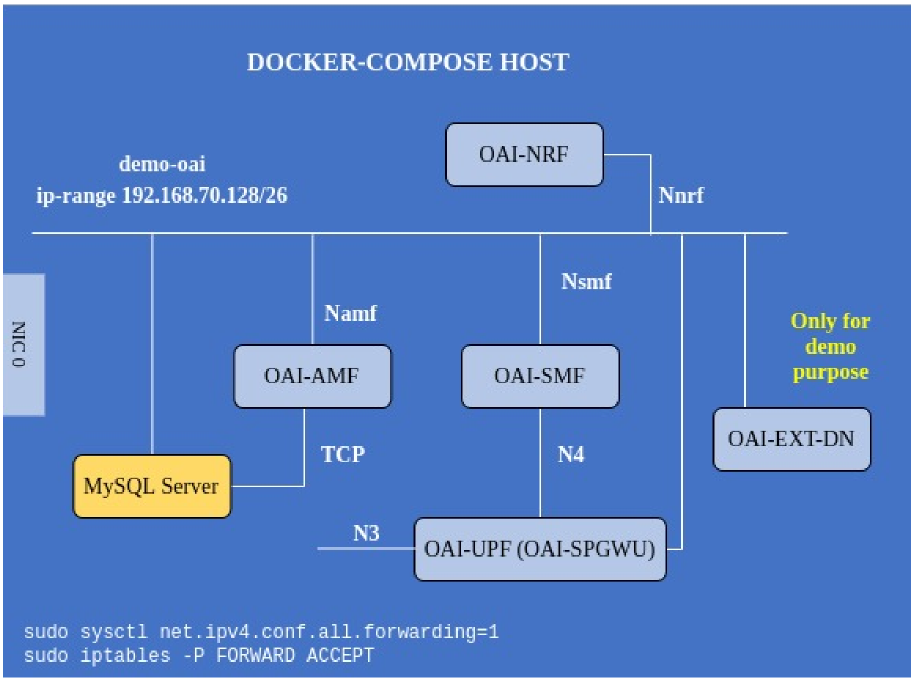


Figure 3.14: Core network IP address scheme [44].

When deploying the 5G Core network, it is important to guarantee that all containers launched using the *docker-compose.yaml* file are healthy and that the UE-related information is the same as the configured in the UE. In Figure 3.15, the *demo-oai* interface configuration is depicted.

```
david@david-Core:~/oai-cn5g-fed/docker-compose$ ifconfig demo-oai
demo-oai: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.70.129  netmask 255.255.255.192  broadcast 192.168.70.191
        inet6 fe80::42:40ff:feef:84a5  prefixlen 64  scopeid 0x20<link>
        ether 02:42:40:ef:84:a5  txqueuelen 0  (Ethernet)
        RX packets 10463  bytes 606258 (606.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18447  bytes 27622966 (27.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 3.15: *Demo-oai* interface configuration.

3.3.2 OAI RAN Configuration

Regarding the access network configuration, for both gNB and UE, a specific branch focused on the USRP B210 from OAI *Gitlab* was used [45]. When changing the gNB configuration file (*gnb.sa.band78.fr1.106PRB.usrpb210.conf*), it was necessary to ensure consistency with the SIM details added to the Core network, which means using the same Mobile Country Code (MCC) and Mobile Network Code (MNC) employed when defining the IMSI. Since the gNB and the Core network run on separate hosts, it was necessary to configure a control link (Wi-Fi or Ethernet) to connect the two hosts, as depicted in Figure 3.16.

```
GNB_INTERFACE_NAME_FOR_NG_AMF          = "wlxb0487a8fe0e1";
GNB_IPV4_ADDRESS_FOR_NG_AMF            = "10.60.11.157/16";
GNB_INTERFACE_NAME_FOR_NGU             = "wlxb0487a8fe0e1";
GNB_IPV4_ADDRESS_FOR_NGU               = "10.60.11.157/16";
GNB_PORT_FOR_S1U                       = 2152; # Spec 2152
```

Figure 3.16: Wi-Fi connection configuration between the gNB and the Core network.

To allow communication between the OAI software and the SDRs, the USRP Hardware Driver (UHD) was built. UHD is a user-space library used to control all USRP device models. It is also responsible for controlling different parameters, such as sampling rate, centre frequency and gains [46]. An example of UHD identifying the used SDR is presented in Figure 3.17.

```
david@david-PC:~$ uhd_find_devices
[INFO] [UHD] linux; GNU C++ version 7.5.0; Boost_106501; UHD_3.15.0.HEAD-0-gaea0e2de
[INFO] [B200] Loading firmware image: /usr/local/share/uhd/images/usrp_b200_fw.hex...
--------------------------------------------------
-- UHD Device 0
--------------------------------------------------
Device Address:
    serial: 307B5FD
    name:
    product: B210
    type: b200
```

Figure 3.17: UHD discovery utility function.

The previous configurations were considered in the *ue.conf* file in the UE host. This file has the default SIM values provided by OAI, which were changed to match the values added to the *oai_db.sql*. The file content considered is depicted in Figure 3.18.

```
ue.conf
uicc0 = {
imsi = "208990000007487";
key = "fec86ba6eb707ed08905757b1bb44b8f";
opc= "C42449363BBAD02B66D16BC975D77CC1";
dnn= "oai";
nssai_sst=1;
nssai_sd=1;
}
```

Figure 3.18: Updated SIM details in the *ue.conf* file.

Finally, a route was created to ensure connectivity between the gNB and the Core network, which were connected through Wi-Fi technology. In Figure 3.19 the command used to configure the route can be observed, where *wlxb0487a8fe0e1* is the name of the Wi-Fi interface attached to the computer running the gNB.

```
david@david-PC:~$
david@david-PC:~$ sudo route add -net 192.168.70.128/26 gw 10.60.9.97 dev wlxb0487a8fe0e1
```

Figure 3.19: Wi-Fi route between the gNB and the Core network.

OAI is composed of two *soft modems*, which are the two executables that implement the UE and the gNB. These *soft modems*, named respectively *nr-uesoftmodem* and *nr-softmodem*, ensure that a UE is capable of connecting to a gNB. In order to implement the *soft modems*, two SDRs, one for the UE and another for the gNB, were used.

3.3.3 Interface and Device Configuration

To start the 5G network components, a sequential order must be followed, starting by the Core network, followed by the gNB and finally the UE. This is done for logical reasons, as the Core network can be seen as the brain of the network, storing UE information and being responsible for many services, such as authentication and session management, which must be in a stable status before launching the gNB.

When launching the *soft modem* responsible for the gNB, the connection with the Core network must be established. This can be inspected through two methods: analyze the output information provided by the corresponding process, or using a packet analyzer tool, such as *Wireshark*, in order to check the NGAP setup request and the respective setup response. In Figure 3.20, a *Wireshark* capture of these packets can be observed, where the first IP source address corresponds to the one presented in Figure 3.16 and the destination IP address corresponds to the AMF, demonstrating a successful communication between the gNB and the Core network.

29	2.169003458	10.60.11.157	192.168.70.132	NGAP	126	NGSetupRequest
31	2.171594020	192.168.70.132	10.60.11.157	NGAP	574	NGSetupResponse

Figure 3.20: NGAP setup request and response.

In order to ensure the 5G connectivity between the UE and the gNB, the two SDRs must be synchronized. In theory, if the two devices are operating at the same frequency (defined when executing each *soft modem*) and at a relatively short distance from each other, they should have no problems synchronizing. However, this can be confirmed visually: if the red and green LEDs of both USRP B210 SDRs are on, synchronization has been achieved, as it can be seen in Figure 3.21.

Figure 3.21: Connection between UE and gNB established.

When all 5G network components are deployed, the UE firstly communicates with the gNB, which forwards the information to the Core network, more specifically to the AMF. This information is mostly transmitted using the NG-AP protocol and is related to the authentication and setup of the radio communications established between the gNB and the UE. After the radio communications are established, the Core network elements exchange messages using HTTP and check if the UE SIM details match the ones presented in Figure 3.18; if they do, the UE is considered registered and a PDU session is established. This process can be observed in Figure 3.22.

10.60.11.157	192.168.70.132	NGAP/N…	146	InitialUEMessage, Registration request
192.168.70.132	10.60.11.157	NGAP/N…	630	DownlinkNASTransport, Authentication request
10.60.11.157	192.168.70.132	NGAP/N…	146	UplinkNASTransport, Authentication response
192.168.70.132	10.60.11.157	NGAP/N…	462	DownlinkNASTransport, Security mode command
10.60.11.157	192.168.70.132	NGAP/N…	174	UplinkNASTransport
192.168.70.132	10.60.11.157	NGAP/N…	1262	InitialContextSetupRequest
10.60.11.157	192.168.70.132	NGAP	122	UERadioCapabilityInfoIndication
10.60.11.157	192.168.70.132	NGAP	86	InitialContextSetupResponse
10.60.11.157	192.168.70.132	NGAP/N…	118	UplinkNASTransport
10.60.11.157	192.168.70.132	NGAP/N…	146	UplinkNASTransport
192.168.70.132	192.168.70.130	HTTP	217	GET /nnrf-disc/v1/nf-instances?target-nf-type=SMF&requester-nf-type=AMF HTTP/1.1
192.168.70.130	192.168.70.132	HTTP/J…	854	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
192.168.70.132	192.168.70.133	HTTP/J…	837	POST /nsmf-pdusession/v1/sm-contexts HTTP/1.1 , JavaScript Object Notation (appli…
192.168.70.133	192.168.70.132	HTTP/J…	234	HTTP/1.1 201 Created , JavaScript Object Notation (application/json)
192.168.70.133	192.168.70.132	HTTP/J…	1578	POST /namf-comm/v1/ue-contexts/imsi-2089900007487/n1-n2-messages HTTP/1.1 , JavaS…
192.168.70.132	192.168.70.133	HTTP	125	HTTP/1.1 200 OK
192.168.70.132	10.60.11.157	NGAP/N…	266	PDUSessionResourceSetupRequest
10.60.11.157	192.168.70.132	NGAP	122	PDUSessionResourceSetupResponse
192.168.70.132	192.168.70.133	HTTP/J…	468	POST /nsmf-pdusession/v1/sm-contexts/1/modify HTTP/1.1 , JavaScript Object Notati…
192.168.70.133	192.168.70.132	HTTP/J…	167	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)

Figure 3.22: *Wireshark* capture regarding UE registration.

After establishing a radio link, an IP address must be assigned to the UE to guarantee an end-to-end connection. The Core component responsible for this assignment is the UPF and it uses the

12.1.1.0/24 IP pool in our setup. In Figure 3.23, a screenshot of part of the *docker-compose.yaml* file regarding the UE IP configuration can be observed.

