

Implementation and Comparative Analysis of Open 5G Standalone Testbeds: A Systematic Approach

by

Maryam Amini

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2024

© Maryam Amini 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Florian Kaltenberger
 Professor, Dept. of Communication Systems
 EURECOM

Supervisor(s): Catherine Rosenberg
 Professor, Dept. of Electrical and Computer Engineering
 University of Waterloo

Internal Member: Amir Keyvan Khandani
 Professor, Dept. of Electrical and Computer Engineering
 Mahesh Tripunitara
 Professor, Dept. of Electrical and Computer Engineering
 University of Waterloo

Internal-External Member: Lukasz Golab
 Professor, Dept. of Management Science and Engineering
 University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contribution

This thesis contains some material that was taken from multi-author papers, in which I am the first author. These papers include:

- M. Amini, A. El-Ashmawy and C. Rosenberg, “Implementing an Open 5G Standalone Testbed: Challenges and Lessons Learnt,” IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)
- M. Amini, A. El-Ashmawy, C. Rosenberg and A. K. Khandani, “5G DIY: Impact of Different Elements on the Performance of an E2E 5G Standalone Testbed,” GLOBE-COM 2023 - 2023 IEEE Global Communications Conference
- M. Amini, and C. Rosenberg. “A Comparative Analysis of Open-source Software in an E2E 5G Standalone Platform,” WCNC 2024 - 2024 IEEE Wireless Communications and Networking Conference
- M. Amini, and C. Rosenberg. “Performance Analysis and Comparison of Full-Fledged 5G Standalone Experimental TDD Testbeds in Single & Multi-UE Scenarios.” arXiv preprint arXiv:2407.02341 (2024).

Abstract

Open-source software and commercial off-the-shelf hardware are finally paving their way into the 5G world, resulting in a proliferation of experimental 5G testbeds. Surprisingly, very few studies have been published on the comparative analysis of testbeds with different hardware and software elements.

This dissertation is a comprehensive study on the implementation of experimental 5G testbeds and the challenges associated with them. We first introduce a precise nomenclature to characterize a 5G-standalone single-cell testbed based on its constituent elements and main configuration parameters. We then build 36 distinct such testbeds and systematically analyze and compare their performance with an emphasis on element interoperability, as well as the number and type of User Equipment (UE), to address the following questions: 1) How is the performance (in terms of bit rate and latency) impacted by different elements? 2) How does the number of UEs affect these results? 3) What is the impact of the user(s)' location(s) on the performance? 4) What is the impact of the UE type on these results? 5) How far does each testbed provide coverage? 6) What is the impact of the available computational resources on the performance of each open-source software? Finally, to illustrate the practical applications of such open experimental testbeds, we present a case study focused on user scheduling. Historically, most research on user scheduling has been conducted using simulations, with a strong emphasis on downlink scheduling. In contrast, our study is fully experimental, and targets enhancements to the uplink scheduler within the open-source Radio Access Network platform, *srsRAN*. We aim to move beyond simulation-based evaluations and explore how improvements in the uplink scheduler translate to real-world performance, specifically by measuring their impact on the user experience in a live experimental testbed.

Acknowledgements

First of all, I must thank my supervisor Professor Catherine Rosenberg, for her support, inspiration, and trust in letting me work on the fascinating experimental aspects of cellular networks. I never imagined something could be so engaging, yet so frustrating to work with. I'm incredibly grateful for this opportunity. Beyond her guidance in my work, she has also taught me so much on a personal level. Thank you, Catherine!

I would also like to extend my sincere appreciation to Professor Amir K. Khandani for his invaluable support in expanding our testbed by providing us hardware and, most importantly, for generously giving his time to discuss challenges and offer helpful suggestions. Also many thanks to my committee members, Professor Florian Kaltenberger, Professor Mahesh Tripunitara, and Professor Lukasz Golab for their time to review this thesis and their valuable comments.

I'd like to acknowledge that implementing an experimental testbed (let alone 36 of them) is not an easy task by any means. This project would have never been completed without the help of my colleague Ahmed El-Ashmawy, Professor Amir Khandani, Professor George Shaker, Eric Praetzel, and the 5G open-source community, especially the srsRAN team, and Raymond Knopp from the OAI community. I do appreciate all your help. Also, I would like to extend my thanks to my fellow collaborator, Haseen Rahman Cheriyaandilakath, for all the 6 AMs, midnights, and weekends we spent conducting the tests.

Thank you to all the amazing people I got to know in Waterloo, my friends and colleagues in the lab: Yuhao, Euan, Paulo, Samir, Kiril, Ahmed, Ahmad, Abdallah, Yekta, Ara, Jonah, Andrew, and Maryam, for all the great memories. I'd also like to mention a few close friends who have been a blessing during the past few years: Samira, Shamisa, Behshid, and Sasha. I would like to take a moment to appreciate my friendship with Sougol, who has always been supportive and there for me throughout all these years. I'm truly grateful to have you in my life.

Last but not least, I would like to thank my family, my dear parents and my baby sister, Yasaman. I have been fortunate to have the best sister in the world. Being only 11 months apart, she has been through it all with me, she is my source of encouragement. Simply put, she is truly awesome. Thank you Aj! Also, I must thank my parents to be such great role models, always believing in me, and for their core values in life. Needless to say, without their love and support, this journey would have not been possible.

"In the black soil emerging from under the ice, the small patch of snow always appears more alluring." - And Quiet Flows the Don - Mikhail Sholokhov.

Dedication

To my loving grandparents...

Table of Contents

Examining Committee	ii
Author's Declaration	iii
Statement of Contribution	iv
Abstract	v
Acknowledgements	vi
Dedication	vii
List of Figures	xii
List of Tables	xiv
List of Abbreviations	xv
1 Introduction	1
1.1 The Movement Towards Openness	2
1.2 System Model & Nomenclature	4
1.3 Research Questions and Contributions	6
1.4 Thesis Outline	9

2	Background & Literature Review	10
2.1	The RAN Evolution	10
2.1.1	D-RAN	10
2.1.2	C-RAN	11
2.1.3	x-RAN	11
2.1.4	Open RAN	12
2.1.5	Open RAN vs. O-RAN	13
2.1.6	5G New Radio Protocol Stack	13
2.2	The Core Evolution	15
2.3	Literature Review	17
2.3.1	Targeted Studies on the performance of a single testbed	17
2.3.2	Comparative studies on the performance of multiple testbeds	18
3	Implementation	22
3.1	Elements Under-Study	25
3.1.1	The RAN	25
3.1.2	The Core	28
3.1.3	The UEs	30
3.1.4	Miscellaneous	31
3.2	Testbeds Under-Study	31
3.3	Implementation Challenges	34
3.3.1	Hardware-related Challenges	35
3.3.2	Software-related Challenges	36
3.4	Performance Metrics	36
3.4.1	Data Rate	36
3.4.2	Latency	37
3.4.3	Coverage	37
3.4.4	Computational resource consumption	38

3.5	Test Scenarios, Methodology and Results for Open-source 5G software Interoperability Assessments	38
3.5.1	Methodology & Results for Interoperability Assessment	40
3.5.2	Methodology & Results for Data Rate and Latency Assessment in interoperability tests	42
3.6	Test Scenarios, Methodology and Results for the RAN Platforms	44
3.6.1	Methodology & Results for Data Rate Assessment	45
3.6.2	Methodology & Results for Latency Assessment	50
3.6.3	Methodology & Results for Coverage Assessment	52
3.6.4	Methodology & Results for Computational Resource Consumption of the Open-source RAN Software	53
3.7	Conclusion	57
4	Uplink Scheduling	59
4.1	Overview of 5G's Physical Channels	60
4.2	PRB, SINR, MCS, and the Rate	60
4.3	Scheduling Process as Specified by 3GPP	64
4.4	Power Control	65
4.5	Uplink Scheduling in the Literature	67
4.6	UL Scheduling in srsRAN	70
4.6.1	Round Robin	70
4.6.2	Proportional Fairness	72
4.7	Single-UE Case: Allocating the right number of PRBs per time slot	73
4.8	Multi-UE Case: Allocating the right number of PRBs to multiple UEs per time slot	74
4.9	Experimental 5G-SA Setup	75
4.10	Methodology	78
4.11	Implementation Challenges	80
4.11.1	Challenges in conducting UL tests with steady SINR measurements	80

4.11.2 Challenges in PRB Allocation	83
4.12 Test Scenarios, and Results for single-UE Case	83
4.13 Test Scenarios, and Results for Two-UE Case	85
4.14 Conclusion	86
5 Conclusions	87
5.1 Future Research Directions	88
References	90

List of Figures

1.1	Typical cellular network architecture	1
1.2	5G-NSA vs. 5G-SA architecture	3
1.3	Single-cell 5G-SA Testbed Design	5
2.1	Simplified D-RAN architecture	11
2.2	Simplified C-RAN architecture	12
2.3	Simplified 3GPP Open RAN architecture	13
2.4	5G NR Protocol Stack	14
2.5	3rd Generation Partnership Project (3GPP) functional split of the 5G protocol stack	15
2.6	5G system architecture	16
3.1	Our 5G-SA platform	22
3.2	Our 5G-SA platform architecture	23
3.3	srsRAN gNB architecture [20]	26
3.4	OAI gNB architecture [15]	26
3.5	USRP X410 [4]	27
3.6	USRP B210 [4]	27
3.7	Open5GS 5GC architecture [16]	29
3.8	OAI-5GC architecture [15]	30
3.9	Quectel 5G modem (RM502Q-AE)	31

3.10	Simplified 5G-NR frame structure	33
3.11	Simplified 5G-SA architecture to illustrate interfaces $N2$ and $N3$	40
3.12	Map indicating the positions where tests were conducted	45
3.13	CPU utilization as a function of number of UEs for OAI and srsRAN in UL and DL	55
4.1	Scheduling flowChart	64
4.2	Achievable rate by Shannon vs. Modulation and Coding Scheme (MCS) function in linear and dB scales	69
4.3	Sample srsRAN log file excerpt	77
4.4	Map indicating the positions where scheduling enhancement assessments tests were conducted	79
4.5	Sample of a good SINR measurement	81
4.6	Sample of a bad SINR measurement	82

List of Tables

1.1	Open Software and Hardware Available for 5G-SA	7
2.1	A Summary of the Literature on 5G-Standalone (5G-SA) testbeds	21
3.1	Our 5G-SA platform elements (We use the same color code as in Fig. 3.2) .	24
3.2	Testbeds implemented for this thesis	32
3.3	Number of OFDM symbols per slot, slots per frame, and slots per subframe for normal cyclic prefix	34
3.4	Parameters of the Open-source Software for each testbed	41
3.5	Downlink and Uplink data rate and latency results for interoperability tests	43
3.6	Downlink and uplink data rate results for different tests corresponding to different locations and S_{RAN} (in Mbps)	47
3.7	Data rate results for two SDRs (in Mbps)	51
3.8	E2E latency between the UE and 5GC (in ms)	52
3.9	Signal strength measurements (in dBm)	54
3.10	The effect of host PC computational power	56
4.1	Maximum transmission bandwidth configuration for Frequency Range 1 (FR1)	61
4.2	The uplink MCS-SINR (64 QAM) mapping used in srsRAN	63
4.3	Elements and sub-elements of testbed $\mathcal{T}_{36}$	76
4.4	Measurements results done in location B1	84
4.5	Measurements results done at location B2	85

List of Abbreviations

3GPP 3rd Generation Partnership Project

5G-NSA 5G-Non Standalone

5G-SA 5G-Standalone

5GC 5G Core Network

AGPLv3 GNU Affero General Public License version 3

AMF Access and Mobility Management Function

ARFCN Absolute Radio Frequency Channel Number

AUSF Authentication Server Function

BBU Base Band Unit

BLER Block Error Rate

BS Base Station

BSF Binding Support Function

C-RAN Centralized Radio Access Network

CLPC Closed Loop Power Control

CO Central Office

COST Commercial Off-the-Shelf

CP Control Plane

CQI Channel Quality Indicator

CSI Channel State Information

CU Central Unit

D-RAN Distributed Radio Access Network

DL Downlink

DU Distributed Unit

E2E End-to-end

EPC Evolved Packet Core

EVM Error Vector Magnitude

FDD Frequency-Division Duplexing

FR1 Frequency Range 1

FR2 Frequency Range 2

GbE Gigabit Ethernet

gNB Next-Generation NodeB

GPSDO GPS Disciplined Oscillator

KPI Key Performance Indicator

LTE Long Term Evolution

MAC Medium Access Control

MCS Modulation and Coding Scheme

ML Machine Learning

MNO Mobile Network Operator

NAS Non Access Stratum

NF Network Function

NGAP Next Generation Application Protocol

NRF NF Repository Function

NSSF Network Slice Selection Function

NWDAF Network Data Analytics Function

O-RAN Open-Radio Access Network

OAI OpenAirInterface

OLLA Outer Link Layer Adaptation

OLPC Open Loop Power Control

OS Operating System

OTA Over-the-air

PBCH Physical Broadcast Channel

PCF Policy and Charging Function

PDCCH Physical Downlink Control Channel

PDCCP Packet Data Convergence Protocol

PDSCH Physical Downlink Shared Channel

PDU Packet Data Unit

PF Proportional Fairness

PHR Power Headroom Report

PHY Physical Layer

PLMN Public Land Mobile Network

PRACH Physical Random Access Channel

PRB Physical Resource Block

PUCCH Physical Uplink Control Channel

PUSCH Physical Uplink Shared Channel

QoS Quality of Service

QSFP28 Quad Small Form-factor Pluggable 28

RAN Radio Access Network

RF Radio Frequency

RL Reinforcement Learning

RLC Radio Link Control

RR Round Robin

RRC Radio Resource Control

RRH Remote Radio Head

RRU Remote Radio Unit

RSRP Reference Signal Received Power

SCP Service Communication Proxy

SCS Subcarrier Spacing

SDAP Service Data Adaptation Protocol

SDR Software-Defined Radio

SEPP Security Edge Protection Proxy

SINR Signal to Interference Noise Ratio

SMF Session Management Function

TDD Time-Division Duplexing

TPC Transmit Power Control

UDM Unified Data Management

UDR Unified Data Repository

UE User Equipment

UHD USRP Hardware Driver

UL Uplink

UP User Plane

UPF User Plane Function

URLLC Ultra-reliable Low Latency Communications

USRP Universal Software Radio Peripheral

x-RAN extensible Radio Access Network

Chapter 1

Introduction

This thesis is on the implementation, and performance analysis of 36 open experimental testbeds for the Fifth Generation (5G) of cellular networks, with a focus on interoperability and multi-user scenarios. We only consider the standalone mode of 5G (5G-SA). We propose a clear and consistent methodology, to describe, analyze, and compare these testbeds. Additionally, as a demonstration of the practicality and flexibility of such experimental testbeds, we focus on the user scheduling mechanism on the Uplink (UL) direction, for one of the testbeds, providing detailed insights into its performance.

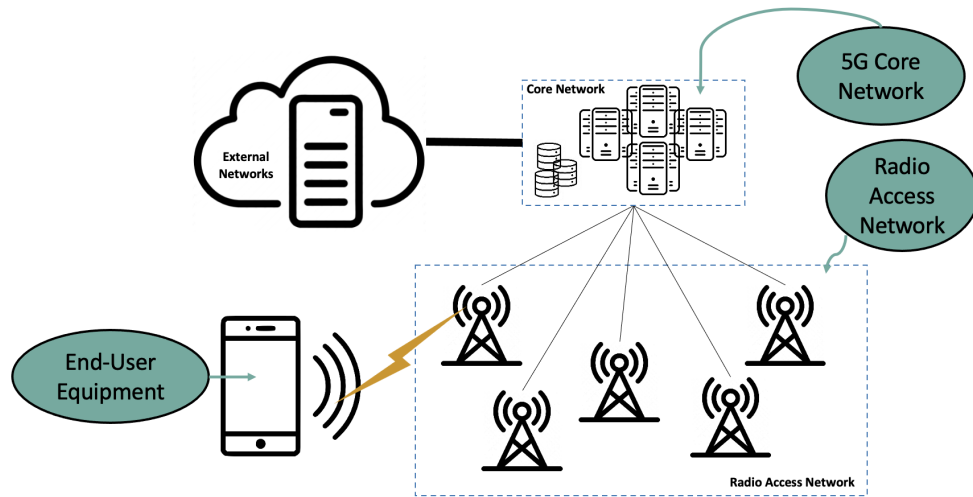


Figure 1.1: Typical cellular network architecture

The main elements of a cellular network, as illustrated in Fig. 1.1, include the Radio Access Network (RAN), the 5G Core Network (5GC), and the User Equipment (UE) devices. UE devices can vary in type (mobile devices, sensors, wearable devices, etc.), but they all have one similarity: they seek connections to the cellular network through their SIM card.

The RAN consists of at least one Base Station (BS). It connects UE devices to the core network through the wireless medium. Depending on the network generation, the BS is referred to by different names. In Long Term Evolution (LTE) networks, it is called evolved Node B (eNodeB), and in 5G networks, it is known as Next-Generation NodeB (gNB).

Finally, the core network is responsible for processing data and providing services to UEs, including connectivity to external networks such as the Internet. It ensures that all connection requirements are met, tracks subscriber usage for billing purposes, and manages overall network operations.

1.1 The Movement Towards Openness

Prior to the 5G era, Mobile Network Operators (MNOs) were struggling with the limitations of traditional single-vendor cellular network infrastructure. Nearly all hardware and software products available were proprietary black-box solutions, meaning that once an MNO opted for a specific vendor, they had very little flexibility. This vendor lock-in came with a lot of challenges, including, higher costs, and lack of competition, all of which slowed advancements in the industry.

Recognizing the need for faster advancements in cellular networks, 3GPP made a bold shift in its approach to standard design. With the release of the first technical reports for 5G standards in 2018, a new paradigm was introduced. *Openness* and *disaggregation* became two of the core principles of 5G networks. Both the RAN, and the core network went through significant architectural transformations. The traditional *monolithic* RAN was replaced by an *OpenRAN*, a modular architecture where each component is connected through well-defined, open interfaces. As a result, the industry experienced a surge of innovations and activities.

This drastic change was not only crucial for the telecom industry, but it also helped the academic community become more relevant by building full-fledged networks, instead of solely relying on simulated studies. The shift towards openness brought about a plethora of activities to build open-source software and white-box hardware, empowering researchers

to implement their experimental testbeds to evaluate their algorithms, rather than relying on simulations.

The migration from LTE to 5G is not straightforward, as both the RAN and core have been significantly changed. Indeed, all LTE base stations and most of the core network should be replaced for a cellular network to be full-fledged 5G. Consequently, MNOs have opted to transition to 5G in two phases, first from LTE to 5G-Non Standalone (5G-NSA) and then from 5G-NSA to 5G-SA. As illustrated in Fig. 1.2, in 5G-NSA, the core and control plane are LTE-based, while the user plane follows 5G standards. This enables MNOs to integrate 5G base stations into their existing LTE network to handle the data plane and gradually transition their networks to a complete 5G-SA setup. In this thesis we are only focused on the standalone mode of 5G cellular networks, where both the data and control planes are 5G.