```
- NETWORK_UE_NAT_OPTION=yes
- NETWORK_UE_IP=12.1.1.0/24
```

Figure 3.23: Screenshot of the *docker-compose.yaml* file regarding UE IP address allocation.

After a successful synchronization with the gNB and correspondence of the SIM details with the Core network configuration, an interface with an address respecting the rules of UE IP address configuration is assigned by the OAI to the UE. In Figure 3.24 this interface can be visualised.

```
david@david-PC:~$ ifconfig oaitun_ue1
oaitun_ue1: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 12.1.1.2  netmask 255.255.255.0  destination 12.1.1.2
        inet6 fe80::625:2abd:7f4:7fb3  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15  bytes 1432 (1.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 3.24: Tunnel interface created by the OAI in the UE.

When synchronization between the UE and the gNB is achieved, OAI provides indicators regarding the connection, such as Modulation and Coding Scheme (MCS) and Block Error Rate (BLER). This information is written in specific files, in order to allow characterizing the quality of communication between the UE and the gNB. This information is obtained by changing an OAI configuration file and constantly updating the files by using the *w+* flag. These adjustments in the configuration file are depicted in Figure 3.25.

```
fPtr = fopen("/home/david/Results_Radio/example.txt", "w+");
fPtr2 = fopen("/home/david/Results_Radio/example_BLER.txt", "w+");
fPtr3 = fopen("/home/david/Results_Radio/example_RSRP.txt", "w+");
fprintf(fPtr, "%d", sched_ctrl->dl_bler_stats.mcs);
fclose(fPtr);
fprintf(fPtr2, "%.5f", sched_ctrl->dl_bler_stats.bler);
fclose(fPtr2);
fprintf(fPtr3, "%d", avg_rsrp);
fclose(fPtr3);
```

Figure 3.25: Adjustments regarding the radio link quality parameters.

3.3.4 ODMMF Network Function

A Web server was created in the computer hosting the Core network using *Node.js*. This server offers a base Hypertext Markup Language (HTML) page containing two hyperlinks: a hyperlink that directs to a Web page displaying the real-time values of the MCS and BLER variables defined by OAI based on the real-world radio conditions experienced, and a second hyperlink that allows to access the Web control page of the Go1 Edu Robot, as illustrated in Figure 3.9.

The principal reason for the creation of this Web server is to fully take advantage of the emphasis that 5G places on virtualisation, leading to the possibility of remote control and monitoring of the 5G network using the HTTP protocol. *Node.js* was the tool chosen to implement this service, especially due to its high efficiency for real-time applications. The main code used to create the Web page with the radio link information can be observed in Figure 3.26. It demonstrates the process of allocating a specific IP address and port, and then the process of reading the files where the variables are stored in a stateless way (*fsreadFileSync* function).

```
var http = require('http');
var fs = require("fs");
const hostname = '192.168.1.248';
const port = 3000;


const server = http.createServer((req, res) => {
    const data_BLER = fs.readFileSync('/home/david/Results_Radio/example_BLER.txt', 'utf8');
    const data_MCS = fs.readFileSync('/home/david/Results_Radio/example.txt', 'utf8');
    console.log(data_BLER);
    res.writeHead(200, {'Content-Type': 'text/plain'});
```

Figure 3.26: Screenshot of part of the code used to create the Web page with the radio link information.

Regarding the Web server located on the Core network (the ODMMF Network Function), the objective is that an MNO only needs to know the URL associated with this Web server. In terms of IP address and port allocation, it is similar to the one presented in Figure 3.26. However, to make it possible to use HTML features such as hyperlinks, the *Express.js* framework was used for this implementation.

Express.js is a backend web application framework for *Node.js* that allows an easy rendering of HTML pages [47]. In our solution, it helps with the display of the HTML page present on the Core network Web server. The code used to show the two hyperlinks can be observed in Figure 3.27, in which, by using the *res.send* function, we can see the HTML code being sent.

```
const express = require("express");
const app = express();

app.listen(3000, '10.60.9.97', () => {
  console.log("Application started and Listening on http://10.60.9.97:3000");
});

app.get("/", (req, res) => {
  res.send('<html><p><a href="https://5Gradiolink.loophole.site/">Radio Stats</a></p><p>
            <a href="https://testeunitree.loophole.site">Dog Mobility</a></p></html>');
});
```

Figure 3.27: *Express.js* code used to create the Core Web server and respective Web page.

Considering that a user might not be familiarized with OAI or the 5G terminology, it was a point of emphasis to create a dynamic and clear HTML Web page that presents the status of the connection between the UE and the gNB. In Figure 3.28 the HTML and *Express.js* implementation can be observed, in which a valid MCS index is the indicator of the establishment of a 5G radio link.

```
res.write(`<!DOCTYPE html>
    <html>
    <head>
    <title>Radio Web Page</title>
    <h1>Radio Statistics Web Page</h1>
    </head>
    <body>`);

if (data_MCS==0) {
    res.write("<h2>You currently DO NOT HAVE a 5G connection!</h2>");
    res.write(`<h3>No Connection at this moment between the UE and the gNB</h3>`);
    res.write(`<p><h1>The Dog must be closer to the terminal!!</h1></p>`);
}
else {
    res.write("<h2>You currently have a 5G connection!</h2>");
    res.write(`<p>The MCS value is currently: ` + data_MCS + `</p>`);
    res.write(`<p>The BLER value is currently:` + data_BLER + `</p>`);
```

Figure 3.28: HTML and *Express.js* code used to create the Web pages for connectivity and non-connectivity cases.

Since the Go1 Edu Robot creates its own private network to which only the computer running the OAI Core is directly connected, it is necessary to make this Web page accessible online. To make this possible, the *Loophole* [48] tool was used to implement a tunnelling solution, where the Web control page is forwarded securely to a URL. The custom domain can be chosen freely by the 5G network owner. In Figure 3.29 the process of making online the Web page created by the Go1 Edu can be observed. Using the URL, a given MNO has now the capability of controlling the mobile robotic platform from anywhere in the world with an Internet connection. The same process happens in Figure 3.30 with the Web page associated with the quality of the radio link.

```
david@david-PC:~/loophole-cli_1.0.0-beta.15_linux_64bit$ ./loophole http 80 192.168.1.248 --hostname testeunitree
Loophole - End to end TLS encrypted TCP communication between you and your clients

Registering your domain... Success!
Starting local proxy server... Success!
I Take graphical screenshot nnel... Success!


Forwarding https://testeunitree.loophole.site -> http://192.168.1.248:80
Press CTRL + C to stop the service
Logs:
3:22PM INF Awaiting connections...
3:22PM INF Succeeded to accept connection over HTTPS
3:22PM INF TLS Certificate successfully provisioned
3:22PM INF Succeeded to accept connection over HTTPS
```

Figure 3.29: *Loophole* Web page creation.

```
david@david-PC:~/loophole-cli_1.0.0-beta.15_linux_64bit$ ./loophole http 80 192.168.1.13 --hostname 5GRadioLink
Loophole - End to end TLS encrypted TCP communication between you and your clients

Registering your domain... Success!
Starting local proxy server...  Success!
Initializing secure tunnel...  Success!


Forwarding https://5GRadioLink.loophole.site -> http://192.168.1.13:80
Press CTRL + C to stop the service
```

Figure 3.30: *Loophole* radio link quality Web page creation.

The real-time aspect of these Web servers is very important, since the proposed 5G network is composed of expensive equipment that has to be controlled very carefully. In order to achieve that, the least amount of latency possible is desired so that changes in the environment are quickly perceived by the MNO controlling the network. To achieve this real-time capability, a code implementation using automatic browser reload was used. In Figure 3.31, the HTML and Java Script code updating the Web page every two seconds can be observed. The *location.reload* method was chosen, due to its ability to constantly reload the Web page without cache and the fact that it is supported by all major Web browsers [49].

```
res.write(`</body>
    <script>location.reload(true);</script>
    </html>`);
res.end();
});
```

Figure 3.31: Screenshot of the code section regarding Web page reload.

Chapter 4

System Validation

This chapter is focused on the validation of the solution proposed by this dissertation. In Section 4.1, the methodology followed for the validation of the proposed solution is described. In Section 4.2, the experimental setup is presented. In Section 4.3, each component is tested. In Section 4.4, the performance of the proposed solution is evaluated. In Section 4.5, a real-world use case was validated. Finally, in Section 4.6, the achieved results are analysed and discussed.

4.1 Methodology

The first set of tests performed was related to **Component Testing**, where each component of the system was evaluated to determine its correct operation and, when possible, its performance evaluation. The second set of tests focused on the **Basic Interoperability Testing**, which was carried out to determine if the components were working jointly properly. The last set of tests, **Use Case Validation**, sought to define and validate specific use cases of the proposed system.

The methodology followed was the same for each set of tests: first, the purpose of the test was defined, followed by the setup it required; then, a functional validation using tools such as a ping and *Iperf* were carried out; finally, if applicable, the performance was evaluated.

4.2 Experimental Setup

This section presents the experimental setup and the testbed architecture. This setup is the real-world representation of the scheme shown in Figure 3.4. In Figure 4.1, a photograph of the experimental scenario can be observed.