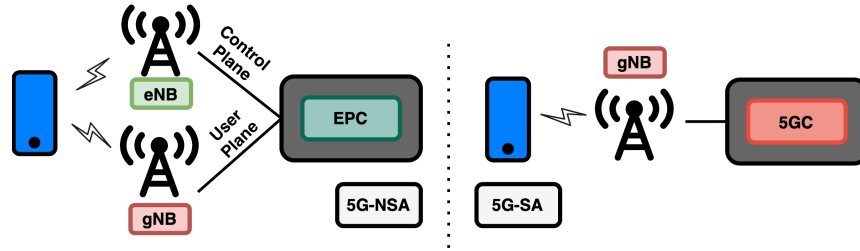


Figure 1.2: 5G-NSA vs. 5G-SA architecture

Over the past couple of years, several open-source software, and Commercial Off-the-Shelf (COST) hardware elements have been released for 5G-SA cellular networks. Although some performance studies have been conducted, there is still a significant gap in evaluating the performance of these open solutions on operational 5G-SA testbeds. More importantly, the topic of interoperability remains overlooked. Most of the testbeds evaluated in the literature so far, utilize elements from only a single vendor (“single-source” testbeds), or from few vendors (“multi-source” testbeds), typically those known to be compatible. However, interoperability should be inherent across all components. Therefore, although open-source software solutions and white-box hardware are expected to follow 3GPP standards, there is a need for studies on the element interoperability, considering different combinations of hardware and software for each element from various vendors.

Moreover, one of the key factors affecting the performance experienced by the UEs is their environment, including their location. The study of performance in experimental testbeds gets even more complicated, in the presence of multiple UEs. These aspects have

not been explored in the literature. The few studies that focus on experimental 5G often aim to create ideal conditions for the UE (e.g., close proximity to the RAN).

This dissertation is a study on the implementation of 5G-SA multi-source testbeds and the challenges associated with them. It proposes a clear methodology to make the study of these testbeds consistent and repeatable. It provides comprehensive insights into the performance achieved by UEs in various scenarios, including multiple locations and single/multi-UE setups. Additionally, to showcase the practicality of 5G-SA testbeds, we have focused on the user scheduling algorithm in the UL, as one of the most important resource management processes in any cellular networks.

To explain our contributions, and to provide some structure and consistency in the way we describe and build testbeds, we first present the necessary context. Therefore, a precise nomenclature to characterize a 5G-SA testbed is provided next. This is followed by an overview of the architecture of a typical 5G-SA testbed. Finally, we outline the main research questions, contributions, and the structure of the thesis.

1.2 System Model & Nomenclature

We define a single-cell, 5G-SA experimental testbed containing n UEs, by the set $\mathcal{T}$ of its software and hardware elements (please see (1.1)) and a set $\mathcal{C}$ containing the configuration parameters (please see (1.2)).

$$\mathcal{T} = \{S_{5GC}, H_{5GC}, S_{RAN}, H_{RAN}, (S_{UE_1}, H_{UE_1}), \dots, (S_{UE_n}, H_{UE_n})\} \quad (1.1)$$

$$\mathcal{C} = \{b, B, T\} \quad (1.2)$$

Where:

- S_{5GC} (resp. S_{RAN} and S_{UE_i}) is the collection of software sub-elements for 5GC (resp. RAN and the i -th UE), and H_{5GC} (resp. H_{RAN} and H_{UE_i}) is the collection of hardware sub-elements for 5GC (resp. RAN and the i -th UE).
- b is the band central frequency used by the system, B is the bandwidth, and T is the transmission mode between the gNB, and the UE (wired or wireless). Kindly note that, the value of b has a one to one mapping to the duplexing mode, i.e., Time-Division Duplexing (TDD) or Frequency-Division Duplexing (FDD).

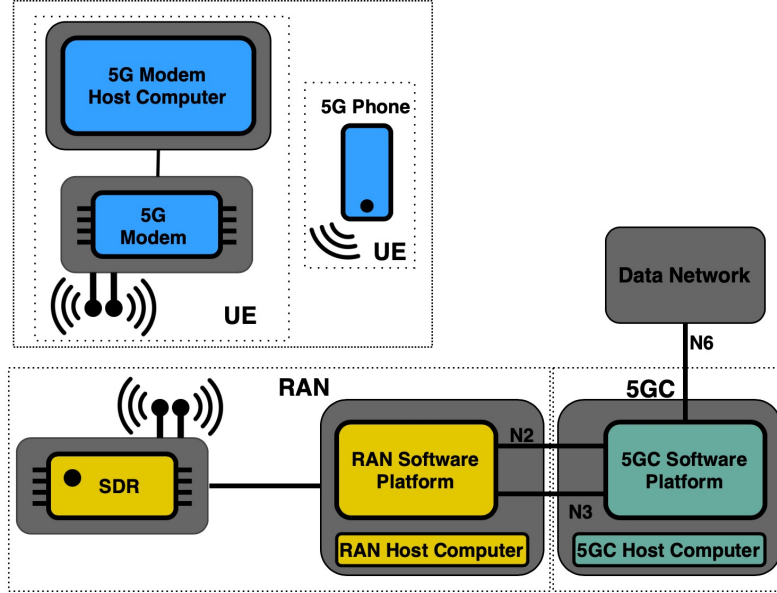


Figure 1.3: Single-cell 5G-SA Testbed Design

Fig. 1.3 shows the different elements of a 5G-SA testbed. Each element, i.e., the 5GC, RAN, and UE(s), can be decomposed into hardware and software sub-elements. The key software sub-element of the RAN includes the open-source platform that runs the 5G protocol stack. The hardware sub-elements consist of a Software-Defined Radio (SDR) device equipped with an antenna system. The SDR is a critical part of the RAN, as it hosts as the Radio Frequency (RF) front-end. It is connected to a computer that executes a software platform that handles the remaining of the 5G protocol stack on the RAN. Recently, there has been an increase in the number of available COST SDR devices. Currently, the three major SDR vendors are: *Ettus Research* [4], *Lime Microsystems* [9], and *Nuand* [11]. As illustrated in Fig. 1.3, the SDR gets connected to the computer hosting the S_{RAN} through a wired connection. It exchanges radio samples with the RAN software platform using a driver installed on the host computer.

In the past couple of years, several open-source software platforms have been proposed for the 5GC, the RAN and the UE. The two predominant open-source RAN solutions for running the 5G protocol stack (except for the RF), are from *srsRAN* [20] and *OpenAirInterface (OAI)* [15]. They are the most widely used and trusted platforms available for the software part of the RAN. Moreover, 5GC solutions developed by OAI [15], *Open5GS* [16] and *free5GC* [6], have gained significant popularity among researchers. Each of these solu-

tions supports different sets of 5GC Network Functions (NFs). However, they all contain the essential NFs required to implement an End-to-end (E2E) experimental testbed with basic functionality. These necessary NFs are: Access and Mobility Management Function (AMF), Session Management Function (SMF), User Plane Function (UPF), Unified Data Management (UDM), Unified Data Repository (UDR), Authentication Server Function (AUSF), and NF Repository Function (NRF), ensuring UE registration, authentication, Packet Data Unit (PDU) session establishment and management, and Non Access Stratum (NAS) security.

Lastly, on the UE front, there are a few possible options. The most obvious one is to use a smart phone. Another option is to use a computer to host some of the 5G protocol stack on the UE-side. First, a computer gets connected to a 5G modem that acts both as an RF front-end and as a host for the lower layer protocols (up to Layer 3). Another computer-based UE option is to use an SDR as an RF front-end while the computer hosts S_{UE} (both *srsRAN* and *OAI* offer UE-based software platforms). So far, 5G modems have proven to be more convenient to work as UEs, than 5G-SA phones.

For completeness, we note that another sub-element of the UE is the SIM card. The use of a programmable SIM card enables the modification of authentication information on the UEs, based on the 5GC's authentication requirements.

Table 1.1 provides several available options for each of the elements of a 5G-SA testbed. The highlighted elements/sub-elements are those analyzed in this thesis (please note that an in-depth description of these elements is presented in Chapter 3).

1.3 Research Questions and Contributions

This thesis is dedicated to 5G-SA testbeds and their constituent elements. In this thesis, we consider and build 36 distinct testbeds, varying in the software/hardware elements of the 5GC, RAN, as well as the number and type of connected UEs. Prior work on 5G-SA open testbeds focus on studying the performance of single-UE scenarios in terms of bit rate and latency, with occasional consideration given to coverage. Very few address interoperability, and to the best of our knowledge, none addresses multi-UE scenarios.

The few work in this domain try to understand and optimize the best performance achieved by a single-UE using experimental 5G-SA testbeds. The study of a single-UE scenario is crucial as it helps verify the capacity, and maximum achievable performance of

Table 1.1: Open Software and Hardware Available for 5G-SA

5GC	Open5GS		Open-source 5G core network based on 3GPP Rel.17. Supporting both 5G standalone and non-standalone modes.
	Free5GC		Open-source 5G core network based on 3GPP Rel.15.
	OAI-5GCore		Open-source 5G core network based on 3GPP Rel.16.
RAN (Software)	srsRAN-Project		Open-source 5G RAN.
	srsRAN-4G		Open-source LTE/5G RAN, provides a monolithic gNB.
	OAI-RAN		Open-source 5G RAN.
RAN (Hardware)	USRP	USRP B210 USRP X410	SDR platforms developed by Ettus Research.
	BladeRF		SDR platforms developed by Nuand.
	LimeSDR		SDR platforms developed by Lime Microsystems.
UE	COTS Phone		e.g., Google Pixel 5, 6 , 6a, 7, Huawei P40, OnePlus Nord, OnePlus 10
	COTS Modem		Quectel 5G modems, Sierra 5G modems
	SDR-Based UE		Software-based UEs with SDRs as their RF front-end. e.g., srsRAN-UE, OAI-UE

a single UE. The study on multi-UE scenarios is overlooked in the literature. Moreover, no prior work considers performance scenarios beyond the best achievable rates or the lowest achievable latency, which raises a critical question: If a specific software or hardware element is selected based on its best performance, can it guarantee to remain the best choice under different conditions? For example, if a RAN software is chosen for its superior performance with a single-UE in good channel conditions, will it still be the best choice when multiple UEs are connected, including some that are far from the gNB?

This work aims to fully explore the capabilities of these experimental platforms and assess the interoperability of components developed by different sources. Specifically, in Chapter 3, we:

- Report on challenges faced and the lessons learned during the implementation process of several “*single-source*”, and “*multi-source*” single-cell 5G-SA testbeds.
- Studied open-source 5G software interoperability. While every open-source software project should inherently be designed to facilitate interoperability, there remains a need to verify if interoperability is possible, easy and whether multi-source testbeds perform well.
- Provided a comparative analysis of two 5GC open-source software platforms, namely,

Open5GS and *OAI5GC* for two different open-source RAN software, and two configuration sets.

- Built and studied 36 single-cell 5G-SA testbeds, each differing by the combination of RAN elements (software and SDR) and number and types of UE(s) being used. We evaluated these testbeds from an interoperability perspective as well as from a performance standpoint, using well-defined qualitative and quantitative metrics, including data rate, latency, and coverage.
- Explored the multi-UE case for different locations systematically.
- Evaluated the computational resource consumption of each the RAN software platform as the number of connected UEs increases, by changing the PC on which the software platforms are hosted.

Next, to demonstrate the practical applications of open-source 5G-SA testbeds, we focus on “*user scheduling*” as a case study. The purpose of user scheduling in cellular networks is to allocate the radio resources to users in an efficient and fair manner, in both UL and Downlink (DL) directions [65]. This is a process that must be performed very quickly, since it is called many times within a frame.

To perform scheduling, information such as the transmitter’s power budget, the Channel State Information (CSI), and details about the configuration of the gNB is required. User scheduling in the UL differs from that in the DL. In the DL, gNB is the only transmitter, and only its power budget needs to be considered. In the UL though, many UEs might send data in a time slot, which makes UL user scheduling more challenging.

To date, almost all of the research on user scheduling has been evaluated through simulations, with majority of studies focusing on DL scheduling. In contrast, our use case targets enhancing the UL scheduler implemented in the open-source RAN software platform, *srsRAN*.

Our goal was to improve the resource allocation process to enhance the achievable data rate of the users. We argue that UEs are not greedy in terms of number of Physical Resource Blocks (PRBs), i.e., the gNB should not allocate a single UE the maximum number of PRBs in a time slot. Indeed, in distant locations they benefit more from being allocated fewer resource blocks. So that the per-PRB power is higher. We successfully validated this on a single-UE 5G-SA testbed using *srsRAN*. However, as discussed later in Chapter 4, due to unexpected behavior of the testbed, we were only able to validate our approach in single-UE scenarios, and the validation for multi-UE scenarios could not be completed. The list of contributions for Chapter 4 is as follows:

- Studied and explained srsRAN’s scheduler (its open-source nature is not a guarantee that the software is written in a well annotated way. In fact, we needed to identify multiple interconnected files affecting its performance, which made it more of a “*grey box*” for us).
- Code modification to enable the scheduler to allocate different numbers of PRBs to different UEs.
- An experimental validation of our claim, “*the UL scheduler should not allocate too many PRBs to a UE.*”
- An attempt at conducting experiment with a different scheduler on multi-UE, which failed because srsRAN yielded inconsistent SINR values.

1.4 Thesis Outline

This thesis is structured as follows: Chapter 2 presents an in-depth study of the related works on 5G-SA testbeds, their implementation and performance studies. Next, Chapter 3 reviews the components and implementation of a 5G-SA testbed, detailing the challenges faced. It assesses the interoperability of various open-source 5GC and RAN software and compares different 5GC, RAN solutions, and SDR options. The chapter also evaluates how these choices affect testbed performance in multi-UE scenarios and different UE locations, referencing our publications [37, 38, 39, 40]. In Chapter 4, we delve into the RAN uplink scheduler, focusing on its implementation in the Medium Access Control (MAC) layer of srsRAN project. We analyze the existing scheduler, we proposed a more efficient approach to PRB allocation, and validated it partially. Finally, Chapter 5 concludes this thesis.

Chapter 2

Background & Literature Review

This chapter begins with a brief history of the evolution of the RAN, and core networks in different generations of cellular networks. This explains how the current RAN/5GC architectures were shaped. Next, we provide a comprehensive overview of the existing literature related to experimental 5G testbeds. While there are a number of publications on this subject, this chapter deliberately narrows its focus to papers featuring full-fledged experimental testbeds equipped with all essential elements for an operational functionality mode, rather than relying on simulations. Specifically, a full-fledged testbed is defined as one where the interface between the UE and the RAN is wireless, and includes all the necessary software and hardware elements required for the UE to establish a connection to the Internet.

2.1 The RAN Evolution

2.1.1 D-RAN

A traditional base station, as shown in Figure 2.1, was typically divided into two main components: the Remote Radio Unit (RRU) or Remote Radio Head (RRH), responsible for the RF front-end; and the Base Band Unit (BBU), which manages the digital signal processing, such as OFDM modulation, and connects to the core network through the backhaul. This setup is known as a Distributed Radio Access Network (D-RAN). In this setting, almost all of the hardware and software are proprietary and sourced from a single vendor.

In this configuration, the BBU was installed in a very close proximity to the antenna, requiring a shelter with air conditioning and power supply to house the equipment, which adds complexity and increases the cost of operating the cells. Moreover, the one-to-one RRU/BBU pairing was very inefficient, as the BBUs were frequently under-utilized, wasting resources.

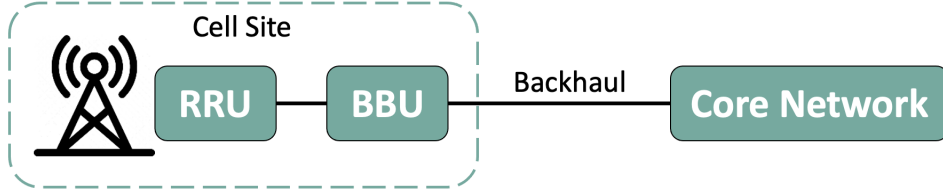


Figure 2.1: Simplified D-RAN architecture

2.1.2 C-RAN

With the increase in network data traffic in the early 2000s, more network development and more base station installation were needed to meet the UEs' new requirements. With such a rapid network deployment, energy concerns were raised. For instance, China's mobile power consumption was doubled as the number of base stations grew from 200,000 to 500,000 from 2005 to 2010 [50].

With the advent of virtualization and cloud networking, to better utilize the resources, BBUs were migrated to a new location called the Central Office (CO) to create a shareable baseband pool. This new design of the access network was called Centralized Radio Access Network (C-RAN). Figure 2.2 illustrates a C-RAN design, where the base stations are split into the RRUs, still located in the cell site and the BBUs, now located in a remote location. This split brought multiple advantages like better utilization of the BBU processing power, energy efficiency, and eliminating the need for leasing a cabinet space near every cell tower [55].

2.1.3 x-RAN

Moving on to 5G networks, MNOs recognized the need for virtualized, cloud-based RAN architectures that offer greater flexibility in selecting vendors and improved visibility into their network operations. In 2016, the extensible Radio Access Network (x-RAN) forum was founded by operators like *AT&T*, *Verizon*, *Deutsche Telekom*, *KDDI*, *NTT Docomo*,

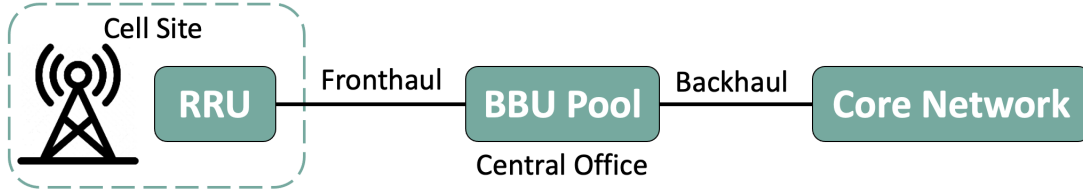


Figure 2.2: Simplified C-RAN architecture

SK Telecom, and *Telstra* to standardize key elements of the RAN and develop new specifications.

The x-RAN forum’s primary focus was set to “achieving interoperability in the fronthaul interface” (i.e., the interface connecting the RRUs and BBUs). Hence, enabling elements from different vendors to work properly together. In early 2018, the x-RAN forum released their first specification [25], dedicated to the fronthaul interface. This specification was a significant milestone as it marked the first attempt to introduce interoperability among different suppliers in the cellular network industry. Within a few months after this release, the x-RAN forum merged with the C-RAN alliance to form the Open-Radio Access Network (O-RAN) alliance.

2.1.4 Open RAN

With the release of the new series of standards on 5G cellular networks by 3GPP in early 2018 [29], *openness* and *disaggregation* became the core principles of the next generation of RANs, commonly referred to as *Open RANs*. Previously, MNOs were bound to a single vendor, with no opportunity to incorporate solutions from other providers, as their infrastructure relied on black-box, proprietary software and hardware accessible only to the original vendor.

The advent of *Open RANs*—a disaggregated architecture composed of vendor-neutral components connected by open, standardized interfaces—marked a major shift in cellular networks. By this new approach, 3GPP aimed for more agile, flexible, and cost-efficient RAN architectures. The main elements of an Open RAN (Fig. 2.3), as defined by 3GPP are:

- **Distributed Unit (DU)**: The DU handles lower-layer processing and real-time tasks closer to the radio interface.