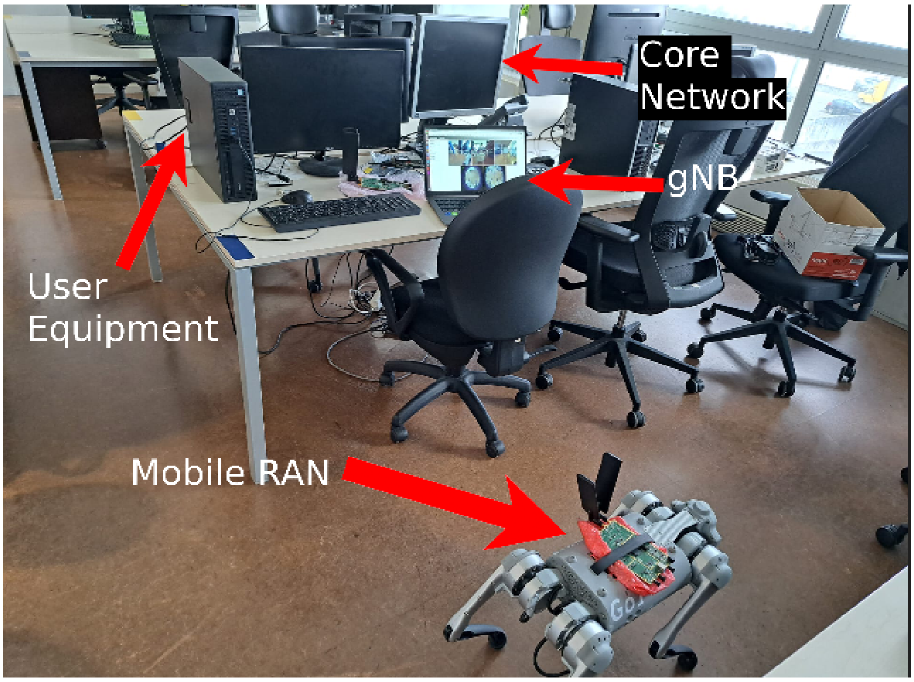


Figure 4.1: Experimental setup.

4.3 Component Testing

4.3.1 Core Network

This test was defined with the main purpose of evaluating the health of the *Docker* containers associated with the 5G Core components and assessing the connectivity between them.

To perform this test, the command shown in Figure 4.2 was launched in the NUC that implements the Core network. Although the command output indicates if the containers are healthy, it is not sufficient to validate the correct operation of the network.

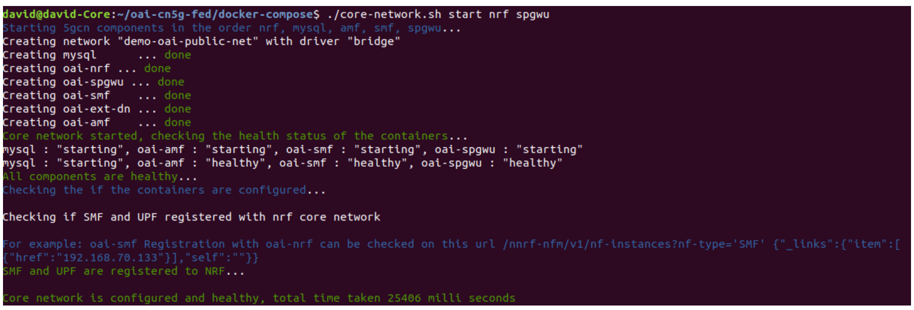
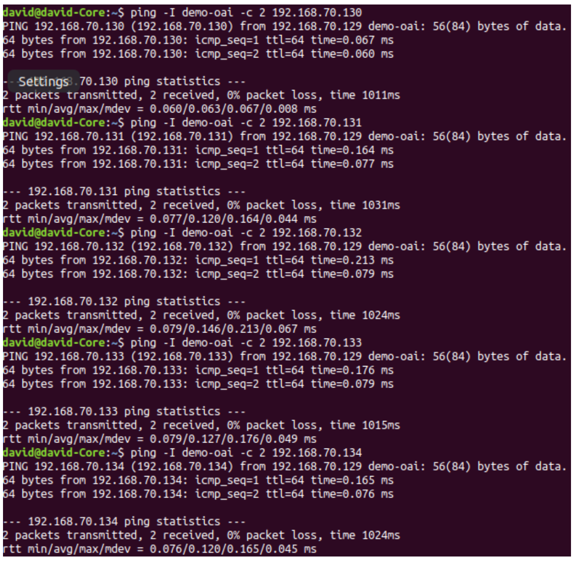


Figure 4.2: 5G Core Network successful launch.

In order to confirm the correct deployment, *demo-oai* must appear as an active interface when running the *ifconfig* command. Moreover, using that interface, we must be able to ping all components. In Figure 4.3 the outcome of the Internet Control Message Protocol (ICMP) packets sent to AMF, SMF, UPF and NRF is depicted.



```
david@david-Core:~$ ping -I demo-oai -c 2 192.168.70.130
PING 192.168.70.130 (192.168.70.130) from 192.168.70.129 demo-oai: 56(84) bytes of data.
64 bytes from 192.168.70.130: icmp_seq=1 ttl=64 time=0.067 ms
64 bytes from 192.168.70.130: icmp_seq=2 ttl=64 time=0.060 ms

- Settings .70.130 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1011ms
rtt min/avg/max/mdev = 0.060/0.063/0.067/0.008 ms
david@david-Core:~$ ping -I demo-oai -c 2 192.168.70.131
PING 192.168.70.131 (192.168.70.131) from 192.168.70.129 demo-oai: 56(84) bytes of data.
64 bytes from 192.168.70.131: icmp_seq=1 ttl=64 time=0.164 ms
64 bytes from 192.168.70.131: icmp_seq=2 ttl=64 time=0.077 ms

--- 192.168.70.131 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1031ms
rtt min/avg/max/mdev = 0.077/0.120/0.164/0.044 ms
david@david-Core:~$ ping -I demo-oai -c 2 192.168.70.132
PING 192.168.70.132 (192.168.70.132) from 192.168.70.129 demo-oai: 56(84) bytes of data.
64 bytes from 192.168.70.132: icmp_seq=1 ttl=64 time=0.213 ms
64 bytes from 192.168.70.132: icmp_seq=2 ttl=64 time=0.079 ms

--- 192.168.70.132 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1024ms
rtt min/avg/max/mdev = 0.079/0.146/0.213/0.067 ms
david@david-Core:~$ ping -I demo-oai -c 2 192.168.70.133
PING 192.168.70.133 (192.168.70.133) from 192.168.70.129 demo-oai: 56(84) bytes of data.
64 bytes from 192.168.70.133: icmp_seq=1 ttl=64 time=0.176 ms
64 bytes from 192.168.70.133: icmp_seq=2 ttl=64 time=0.079 ms

--- 192.168.70.133 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1015ms
rtt min/avg/max/mdev = 0.079/0.127/0.176/0.049 ms
david@david-Core:~$ ping -I demo-oai -c 2 192.168.70.134
PING 192.168.70.134 (192.168.70.134) from 192.168.70.129 demo-oai: 56(84) bytes of data.
64 bytes from 192.168.70.134: icmp_seq=1 ttl=64 time=0.165 ms
64 bytes from 192.168.70.134: icmp_seq=2 ttl=64 time=0.076 ms

--- 192.168.70.134 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1024ms
rtt min/avg/max/mdev = 0.076/0.120/0.165/0.045 ms
```

Figure 4.3: Successful pings from the computer that implements the Core to each Core component.

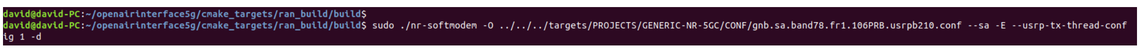
The system performed as expected, thus allowing us to conclude that the 5G Core Network is fully operational.

4.3.2 Mobile RAN

4.3.2.1 gNB

The gNB is the component of the mobile radio access network that is responsible for connecting the UE to the Core Network, establishing a 5G radio link with the UE and a Wi-Fi radio link with the Core.

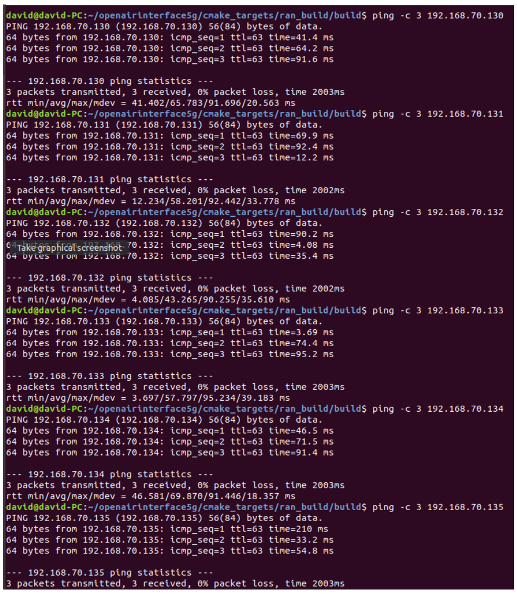
After connectivity is established with the Core Network through the creation of a static route, the gNB must be able to ping every component in the Core. For the radio communication performed over-the-air with the UE, this is done with the execution of the command observed in Figure 4.4. The *-O* flag specifies the location of the configurations file used, the *–sa* flag indicates that the gNB will run in standalone mode, the *–E* flag specifies that 3/4 sampling frequency must be applied, as recommended for the USRP B210 SDR [50]. Finally, the *–usrp-tx-thread-config 1* option tells the OAI software that an extra thread shall be used for the transmissions.



```
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$ sudo ./nr-softmodem -O ../../../targets/PROJECTS/GENERIC-NR-5GC/CONF/gnb.sa.band78.fr1.106PRB.usrpb210.conf --sa -E --usrp-tx-thread-conf
ig 1 -d
```

Figure 4.4: Command used to launch the gNB using OAI software.