- **Central Unit (CU):** The CU is responsible for managing the high-layer processing functions of the RAN.

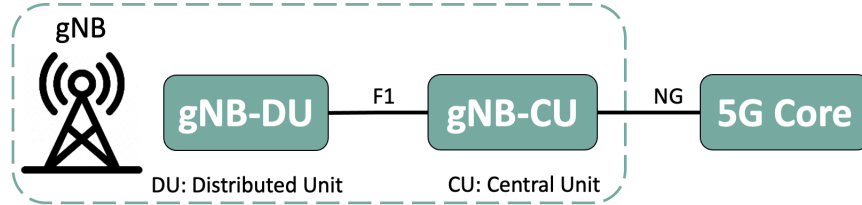


Figure 2.3: Simplified 3GPP Open RAN architecture

2.1.5 Open RAN vs. O-RAN

Before moving on to the rest of this chapter’s material, a clarification should be made on the different terms that we use in this document.

First, *Open RAN* is a technology movement or paradigm. When we talk about *Open RAN* we are talking about the new direction that the next-generation of RAN is taking. The direction of disaggregation and openness.

On the other hand, the term *O-RAN* refers to an operator-initiated alliance [12]. As mentioned earlier, in August 2018, the x-RAN forum and C-RAN alliance merged to establish the O-RAN alliance with *AT&T*, *China Mobile*, *Deutsche Telekom*, *NTT DOCOMO*, and *Orange* as the founding members. The primary goal of this alliance is to establish a modular architecture with open interfaces that realize the Open RAN paradigm, promoting more intelligent, open, virtualized, and interoperable radio access networks. Although, at the time of this writing, this alliance is the primary contributor to the *Open RAN* movement, this thesis focuses on the RAN architecture defined by 3GPP, rather than the one defined by the O-RAN alliance.

2.1.6 5G New Radio Protocol Stack

Communication between the UE and any element of a cellular network is governed by a set of protocols at all times. These protocols, defined by 3GPP, determine how to establish, manage, and terminate message exchanges between network elements. In 5G networks, these protocols are organized into a layered stack. Depending on the type of data being

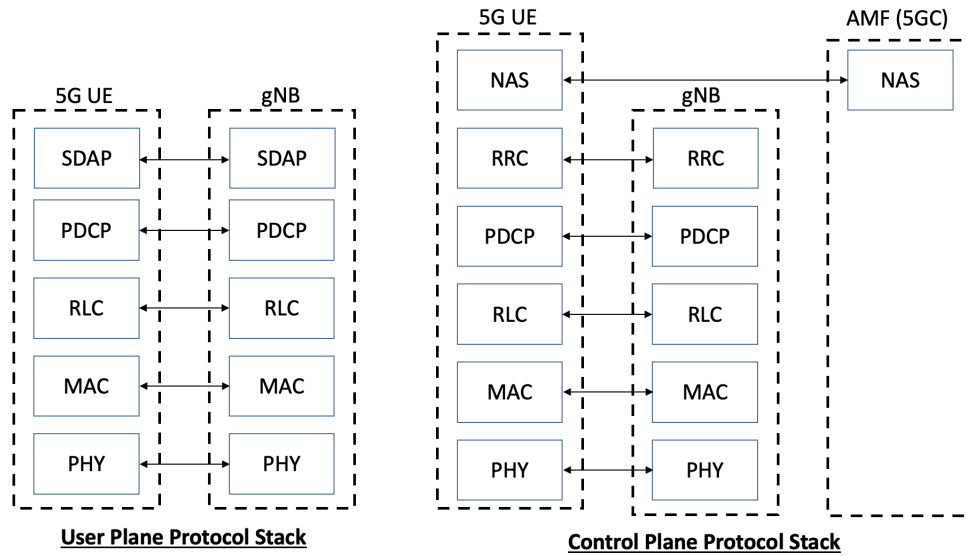


Figure 2.4: 5G NR Protocol Stack

exchanged—whether control signals or user data—different categories of the protocol stack are utilized, as illustrated in Fig. 2.4.

- **Physical Layer (PHY):** in charge of modulation and coding, and transmitting/receiving the data over the air interface.
- **MAC:** in charge of multiplexing/demultiplexing data units into transport blocks, error correction, and scheduling the of the UEs.
- **Radio Link Control (RLC):** in charge of segmentation, flow control, and data correction.
- **Packet Data Convergence Protocol (PDCP):** in charge of in-sequence delivery of data units, security, duplicate detection, and time-based discarding of data units.
- **Radio Resource Control (RRC):** in charge of broadcasting system information, establishing and maintaining the RRC connections, mobility functions, and UE measurement reports.
- **Non Access Stratum (NAS):** in charge of UEs' connection management, data session management, and UEs' authentication and authorization.

- **Service Data Adaptation Protocol (SDAP):** SDAP, which is a new protocol to 5G, is in charge of managing the Quality of Service (QoS) in the user plane.

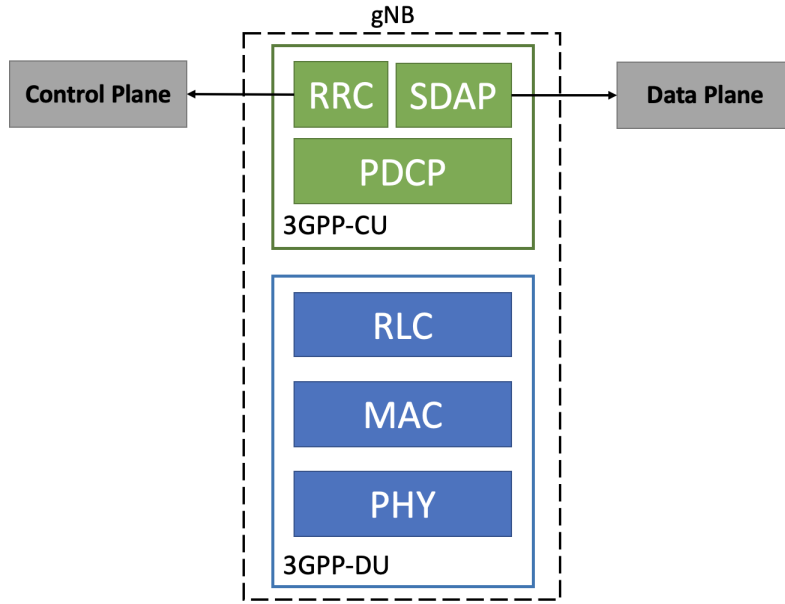


Figure 2.5: 3GPP functional split of the 5G protocol stack

5G Functional Splits

Traditionally, the entire protocol stack on the RAN side was located in a monolithic base station. However, 3GPP Release 15 initiated the disaggregation of the base station into the two elements introduced earlier, i.e., the Distributed Unit (DU): which manages the RF, PHY, MAC, and RLC layers; and 3) the Central Unit (O-CU), which hosts the remaining layers of the protocol stack. Figure 2.5 illustrates the functional split approved by 3GPP for the RAN [29].

2.2 The Core Evolution

The 5G core network was also specified in 3GPP Release 15, and introduced an innovative Service-Based Architecture (SBA) model [28]. Its cloud-native architecture promotes a high

decoupling of NFs. Unlike the 4G core network (EPC), which consists of four primary NFs, the 5GC architecture, shown in Fig 2.6, includes many NFs. These NFs offer a variety of services, from the Network Slice Selection Function (NSSF) to analytics via the Network Data Analytics Function (NWDAF). The main NFs in charge of communication with the RAN will be the AMF, Session Management Function (SMF), and the UPF, directly involved with the NAS and Next Generation Application Protocol (NGAP) protocols.

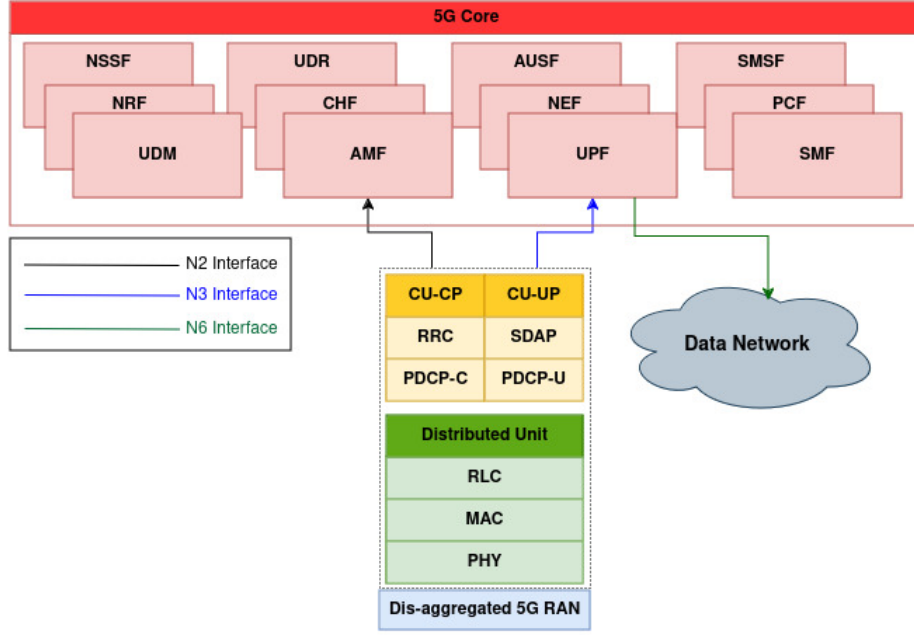


Figure 2.6: 5G system architecture

The AMF is responsible for managing all signaling not related to the user data, such as mobility and security, and serves as one of the endpoints for Control Plane (CP) communications. The SMF handles control signaling related to user data traffic, such as session establishment. User data is sent to the RAN via the wireless link, and from the RAN to the AMF through the N2 interface. The SMF which is not directly connect to the RAN, gets the necessary messages from the AMF. The UPF, manages user data, directly connecting to the RAN using GPRS Tunneling Protocol (GTP)-U, a tunneling protocol for the User Plane (UP) traffic. The UPF establishes GTP tunnels for each UE, processes user data traffic, shapes network traffic, and collects performance measurements [69].

2.3 Literature Review

Now that we have seen how the system has evolved over time and the reasons behind the introduction of new elements, we can explore the literature in more detail. We first review papers that have focused on a single full-fledged 5G-SA testbed and then those that dealt with comparisons of full-fledged testbeds.

2.3.1 Targeted Studies on the performance of a single testbed

In [48], Haakegaard *et al.* focus on a 5G-SA testbed, operating in TDD mode, employing *Open5GS* and *srsRAN* as the software platform of RAN, and 5GC, respectively. This study compares the theoretical and achieved performance of the testbed in terms of UL & DL bit rates, latency, and coverage. Additionally, the authors study the effect of several radio parameters, such as the number of SDR antennas, the bandwidth, and the TDD frame structure, on the performance of the testbed.

In [42], Bozis *et al.* present their 5G-SA testbed, operating in TDD mode over band *n78* which features the 5GC, RAN, and UE solutions from the OAI project. In this testbed, two Universal Software Radio Peripheral (USRP) N310 devices are utilized as the SDR-based UE and the RAN. The RAN and UE are connected through RF cables instead of a wireless connection, over-the-air. This study reports the latency and DL/UL throughput of a single-UE for two bandwidth scenarios.

In the evaluation done by Chepkoech *et al.* [45], the performance of six testbeds operating in LTE, 5G-NSA, and 5G-SA modes are studied with a focus on metrics such as throughput, latency, and signal strength. Notably, the only testbed operating in 5G-SA mode is implemented using *srsRAN_4G*, and *Open5GS*, and over band *n7*, which is an FDD 5G-NR band.

Sahbafard *et al.* provide a comprehensive assessment of a 5G-SA testbed, operating on TDD mode and utilizing OAI for both the RAN and the 5GC platforms in [66]. This testbed, uses a Quectel 5G modem as the UE. The authors compare the modem's achieved performance while using USRP B210 or N310 as the SDR of choice. They also conduct an analysis of the signal strength, in both single-user and multi-user scenarios, to evaluate the testbed's coverage.

The authors of [58] provide a tutorial on establishing a slice-aware 5G-SA testbed utilizing *srsRAN_Project* and *Open5GS* for the RAN and 5GC software platforms. This paper provides insights into the challenges of integration of different elements to the testbed.

Furthermore, it offers valuable information on potential issues faced during the implementation phase, along with troubleshooting strategies for these scenarios. Note that, the authors have not mentioned what specific band they are using for their tests. They are using 30 KHz for Subcarrier Spacing (SCS) which is mostly used for TDD bands.

The authors of [64] utilized an OAI-based 5G-SA testbed to conduct an experimental assessment of their proposed 5G positioning framework. They extracted information from DL Positioning Reference Signal (PRS) transmissions via the gNB and validated their approach in two scenarios: single gNB and multi-gNB connections. For this purpose, three USRP X310 were used as for the gNB, and the UE, on an 80 MHz, TDD channel with 30 KHz of SCS.

In [74], Vazquez *et al.*, have implemented a 5G-SA testbed using the RAN, and 5GC solutions from OAI. The testbed's performance is then analyzed in terms of UE's UL/DL throughput, and the CPU utilization of the testbed's host PC. For this purpose, the authors have worked with two types of UE-gNB connections, namely, using RF cables to connect the UE's antenna ports to the SDR's, and, Over-the-air (OTA) transmissions between the UE, and the SDR device.

Finally, the authors of [43], have introduced an easier way to configure and deploy various scenarios for the open-source OAI RAN platform. The authors have made their *Linux* shell scripts publicly available. Moreover, using the shells, the authors have provided some insights on the minimum achievable E2E latency of their testbed, and specifically, the OAI platform.

2.3.2 Comparative studies on the performance of multiple testbeds

The literature addressing a comparative analysis of 5G-SA testbeds with different combinations of elements is quite scarce. While the inherent design of open-source software platforms aims to facilitate interoperability, it is crucial to verify its feasibility, simplicity, and performance. To the best of our knowledge, the only existing studies on this subject are [36, 44, 46, 60].

The authors of [36] have provided a comparison between the performance of *srsRAN* and *OAI-RAN* in three aspects, namely, UE's DL bit rate, latency, and a qualitative comparison of the quality of a video call made by the UE. More so, they assessed the interoperability of the employed open-source RAN and 5GC software. The authors have configured the four implemented testbeds to be operating over band *n78*, in TDD mode. Noteworthy is the authors' effort to install OAI on an *ArchLinux* [2] system, deviating from the recommended *Ubuntu* [23] or *Fedora* [5] distributions of Linux operating system. The

results reveal differences in video call quality, download bit rates, and latency, with specific patterns observed for each platform. Additionally, the study highlights the differences in performance between SDR-based UEs and COST UEs.

[46] presents a comprehensive assessment of the performance of three 5GC software: *Open5GS*, *Open5GCore*, and *Amarisoft 5G Core* for three types of 5G modems, in terms of both throughput and latency, when the same RAN, *Amarisoft 5G RAN* is used. For the purpose of these tests, band *n78*, TDD mode was selected.

In [60], Mubasier *et al.* have implemented two distinct testbeds. both operating on band *n77*, which is a TDD band. The first testbed features *OAI* 5GC and RAN, along with USRP B210 and a host laptop. The second is a testbed utilizing USRP X300 in conjunction with *srsRAN* and *Open5GS*. This study evaluates network connectivity, the performance of the testbed seen by the UE, and also computing resource utilization of the open-source software.

[44] is another comparative study on the performance of testbeds comprising different elements. In this study *OAI-RAN*, and *srsRAN* were the focus, and a comparative analysis of their features, as well as quantitative results in terms of throughput, signal strength and latency were discussed for two 5GC software, namely, *Open5GS*, and *Free5GC*, in single-UE scenario. All the tests were set to be conducted over band *n78*.

Table 2.1 presents a comprehensive overview of the testbeds implemented in the existing literature, highlighting their key components and the measurement tools utilized for evaluating the relevant Key Performance Indicators (KPIs).

The main characteristics of the current literature are:

- **The scarcity of comparative studies** on different elements of 5G-SA experimental testbeds. There have been very few studies on the subject, and there is still a huge gap in understanding the full potential of different components.
- **The lack of a systematic approach** in experiments conducted within this domain. Throughout this thesis, we have tried our best to provide a clear definition of performance metrics and the corresponding measurement processes. Our work introduces a systematic approach to presenting the test scenarios and methodology.
- Although a few studies on comparative aspects overlap with this thesis in terms of KPIs such as data rate and E2E latency from the UE's perspective, **we expand upon these by introducing additional KPIs** like coverage and the computing resource consumption of the studied elements.

- **The lack of studies featuring multi-UE scenarios.**
- **The lack of studies considering UEs' location** as a factor affecting the performance.

Table 2.1: A Summary of the Literature on 5G-SA testbeds

Ref.	S_{5GC}	S_{RAN}	H_{RAN}	Duplexing	Connection Type	single/multi UE	UE Type	KPIs	Measurement Tools
[48]	Open5GS	srsRAN	USRP B210	TDD (n77)	OTA	multi-UE	Oneplus Nord CE 2 Lite	Coverage	Network Cell Info. Lite
							Simcom 5G modem	Throughput Latency	iperf3 ping
[42]	OAI-5GC	OAI	USRP N310	TDD (n78)	RF Cable	single-UE	OAI UE	Throughput Latency	iperf ping
[45]	Open5GS	srsRAN_4G	USRP X300	FDD (n7)	OTA	single-UE	COST UE	Throughput Latency	Iperf3
								Signal Strength	cellular-Z
[66]	OAI-5GC	OAI	USRP N310	TDD (n77)	OTA	multi-UE	Quectel RM500Q-GL	Throughput	iperf
			USRP B210					Latency Covergae	ping gNB logs
[58]	Open5GS	srsRAN	USRP X300	-	OTA	single-UE	Huawei P40 pro	-	-
[64]	OAI-5GC	OAI	USRP X310	TDD (n78)	OTA	Single-UE	OAI UE	UE Position	-
[74]	OAI-5GC	OAI	USRP X310	TDD (n78)	OTA	multi-UE	Samsung A33	Throughput	iperf3
			USRP B210		RF Cable		OAI UE	CPU Utilization	System Monitor
[43]	OAI-5GC	OAI	USRP N310	TDD (n78)	OTA	Single-UE	OAI UE	Throughput Latency	iperf ping
				TDD (n77)	RF Cable			Retransmission	gNB Logs
[36]	Open5GS	srsRAN	USR B210	TDD (n78)	OTA	single-UE	Moto G200 5G	Throughput Latency	SpeedTest iperf
	OAI-5GC	OAI						Video Call	qualitative measures
[46]	Amarisoft	Amarisoft	Amarosift CALLBOX PRO	TDD (n78)	OTA	Single-UE	Telit FN980	Throughput	SpeedTest
	Open5GCore						Sierra EM9191	Latency	ping
[60]	OAI-5GC	OAI	USRP B210	TDD (n77)	OTA	Single-UE	Oneplus 10 Pro	Throuput Latency	Nettfart ping
	Open5GS	srsRAN	USRP X300					CPU, Memory Utilization	htop
[44]	Open5GS	OAI	USRP B210	TDD (n78)	OTA	Single-UE	Google Pixel5	Throughput	Fusion Connect speed test
	Free5GC	srsRAN					Asus Zenfone9	Latency	ping
This thesis	Open5GS	srsRAN	USRP X410	TDD (n78)	OTA	multi-UE	OpenPlus Nord CE2	Throughput Latency	SpeedTest ping
	OAI-5GC	OAI	USRP B210	FDD (n1, n3)	RF Cable		Quectel RM502Q-AE	Coverage CPU, memory utilization	5G Switch - Force 5G Only top

Chapter 3

Implementation



Figure 3.1: Our 5G-SA platform

This chapter focuses on the implementation of 35 distinct 5G-SA testbeds. We have developed a flexible platform (shown in Fig. 3.1) that allows for the integration and configuration of various elements within a 5G-SA testbed, enabling a comprehensive evaluation of their impact on the performance experienced by the UEs.