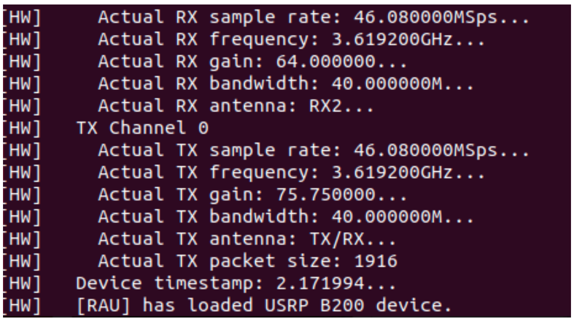
With the static route created, in Figure 4.5 successful connectivity test results from the gNB to the Core are depicted.



```
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$ ping -c 3 192.168.70.130
PING 192.168.70.130 (192.168.70.130) 56(84) bytes of data.
64 bytes from 192.168.70.130: icmp_seq=1 ttl=63 time=41.4 ms
64 bytes from 192.168.70.130: icmp_seq=2 ttl=63 time=64.2 ms
64 bytes from 192.168.70.130: icmp_seq=3 ttl=63 time=91.6 ms

--- 192.168.70.130 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 41.402/65.783/91.696/20.563 ms
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$ ping -c 3 192.168.70.131
PING 192.168.70.131 (192.168.70.131) 56(84) bytes of data.
64 bytes from 192.168.70.131: icmp_seq=1 ttl=63 time=69.9 ms
64 bytes from 192.168.70.131: icmp_seq=2 ttl=63 time=92.4 ms
64 bytes from 192.168.70.131: icmp_seq=3 ttl=63 time=12.2 ms

--- 192.168.70.131 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 12.234/58.201/92.442/33.778 ms
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$ ping -c 3 192.168.70.132
PING 192.168.70.132 (192.168.70.132) 56(84) bytes of data.
64 bytes from 192.168.70.132: icmp_seq=1 ttl=63 time=90.2 ms
64 bytes from 192.168.70.132: icmp_seq=2 ttl=63 time=4.08 ms
64 bytes from 192.168.70.132: icmp_seq=3 ttl=63 time=35.4 ms

--- 192.168.70.132 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 4.085/43.265/90.255/35.610 ms
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$ ping -c 3 192.168.70.133
PING 192.168.70.133 (192.168.70.133) 56(84) bytes of data.
64 bytes from 192.168.70.133: icmp_seq=1 ttl=63 time=3.69 ms
64 bytes from 192.168.70.133: icmp_seq=2 ttl=63 time=74.4 ms
64 bytes from 192.168.70.133: icmp_seq=3 ttl=63 time=95.2 ms

--- 192.168.70.133 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 3.697/57.797/95.234/39.183 ms
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$ ping -c 3 192.168.70.134
PING 192.168.70.134 (192.168.70.134) 56(84) bytes of data.
64 bytes from 192.168.70.134: icmp_seq=1 ttl=63 time=46.5 ms
64 bytes from 192.168.70.134: icmp_seq=2 ttl=63 time=71.5 ms
64 bytes from 192.168.70.134: icmp_seq=3 ttl=63 time=91.4 ms

--- 192.168.70.134 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 46.581/69.870/91.446/18.357 ms
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$ ping -c 3 192.168.70.135
PING 192.168.70.135 (192.168.70.135) 56(84) bytes of data.
64 bytes from 192.168.70.135: icmp_seq=1 ttl=63 time=210 ms
64 bytes from 192.168.70.135: icmp_seq=2 ttl=63 time=33.2 ms
64 bytes from 192.168.70.135: icmp_seq=3 ttl=63 time=54.8 ms

--- 192.168.70.135 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
```

Figure 4.5: Successful pings from the gNB to the Core components.

Since communication with the Core is successful, synchronization of the gNB and the UE is the following requirement the proposed system must meet. Besides the distance between the two SDRs, the carrier frequency used is the most important factor in the process of synchronization. The transmission (TX) and reception (RX) frequency in which the USRP B210 is operating is depicted in Figure 4.4 (cf. 3.6 GHz).



```
[HW]     Actual RX sample rate: 46.080000MSps...
[HW]     Actual RX frequency: 3.619200GHz...
[HW]     Actual RX gain: 64.000000...
[HW]     Actual RX bandwidth: 40.000000M...
[HW]     Actual RX antenna: RX2...
[HW]   TX Channel 0
[HW]     Actual TX sample rate: 46.080000MSps...
[HW]     Actual TX frequency: 3.619200GHz...
[HW]     Actual TX gain: 75.750000...
[HW]     Actual TX bandwidth: 40.000000M...
[HW]     Actual TX antenna: TX/RX...
[HW]     Actual TX packet size: 1916
[HW]   Device timestamp: 2.171994...
[HW]   [RAU] has loaded USRP B200 device.
```

Figure 4.6: Values regarding transmission and reception of data defined by the gNB.

4.3.2.2 Mobile Robotic Platform

In terms of setup, the USRP B210 SDR was connected to the computer implementing the gNB. The SDR was placed on-board the Go1 Edu Robot. For security purposes, the mobile robotic platform has a rubber handle to hold the SDR, allowing the Go1 Edu Robot to move in a fast but secure manner, without damaging the SDR. In Figure 4.7, a photograph of the SDR positioned on top of the mobile robotic platform is presented.

Figure 4.7: USRP B210 positioned on top of the mobile robotic platform.

With the setup ready, a computer, placed close to the mobile robotic platform, connected to the Wi-Fi created by the Go1 Edu Robot and accessed its Web control page. After accessing the *Vision* page and selecting the *Walk* mode, the user gave movement commands using the AWSD keys so that the mobile robotic platform was able to take a step in each direction.

This experiment was successful with the mobile robotic platform showing stability and with low latency between the instant when the user gave the mobile robotic platform an order to move in a certain direction and the actual movement was performed. With the mobile robotic platform's performance validated, the gNB was ready to be launched and positioned on-demand.

4.3.3 User Equipment

With the gNB looking for devices to synchronize with at the frequency of 3.6 GHz, the next component to be deployed was the User Equipment. In order to conduct this test, the USRP B210 SDRs were positioned one meter apart from each other, on top of a table in the third floor of the INESC TEC building. In Figure 4.8, a photograph of the setup used in this test can be observed.

Figure 4.8: Setup with the SDRs positioned one meter apart from each other.

The command executed to deploy the UE using the OAI software is shown in Figure 4.9. The *–band 78* option specifies the band index, being the same as the one specified in the configurations file of the gNB. The *-r 106* option defines the channel bandwidth in terms of Resource Blocks (RBs), with 106 being the default value. The *–numerology 1* flag specifies the type of numerology, for which 5G allows multiple types (from 0 to 4) of subcarrier spacing [51]; the gNB configurations file in OAI uses numerology type 1. The *–sa* and *-E* are the same used when launching the gNB, while the *—nokrnmod 1* option indicates OAI to use a different module than the default one. The *–dlsch-parallel 8* flag defines that 8 threads are required for the downlink shared channel processing. Finally, the *-O* flag specifies the path to the UE's configurations file.

```
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$ sudo ./nr-uesoftmodem -C 3619200000 --band 78 -r 106 --numerology 1 --sa -E --nokrnmod 1 --dlsch-parallel 8 -O ../../../targets/PROJECTS/
GENERIC-NR-5GC/CONF/ue.conf
```

Figure 4.9: Command used to launch the UE using OAI software.

For deploying all 5G components, the correct synchronization and communication between the UE and the gNB must be ensured, which occurs when the two SDRs are within a certain range of each other and the SIM details are correct. In Figure 4.10, we can observe the UE information regarding the time instant when synchronization occurs and the SIM details are exchanged with the Core network (cf. Figure 3.22).

```
[NR_RRC]   [UE 0] State = NR_RRC_CONNECTED (gNB 0)
[LIBCONFIG] uicc0: 9/9 parameters successfully set, (3 to default value)
[SIM]   UICC simulation: IMSI=2089900007487, Ki=fec86ba6eb707ed08905757b1bb44b8f, OPc=C42449363BBAD02B66D16BC975D77CC1
```

Figure 4.10: UE communication and registration with the gNB.

In terms of functional validation, when synchronization occurs and the tunnel interface *oaitun_ue1* is created, the UE must be able to ping every Core Network component and the Internet, with *Google's* public Domain Name System (DNS) server (8.8.8.8) serving as the online IP address. In Figures 4.11 and 4.12, the outcome of the ping tests to respectively the Core and to the Internet is depicted.

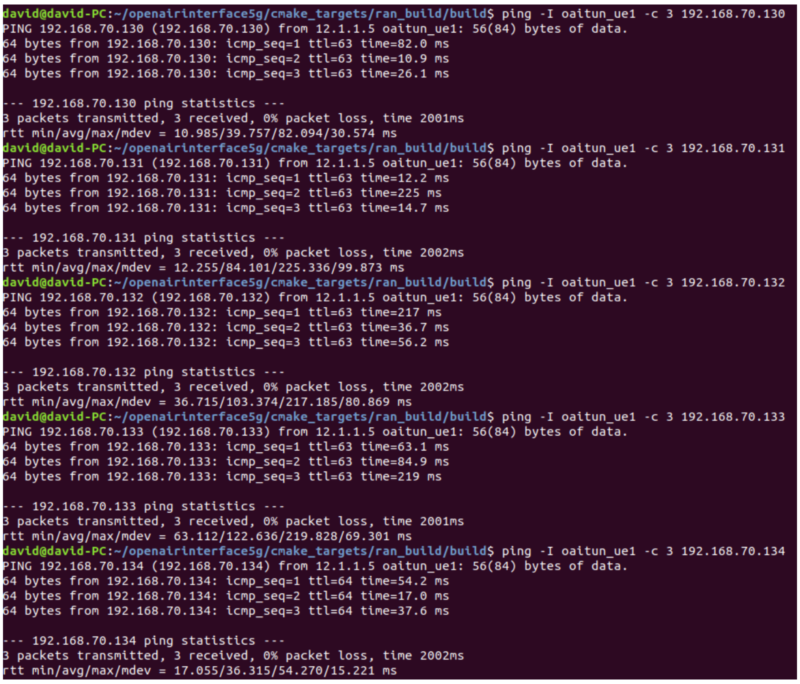


Figure 4.11: Successful pings from the UE to the Core components using the tunnel interface.