Although we already have introduced our nomenclature and the elements of a typical 5G-SA system in Section 1.2 of Chapter 1, for the sake of completeness of this chapter, we

will re-introduce them. This chapter includes the main contributions of this thesis, and it is worthwhile introducing the system architecture once more.

The elements and sub-elements that constitute our operating 5G-SA platform are illustrated in Fig. 3.2, and listed in Table 3.1.

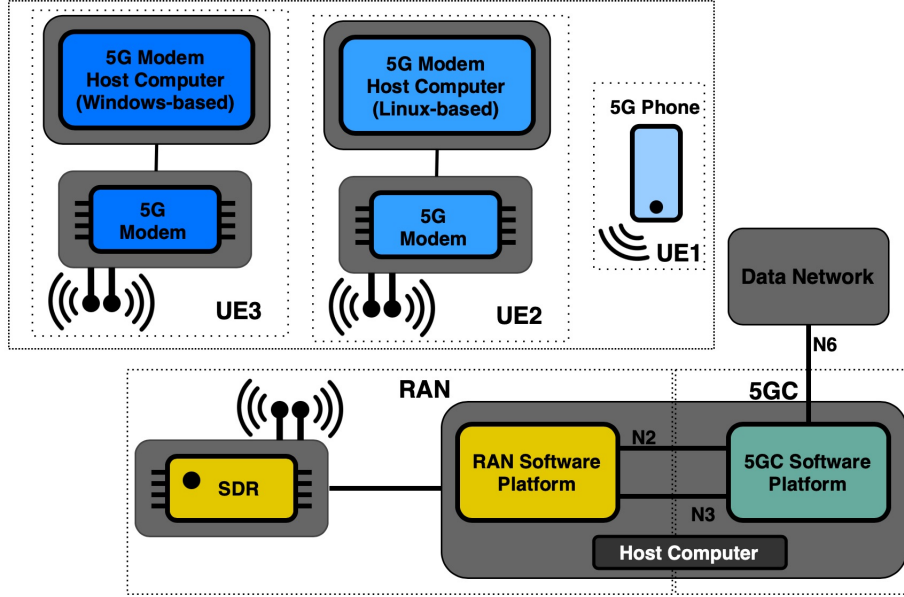


Figure 3.2: Our 5G-SA platform architecture

A single-cell, 5G-SA experimental testbed containing n UEs can be defined by the set $\mathcal{T}$ of its software and hardware elements and a set $\mathcal{C}$ containing the configuration parameters.

$$\mathcal{T} = \{S_{5GC}, H_{5GC}, S_{RAN}, H_{RAN}, (S_{UE_1}, H_{UE_1}), \dots, (S_{UE_n}, H_{UE_n})\} \quad (3.1)$$

$$\mathcal{C} = \{b, B, T\} \quad (3.2)$$

Where:

- S_{5GC} (resp. S_{RAN} and S_{UE_i}) is the collection of software sub-elements for 5GC (resp. RAN and the i -th UE), and H_{5GC} (resp. H_{RAN} and H_{UE_i}) is the collection of hardware sub-elements for 5GC (resp. RAN and the i -th UE).

- b is the band central frequency used by the system, B is the bandwidth, and T is the transmission mode between the gNB, and the UE (wired or wireless).

On the RAN side, the hardware elements include an SDR device equipped with an antenna system, acting as the RF front-end, connected to a host computer. The software aspects include an open-source platform for the 5G protocol stack. Additionally, the SDR requires a driver to communicate with the host computer's operating system.

For the 5GC, the hardware is a host computer, while the software includes an open-source platform that manages core network functions.

On the UE side, hardware options include smartphones, or computers with modems. In case of a phone-based UE, the software includes the operating system of the smartphone (we do not have access to the phone's protocol stack). In case of a modem-based UE, the software aspect consists of the operating system of the computer (with the first two layers of the 5G protocol stack being executed on the modem, and the rest on the computer).

Please note that for all software elements involving an Operating System (OS), the kernel used by the OS, particularly if it is *Linux*, is also a critical factor. As we will see later, setting up drivers to enable communication between hardware pieces and the computer was challenging when the host computer used *Linux*.

Table 3.1: Our 5G-SA platform elements
(We use the same color code as in Fig. 3.2)

Software Elements			Hardware Elements	
S_{5GC}	$S_{5GC} \in \{\text{Open5GS, OAI-5GC}\}$	H_{5GC}	5GC/RAN Host PC $\in \{\text{Host Computer \#1 \& \#2}\}$ (both S_{5GC} and S_{RAN} run on the same host PC)	
	OS: Ubuntu 20.04.6 LTS			
	Kernel: 5.15.0-xx-lowlatency			
S_{RAN}	$S_{RAN} \in \{\text{srsRAN, OAI-RAN}\}$	H_{RAN}	5GC/RAN Host PC $\in \{\text{Host Computer \#1 \& \#2}\}$	
	OS: Ubuntu 20.04.6 LTS			
	Kernel: 5.15.0-xx-lowlatency		SDR $\in \{\text{USRP B210 \& USRP X410}\}$	
	Driver: UHD v4.5.0		Antenna: Molex 5GHz compatible	
S_{UE_1}	OS: Android 11	H_{UE_1}	5G Phone: OnePlus Nord CE 2	
			SIM Card: sysmoISIM-SJA2	
S_{UE_2}	OS: Ubuntu 20.04.6 LTS	H_{UE_2}	Host PC: Intel® CoreTM i7-3770 CPU @ 3.4GHz	
	Kernel: 5.14.0-051400-generic		5G Modem: Quectel RM502Q-AE	
			SIM Card: sysmoISIM-SJA2	
S_{UE_3}	OS: Windows 11	H_{UE_3}	Host PC: Intel® Core i7-1255U CPU @ 1.70 GHz	
			5G Modem: Quectel RM502Q-AE	
			SIM Card: sysmoISIM-SJA2	

3.1 Elements Under-Study

This section provides an in-depth introduction to all the elements incorporated into our platform, enabling us to build 35 distinct testbeds.

3.1.1 The RAN

RAN Open-source Software Platforms

- *Platform #1- srsRAN*: It comprises open-source 4G and 5G software radio suites developed by the *Software Radio Systems* team. The project includes two main repositories, namely, *srsRAN_4G* and *srsRAN_Project*, both available under the GNU Affero General Public License version 3 (AGPLv3). While *srsRAN_4G* provides a prototype for 5G-SA, the supported features are minimal, and there will be no further updates. *srsRAN_Project* though, offers a full 5G-SA solution based on a complete codebase. In this study we have used *srsRAN_Project*, and we will refer to it as *srsRAN* in the following. It supports all TDD and FDD bands on FR1. The latest release of the software (*srsRAN_Project* 23.10.1) offers the flexibility to configure over 400 parameters in a user-friendly way, which has made working with *srsRAN* particularly convenient.
- *Platform #2- OAI-RAN*: Developed by the *Eurecom* team, the OAI software platform provides LTE, and 5G solutions for the RAN, and unlike *srsRAN*, the 5G core. For clarity, we will refer to the RAN solution from OAI, as *OAI-RAN*, and the 5GC as *OAI-5GC*. This project is distributed under the *OAI 5G Public License*. Compared to *srsRAN*, *OAI-RAN* provides more features such as more subcarrier spacing options and support for Frequency Range 2 (FR2). However, configuring OAI-RAN is more complex than configuring *srsRAN* since it requires modifying code blocks in the configuration file. For instance, changing the frequency band in *srsRAN* requires only changing the value assigned to the frequency band parameter, while in *OAI-RAN*, a large code block in the configuration file needs to be modified. Therefore, working with *OAI-RAN* requires a deep understanding of each parameter's function within the gNB and meticulous adjustment of specific values associated with those parameters. In this regard, we highly recommend reading [68], where the authors have described how to work with the *OAI-RAN* configuration file. We have used the “2024.w09” version of the *develop* branch of OAI’s GitLab [13].

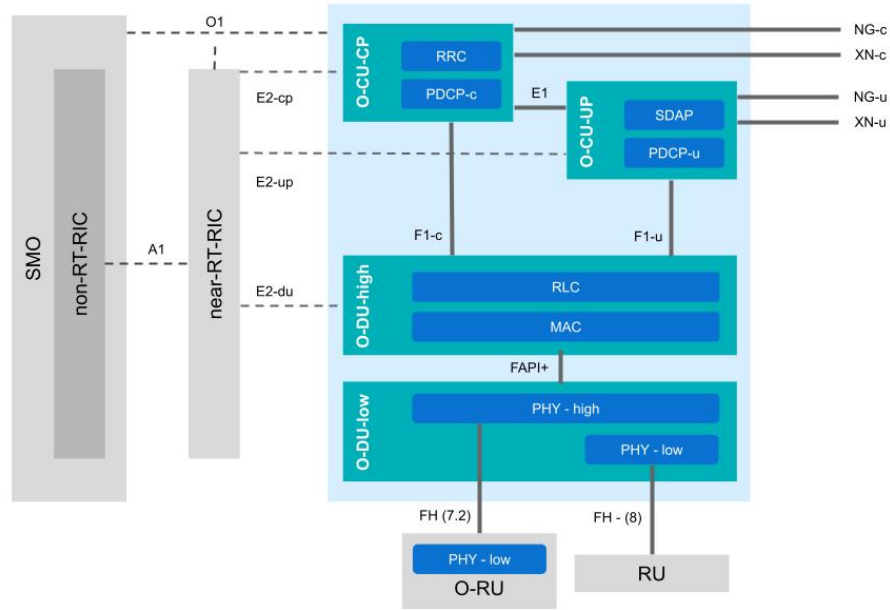


Figure 3.3: srsRAN gNB architecture [20]

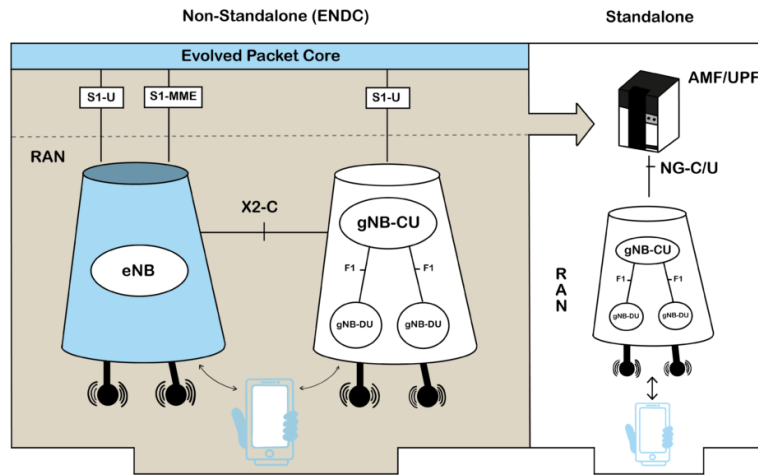


Figure 3.4: OAI gNB architecture [15]

RAN Hardware Platforms (SDR)

- *RAN SDR #1- USRP X410*: it is a high-end, all-in-one SDR (Fig. 3.5). It comes with advanced features like four independent Tx and Rx channels, each capable of 400

MHz of bandwidth. The X410 model is equipped with a built-in GPS Disciplined Oscillator (GPSDO) for improved timing synchronization. Additionally, it offers multiple networking interfaces for data and control offloading, such as two Quad Small Form-factor Pluggable 28 (QSFP28) ports supporting data transfer rates of up to 100 Gigabit Ethernet (GbE), along with standard interfaces like Ethernet and USB-C. In our experiments, we utilize one USRP X410 connected to the RAN host computer via two QSFP28-10GB connections and one Ethernet connection to the network.



Figure 3.5: USRP X410 [4]

- *RAN SDR #2- USRP B210*: This single-board, low-cost USRP (Fig. 3.6), is a dual-channel transceiver, providing up to 56 MHz bandwidth. B210 comes with a USB 3.0 connector to enable a connection to the RAN host PC. Since this USRP lacks a built-in GPSDO, maintaining synchronization might become a challenge. We have chosen to use a USRP B210 in our tests, as it is arguably the most popular SDR in the research community as of now. Hence, we can gain a clear understanding of what this USRP model offers compared to the high-end X410.



Figure 3.6: USRP B210 [4]

RAN Host Computers

To investigate the influence of computing resources on the testbed’s performance, we utilize the two PCs listed in Table 3.1, featuring different levels of computational power to host the RAN software.

- *PC #1*: The first host computer utilized in our tests is equipped with an 11th Gen Intel® Core™ i9-11900K processor, running at the base frequency of 3.50GHz. This system operates on Ubuntu 20.04.6 LTS, featuring kernel version 5.15.0-60-low-latency.
- *PC #2*: The second host is a mini PC featuring Intel® Core™ i7-10700 CPU @ 2.90GHz. This PC also runs Ubuntu 20.04.6 LTS, with kernel version 5.15.0-84-low-latency.

Note that the SDR requires a driver installed on the RAN host computer, so that they both can communicate. All USRP products from *Ettus* use the same hardware driver, called USRP Hardware Driver (UHD). In this study, we have installed UHD_4.5.0.0 on both RAN host computers. Utilizing two RAN host computers, allows us to assess the impact of computational power on the final performance of the testbed.

3.1.2 The Core

5GC Software Platforms

Our study focuses on two 5GC open-source software, namely Open Air Interface-5GC (OAI-5GC) and Open5GS. A 5GC software comprises a large number of different NFs. It is important to note that these two 5GC software, have different architectures, as illustrated in figures 3.7 and 3.8. To perform consistent and fair comparisons, we have used the same procedures across all our testbeds, such as UE attachment, UE authentication, UE PDU session establishment, and more.

- *Platform #1- Open5GS*: It is a popular core network solution that not only offers a 5GC but also an Evolved Packet Core (EPC) solution, enabling the implementation of 5G-SA, 5G-NSA, and LTE networks. The 5GC solution is based on 3GPP-Rel.17, and contains the following network functions: NF Repository Function (NRF), Service Communication Proxy (SCP), Security Edge Protection Proxy (SEPP), AMF, SMF,

UPF, Authentication Server Function (AUSF), Unified Data Management (UDM), Unified Data Repository (UDR), Policy and Charging Function (PCF), NSSF, and Binding Support Function (BSF). It is open-source and available under AGPLv3. For our test scenarios, we have used *Open5GS* v2.7.0.

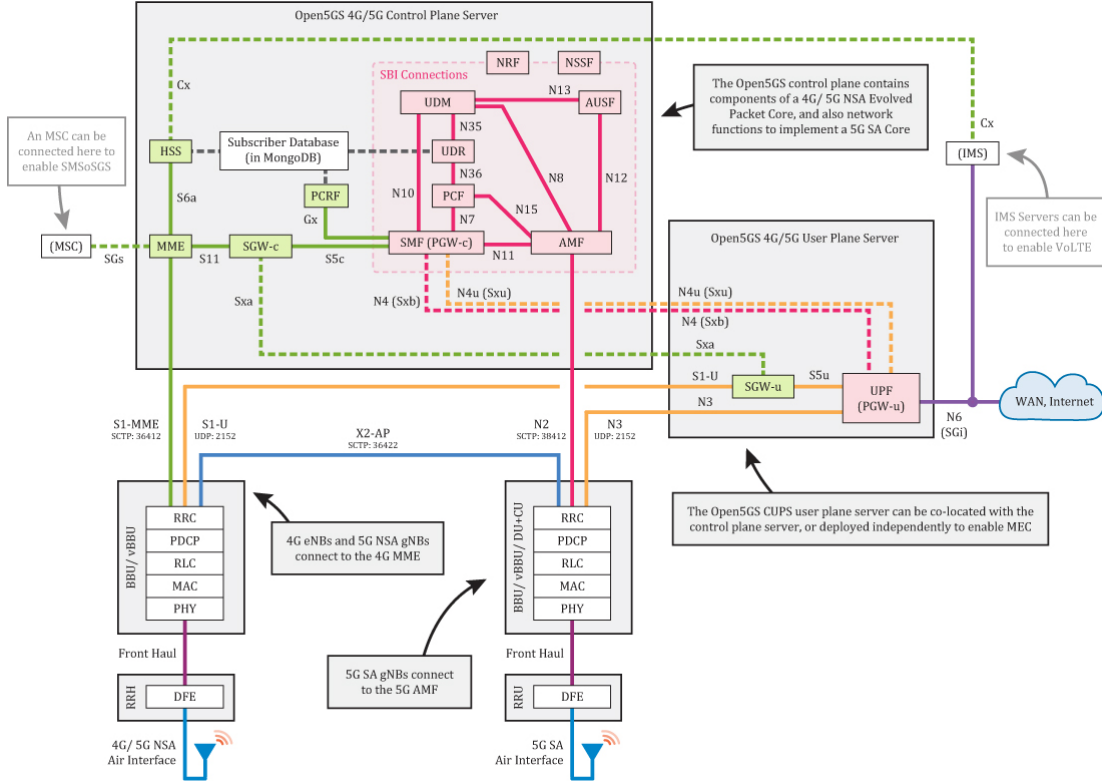


Figure 3.7: Open5GS 5GC architecture [16]

- *Platform #2- OAI-5GC*: It is an open-source software developed by the Eurecom team. Similar to *Open5GS*, this platform also offers solutions for 5GC and EPC. *OAI-5GC* is 3GPP compliant and provides various deployment options. Each deployment option includes a specific set of NFs, allowing users to choose the most suitable one based on their needs. We have chosen to deploy the *basic* version of *OAI-5GC* (branch develop, tag: v1.5.1), which includes the following NFs: AMF, SMF, UPF, NRF, UDM, UDR, and AUSF. This code is licensed under the *OAI Public License v1.1*.

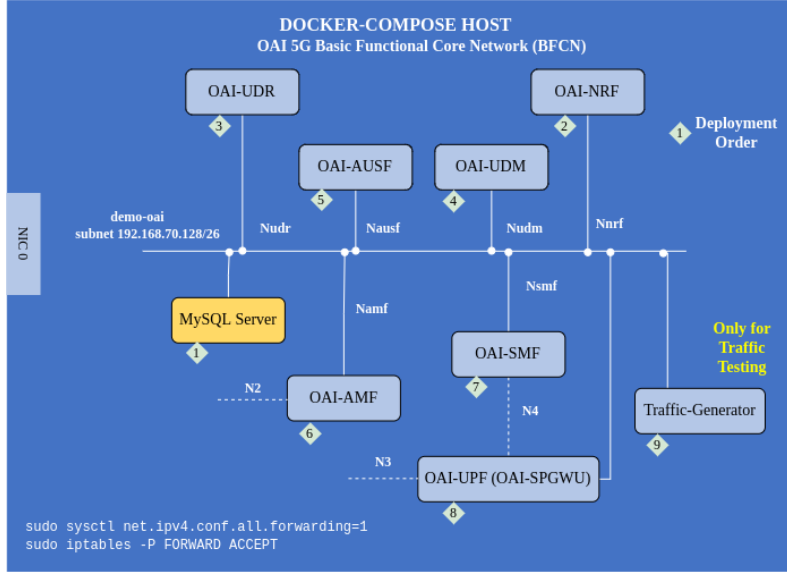


Figure 3.8: OAI-5GC architecture [15]

5GC Hardware Platforms

The core hardware platform is one of the two PCs described above since we execute both S_{5GC} and S_{RAN} on the same host computer.