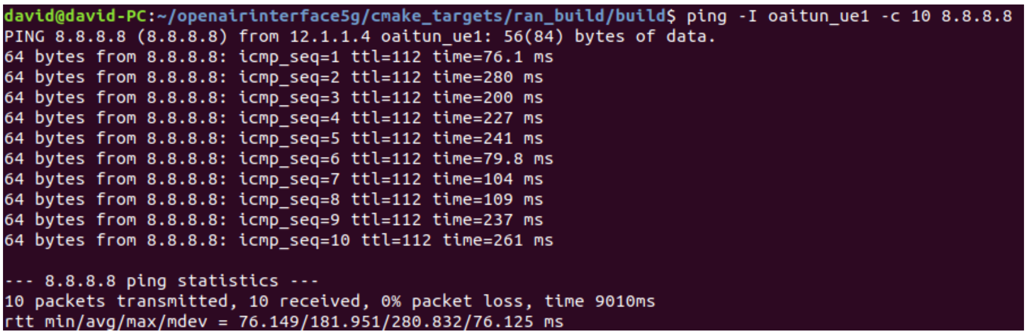
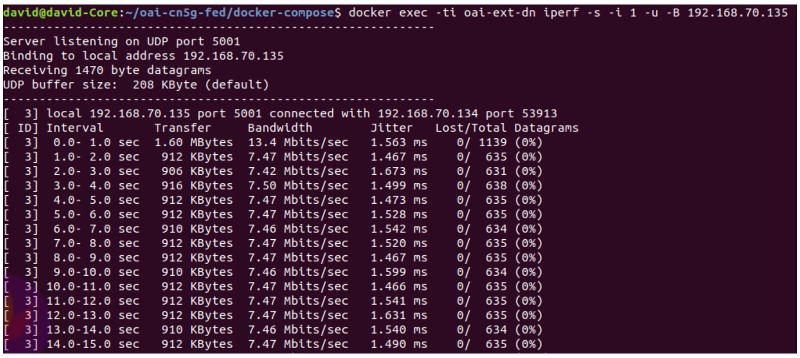


Figure 4.12: Successful pings from the UE to *Google's* public DNS server.

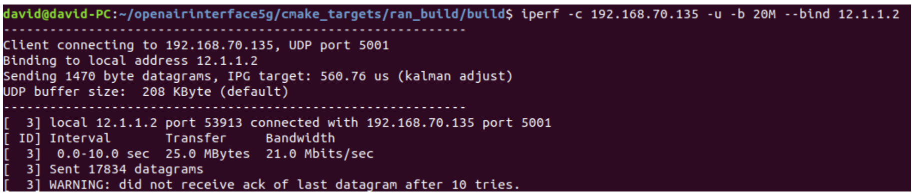
The round trip time (RTT) values, which represent the amount of time in milliseconds from when a request is sent to when the sender receives a response, were higher when accessing the Internet compared to the Core network. The results show 0% packet loss in both cases.

To gather network performance measurements such as throughput, the *Iperf* [52] tool was used. It allows to generate TCP and UDP flows between a client and a server. This test aimed at assessing the maximum throughput achieved in our 5G network. In this test scenario the setup of the components was the same as in the previous experiment, with the UE acting as the client and the Core Network as the server. As it can be visualised in Figure 4.13 and Figure 4.14, two commands were executed to generate UDP (specified by the *-u* flag) traffic flows. The UDP traffic was characterized by a bit rate of 20 Mbit/s.



```
david@david-Core:~/oai-cn5g-fed/docker-compose$ docker exec -ti oai-ext-dn iperf -s -i 1 -u -B 192.168.70.135
------------------------------------------------------------
Server listening on UDP port 5001
Binding to local address 192.168.70.135
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 192.168.70.135 port 5001 connected with 192.168.70.134 port 53913
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  1.60 MBytes  13.4 Mbits/sec   1.563 ms    0/ 1139 (0%)
[  3]  1.0- 2.0 sec   912 KBytes  7.47 Mbits/sec   1.467 ms    0/  635 (0%)
[  3]  2.0- 3.0 sec   906 KBytes  7.42 Mbits/sec   1.673 ms    0/  631 (0%)
[  3]  3.0- 4.0 sec   916 KBytes  7.50 Mbits/sec   1.499 ms    0/  638 (0%)
[  3]  4.0- 5.0 sec   912 KBytes  7.47 Mbits/sec   1.473 ms    0/  635 (0%)
[  3]  5.0- 6.0 sec   912 KBytes  7.47 Mbits/sec   1.528 ms    0/  635 (0%)
[  3]  6.0- 7.0 sec   910 KBytes  7.46 Mbits/sec   1.542 ms    0/  634 (0%)
[  3]  7.0- 8.0 sec   912 KBytes  7.47 Mbits/sec   1.520 ms    0/  635 (0%)
[  3]  8.0- 9.0 sec   912 KBytes  7.47 Mbits/sec   1.467 ms    0/  635 (0%)
[  3]  9.0-10.0 sec   910 KBytes  7.46 Mbits/sec   1.599 ms    0/  634 (0%)
[  3] 10.0-11.0 sec   912 KBytes  7.47 Mbits/sec   1.466 ms    0/  635 (0%)
[  3] 11.0-12.0 sec   912 KBytes  7.47 Mbits/sec   1.541 ms    0/  635 (0%)
[  3] 12.0-13.0 sec   912 KBytes  7.47 Mbits/sec   1.631 ms    0/  635 (0%)
[  3] 13.0-14.0 sec   910 KBytes  7.46 Mbits/sec   1.540 ms    0/  634 (0%)
[  3] 14.0-15.0 sec   912 KBytes  7.47 Mbits/sec   1.490 ms    0/  635 (0%)
```

Figure 4.13: *Iperf* test with the server launched from the Core network.



```
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$ iperf -c 192.168.70.135 -u -b 20M --bind 12.1.1.2
------------------------------------------------------------
Client connecting to 192.168.70.135, UDP port 5001
Binding to local address 12.1.1.2
Sending 1470 byte datagrams, IPG target: 560.76 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 12.1.1.2 port 53913 connected with 192.168.70.135 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  25.0 MBytes  21.0 Mbits/sec
[  3] Sent 17834 datagrams
[  3] WARNING: did not receive ack of last datagram after 10 tries.
```

Figure 4.14: *Iperf* test with the server launched from the UE.

The results obtained, where Figure 4.13 depicts the *Iperf* server running in the Core Network and Figure 4.14 depicts the *Iperf* client, allowed us to conclude that the throughput achieved in our 5G network is approximately 13 Mbit/s. The value of throughput was lower than expected, but this can be explained by the hardware used. Even though the low throughput, this allows validating the proof of concept.

The following test performed aimed at evaluating the amount of time it took the 5G Core Network to realize that the UE had lost its 5G wireless connection. Regarding the setup, the positioning of the components was equal to the previous two experiments. Concerning the methodology to measure this amount of time, we used the ping tool to assess the connectivity between the UE to the Internet when the UE had a stable 5G wireless link established with the gNB, followed by the termination of the execution of OAI's UE. We considered *Wireshark* timestamps to measure the time it took for the Core Network to receive an NG-AP packet indicating the release of the UE after the first unsuccessful ping. The results obtained are shown by the underlined timestamps in Figure 4.15. The 5G Core takes approximately 20 seconds to detect the release of the UE.

345	80.398677431	12.1.1.2	8.8.8.8	GTP <I…	134	Echo (ping) request id=0x0d28, seq=14/3584, ttl=64 (reply in 348)
346	80.398735585	192.168.70.134	8.8.8.8	ICMP	98	Echo (ping) request id=0x0d28, seq=14/3584, ttl=63 (reply in 347)
347	80.411541217	8.8.8.8	192.168.70.134	ICMP	98	Echo (ping) reply id=0x0d28, seq=14/3584, ttl=113 (request in 346)
348	80.411583771	8.8.8.8	12.1.1.2	GTP <I…	142	Echo (ping) reply id=0x0d28, seq=14/3584, ttl=112 (request in 345)
401	100.600686465	10.31.2.186	192.168.70.132	NGAP	98	UEContextReleaseRequest
402	100.601964002	192.168.70.132	10.31.2.186	NGAP	270	SACK (Ack=9, Arwnd=106496) , UEContextReleaseCommand
403	100.630588511	10.31.2.186	192.168.70.132	NGAP	738	SACK (Ack=5, Arwnd=106304) , UEContextReleaseRequest, UEContextReleaseRequest, UECc
404	100.632552358	192.168.70.132	10.31.2.186	NGAP	270	SACK (Ack=22, Arwnd=106146) , UEContextReleaseCommand
405	100.636057011	10.31.2.186	192.168.70.132	NGAP	102	SACK (Ack=6, Arwnd=106304) , UEContextReleaseComplete

Figure 4.15: *Wireshark* screenshot with the amount of time necessary for UE release detection.

The following test performed aimed at determining the maximum possible distance between the SDRs in which communication still occurs and the UE is able to reach the Internet. This test was also very important to determine how OAI and the SDRs reacted to mobility changes during the execution of the software.

Once the UE was able to synchronize with the gNB and therefore communicate with the Core Network, we started to slowly increasing the distance between the USRP B210 SDRs, until the pings from the UE to the Internet began to fail. A photograph depicting the maximum distance achieved between the SDRs is shown in Figure 4.16.

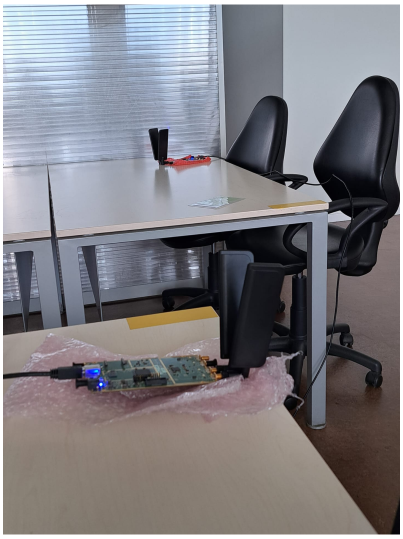


Figure 4.16: Maximum possible distance between the two USRP B210 SDRs so that a wireless link is established.

The test allowed us to conclude that the maximum distance enabling 5G wireless connectivity was approximately two meters from each other. This distance decreases if there are obstacles in the line of sight of the USRPs and increases in more isolated radio environments, such as laboratories. The low power provided by the USRP B210 SDR is also a cause of this short range of communications achieved, since this hardware had already proven to be unstable, even in scenarios with a strong link [53].

With the on-demand private 5G network fully functional and presenting good results, the ODMMF Network Function was ready for deployment.

4.3.4 On-Demand Mobile Management Function

With the 5G network and mobile robotic platform basic functionality validated and its basic performance estimated, it was time to incorporate the new Network Function developed, and validate its basic functionality.