3.1.3 The UEs

We consider three different UEs.

- UE_1 : Our first UE is a *OnePlus Nord CE 2 5G* COST phone, which is 5G-SA compatible. This phone runs *Android 11* and supports 11 5G-SA bands.
- UE_2 : The second UE comprises a Quectel 5G modem (shown in Fig. 3.9), the RM502Q-AE, connected to a host PC. Together, the modem and the PC will form the second *UE*. The PC (*Intel^(R) CoreTM i7 – 3770* CPU @ 3.4GHz) is running Ubuntu 20.04 with kernel version v5.14.0. Note that this UE is not easily movable.
- UE_3 : The third UE is composed of another Quectel 5G modem (RM502Q-AE) connected to a Dell laptop equipped with a 12th Generation Intel Core i7-1255U processor

running at 1.70 GHz and Windows 11. This choice allows us to investigate potential performance differences between the second and third UEs and determine if these differences can be attributed to their respective host computer operating systems (Ubuntu vs. Windows).

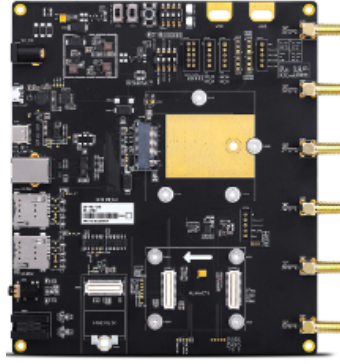


Figure 3.9: Quetel 5G modem (RM502Q-AE)

3.1.4 Miscellaneous

Last but not least, we utilized *sysmoISIM-SJA2* programmable SIM cards from *sysmo-com* [22] in this study. These SIM cards are 3GPP-Rel.16 compliant and come with the credentials required for modifying them. Additionally, for configuring our testbed's Public Land Mobile Network (PLMN), we assigned the Mobile Country Code (*MCC*), and Mobile Network Code (*MNC*) values as *001* and *01*, respectively.

3.2 Testbeds Under-Study

Now that we have introduced all the elements under study, we can present the testbeds that we have built, with respect to the definitions of the sets $\mathcal{T}$ and $\mathcal{C}$ in (1.1) and (1.2) respectively.

For the purposes of this thesis and to conduct a thorough study of the performance achieved by the UE(s), we built nearly all possible combinations of the available system elements. This resulted in the implementation of 36 distinct testbeds and the execution of

Table 3.2: Testbeds implemented for this thesis

<i>Testbed ID</i>	<i>S_{5GC}</i>	<i>S_{RAN}</i>	<i>Host PC</i>	<i>SDR</i>	<i>UE(s)</i>	<i>Configuration Set</i>
$\mathcal{T}_1$	Open5GS	srsRAN_4G	PC #2	USRP B210	{UE ₂ }	{n3, 20 MHz, OTA}
$\mathcal{T}_2$	Open5GS	srsRAN	PC #2	USRP B210	{UE ₂ }	{n78, 20 & 40 MHz, OTA}
$\mathcal{T}_3$	OAI-5GC	srsRAN	PC #2	USRP B210	{UE ₂ }	{n78, 20 & 40 MHz, OTA}
$\mathcal{T}_4$	OAI-5GC	OAI-RAN	PC #2	USRP B210	{UE ₂ }	{n78, 20 & 40 MHz, OTA}
$\mathcal{T}_5$	Open5GS	OAI-RAN	PC #2	USRP B210	{UE ₂ }	{n78, 20 & 40 MHz, OTA}
$\mathcal{T}_6$	Open5GS	srsRAN	PC #1	USRP X410	{UE ₁ }	{n78, 40 MHz, OTA}
$\mathcal{T}_7$	Open5GS	srsRAN	PC #1	USRP B210	{UE ₁ }	{n78, 40 MHz, OTA}
$\mathcal{T}_8$	Open5GS	srsRAN	PC #1	USRP X410	{UE ₂ }	{n78, 40 MHz, OTA}
$\mathcal{T}_9$	Open5GS	srsRAN	PC #1	USRP B210	{UE ₂ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{10}$	Open5GS	srsRAN	PC #1	USRP X410	{UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{11}$	Open5GS	srsRAN	PC #1	USRP B210	{UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{12}$	Open5GS	srsRAN	PC #1	USRP X410	{UE ₁ , UE ₂ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{13}$	Open5GS	srsRAN	PC #1	USRP B210	{UE ₁ , UE ₂ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{14}$	Open5GS	srsRAN	PC #1	USRP X410	{UE ₁ , UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{15}$	Open5GS	srsRAN	PC#1	USRP B210	{UE ₁ , UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{16}$	Open5GS	srsRAN	PC#1	USRP X410	{UE ₂ , UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{17}$	Open5GS	srsRAN	PC#1	USRP B210	{UE ₂ , UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{18}$	Open5GS	srsRAN	PC#1	USRP X410	{UE ₁ , UE ₂ , UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{19}$	Open5GS	srsRAN	PC#1	USRP B210	{UE ₁ , UE ₂ , UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{20}$	Open5GS	OAI-RAN	PC#1	USRP X410	{UE ₁ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{21}$	Open5GS	OAI-RAN	PC#1	USRP B210	{UE ₁ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{22}$	Open5GS	OAI-RAN	PC#1	USRP X410	{UE ₂ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{23}$	Open5GS	OAI-RAN	PC#1	USRP B210	{UE ₂ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{24}$	Open5GS	OAI-RAN	PC#1	USRP X410	{UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{25}$	Open5GS	OAI-RAN	PC #1	USRP B210	{UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{26}$	Open5GS	OAI-RAN	PC #1	USRP X410	{UE ₁ , UE ₂ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{27}$	Open5GS	OAI-RAN	PC #1	USRP B210	{UE ₁ , UE ₂ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{28}$	Open5GS	OAI-RAN	PC #1	USRP X410	{UE ₁ , UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{29}$	Open5GS	OAI-RAN	PC #1	USRP B210	{UE ₁ , UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{30}$	Open5GS	OAI-RAN	PC #1	USRP X410	{UE ₂ , UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{31}$	Open5GS	OAI-RAN	PC #1	USRP B210	{UE ₂ , UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{32}$	Open5GS	OAI-RAN	PC #1	USRP X410	{UE ₁ , UE ₂ , UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{33}$	Open5GS	OAI-RAN	PC #1	USRP B210	{UE ₁ , UE ₂ , UE ₃ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{34}$	Open5GS	srsRAN	PC #2	USRP B210	{UE ₁ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{35}$	Open5GS	OAI-RAN	PC #2	USRP B210	{UE ₁ }	{n78, 40 MHz, OTA}
$\mathcal{T}_{36}$	Open5GS	srsRAN	PC #2	USRP B210	{UE ₂ , UE ₃ }	{n77, 40 MHz, OTA}

around 350 different tests. You can see the details of the elements, and configuration sets of those 36 testbeds in Table 3.2. Additionally, a demo of our initial testbed implementation ($\mathcal{T}_1$), as reported in [37], can be viewed at <https://bit.ly/3KZPXZW>.

We have configured all the tests (except for our initial testbed, $\mathcal{T}_1$), to be done on $b = n78$, which operates in TDD mode. In a TDD-based communication system, time is divided into discrete periodic intervals, referred to as a *frame*. 3GPP has set the frame length to be fixed as 10 ms [27]. Each frame consists of a set of T *subframes*. A subframe consists of one or more consecutive *slots*, with each slot containing 14 *symbols*. A frame structure may look like what is illustrated in Fig. 3.10.

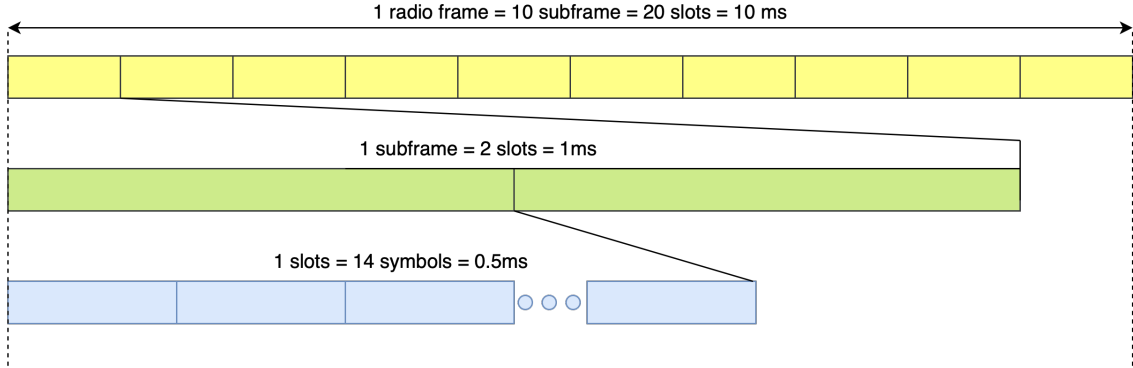


Figure 3.10: Simplified 5G-NR frame structure

Numerology (μ), refers to the set of parameters that define the physical layer structure, specifically, the subcarrier spacing (SCS), symbol duration, and cyclic prefix length in a communication system. Also, SCS refers to the frequency difference between adjacent subcarriers in a multi-carrier communication system. Since 5G aims to cover a wide range of operating frequencies, including sub-3 GHz, sub-6 GHz, and mmWave (over 25 GHz), it is very difficult to create a single numerology (subcarrier spacing) that maintains efficiency and performance across all these frequencies. Therefore, unlike LTE, where the SCS was fixed to 15 KHz, 5G introduces multiple SCS options as shown in Table 3.3 [31]. Please note that not every SCS can be used for every frequency band, and bandwidth (you can refer to 3GPP TS38.300-Table 5.1-1 for more information [30]).

Moreover, it is crucial to precisely define when to expect transmission and reception. In LTE networks, a predefined pattern for UL and DL slot allocation in a radio frame was established. 5G NR comes with more flexibility in terms of allocating slots to UL or DL. For instance, imagine a frame structure that looks like “DDDDDFUUUU”, where

Table 3.3: Number of OFDM symbols per slot, slots per frame, and slots per subframe for normal cyclic prefix

μ	N_{slot}^{symbol}	$N_{slot}^{frame,\mu}$	$N_{slot}^{subframe,\mu}$
0	14	10	1
1	14	20	2
2	14	40	4
3	14	80	8
4	14	160	16
5	14	320	32
6	14	640	64

“*D*” represents downlink slots, “*U*” represents uplink slots, and “*F*” stands for a flexible slot. The flexible slot must include a Guard Period (GP) between the downlink and uplink symbols. For small cell sizes, such as those in experimental testbeds, even 1 GP symbol is sufficient. The flexible slot can, for instance, consist of eight downlink symbols, 1 GP symbol, and five uplink symbols, denoted as “8:1:5” [48].

When comparing the performance of different RAN platforms operating in TDD mode, it is imperative to configure the same TDD slots and symbols for a meaningful comparison. Unfortunately, this aspect is often overlooked in the few comparative studies on experimental testbeds. In our tests, however, we ensured that both S_{RAN} platforms were set with the same TDD configuration (as **the default TDD configuration of srsRAN and OAI-RAN are not the same**). We set the frame structure as: “*DDDDDDFUUU*”, which accounts for 6 DL slots, 3 UL slots, and 1 flexible slot. Moreover, we are using Numerology 1, with an SCS value of 30 kHz.

3.3 Implementation Challenges

In this section, we describe the challenges we faced while building these testbeds, and the reasons for some of our choices.

3.3.1 Hardware-related Challenges

The hardware-related challenges mostly arise from the limitations of the hardware elements in supporting the 5G-SA technology.

- *Phones*: The availability of 5G SA-enabled phones is still limited. For our platform, we used multiple phones, such as *LG Velvet 5G*, *Google Pixel 4a*, *Google Pixel 5 5G*, and *OnePlus Nord CE2*, and could only get *OnePlus Nord CE2* connected to it. The main reasons for the challenges regarding connecting 5G-enabled phones to the experimental testbeds, include:
 - In the initial phase of our project, the RAN software platforms offered limited support for 5G-SA frequency bands, making it difficult to find a phone compatible with the same bands supported by the RAN platforms. However, as both the open-source RAN software matured over time, this challenge gradually faded¹.
 - Next, is the challenge of forcing 5G phones to operate in the 5G-SA mode. Although some phones do support the SA mode, manufacturers often set them up to operate on the 5G-NSA mode by default, which requires additional steps to switch to 5G-SA. To ensure our phone operated exclusively in 5G-SA mode, we installed an Android application called *5G Switch - Force 5G Only* [1]. This application is free and does not require the phone to be rooted. There is also another Android application called *Network Signal Guru* [10] which can be used to force the phone to operate on the SA mode. However, *Network Signal Guru* requires root access to the phone.
 - Additionally, using the correct PLMN configuration is sometimes crucial for making experimental testbeds visible to the phone. Manufacturers often configure phones to exclusively connect to commercial networks, not private or experimental ones. The PLMN serves as the unique identifier for each network, and private networks are generally assigned the PLMN “00101” for their configuration. As a result, the PLMNs used by them are not recognized by the phone, leaving the testbeds invisible. In such cases, multiple PLMN configurations should be tested to find one that works across all UEs.
- *Quectel modem*: This 5G SA enabled modem is used as a user equipment (UE) instead of commercial mobile phones since it has more flexibility in controlling crucial

¹As of now, both OAI-RAN, and srsRAN provide a list of tested COST UEs that have successfully get connected to their platforms [18, 14].

operational parameters such as the band and the mode. However, it has software-related drawbacks that will be discussed shortly.

- *Antennas (for the modem and the USRP)*: They have to operate at the required bands for successful attachment.
- *USIM*: Re-programmable universal SIM cards are needed to successfully evaluate and control the testbed’s parameters, since acquiring encryption keys and security parameters from commercial SIM cards is impossible.

3.3.2 Software-related Challenges

Software-related challenges arise from the lack of compatibility of the different software elements.

- *Quectel modem*: When connected to a Windows-based host PC or laptop, it relies on the Windows Connection Manager to manage the 5G connection. However, on Ubuntu, several packages and drivers need to be installed, and in our case, we were only able to successfully install them with Kernel version 5.14.0, but not with any higher versions.

3.4 Performance Metrics

Before presenting our test scenarios and results, a detailed description of the selected performance metrics is required. In the following paragraphs, we define each metric, introduce the measurement tool used for each, and explain the measurement approach.

3.4.1 Data Rate

We measure the UL and DL average data rate in Mbps, using *iperf3* [8]. Each experiment runs for three minutes, and we report the average of the achieved UL and DL data rates. Please note that, We selected a three-minute duration for our experiments to assess the stability of the wireless connection. Using such a prolonged duration allowed us to verify not only the data rate and latency but also the stability of the 5G-NR connection between the UE and the gNB. Moreover, to run *iperf3* on UE₁ the Android phone, we installed *he.net - Network Tools* [7]. The data rate assessment procedure is detailed in Algorithm 1.

Algorithm 1 Data Rate Assessment using *iperf3*

- 1: Start S_{5GC}
 - 2: Start S_{RAN}
 - 3: Power on the UE(s) and connect them to the testbed.
 - 4: **DL Rate Assessment:**
 - 5: Set up an *iperf3* server on the UE(s).
 - 6: Run *iperf3* client(s) on the host PC to send full buffer traffic from the 5GC to the UE(s) for 3 minutes.
 - 7: **UL Rate Assessment:**
 - 8: Set up *iperf3* server(s) on the 5GC.
 - 9: Run an *iperf3* client on each UE to send full buffer traffic from the UE(s) to the 5GC for 3 minutes.
 - 10: **return** Average rates achieved for both DL and UL directions.
-

3.4.2 Latency

The E2E latency is measured in milli-second (ms), by using the *ping* command at the UE side. We conducted each test for three minutes and report the average latency between the UE and the 5GC. The latency assessment procedure is detailed in Algorithm 2.

Algorithm 2 E2E Latency Assessment using *ping*

- 1: Start S_{5GC}
 - 2: Start S_{RAN}
 - 3: Power on the UE(s) and connect them to the testbed.
 - 4: run “*ping 5GC-IP -w 180*” on the UE(s).
 - 5: **return** Average & Min. latency values.
-

3.4.3 Coverage

We consider the Reference Signal Received Power (RSRP) in dBm to be the measure of the coverage. The coverage tests were conducted for UE₁. We used the Android application *5G Switch - Force 5G Only* [1], on UE₁, the mobile phone, to report the RSRP.

Algorithm 3 Coverage Assessment using the Android Phone

```
1: *Loc  $\leftarrow \{A2, B, C, D, E, F, G, H, I, J, K, L, M, N, O\}$ 
2: Start  $S_{5GC}$ 
3: Start  $S_{RAN}$ 
4: for  $l$  in Loc do
5:   Turn UE1 on at  $l$ 
6:   Get UE1 connected to the 5G-SA testbed
7:   Open “5G Switch - Force 5G Only”
8:   Report the Signal Strength at  $l$ 
9: end for
10: return
```

*For more details about the list of locations please refer to Fig. 3.12

3.4.4 Computational resource consumption

Finally, to monitor how each RAN software platform consumes the computational resources of its host computer, we use the *top* command. This allows us to observe the running processes, and the overall host computer resource utilization (CPU and memory). We run this command on the host PC that executes both 5GC and RAN, for three minutes and report the maximum percentage of CPU and memory utilization for each software platform.

Algorithm 4 Computational Resource Consumption Assessment using *top*

```
1: Start  $S_{5GC}$ 
2: Start  $S_{RAN}$ 
3: Power on the UE(s) and connect them to the testbed.
4: Run the top command on the  $S_{5GC}$  and  $S_{RAN}$  host PC.
5: Repeat the “Data Rate Assessment” defined earlier in Algorithm 1.
6: Report the maximum CPU and memory utilization.
```

3.5 Test Scenarios, Methodology and Results for Open-source 5G software Interoperability Assessments

In this section, we first study open-source 5G software interoperability. While every open-source software project should inherently be designed to facilitate interoperability, there

remains a need to verify if interoperability is possible, easy and performing well. To assess the impact on performance of one software over another, we present our test scenarios and the corresponding methodology to analyze the performance of single-cell, single-UE testbeds with different combinations of RAN and 5GC including the quantitative and qualitative metrics used for the comparison.

For our comparative analysis, we have implemented four testbeds (please refer to Table 3.2 for details of these testbed) and experimented with two sets of configurations as follows (abusing our notation to focus only on the RAN and 5GC open-source software).

$$\begin{aligned}\mathcal{T}_2 &= \{Open5GS, srsRAN\} \\ \mathcal{T}_3 &= \{OAI - 5GC, srsRAN\} \\ \mathcal{T}_4 &= \{OAI - 5GC, OAI - RAN\} \\ \mathcal{T}_5 &= \{Open5GS, OAI - RAN\} \\ \mathcal{C}_1 &= \{n78, 20MHz\}, \mathcal{C}_2 = \{n78, 40MHz\}\end{aligned}$$

We have build and analyzed the performance of these four testbeds, while both RAN, and 5GC software platforms are executed on *PC #2*, using *UARP B210* as our SDR device.

We have tested all three UEs for interoperability tests; however, we were unable to get UE₁, the 5G phone, connected to $\mathcal{T}_3$ and $\mathcal{T}_4$, which use OAI-5GC as their core network. We tried multiple PLMN combinations, including:

- “00101” which is used for private networks.
- “20895” used for commercial French MNOs.
- “302720” used for commercial Canadian MNOs.

We were unable to connect this phone to those testbeds, primarily because $\mathcal{T}_3$ and $\mathcal{T}_4$ were invisible from the phone’s perspective. As a result, we relied on UE₂ and UE₃, both of which successfully connected to all four testbeds.

Since the goal of this section is to assess the interoperability of the four open-source 5G software platforms, we conducted all our interoperability tests using only UE₂, the *Quectel* 5G modem connected to a Linux-based PC. However, further details about the impact of different UE types on the achieved performance are provided in Section 3.6.

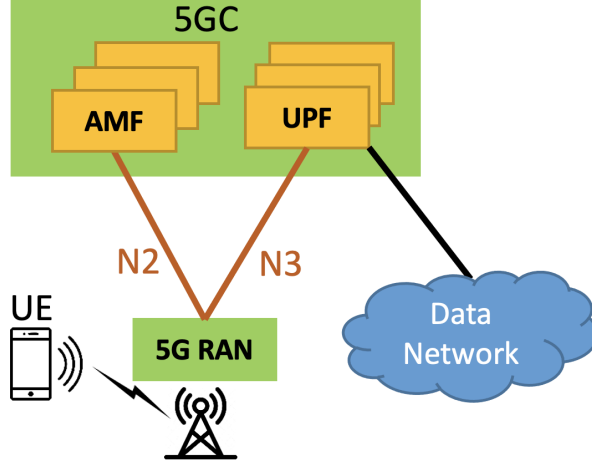


Figure 3.11: Simplified 5G-SA architecture to illustrate interfaces $N2$ and $N3$

3.5.1 Methodology & Results for Interoperability Assessment

Methodology

Fig. 3.11 depicts the simplified architecture of 5G-SA, highlighting two important features: 1) the RAN that we use is not dis-aggregated, and 2) the interfaces $N2$ and $N3$ that need careful configuration for establishing a connection between the RAN and the 5GC.

In order to establish connections, the open-source RAN software must be configured with the IP addresses of the AMF and UPF in the 5GC. Conversely, the 5GC requires the RAN IP address for this setup. Once these connections are successfully established, both software elements can set up the interfaces and exchange messages using the NGAP protocol [26].

Generally, each open-source software includes a configuration file with default values for their addresses. For instance, for $\mathcal{T}_4$, the configuration file in *OAI-RAN* is pre-configured with values that align with the IP addresses of AMF and UPF in *OAI-5GC* (as this is the intended combination of RAN and 5GC envisioned by the developers) and that is the configuration file that we use. However, for $\mathcal{T}_2$, $\mathcal{T}_3$, and $\mathcal{T}_5$, the configuration files of the software elements needed to be modified. srsRAN provides a documentation on how such a modification should be done for $\mathcal{T}_2$. For $\mathcal{T}_3$, and $\mathcal{T}_5$, though, no configuration was recommended by Open5GS, srsRAN, or OAI. Consequently, we had to create our own set of addresses for the interfaces connecting the RAN to the 5GC so that they can successfully communicate. Table 3.4 provides the addresses and their respective values, which were

Table 3.4: Parameters of the Open-source Software for each testbed

	Open-source 5GC software		Open-source RAN software	
$\mathcal{T}_2$	Open5GS		srsRAN	
	AMF	sbi: 127.0.0.5 ngap: 127.0.0.2	AMF	addr: 127.0.0.2
	UPF	pfcf: 127.0.0.7 gtpu: 127.0.0.2		bind_addr: 127.0.1.1
$\mathcal{T}_3$	OAI-5GC		srsRAN	
	AMF	ipv4_address: 192.168.70.132	AMF	addr: 192.168.70.132
	networks	subnet: 192.168.70.128/26		bind_addr: 192.168.70.129
$\mathcal{T}_4$	OAI-5GC		OAI-RAN	
	AMF	ipv4_address: 192.168.70.132	AMF	ipv4: 192.168.70.132
	networks	subnet: 192.168.70.128/26		GNB_IPV4_ADDRESS_FOR_NG_AMF: 192.168.70.129/26
$\mathcal{T}_5$	Open5GS		OAI-RAN	
	AMF	sbi: 127.0.0.5 ngap: 127.0.0.5	AMF	ipv4: 127.0.0.5
	UPF	pfcf: 127.0.0.7 gtpu: 127.0.0.7		GNB_IPV4_ADDRESS_FOR_NG_AMF: 127.0.1.1/24

employed in each software across all four testbeds.

Please note that, beyond the aforementioned modifications made to the configurations of the interfaces, adjustments of the parameters of the PLMN (00101) are also required. These adjustments, not only include the RAN and 5GC, but also extend to the SIM card, to ensure that all three elements — RAN, 5GC, and the UE — utilize identical authentication information, to enable seamless interactions among them.

Results on tests conducted on interoperability of open-source 5G

We successfully verified the interoperability of the open-source 5G software platforms for $\mathcal{T}_2$, $\mathcal{T}_3$, $\mathcal{T}_4$, and $\mathcal{T}_5$, using the configuration details outlined in Table 3.4.

Observations on tests conducted on interoperability of open-source 5G

The observation is that interoperability is often very challenging because it requires careful configuration.

- *Ease-of-Use*: Among the four open-source software we have worked with, *srsRAN*, and *Open5GS*, proved to be the most user-friendly, as their configuration files are

straightforward and easy to read. In contrast, both *OAI-RAN* and *OAI-5GC* were much more difficult to deploy. For example, when adjusting the bandwidth within the RAN software, *srsRAN* offers a single parameter to adjust, “channel_bandwidth_MHz” within its configuration file. On the other hand, *OAI-RAN* requires changes in a specific code blocks within the configuration file for bandwidth adjustments. Furthermore, in the context of frequency configuration, *srsRAN* only requires two parameters: the New Radio (NR) band number and the Absolute Radio Frequency Channel Number (ARFCN). In contrast, *OAI-RAN* needs careful adjustments within an even bigger code block (compared to changing the bandwidth) in the configuration file to accurately define the frequency ranges.

- *Robustness*: We observed that the two OAI software suites show a higher sensitivity to unexpected messages or errors. In many scenarios, the software stack crashed, necessitating a restart. *srsRAN* and *Open5GS* both are more resilient to such events. For instance, consider a situation where a UE unexpectedly drops its connection with the RAN. We have noticed that in such cases, when the UE attempts to re-establish the connection, *OAI-5GC* often experiences disruptions, necessitating a complete software restart. In contrast, we have not observed this behavior with *Open5GS*.

3.5.2 Methodology & Results for Data Rate and Latency Assessment in interoperability tests

Methodology

To analyze the E2E latency and UE’s achieved data rate in UL and DL, we used the four testbeds mentioned earlier ($\mathcal{T}2$, $\mathcal{T}3$, $\mathcal{T}4$, and $\mathcal{T}5$). In our experiments, the goal is to compare the performance on the DL and the UL of these selected open-source software solutions, from the perspective of a UE in the simplest scenario: a single, stationary UE that located in near proximity of the SDR device, and is registered and connected to the internet. Hence, our comparison while being very detailed for the scenario that we consider, does not include all the features of the different 5GCs and RANs, e.g., mobility, localization, etc.

For interoperability tests, other than Algorithm 1, we assessed data rates using *Speedtest* by Ookla [17]. After conducting five tests and averaging the results, we did not observe any noticeable differences between the measurements from the two tools, *iperf3* and *SpeedTest*. Moreover, for each latency assessment, we conducted two types of tests: (i) the E2E latency from the UE to the 5GC, and (ii) the E2E latency from the UE to an external server. For

Table 3.5: Downlink and Uplink data rate and latency results for interoperability tests

	S_{5GC}	S_{RAN}	BW	$DL\ rate$	$UL\ rate$	$Latency$	AVG	$Min.$
$\mathcal{T}_2$	<i>Open5GS</i>	<i>srsRAN</i>	20	41.3	17.4	Core	31.4	20.3
						External	32.4	22.7
			40	91.7	40.7	Core	30.3	19.2
						External	33	22.6
$\mathcal{T}_3$	<i>OAI-5GC</i>	<i>srsRAN</i>	20	41.2	17.3	Core	32.4	21.3
						External	33.6	23.7
			40	91.5	40	Core	30.8	19.3
						External	33.1	23.2
$\mathcal{T}_4$	<i>OAI-5GC</i>	<i>OAI-RAN</i>	20	55.5	9.4	Core	9.9	7.1
						External	14.4	11.5
			40	113.6	19.7	Core	9.9	6.9
						External	14.6	11.6
$\mathcal{T}_5$	<i>Open5GS</i>	<i>OAI-RAN</i>	20	55.1	9.4	Core	9.7	7.2
						External	14.4	11.6
			40	113.5	19.7	Core	10	7.4
						External	14.4	11.6

Data rate values are in Mbps. Latency values are expressed in milliseconds (ms).

the former, a *ping* command was sent from UE₂ to the IP address of the AMF in the core network (as outlined in Table 3.4). For the latter, a *ping* command was sent from the UE to the external server. Both the average and minimum measured latency values are reported.

Results of Data Rate and Latency Assessment for Interoperability Tests

The performance results for the four testbeds are reported in Table 3.5 for two bandwidth values: 20 MHz and 40 MHz. Focusing on the comparisons, we make the following observations:

- *Impact of the RAN software:* For this comparison, we consider $\mathcal{T}_3$, and $\mathcal{T}_4$, where the common S_{5GC} is *OAI-5GC*.
 - *DL data rate:* Clearly, there is a difference in the achieved DL rate. With a 40 MHz bandwidth, *srsRAN* offers 91.5 Mbps, whereas *OAI-RAN* delivers an impressive 113.6 Mbps.

- *UL data rate*: On the UL, *srsRAN* is the one delivering better performance. With both 20, and 40 MHz bandwidth, *srsRAN* offers about double the UL rate of *OAI-RAN*.
- *E2E Latency*: *OAI-RAN* outperforms *srsRAN* in terms of latency. It is also worth highlighting that throughout our experiments, latency measurements with *OAI-RAN* exhibited a greater stability, whereas with *srsRAN*, latency values showed considerable variations. These variations are evident when comparing the average and minimum latency values.
- *Impact of the 5GC software*:: For this comparison, we focus on $\mathcal{T}_2$ and $\mathcal{T}_3$, both using *srsRAN*.
 - *Data rate*: There is no significant difference in the UL and DL rates between the two 5GC software.
 - *Latency*: In this context as well, there is not a huge difference between the average latencies achieved by the UE from both *Open5GS*, and *OAI-5GC*.

3.6 Test Scenarios, Methodology and Results for the RAN Platforms

In this section, we introduce the test scenarios and the corresponding methodology to analyze the performance of the RAN elements, including the SDR devices and the two open-source software platforms, followed by a presentation of the results.

We have build and analyzed the performance of 28 testbeds that consist of *all the possible combinations* of the other elements described above with either (any) one, (any) two or the three UEs. Specifically, considering the three UE devices that we have described above, we created seven different UE combinations based on their number and types, i.e., $\{\text{UE}_1\}$, $\{\text{UE}_2\}$, $\{\text{UE}_3\}$, $\{\text{UE}_1, \text{UE}_2\}$, $\{\text{UE}_1, \text{UE}_3\}$, $\{\text{UE}_2, \text{UE}_3\}$, $\{\text{UE}_1, \text{UE}_2, \text{UE}_3\}$. Recall that we consider two SDR devices, i.e., $\{\text{USRP B210}, \text{USRP X410}\}$ and two S_{RAN} platforms, i.e., $\{\text{srsRAN}, \text{OAI-RAN}\}$. Thus, using all possible combinations of these three groups, $(7 \times 2 \times 2)$, we built 28 testbeds.

Additionally, to assess the computational resource consumption of the two 5G open-source software, we have built two additional testbeds using a less powerful PC, PC #2, and the two RAN platforms. This setup allows us to evaluate how the performance of

each open-source software is impacted by the host PC, and the number of connected UEs. Hence in total, we have build and studied 30 testbeds for this section.

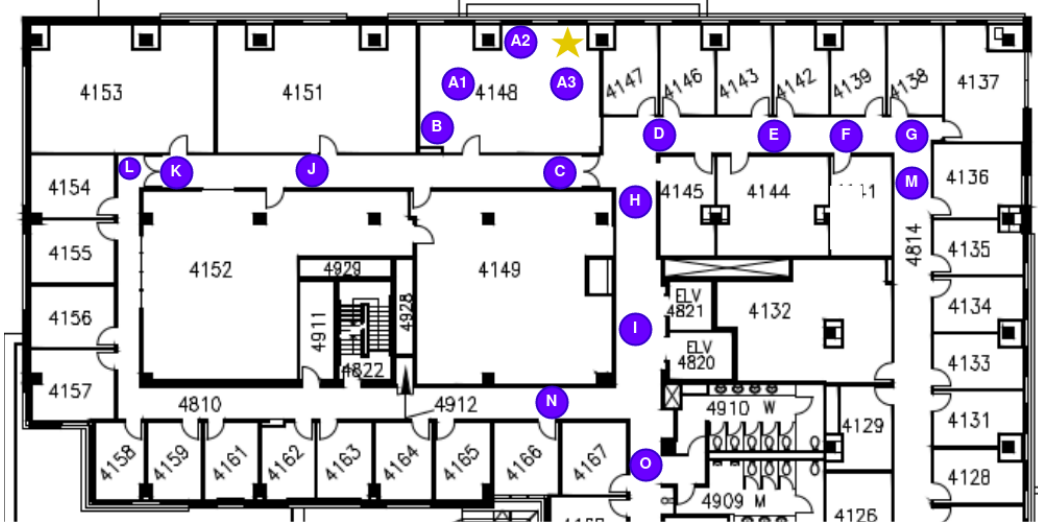


Figure 3.12: Map indicating the positions where tests were conducted

3.6.1 Methodology & Results for Data Rate Assessment

Methodology

For the assessment of the data rate of each UE within each testbed, we had to carefully take the location of each UE into account. The first tests were done when all the UEs of each testbed were located in “good” positions, i.e., at positions where the downlink data rate of a single UE was consistently at its peak (characterized by the highest rate of the existing MCS). The comparison of the performance of the different testbeds when all UEs are in those positions, gives us valuable information on how best the testbeds can perform.

Fig. 3.12 illustrates the map of the fourth floor of the *Centre for Environmental and Information Technology* building on the main campus of University of Waterloo where we conducted our tests. Our lab is in room 4148, and we have indicated the location of the SDR by a star sign on the map.

We first identified three good positions in our lab, all in the vicinity of the SDR device. We have marked the three selected positions for the UEs in Fig. 3.12 as A1, A2, A3. UE₁

was placed at A_1 , UE_2 at A_2 , and UE_3 at A_3 . Recall that UE_2 is not easily movable and hence was kept at A_2 for all tests. Hence, for all other locations, we only checked the rates seen by UE_1 and UE_3 . The results of this initial round of tests where the UEs are in good positions, with either one, two or three UEs are presented in Table 3.6 (please refer to tests $\{T1, T2, T3, T8, T9, T10, T15\}$).

Next, to study the impact of locations on the data rate observed by the UE(s), we selected two additional positions where the drop in the data rate was significant enough to categorize the positions as “*fair*”, and “*bad*”. In this regard, after multiple preliminary tests conducted using UE_1 , and UE_3 , position D on the map was selected as the position which would yield a “*fair*” rate, with MCS values observed between 15 to 17 and position E as the “*bad*” position, with MCS values observed between 9 to 11. The single and multi-UE results corresponding to these tests are presented in Table 3.6, with test ids $\{T4, T5, T6, T7, T11, T12, T13, T14, T16, T17\}$. Note that the other positions in Fig. 3.12 are used for our coverage study.

In order to keep the number of tests reasonable and the size of the tables manageable, we only used 14 of the 28 testbeds to study the impact of location on performance, by fixing the SDR to USRP B210 in this first round of tests. A second round of tests to compare the SDR, is described later in the paper.

Results on tests conducted on “*good*” locations

- *Impact of the RAN software:* Throughout our tests we observed that *srsRAN* delivers much higher UL rates, while *OAI-RAN* performs better on the DL, regardless of the type of the SDR and the number or the type of connected UEs.
- *Impact of the type of UE:* Our tests indicate that for almost all of the single/multi-UE scenarios, irrespective of the RAN software, the two modem-based UEs (UE_2 , and UE_3), outperform the phone in terms of DL rates. Focusing on *srsRAN*, we observe that in single-UE scenarios ($T1$ vs. $T2$ & $T3$), UE_1 is receiving 93% of the DL rate of UE_2 and UE_3 . Similarly, UE_1 receives 92% and 90% of the DL rate of UE_2 and UE_3 , when two UEs are connected at the same time, in $T8$ & $T9$. When it comes to $T15$ (corresponding to the three UEs case), UE_1 is receiving only 84% of the DL rate achieved by the other two modem-based UEs. This pattern is also evident in the results achieved by *OAI-RAN* even if in the single-UE scenarios, the difference between the DL rates of UE_1 and UE_3 is negligible. Indeed, there is an 8% gap between the DL rates of UE_1 and UE_2 . Moving to the multi-UE scenarios with *OAI-RAN*, we see that the gap between the DL rate of UE_1 and the other two

Table 3.6: Downlink and uplink data rate results for different tests corresponding to different locations and S_{RAN} (in Mbps)

Test #	S_{RAN}		$srsRAN$		OAI	
	<i>Location</i>	<i>UE Type</i>	<i>DL</i>	<i>UL</i>	<i>DL</i>	<i>UL</i>
Single-UE scenarios						
T1	A1	UE1	86	39.1	101	24.5
T2	A2	UE2	92.1	39.9	109	24.4
T3	A3	UE3	92.3	41.3	102	20.3
T4	D	UE1	7.66	3.23	48	15.9
T5	D	UE3	51.9	30.3	66	19.5
T6	E	UE1	3.03	1.5	23.3	15.6
T7	E	UE3	23.1	29.3	35.1	14.3
Two-UE scenarios						
T8	A1	UE1	42.5	18.9	58.8	13.8
	A2	UE2	45.7	17.9	65.1	11.8
T9	A1	UE1	41.2	17.6	62.9	15.5
	A3	UE3	45.5	19.9	66.8	14.3
T10	A2	UE2	46.9	16.8	57.7	12.8
	A3	UE3	46.8	21.2	60.2	15
T11	D	UE1	6.51	5.28	19.4	10.2
	A3	UE3	44.5	19.9	52.9	12.9
T12	A1	UE1	31.8	17.7	49.6	14.2
	D	UE3	27.8	9.87	33.9	11
T13	D	UE1	2.18	3.02	14.3	10.1
	A2	UE2	34	13	50.3	11
T14	A2	UE2	43.7	16.8	53.7	11.8
	D	UE3	16.3	14.9	35.8	12.6
Three-UE scenarios						
T15	A1	UE1	26	10.2	41.2	11.3
	A2	UE2	30.6	10.9	51.7	8.44
	A3	UE3	30.9	14.2	53.1	9.61
T16	D	UE1	5.19	4.22	14.3	6.5
	A2	UE2	20.1	12.6	38.9	5.88
	E	UE3	8.16	7.22	13.3	7.79
T17	E	UE1	X	X	X	X
	A2	UE2	X	X	X	X
	D	UE3	X	X	X	X

modem-based UEs increases. In two-UE scenarios, T8 and T9, UE₁ achieved 90% and 94% of the DL rate of UE₂ and UE₃, respectively. When all UEs are connected in T15, UE₁ is only able to receive 79% and 77% of the DL rate achieved by UE₂ and UE₃, respectively. The results on the UL are difficult to interpret since on single-UE scenarios, UE₃ does better than the other UEs for *srsRAN* and worse for *OAI-RAN*.