ODMMF is composed of a home page with two hyperlinks that redirect the user to the two Web pages available: 1) a Web page that displays in real-time the existence of the 5G connections and the statistics associated with these connections; 2) a Web page that allows the remote control of the mobile robotic platform.

To evaluate the performance of the ODMMF, the tests performed focused on the radio link quality Web page and its synchronization with the real-time output of the gNB command that displays the statistics of the 5G connection.

With the two Web pages online, ODMMF was ready to be launched. To setup the experiment, a *Node.js* command was executed to deploy the ODMMF. Following the execution of this command, a user was able to reach the IP associated with this server, *10.31.3.82*, and access the home page with the two hyperlinks that redirect to each Web page. In Figure 4.17, the home page of the ODMMF Web service is shown.

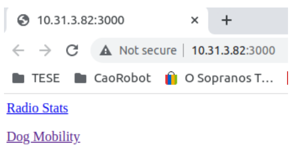


Figure 4.17: ODMMF home page with the two hyperlinks.

The radio link quality Web page provides two messages, according to the availability of 5G wireless connectivity. If the UE was able to connect with success, the Web page indicates the establishment of 5G connectivity and displays the details of the SIM card that joined the network; if the UE was not able to connect or lost connection, the Web page updates itself and informs the user (i.e., the MNO) of the change, advising him to position the mobile robotic platform closer to the UE. The two possible displays are depicted side by side in Figure 4.18.

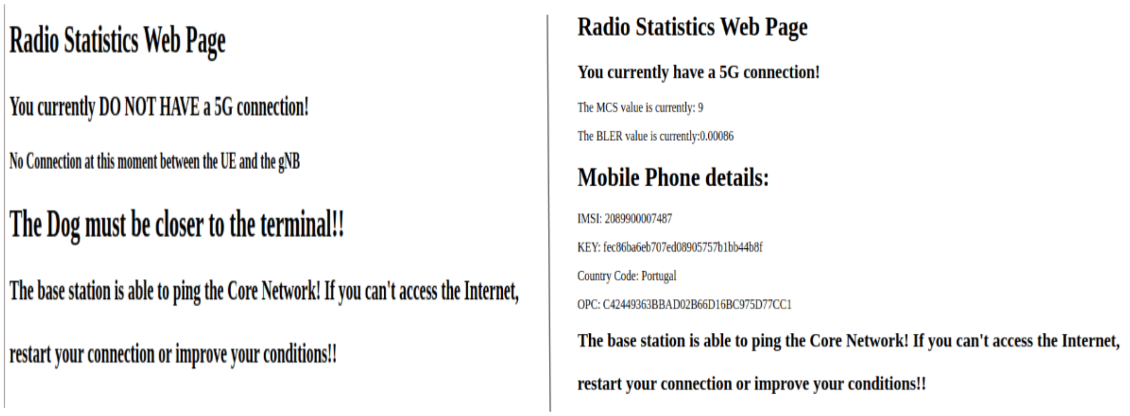


Figure 4.18: Two possible displays for the radio link quality Web page.

To evaluate the real-time synchronization of the ODMMF NF, Figure 4.19 depicts the comparison between the output of the OAI *nr-softmodem* with the information provided by the radio link Web page.

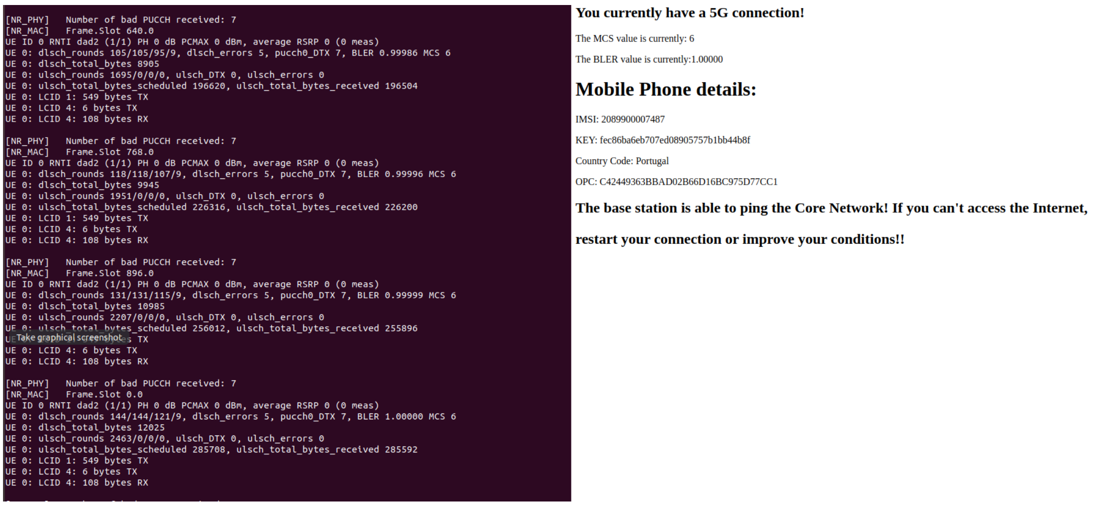


Figure 4.19: Synchronization between the OAI software output and the radio Web page.

As shown in Figure 4.19, the *location.reload* function proved to be a successful choice in terms of performance, with our radio link quality Web page being capable of updating itself autonomously with a periodicity that enables it to be synchronized with the OAI software.

4.4 Basic Interoperability Testing

This test aimed at jointly assessing the performance of the components that were individually evaluated in Section 4.3.

In terms of setup, to analyze the results of the fully operational testbed, experimental tests were performed in the INESC TEC building, using the experimental setup depicted in Figure 4.1.

To validate our experimental setup, a representative networking scenario was defined. Firstly, we positioned the mobile robotic platform carrying the gNB ten meters apart from the UE, which is a high enough distance so that the UE is outside the range of the communications cell enabled by the gNB; then, the Web control page was accessed to position the platform closer to the UE. With the SDRs capable of communicating and synchronizing, the UE was able to join the on-demand private 5G network and take advantage of 5G wireless connectivity to the Internet. The radio link quality Web page updated the information displayed accordingly, reporting a successful 5G connection and showing the statistics associated with it.

In Figure 4.20, the Go1 Edu Web control page showing the mobile robotic platform ten meters from the connectionless UE firstly, and then in a position closer to the UE, after moving according to the HTTP commands sent, is depicted.

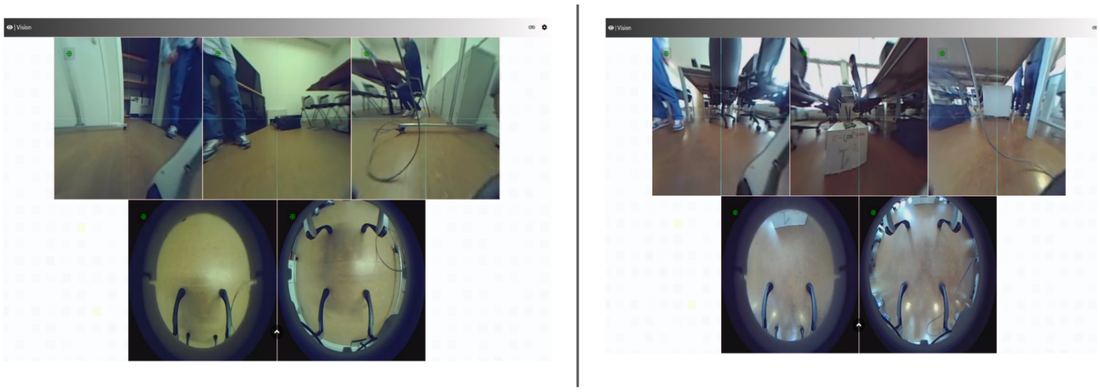


Figure 4.20: Side by side approximation of the mobile robotic platform to the UE.

To validate the capability of the UE accessing the Internet via the tunnel interface created, several ICMP ping packets were sent to *Google's* public DNS server, which are depicted in Figure 4.21.

```
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$ ping -I oaitun_ue1 -c 10 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 12.1.1.4 oaitun_ue1: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=76.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=280 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=200 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=112 time=227 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=112 time=241 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=112 time=79.8 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=112 time=104 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=112 time=109 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=112 time=237 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=112 time=261 ms

--- 8.8.8.8 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9010ms
rtt min/avg/max/mdev = 76.149/181.951/280.832/76.125 ms
```

Figure 4.21: Successful pings from the UE to *Google's* public DNS server.

Following the successful demonstration of an Internet connection, another computer, connected to the Wi-Fi that the gNB and Core Network use to communicate accessed the ODMMF home page. This computer was also able to access the radio link quality Web page, whose display can be observed in Figure 4.22.

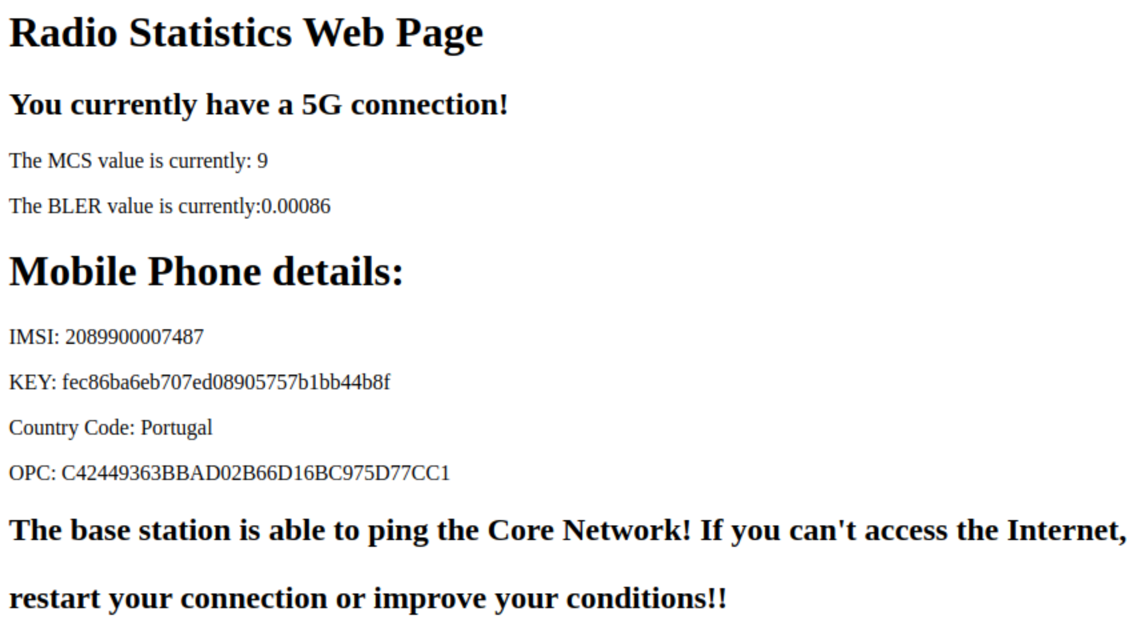


Figure 4.22: View of the online radio link quality Web page indicating a successful 5G connection.

This allowed to attest that the radio Web page is capable of updating itself, taking into account the conditions of the network and the attachment of a UE to the private on-demand 5G network.

4.5 Use Case Validation

An MNO, fictionally referred as *5GVentures*, is looking to provide 5G SA connectivity to a customer, the collaborators at the third floor of the INESC TEC building. To accomplish the terms agreed in their customer-supplier contract, *5GVentures* deployed the Go1 Edu Robot on the third floor. When an INESC TEC collaborator in the third floor wants to establish or restore 5G connectivity for a 4K video call, the Go1 Edu Robot is placed closer to the collaborator's device.

With our testbed fully operational, a scenario to validate it using a real-life use case application was considered. The scenario considers the possibility of multiple OAI UEs take advantage of the wireless 5G connectivity.

4.5.1 Multiple UEs

Similar to what was done for the first UE, the SIM details of the second UE were included in the *oai_db.sql*, which is a database of SIM cards that are allowed to connect to the 5G Core Network. In Figure 4.23, the inclusion of the new SIM details in this database is depicted.

```
INSERT INTO `users` VALUES ('2089900007486','380561234567','55000000000001',NULL,'PURGED'
  ,50,40000000, 100000000,47,0000000000,1,0xfec86ba6eb707ed08905757b1bb44b8f,0,0,0x40,
  'ebd07771ace8677a',0xc42449363bbad02b66d16bc975d77cc1);
```

Figure 4.23: Addition of the second UE SIM details to the *oai_db.sql* file.

When launching the OAI software in the computer that implements multiple UEs, OAI distinguishes the several SIM cards by allowing the user to use a flag that specifies the IMSI of the SIM card demanding wireless 5G connectivity. Since the IMSI value must be unique for each SIM card, the *–uicc0.imsi* flag plus the IMSI value must be used, which in this case is *2089900007486*. A screenshot of the command using this flag to indicate the UE that aims at establishing a 5G connection can be observed in Figure 4.24.

```
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$
david@david-PC:~/openairinterface5g/cmake_targets/ran_build/build$ sudo ./nr-uesoftmodem -C 3619200000 --uicc0.imsi 2089900007486 --band 78 -r 106 --numerology 1 --sa -E --nokrnmod 1 --dlsch-parallel 8
```

Figure 4.24: Command used to deploy a specific UE using OAI software.

Using *Wireshark*, we were able to verify that the UE was able to connect to the 5G network. This was done by analyzing the HTTP messages that were exchanged between the AMF and SMF. These messages, sent using HTTP/1.1, include multiple information regarding the UE, including IMSI, as shown in Figure 4.25.


```
------Boundary
Content-Type: application/json

{"n1MessageContainer":{"n1MessageClass":"SM","n1MessageContent":
{"contentId":"n1SmMsg"}},"n1n2FailureTxfNotifURI":"192.168.70.132/nsmf-pdusession/v1/callback/
N1N2MsgTxfrFailureNotification/imsi-2089900007486","n2InfoContainer":{"n2InformationClass":"SM","ranInfo":"SM","smInfo":
{"n2InfoContent":{"ngapData":{"contentId":"n2msg"},"ngapIeType":"PDU_RES_SETUP_REQ"},"pduSessionId":10,"sNssai":
{"sd":"1","sst":1}}},"pduSessionId":10}
------Boundary
Content-Type: application/vnd.3gpp.5gnas
Content-Id: n1SmMsg

.
....        ...11..........
)......".....y... A...{...%..oai.mnc099.mcc208.gprs
------Boundary
Content-Type: application/vnd.3gpp.ngap
Content-Id: n2msg

........... ......
....F.....................
------Boundary--
HTTP/1.1 200 OK
Connection: Close
Content-Length: 2

OK
```


Figure 4.25: *Wireshark* screenshot of the HTTP packet sent from the AMF to SMF regarding the UE.

4.6 Discussion

The tests carried out allowed to evaluate the basic functionality and performance of the designed system. The problem of this dissertation was addressed by means of the on-demand positioning of a 5G radio station carried by a mobile robotic platform, which is able to overcome the communications challenges associated with several use cases on different scenarios, such as a UE losing connectivity, by enabling network reconfiguration to provide wireless connectivity to the UE.

As emphasized in the SoA, the SDN and NFV paradigms are being increasingly applied and it has become almost mandatory for a service to be available anywhere, anytime. With this, the Network Function developed, ODMMF, enables an on-demand network compliant with 5G and beyond network architectures.

The first testing phase, Component Testing, sought to evaluate and validate the individual performance of each component that composes our testbed. Following the successful validation, the second test scenario, Basic Interoperability Testing, focused on evaluating and validating the correct behaviour of a private on-demand 5G network, with all individual components implemented together to form the testbed. Finally, the third set of tests, Use Case Validation, considered an experimental scenario focused on real-world applications such as the use of multiple UEs; this tests was performed taking into account the perspective of an MNO providing 5G connectivity to a customer.

The networking scenarios evaluated allowed us to conclude that our solution is a valuable contribution for an MNO that aims at taking advantage of the capability provided by a mobile robotic platform to be positioned remotely using a browser, while carrying a gNB. Another conclusion is that 5G is incredibly demanding in terms of hardware requirements, requiring computers and SDRs with very high computational power.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The main goal of the dissertation was to develop a solution to control and position a 5G radio access node carried by a mobile robotic platform, while deploying a standalone private on-demand 5G network. The proposed solution allows to overcome the limitations in terms of flexibility and customization provided by commercial network deployments.

On the one hand, 5G plays a crucial role in improving the Quality of Service (QoS) and Quality of Experience (QoE) offered to mobile users by providing higher data rates, ultra-low latency, more reliability, and optimized management of the communications resources. Combined with mobile networks, which can be dynamically positioned according to the users' needs, 5G private networks are capable of promoting better QoS and QoE for an enterprise or entity in certain scenarios.

On the other hand, the dynamic placement and resource management in high mobility networks allows to enhance the overall performance achieved by improving the flexibility for dealing with unexpected bursts of traffic that may preclude the QoS offered by static network deployments. When taking advantage of the capabilities provided by 5G, high mobility networks have characteristics that make them suitable for temporary networking scenarios, as they provide flexibility to be deployed anywhere, anytime.

Another characteristic that should be taken into consideration when addressing on-demand 5G networks is its service-based architecture, one of the pillars of the 5G Core network. With this architecture deployed, the 5G Core is flexible, agile and scalable, with multiple network functions that can be managed independently, while allowing a faster and optimized integration of new services.

Our solution proposes a 5G SA network architecture where the 5G radio station is carried by a mobile robotic platform. This solution considers that the gNB, which is placed on top of a mobile robotic platform, establishes communication with the 5G Core Network using a wireless connection. It is able, by means of movement commands sent to the mobile robotic platform by a proposed On-Demand Mobility Management Function (ODMMF) deployed into the Core network, to provide on-demand 5G connectivity for UEs. The open-source software used for the implementation of the 5G architecture was OpenAirInterface, while the hardware used to enable the radio communications between the UE and the gNB was the USRP B210, which uses software for the modulation and demodulation of radio signals.

The proposed Network Function (ODMMF) provides two Web pages so that a Mobile Network Operator can monitor the statistics regarding the radio conditions and control the mobile robotic platform movements, taking advantage of the video feed provided, both in real-time. Our solution enables a Mobile Network Operator to quickly reconfigure the 5G network and restore connectivity for UEs outside the radio coverage provided by the fixed wireless network infrastructure.

An implementation challenge faced was the time spent to get familiar with the OAI software. Since OAI is very complex in terms of code and implementation, it took long time to achieve a fully functional network configuration. OAIs' implementation of 5G SA is recent and an ongoing work by the community, resulting in bugs and malfunctioning. OAI is also very demanding in terms of hardware requirements, which led to multiple improvements on the computing platforms used during the development phase of this dissertation, to mitigate performance issues.

Additional challenges were imposed by the SDRs and the mobile robotic platform (Go1 Edu Robot) used. On the one hand, the SDRs are extremely sensitive and must be dealt with caution, resulting in several problems receiving and delivering samples and leading to software crashing. To address this, several settings in the Operating System and in the BIOS of the host machine had to be fine-tuned, leading to inferior performance, for the sake of a more stable testbed. On the other hand, since the mobile robotic platform is capable of achieving high velocity, its integration and testing had to be done with the necessary prudence, in order to avoid any damage to the platform itself and surrounding obstacles.

Moreover, for implementing ODMMF NF, the *Express.js* framework did not allow us to use HTTP2 as the 5G standard defines. However, with the use of the TLS protocol, the ODMMF is still secure. In addition, since implementing the HTTP status is considerably harder using the *Express.js* framework, when compared to the *Node.js* framework, this is left for future work.

After the implementation was concluded, many tests were defined and carried out to evaluate individually each component of our solution. The first set of tests, related with the evaluation of each network component, allowed us to conclude that each component was working correctly. The second set of tests aimed at ensuring the correct operation of the proposed solution. Finally, the last set of tests aimed at validating a real-world use case.

Although there are still some improvements that can be made to the proposed solution, the main objectives of this dissertation were fully achieved, which resulted in two main original contributions: 1) a 5G SA network, that allows remote control and positioning of a mobile robotic platform that can be used to provide 5G high-speed connectivity for UEs, while also possessing the capability of quickly restoring a lost connection to the Internet; 2) an extension to the 5G architecture to support a mobile Radio Access Network, including a new Network Function (ODMMF) and a new network interface (N_M), which enable monitoring the QoS radio link established with a UE and controlling in real-time a mobile gNB, taking advantage of visual information of the mobile platform.