- *Cases with multiple UEs:* Throughout our multi-UE tests in “good” locations, we observed that both S_{RAN} do seem to share the resources *roughly* equally among the UEs in both UL, and DL directions. For instance, comparing T3, T8, and T15 in the UL direction, we see that if *srsRAN* is used, the maximum UL rate achieved in a single-UE scenario is 41.3 Mbps. In the two-UE scenario (T8), UE₁ and UE₂ receive 45% and 43% of this rate, respectively. Moreover, in T15, when all UEs are connected, UE₁, UE₂, and UE₃, they can send 24%, 26%, and 34% of the maximum achieved UL rate in the single-UE scenario. If *OAI-RAN* is used, the maximum UL rate achieved in single-UE scenarios is 24.5 Mbps. We see that in T8, UE₁ and UE₂ each can send 56% and 48% of this maximum rate, respectively. In T15, UE₁, UE₂, and UE₃ each can transmit 46%, 34%, and 39% of the maximum UL rate. While *OAI-RAN* performs poorly in the UL, it shows better resource sharing capabilities in multi-UE scenarios.

Observations on tests conducted on “good” locations

- *Impact of the type of UE:* Overall, we found it easier to work with modem-based UEs. While *srsRAN* did not exhibit any apparent differences in the attachment process of the UEs, when working with *OAI-RAN*, we observed that the phone, UE₁, had a harder time attaching to the testbed and maintaining its connection for three minutes during the tests. We did not observe such a behaviour with the two modem-based UEs, while connected to *OAI-RAN*.
- *Effect of multiple UEs:* Throughout our multi-UE tests in “good” locations, we noticed that *OAI-RAN* crashed several times, specifically when the tests were done on the UL direction.

Results on tests conducted on “fair” and “bad” locations

- *Impact of location:* The overall observation is that there is a significant difference between the two RAN software platforms: while *OAI-RAN* adjusts the power automatically, *srsRAN* provides a static power setting mechanism, which is through

setting the Tx/Rx gain values. This is difficult to adjust in scenarios with multiple UEs. We carefully selected the Tx and Rx gains that resulted in the best achieved rates in “good” positions, and maintained those values for the gNB throughout all our tests for all testbeds, and in that case, *srsRAN* loses its superior performance, in the presence of multi-UEs, each located on a different type of position. As for *OAI-RAN*, the automatic power adjustment feature provides some consistency in the results based on the locations of the UEs. We see from T1, T4, and T6 that moving UE₁ from A1 to D, and to E causes a 52% and 76% drop in the DL data rate, and 35%, and 36% in the UL rate, respectively. Looking at UE₃ (T3, T5, T7), we see that the drop in the DL (UL respectively) rates from point A3 to D is 40% (20% respectively), and from A3 to E is 67% (40% respectively). *OAI-RAN*’s performance is thus not impacted by the location of the connected UE in the UL direction as much as it is in the DL. The same trend is also seen in the multi-UE scenarios with *OAI-RAN*.

- *Impact of the type of UE*: The most unexpected observation for us was the fact that *srsRAN* seems to provide very limited coverage for UE₁. Comparing T4 and T5, as well as comparing T6 and T7, you can see that while UE₃ is able to achieve 51 and 23 Mbps in DL, the phone is only getting around 7 and 3 Mbps respectively. This trend is also evident in multi-UE scenarios with *srsRAN*. On the *OAI-RAN* side, we did observe this trend, but it was much less pronounced. For instance, comparing T4 and T5, we see that the phone is getting 48 Mbps in the DL, while the modem is achieving around 66 Mbps. There is certainly a difference, but it is nowhere near as drastic as the gap observed with *srsRAN*.
- *Effect of multiple UEs*: As mentioned in the previous points, since *srsRAN* does not support automatic power adjustment, the results in “*fair*” and “*bad*” locations are very poor. Additionally, the unforeseeable discrepancy between the performance seen by UE₁ and UE₃ in those locations has made this category of results for *srsRAN* to be haphazard and inconsistent, not revealing any kind of pattern. However, *OAI-RAN* was more reliable. *OAI-RAN* seems to share the Physical Resource Blocks (PRBs) equally among users, and the final rate achieved by the UE is then determined by the MCS value. For instance, taking a look at T3, T4, and T11, we see that in the case of T11, the two-UE scenario, each UE is receiving roughly half of what they used to get at the same location in the single-UE scenario, e.g., UE₁ which was receiving about 48 Mbps at D in the single-UE scenario, is now receiving 19.2 Mbps when UE₃ is also connected from location A3, and UE₃ which was receiving about 102 Mbps at A3 in the single-UE scenario, is now receiving 52.9 Mbps at location D, in the

two-UE scenario.

Observations on tests conducted on “fair” and “bad” locations

- *Effect of multiple UEs:* We have to mention that for both software platforms, we were unable to conduct T17. Despite attempting the test more than seven times, UE₁ was unable to maintain its connection for the duration of the test (three minutes) for each data rate test at location *E*.

Second round of tests to compare the SDRs

For the second round of tests, we configured all the tests to be done on “good” locations. The results are presented in Table 3.7. In our tests, we observed that between the two SDRs, *USRP X410* yielded better UL rates. Based on the results presented, *OAI-RAN* shows the largest improvement in UL rates when *USRP X410* is utilized (up to 48% improvement). For instance, in a single-UE scenario with UE₃, using {*USRP B210*, *OAI-RAN*} results in 20.3 Mbps in UL, whereas using {*USRP X410*, *OAI-RAN*} results in a 45% improvement, reaching 29.5 Mbps. In the same scenario, if *srsRAN* is used, changing the SDR from *USRP B210* to *USRP X410* results in no improvement. Additionally, during single-UE scenarios, we observe the best DL performance with *USRP X410*. As an example, when UE₁ is the only connected UE, the combination of {*USRP X410*, *OAI-RAN*} results in 13% improvement in DL rate, compared to {*USRP B210*, *OAI-RAN*}. For the same scenario and UE, the combination of {*USRP X410*, *srsRAN*} improves the DL rate by 8% compared to {*USRP B210*, *srsRAN*}.

3.6.2 Methodology & Results for Latency Assessment

Methodology

To analyze the E2E latency, we followed the same methodology as for the data rate assessments, as in we used the same testbeds, elements and configurations described in the data rate assessments. We also used the same single/multi-UE scenarios with different types of locations. The latency tests were done using the *ping* command. For each assessment, a *ping* command from the measuring UE to the IP address of the AMF in the core network was conducted. Based on our observations, having multiple connected UEs did not affect the E2E latency experienced by each UE. Furthermore, the positions of the connected

Table 3.7: Data rate results for two SDRs (in Mbps)

<i>SDR</i>	<i>USRP B210</i>				<i>USRP X410</i>			
S_{RAN}	<i>srsRAN</i>		<i>OAI</i>		<i>srsRAN</i>		<i>OAI</i>	
	<i>DL</i>	<i>UL</i>	<i>DL</i>	<i>UL</i>	<i>DL</i>	<i>UL</i>	<i>DL</i>	<i>UL</i>
Single-UE scenarios								
UE1	86	39.1	101	24.5	92.9	40.4	115	25.5
UE2	92.1	39.9	109	24.4	92.1	40.9	115	33.3
UE3	92.3	41.3	102	20.3	92.9	41.3	115	29.5
Two-UE scenarios								
UE1	42.5	18.9	58.8	13.8	45.3	17.7	66.9	17.2
UE2	45.7	17.9	65.1	11.8	46.8	21.3	66.9	17.5
UE1	41.2	17.6	62.9	15.5	48	16.7	62.2	18.2
UE3	45.5	19.9	66.8	14.3	44.5	20.2	62.6	16.5
UE2	46.9	16.8	57.7	12.8	48.5	21.3	61.3	16.4
UE3	46.8	21.2	60.2	15	45.1	20	61.4	14.9
Three-UE scenario								
UE1	26	10.2	41.2	11.3	28.4	13	46.1	8.1
UE2	30.6	10.9	51.7	8.44	24.3	13.8	46.1	11.3
UE3	30.9	14.2	53.1	9.61	30.8	13.9	46	10.3

UEs also did not appear to influence the final result. Therefore, to simplify comparisons, we only report on the latencies achieved by different types of UEs in single-UE scenarios, where each UE is located in a “good” position. Table 3.8 presents our results.

Impact of the RAN software

Similar to the data rate tests, we observe the impact of the choice of S_{RAN} on the E2E latency. Although recent updates for both software platforms have brought the data rates closer together, a significant difference in the latency results persists. *OAI-RAN* outperforms *srsRAN* in this regard. Comparing the latency results achieved by UE₁, when connected to *USRP B210*, we find that when we have the shortest E2E latency results for both S_{RAN} , the result achieved by *OAI-RAN* is 70% shorter than that of *srsRAN*.

Table 3.8: E2E latency between the UE and 5GC (in ms)

SDR	<i>USRP B210</i>		<i>USRP X410</i>	
S_{RAN}	<i>srsRAN</i>	<i>OAI-RAN</i>	<i>srsRAN</i>	<i>OAI-RAN</i>
UE1	27	8	29	8
UE2	31.8	10.4	31.2	11.3
UE3	33	13	34	15

Impact of the type of UE

Table 3.8 shows that the type of UE plays some role in the final E2E latency. Notably, UE₁ consistently achieved lower latency compared to the modem-based UEs. Additionally, within the modem-based UEs, UE₂, which is a modem connected to a Linux system shows slightly superior performance compared to UE₃, the same modem, connected to a Windows laptop.

Comparison between the SDRs

In our tests, we found that other than for two scenarios, the measured latencies were relatively consistent regardless of the SDR device used. The two exceptions are, the latency result achieved by UE₁ with $\{USRP\ B210, srsRAN\}$ is 6.9% less than with $\{USRP\ X410, srsRAN\}$, the latency result achieved by UE₃ with $\{USRP\ B210, OAI-RAN\}$ is 13% less than with $\{USRP\ X410, OAI-RAN\}$.

3.6.3 Methodology & Results for Coverage Assessment

Methodology

We assess the coverage of four testbeds using UE₁, specifically focusing on the impact of the SDR and S_{RAN} . These included combinations of $\{srsRAN, OAI-RAN\}$ for the S_{RAN} , alongside $\{USRP\ B210, USRP\ X410\}$. All the measurement locations are indicated on the map, as shown in Fig. 3.12, by purple circles. At each location, we utilized the *5G Switch - Force 5G Only* application on UE₁ and recorded the RSRP values in Table 3.9 (Please note that any cell with X as their value in the table, indicate a disconnection).

Impact of the software

Due to the automatic power adjustment feature, *OAI-RAN* outperforms *srsRAN* in terms of coverage support. A notable observation was that in areas with very poor signal strength, such as points *L*, *G*, or *M*, *OAI-RAN* exhibited a tendency to crash frequently. In contrast, *srsRAN* gNB at such points continue to run, but the UE is unable to find any connections.

Comparison between the SDRs

Comparing the coverage maps of the two SDR devices, it is evident that *USRP X410* exhibits a superior performance. However, the impact of the S_{RAN} is more influential. Specifically, the testbed utilizing $\{OAI-RAN, USRP B210\}$ exhibits a better efficacy compared to $\{srsRAN, USRP X410\}$.

3.6.4 Methodology & Results for Computational Resource Consumption of the Open-source RAN Software

Methodology

There are two aspects to the analysis of resource consumption for the open-source software platforms. First, how each RAN software consumes the available computational resources, and second, how the choice of the host PC affects the performance achieved by the UEs. To answer the first question, we set up testbeds using $\{srsRAN, OAI-RAN\}$ as their S_{RAN} , *Open5GS* as S_{5GC} , and set the host PC for the software platforms as PC #1. Please note that PC #1 is our more powerful computer with 16 CPU cores, running at the frequency of 3.5 GHz. We are specifically interested in determining the maximum resource consumption by each S_{RAN} in a full buffer scenario in both UL and DL transmissions, for single and multi-UE scenarios. This allows us to gain insight into the worst-case computational consumption scenario for each software platform.

Observations on CPU utilization

Please see Fig. 3.13, that shows the CPU utilization as a function of the number of connected UEs.

Table 3.9: Signal strength measurements (in dBm)

S_{RAN}	<i>srsRAN</i>		<i>OAI-RAN</i>	
<i>SDR</i>	<i>B210</i>	<i>X410</i>	<i>B210</i>	<i>X410</i>
A2	-100	-84	-92	-74
B	-109	-100	-107	-85
C	-109	-101	-107	-91
D	-119	-110	-113	-102
E	-125	-114	-119	-106
F	-131	-121	-122	-106
G	X	-127	X	-108
H	-124	-115	-116	-106
I	-130	-119	-120	-111
J	-118	-113	-115	-99
K	-127	-120	-125	-111
L	-135	-127	X	-114
M	X	X	X	-117
N	X	X	X	-116
O	X	X	X	-120

- *Traffic direction effect, UL vs. DL:* At first glance, we observe a trend with *OAI-RAN*: the DL traffic appears to be less CPU-hungry than the UL. This trend is reversed for *srsRAN*, i.e., the UL traffic seems to require fewer CPU cores. Another important observation is the significant difference in computational resource consumption between *srsRAN* and *OAI* in the DL direction. With a single UE, *OAI* consumes 73% of one CPU core, whereas *srsRAN* consumes 1.2 CPU cores, indicating a drastic gap in resource utilization (Please note that none of the S_{RAN} utilized more than 2 cores of PC #1).
- *Effect of multiple UEs:* There is a gradual increase in the resource consumption for both S_{RAN} as the number of connected UEs increases.

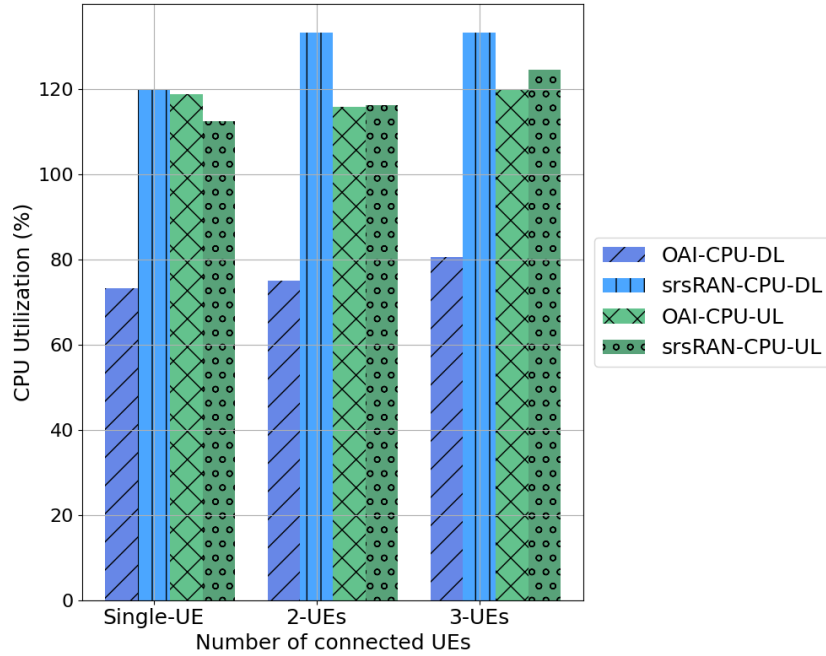


Figure 3.13: CPU utilization as a function of number of UEs for OAI and srsRAN in UL and DL

Observations on memory utilization

Our results indicate that the CPU is the primary bottleneck resource, while memory usage remains relatively consistent. In all scenarios, *OAI-RAN* consumed a maximum of 3% of

Table 3.10: The effect of host PC computational power

Performance Metric		Data Rate (in Mbps)		Latency (in ms)	
S_{RAN}	Host PC	DL	UL	Min	AVG
srsRAN	PC #1	92.1	39.9	12	27
srsRAN	PC #2	46.9	39.9	11	37
OAI-RAN	PC #1	109	24.4	7.6	10.1
OAI-RAN	PC #2	99.4	24.5	7.1	10.5

the memory, whereas *srsRAN* never exceeded 10.2% of the memory.

Observations on the effect of the host PC

For this round of tests, utilizing the less powerful PC, PC #2, we established two additional testbeds with the two open-source RAN platforms and UE₁. This setup aims to compare the data rate and latency results with those obtained previously using PC #1 as the host PC. Our primary interest lies in observing the performance of each S_{RAN} under varying CPU budgets, as highlighted in the previous section, where it was noted that not all CPU cores were fully utilized by any of the S_{RAN} platforms. Please note that, PC #1 operates at a base frequency of 3.5 GHz, while PC #2 runs at a frequency of 2.9 GHz. The results are presented in Table 3.10.

- *Impact of the host PC on srsRAN:* There is a significant difference in the achieved performance of *srsRAN* when changing the host PC. This effect is mainly evident in the DL rate and the E2E latency experienced by the UE. When *srsRAN* is running on PC #2, the DL rate drops to almost half of what is achievable when running *srsRAN* on PC #1 (from 92.1 Mbps to 46.9 Mbps). Additionally, the average latency increases by 37%. These results are consistent with the findings reported in [48], where the authors report achieved DL and UL rates of 123 Mbps and 39 Mbps, respectively, using a host PC more powerful than our PC #1.
- *Impact of the host PC on OAI-RAN:* *OAI-RAN* appears to be less sensitive to the host PC, as the change in achieved performance when switching the host PC is not as significant as it is with *srsRAN*. Although there is about a 9% increase in the DL rate, when using the more powerful PC, the UL rate and average latency remain almost the same.