5.2 Known Limitations & Future Work

Even though the present dissertation achieved all the proposed objectives, there are some known limitations and there is still room for improvements in terms of performance.

The first limitation is the fact that our solution does not identify UEs autonomously. By integrating into ODMMF state of the art algorithms able to detect UEs autonomously, it will allow an MNO to access the mobile robotic platform remotely only for periodic security purposes or specific tasks, reducing the work load. As future work, it would be useful to use a machine learning algorithm capable of identifying UEs and avoid the mobile robotic platform collide with obstacles when looking for UEs to provide wireless connectivity to.

The second known limitation is related to our network architecture not implementing a full 5G SBA. Although the developed NF (ODMMF) was integrated into the Core Network, the selected OAI core network is not fully compliant with the SBA architecture, which enables scalability, independent services, and network slicing with dynamic and efficient resource utilization. In future works, the deployment of a 5G SBA Core Network would be useful, especially in terms of comparing its performance with the current version.

The third limitation is associated with the OpenAirInterface software. When the quality of the radio link is close to the minimum value required for establishing a wireless link, OAI assumes the radio wireless link is always-on, even though it may be unstable. OpenAirInterface creates the tunnel interface (cf. Figure 3.24), assuming the wireless link is established, passing that information to the ODMMF, which mistakenly informs the user that it currently has 5G connectivity. In future work, taking into account that 5G is a recent technology and OpenAirInterface implementation is not mature enough, this problem should be solved.

The fourth limitation is related to the SDR model chosen to enable the radio communication between the UE and the gNB. The USRP B210 is one of the SDR models recommended by OpenAirInterface to deploy a 5G NR network, but it is the model that enables the lowest performance among the ones recommended. Its main limitation is associated with the fact that it uses a USB connection to the host computer, which leads to problems when other USB devices are connected to the same machine. Other USRPs, such as USRP N310 or USRP X310, are considerably more expensive than the USRP B210 but use an Ethernet connection, which enables improved performance. For future work, the use of one of those SDRs is recommended.

References

[1] Reuters. S.Korea First to Roll Out 5G Services, beating U.S. and China. Available from https://www.reuters.com/article/southkorea-5g-idUSL3N21K114, 4 2019.

[2] Alan Seal. Five 5G Statistics You Need to Know. Available from https://www.vxchnge.com/blog/5g-statistics, 04 2020.

[3] Opensignal. South Korea- 5G Experience Report December 2021. Available from https://www.opensignal.com/reports/2021/12/southkorea/mobile-network-experience-5G, 12 2021.

[4] Chai Li and Arda Akman. O-RAN Use Cases and Deployment Scenarios. 02 2020.

[5] Thushan Sivalingam, Maheshi B Dissanayake, Shashika Badalge, and Nandana Rajatheva. Positioning of Multiple Unmanned Aerial Vehicle Base Stations in future Wireless Network. Master's thesis, University of Oulu and University of Peradeniya, 8 2019.

[6] Erik Dahlman, Stefan Parkvall, and Johan Skold. *5G NR: The Next Generation Wireless Access Technology*. Elsevier, 125 London Wall, London EC2Y 5AS, United Kingdom, 2018.

[7] Rohit Mehta. Pros and cons of 5G technology. Available from https://timesofindia.indiatimes.com/blogs/digital-mehta/pros-and-cons-of-5g-technology/.

[8] Kleber Vieira Cardoso, Cristiano Bonato Both, Lúcio Rene Prade, Ciro J. A. Macedo, and Victor Hugo L. Lopes. A softwarized perspective of the 5g networks, 2020. arXiv:2006.10409.

[9] everythingRF. Distributed RAN. Available from https://www.everythingrf.com/community/what-is-a-remote-radio-head.

[10] Artiza Networks. Cloud/Centralized Radio Access Network (C-RAN). Available from https://www.artizanetworks.com/resources/tutorials/cran.html.

[11] Parallel Wireless. Everything you need to know about open ran. Available from https://www.parallelwireless.com/wp-content/uploads/Parallel-Wireless-e-Book-Everything-You-Need-to-Know-about-Open-RAN.pdf.

[12] Telecom Infra Project. OpenRAN. Available from https://telecominfraproject.com/openran/, 2021.

[13] Parallel Wireless. Open RAN Network Software. Available from https://www.parallelwireless.com/products/5g-openran/.

[14] Matt Kapko. Rakuten's US Leader Defends Open RAN Platform. Available from https://www.sdxcentral.com/articles/news/rakutens-us-leader-defends-open-ran-platform/2020/12/.

[15] Nokia. 5G Core (5GC). Available from https://www.nokia.com/networks/portfolio/5g-core/.

[16] Resurchify. 5G NR Tutorial. Available from https://www.resurchify.com/5G-tutorial/5G-tutorial.php.

[17] ETSI, 650 Route des Lucioles. *System architecture for the 5G System (5GS)*, 9 2021.

[18] Wikipedia. List of HTTP status codes. Available from https://en.wikipedia.org/wiki/List_of_HTTP_status_codes.

[19] ETSI, 650 Route des Lucioles. *Technical Realization of Service Based Architecture*, 5 2022.

[20] Free5GC. What is free5GC? Available from https://www.free5gc.org/.

[21] Labora Research Group. IEEE NetSoft2020-Tutorial4-Demo2-Exp2. Available from https://corp.mobile.rakuten.co.jp/english/news/press/2021/0712_01/, 7 2021.

[22] OpenAirInterface. 5G Core Network. Available from https://openairinterface.org/oai-5g-core-network-project/.

[23] Open5GCore. Open5GCore – The Next Mobile Core Network Testbed Platform. Available from https://www.open5gcore.org/.

[24] Open5GS. Open5GS - Open source project of 5GC and EPC. Available from https://open5gs.org/.

[25] Magma. An open source platform for building carrier-grade networks. Available from https://magmacore.com/.

[26] ALİ GÜNGÖR. Ueransim. Available from https://github.com/aligungr/UERANSIM.

[27] Free5G. free5gran. Available from https://github.com/free5G/free5GRAN.

[28] srsRAN. The srslte project is evolving. Available from https://www.srslte.com/srslte-srsran.

[29] Boston Dynamics. Spot. Available from https://www.bostondynamics.com/products/spot.

[30] Unitree. Unitree Go1. Available from https://www.unitree.com/products/go1.

[31] Dji. Phantom 2. Available from https://www.dji.com/pt/phantom-2.

[32] André Coelho, Nuno Almeida, Pedro Silva, José Ruela, Rui Campos, and Manuel Ricardo. Redefine: Centralized routing for high-capacity multi-hop flying networks. *2018 Eleventh International Workshop on Selected Topics in Mobile and Wireless Computing*, 2018.

[33] INESC TEC. Wise. Available from http://wise.inesctec.pt/.

[34] André Coelho, Hélder Fontes, Rui Campos, and Manuel Ricardo. Traffic-aware Gateway Placement for High-capacity Flying Networks. *2021 IEEE 93rd Vehicular Technology Conference*, 2021.

[35] Lorenzo Bertizzolo, Tuyen X. Tran, John Buczek, Bharath Balasubramanian, Rittwik Jana, Yu Zhou, and Tommaso Melodia. Streaming from the Air: Enabling Drone-sourcedVideo Streaming Applications on 5G Open-RAN Architectures. 11 2021.

[36] Mariam Ishtiaq, Nasir Saeed, and Muhammad Asif Khan. Edge Computing in IoT: A 6G Perspective. 11 2021.

[37] Samira Hayat, Evsen Yanmaz, and Raheeb Muzaffar. Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint. *IEEE COMMUNICATIONS SURVEYS TUTORIALS, VOL. 18, NO. 4, FOURTH QUARTER 2016*, 2016.

[38] Free5gc. Free5gc support the SDR device? Available from https://forum.free5gc.org/t/free5gc-support-the-sdr-device/551.

[39] OpenAirInterface. TESTING 5G SA Setup. Available from https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/doc/TESTING_5GSA_setup.md.

[40] OpenAirInterface. OpenAirSystemRequirements. Available from https://gitlab.eurecom.fr/oai/openairinterface5g/-/wikis/OpenAirSystemRequirements.

[41] OpenAirInterface. 5G NR Development And Setup. Available from https://gitlab.eurecom.fr/oai/openairinterface5g/-/wikis/5g-nr-development-and-setup.

[42] Farnell. TG.30.8113. Available from https://pt.farnell.com/taoglas/tg-30-8113/dipole-antenna-3-4-3-6ghz-3-6dbi/dp/2919859?st=tg.30.8113.

[43] rdefosseoai. rdefosseoai. Available from https://hub.docker.com/u/rdefosseoai.

[44] OpenAirInterface. OpenAirInterface 5G Core Network Deployment using Docker-Compose and Testing with dsTest. Available from https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed/-/blob/v1.1.0/docs/DEPLOY_SA5G_WITH_DS_TESTER.md.

[45] OpenAirInterface. OpenAirInterface5G. Available from https://gitlab.eurecom.fr/oai/openairinterface5g/-/tree/USRP_B210_DLSCH_Improvements.

[46] Ettus Research. UHD. Available from https://kb.ettus.com/UHD.

[47] Code for Geek. Render HTML file in Node.js and Express.js framework. Available from https://codeforgeek.com/render-html-file-expressjs/.

[48] Loophole. Loophole. Available from https://loophole.cloud/.

[49] W3Schools. Window location.reload(). Available from https://www.w3schools.com/jsref/met_loc_reload.asp.

[50] OpenAirInterface. 5g nr development and setup. Available from https://gitlab.eurecom.fr/oai/openairinterface5g/-/wikis/5g-nr-development-and-setup.

[51] Tech Playon. 5G NR Numerology – Subcarrier Spcaing (SCS). Available from https://www.techplayon.com/5g-nr-numerology-subcarrier-spcaing-scs/.

[52] Wikipedia. Iperf. Available from https://en.wikipedia.org/wiki/Iperf.

[53] Rúben Queirós. Multi-technology Flying Access Network for Disaster Management Scenarios. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2020.