3.7 Conclusion

In this chapter, we presented a comprehensive assessment of the performance achieved by UEs connected to 5G-SA single-cell experimental testbeds. We built, analyzed, and compared 35 distinct testbeds using well defined metrics, and measurement tools. Moreover, we outlined a precise methodology for conducting tests in each section. Our discussions into the performance of each testbed in both single and multi-UE scenarios, highlighted:

- The challenges we faced while implementing our initial testbed and the lessons we learned.
- The interoperability assessments of open-source 5G software platforms, with respect to the two open-source RAN and 5GC software platforms were **successful**. Moreover, we provided the configuration parameters, that allowed for successful implementation of “multi-source” testbeds.
- The performance achieved by the UE attached to testbeds with two different 5GC software platform, while fixing all other elements in all other parts, does not show a noticeable dependency on the specific open-source 5GC software.
- The performance of 5G open-source software and COST hardware for the RAN, across 30 5G-SA testbeds. We observed that between the two open-source RAN platforms, srsRAN provides a more robust UL connection and higher rates. However, in the DL, OAI-RAN outperforms srsRAN. Additionally, OAI-RAN leads in terms of E2E latency. For the two SDR devices, we found that USRP X410 provided more coverage and better performance for the UE compared to USRP B210.
- How the type of RAN software and location of each connected UE impact the performance. Specifically, we observed that, due to the lack of automatic power adjustment in srsRAN, OAI-RAN provides superior coverage and data rates for the UEs located far from the SDR.
- Additionally, we explored the interoperability of different UE types with various hardware and software elements of the RAN. In this regard, we found that UE₁, the Android phone, often exhibits lower data rates compared to the other two UEs, which are both modem-based. Conversely, UE₁ achieved lower E2E latency in most scenarios.”

- Finally, we evaluated the computational resource consumption of each software platform in both single and multi-UE scenarios. We saw that, with close proximity between the UE and the SDR device, srsRAN can deliver exceptional data rates when hosted on a powerful computer. On the other hand, OAI-RAN's performance appeared to be less dependent on the computational resources of its host PC.

Chapter 4

Uplink Scheduling

This chapter focuses on enhancing the “*user scheduling*” process in *srsRAN*, one of the two open-source RAN platforms discussed in the previous chapter. This chapter also serves as a showcase of the practicality of experimental testbeds, particularly for the research community.

“*User scheduling*” is a critical procedure that allocates radio resources and power to UEs on a very short time scale. It is performed in both DL and UL, but the implementation is different for each direction, primarily due to differences in power budgets and interference management [63].

Unfortunately so far, the literature has been more directed on improving user scheduling for DL, with most studies relying heavily on simulations for their evaluations. To the best of our knowledge, only two papers [75, 41] have utilized experimental 5G-SA testbeds to assess or enhance scheduling algorithms. Notably, both have been limited to (i) the scheduling process on the DL direction, and (ii) employed *OAI-RAN* as their chosen RAN software platform. In the previous chapter, we observed that *OAI-RAN* exhibited poor performance in the UL direction compared to *srsRAN*. Therefore, considering the two points mentioned earlier, along with *OAI-RAN*’s UL performance, we decided to focus on studying the UL scheduling mechanism implemented in *srsRAN* and explore avenues to enhance it.

The existing scheduler implemented in *srsRAN* is a basic scheduler that considers only the number of users with UL traffic, and the number of bits to be transmitted when deciding how many resource blocks to allocate to each UE. In the case of a single-UE, assuming a full buffer scenario, this scheduler allocates all available channels to the UE.

In this chapter, we present our study to improve the UL scheduler implemented in the *srsRAN* platform by showing that a single UE might do better if allocated less PRBs in a time slot. We must inform the reader that the outcome of this chapter was a mitigated success. Due to unexpected limitations of the testbed, we were only able to evaluate our approach in single-UE scenarios; unfortunately, multi-UE tests could not be conducted in a satisfying way (i.e., with consistent results). Before diving into the details of our proposal, it is important to provide some context and background. The following sections are dedicated to introducing a few important concepts, followed by the literature survey.

4.1 Overview of 5G's Physical Channels

5G defines different types of physical channels for the exchange of data or control messages between the gNB and the UEs:

1. *Physical Downlink Shared Channel (PDSCH)* is mainly used to transmit data from the gNB to the UEs in the downlink.
2. *Physical Uplink Shared Channel (PUSCH)* is mainly used to transmit data from the UE to the BS.
3. *Physical Downlink Control Channel (PDCCH)* carries DL control information such as resource allocation, and scheduling information.
4. *Physical Uplink Control Channel (PUCCH)* carries UL control information such as the scheduling requests, or CSI.
5. *Physical Broadcast Channel (PBCH)* broadcasts the parameters from the base station to UEs to synchronize with the cell and establish the initial communication.
6. *Physical Random Access Channel (PRACH)* is used for initial access and random access procedures, allowing UEs to initiate communication with the network.

4.2 PRB, SINR, MCS, and the Rate

-*PRB*: A PRB is the smallest unit of the radio resources that can be allocated, in the 5G resource grid. For each SCS configuration, with its corresponding numerology, PRBs are numbered in each time slot, starting from 0 up to a maximum value determined by the size of the bandwidth in use. For instance, where the SCS is 30 kHz, a 40 MHz bandwidth consists of 106 PRBs. Please note that, in the UL out of the 106 PRBs, the last three are set for the exchange of control messages over PUCCH. 3GPP has listed the number of PRBs in [34], as in Table 4.1:

Table 4.1: Maximum transmission bandwidth configuration for FR1

SCS (kHz)	Bandwidth (MHz)														
	5	10	15	20	25	30	35	40	45	50	60	70	80	90	100
15	25	52	79	106	133	160	188	216	242	270	N/A	N/A	N/A	N/A	N/A
30	11	24	38	51	65	78	92	106	119	133	162	189	217	245	273
60	N/A	11	18	24	31	38	44	51	58	65	79	93	107	121	135

-*SINR*: Signal to Interference Noise Ratio (SINR) represents the ratio of the useful received signal power to the combined interference and noise power per-PRB, and serves as an indicator of the quality of the received signal. It is defined for user u , on subchannel c , in time slot t as:

$$\gamma_u^{c,t} = \frac{P_u^{c,t} \times G_u^{c,t}}{\sigma^2} \quad (4.1)$$

Where $P_u^{c,t}$ is the transmit power used by user u , on subchannel c , in time slot t , and $G_u^{c,t}$ is the channel gain between user u and BS, on subchannel c , in time slot t , and σ^2 is the noise plus interference at BS. Obviously, the higher the SINR, the better the signal quality and channel conditions. In srsRAN, there is one value of SINR per user, per time slot, which means that:

$$G_u^{c,t} = G_u^t, \quad \text{and} \quad P_u^{c,t} = P_u^t$$

Moreover, the per-timeslot power budget of user u in time slot t , P_u^t is distributed equally on all the subchannels allocated to user u , in time slot t . Hence, if user u is allocated m_u^t PRBs, in time slot t , then:

$$\gamma_u^{c,t} = \frac{P_u^t \times G_u^t}{m_u^t \times \sigma^2} \quad (4.2)$$

Note that m_u^t is selected by the UL scheduler implemented in the gNB given an SINR. The rate can be obtained using what is described next. Please note that, we will remove the index for time slot t in the following.

-*MCS*: Modulation and Coding Scheme (MCS) determines the modulation type and coding rate to use for a data transmission based on the channel quality. When the SINR is high, higher-order modulation schemes (e.g., 64-QAM or 256-QAM) and higher coding rates are selected to transmit more data bits per PRB. Conversely, when SINR is low, indicating poor channel conditions, the system switches to lower-order modulation schemes and lower coding rates, which are more robust to noise and interference but transmit fewer bits per PRB. This dynamic adjustment ensures data is efficiently transmitted, taking the environment condition into account.

To ensure that the UE follows a well-defined procedure to determine the appropriate MCS for the PUSCH, the mapping between the MCS and the corresponding modulation order (Qm) and target code rate, is standardized by 3GPP, as defined in [33] and specified in Table 5.1.3.1-1. However, the mapping between specific MCS values and their corresponding SINR ranges is not standardized by 3GPP and is vendor-specific. Table 4.2 is the MCS table which has been implemented in *srsRAN*. This table can be found in the source code at:

`srsRAN_Project/lib/scheduler/support/mcs_calculator.cpp`, available at [19].

This table shows the minimum required SINR for each MCS index. As mentioned by the *srsRAN* developers, the following steps have been taken to derive the minimum required SINR value for a given MCS index, using a test scenario with Single Input Single Output (SISO) configuration over a 20 MHz bandwidth with TDD duplexing mode:

1. To determine the minimum required SINR value for a specific data setting, the gNB and UE are connected using *ZeroMQ* [24]. *ZeroMQ* is a virtual radio interface that transfers radio samples between two applications, i.e., the gNB, and the UE. This connection is set to have varying SINR and channel conditions.
2. Initiate UL traffic from the UE to the 5GC.
3. Monitor the Block Error Rate (BLER) during periods when the SINR remains constant and confirm that the BLER is 0% for the selected MCS.
4. If errors are detected (BLER is not 0%) in step 3 then, increase the SINR value in the table corresponding to the MCS chosen.

It is important to note that Table 4.2 has a critical impact on the performance.

Table 4.2: The uplink MCS-SINR (64 QAM) mapping used in srsRAN

MCS Index	SNR (dB)	Rate (bps/Hz)	MCS index	SNR (dB)	Rate (bps/Hz)
0	-5.7998	0.2344	15	10.4000	2.4063
1	-3.5500	0.3066	16	11.1540	2.5703
2	-2.9250	0.3770	17	12.1070	2.5664
3	-2.5625	0.4902	18	12.5250	2.7305
4	-1.0500	0.6016	19	13.0625	3.0293
5	0.98266	0.7402	20	13.5250	3.3223
6	1.6250	0.8770	21	13.9375	3.6094
7	2.5425	1.0273	22	14.1160	3.9023
8	3.4175	1.1758	23	14.5525	4.2129
9	4.3548	1.3262	24	14.9725	4.5234
10	5.3695	1.3281	25	15.3450	4.8164
11	5.8250	1.4766	26	15.9175	5.1152
12	6.6375	1.6953	27	16.0425	5.3320
13	7.6375	1.9141	28	16.591	5.5547
14	9.5875	2.1602			

4.3 Scheduling Process as Specified by 3GPP

Scheduling, in its simplest form, can be defined as the coordination process between the gNB and the UEs to facilitate data transmission.

Scheduling plays a critical role in determining the achieved throughput by the UEs. While scheduling algorithms are proprietary and not defined by 3GPP [51], an efficient scheduler is crucial for maximizing radio resource (i.e., PRB) utilization and ensuring high throughput for UEs. The scheduling procedure involves the UE sending measurements to the gNB, providing the gNB with information about their data buffer status and QoS requirements.

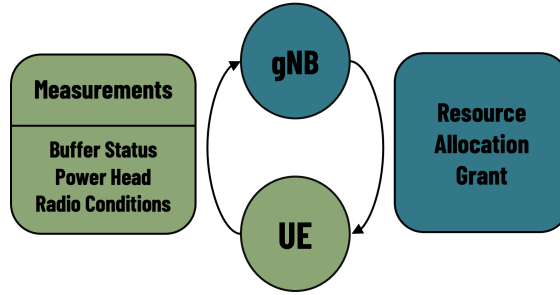


Figure 4.1: Scheduling flowChart

The scheduling algorithm then takes this information along with information measured locally such as the UE radio conditions as input, and then decides how to assign PRBs to specific UEs in a given time slot.

Measurement reports are essential for enabling the scheduler to operate effectively. For instance, when it comes to the buffer status, i.e., the indicator of how much data is going to be transmitted, in the DL, the gNB inherently knows the buffer status for each UE, while in the UL, the UE is the node with the data and has to transfer its buffer status to the gNB to inform the scheduler about its transmission. The radio condition experienced by the UE is also an important factor, since it dictates the SINR along with the number of allocated PRBs.

Downlink Scheduling: In the downlink, the gNB dynamically allocates resources to each UE and announces the scheduling outcome via the PDCCH. Each UE continuously monitors the PDCCH for potential assignments. In the case of latency-sensitive data, the gNB may pause an ongoing PDSCH transmission to a UE and immediately schedule the transmission of the latency-sensitive data to the designated UE.

Uplink Scheduling: The core concept of scheduling is similar in both the UL and DL. However, there are key differences: i) while in the downlink, the scheduler has extensive information readily available (e.g., data buffer status), in the uplink, the gNB relies on measurements reported by the UEs; and ii) the power budget for transmission is also a crucial factor. In the downlink, this information is known to the scheduler since the power comes from the gNB. In contrast, in the uplink, each UE has its own power budget. Similar to the downlink, the gNB dynamically allocates resources to UEs in the uplink via the PDCCH. Each UE continuously monitors the PDCCH for potential grants for uplink transmission.

It is important to emphasize that scheduling algorithms are not defined in the 3GPP specifications. While the standard outlines the signaling flow between the gNB and the UEs to support scheduling processes and includes provisions for measurement collection on data buffers and radio conditions, it deliberately avoids prescribing how this information should be used. The design and implementation of scheduling algorithms are left to vendors, who are tasked with developing solutions that not only maximize data throughput within the cell but also address specific priorities, unique to each of them (such as higher priority for low-latency services).

4.4 Power Control

Power control is a technique used to adjust the transmit power of a UE to maintain signal quality at the gNB, while limiting inter-cell interference, and control the UE's power consumption [32]. Like 3G and 4G, 5G uses open-loop and closed-loop mechanisms to adapt power levels based on radio conditions and feedback.

Open Loop Power Control (OLPC) refers to the UE u 's ability to set its transmit power based on the values m_u and α_u , which are provided by the gNB, and represent the number of channels allocated to the UE, and the path loss compensation factor (assuming channel reciprocity), respectively. This means that the power budget P_u , available for use by the UE, is determined as follows:

$$P_u = \min(P_{\max}, \Delta(P_{PL}^u)^{\alpha_u} m_u) \quad (4.3)$$

where $P_{\max}$ is the UE's maximum power per time slot, Δ is the target receive power, and P_{PL}^u is the path loss power. Please note that the values of P_{PL}^u and Δ are known to both the gNB and the UE.

This method allows the UE to estimate its appropriate transmit power level using only m_u , and α_u , which facilitates initial access and helps maintaining communication quality in varying radio conditions.

Closed Loop Power Control (CLPC), on the other hand, involves a dynamic adjustment of a UE's transmit power budget based on a special feedback from the gNB. In this approach, the base station measures the SINR, compares it to the target SINR, considers the reported Power Headroom Report (PHR) from the UE, and sends a Transmit Power Control (TPC) command instructing the UE to increase or decrease its transmit power accordingly. PHR is a UE measurement which shows the difference between the nominal UE maximum transmit power and the estimated power for uplink transmission, and can be used to provide support for power-aware scheduling.

In CLPC, the power budget P_u , that can be used by the UE is determined by:

$$P_u = \min(P_{\max}, \Delta(P_{PL}^u)^{\alpha_u} m_u + TPC_u) \quad (4.4)$$

where $P_{\max}$ is the UE's maximum power per time slot, Δ is the target receive power, P_{PL}^u is the path loss power, α_u is the path loss compensation factor, m_u is the number of channels allocated to UE u , and TPC_u is the TPC correction factor. Here, m_u , α_u , and TPC_u are all provided to the UE by the base station. CLPC enables real-time adjustment of transmit power, effectively compensating for rapid changes in the radio environment.

The combination of OLPC and CLPC mechanisms provides a strategy for managing transmit power in wireless networks. While open loop control offers a quick power setting, closed loop control and its feedback-based adjustments, is great for fine-tuning the transmit power to account for real-time variations in the communication channel, such as fading.

Please note that srsRAN_Project v23.10.1 does not include any power control among its supported features, i.e., the UE always uses its full power budget in a time slot when it is allocated PRBs. In fact, to verify the status of the power control mechanism in srsRAN, we delved deeper into the code and discovered that the specific version of srsRAN we used implements OLPC with a very large value for “*target receive power*”, which is Δ in Equation 4.3. Please take a look at line 440 in:

```
srsRAN_Project/lib/scheduler/config/serving_cell_config_factory.cpp
```

We see that the value set for Δ is extremely high (-76 dBm). With such a high value, equation 4.3 always returns $P_{\max}$, which effectively disables OLPC, and forces the UE to transmit at maximum power. Our measurements confirm this.

4.5 Uplink Scheduling in the Literature

Scheduling algorithms have been extensively studied for many years, resulting in a rich body of literature. This includes a wide range of publications, ranging from tutorials, and surveys [63, 49, 59, 57, 76], to new algorithm proposals [78, 67, 72, 56, 70, 52, 53, 47, 61, 63, 71, 54].

As mentioned before, the literature is more abundant for the DL (e.g., [77, 35]), with many papers introducing innovative downlink schedulers over the years. In contrast, the research on the UL is considerably more limited, which can be attributed to the inherent complexity of UL user scheduling. Moreover, the number of experimental studies [75, 41, 73] remains exceptionally scarce and focuses predominantly on the DL direction.

Scheduling aims to balance throughput, fairness, and QoS requirements. Over the years, various scheduling algorithms have been developed, ranging from simple metric-based methods to more sophisticated Machine Learning (ML), and optimization-based approaches.

Metric-based scheduling [78, 67, 72, 56, 70] is one of the simplest and most widely implemented techniques. It relies on predefined metrics, which may include UE-side measurements, such as SINR, or Head-of-Line delay (the time a packet spends in the buffer before being sent), to make scheduling decisions. The scheduling process involves ranking the users according to their metric priority and generating a mapping table that links each PRB to a specific UE. For example, Max-Channel Quality Indicator (CQI) scheduling prioritizes users with the best channel quality for maximizing throughput [67], while Round Robin (RR) scheduling is a simple scheme that treats all users equally, ensuring fairness without taking channel conditions into account [78]. Moreover, another popular metric-based scheduling approach is max-rate, where the scheduler's goal is to maximize the system's throughput without taking the performance of individual UEs into account [72, 56]. Also, delay-based techniques like Largest Weighted Delay First (LWDF) [70], are channel-unaware, and designed for use-cases like Ultra-reliable Low Latency Communications (URLLC). They prevent deadline expiration by incorporating the Head-of-Line delay as the scheduling metric. Metric-based scheduling is computationally efficient and highly suitable for real-time systems due to its low complexity. However, it may lead to sub-optimal system performance as it focuses on local, user-specific decisions rather than the global network state.

Optimization-based scheduling [52, 53, 47, 61, 63] takes a more systematic approach, formulating resource allocation as an optimization problem to maximize or minimize specific objectives, such as network throughput, fairness, or energy efficiency. This approach

considers the entire system state (i.e., all relevant network parameters and conditions) to be known. Therefore, by using tools like linear programming, convex optimization, or heuristic algorithms, optimal solutions under constraints like power limits and QoS requirements can be obtained. For example, an optimization-based scheduler may allocate resource blocks in a way that maximizes the total throughput of the system [52, 53], while another approach can add constraints like ensuring minimum bandwidth for all users to the same objective [61, 47]. While optimization-based scheduling techniques deliver superior performance in theory, they are computationally intensive and challenging to implement in real-time, especially in dynamic networks with many users.

ML-based scheduling [62, 71, 54] introduces a data-driven perspective, utilizing ML models to predict and make scheduling decisions based on historical and real-time network data. Techniques such as supervised learning can be employed to predict user throughput or delay, while Reinforcement Learning (RL) models can learn policies to optimize long-term network performance, e.g., [62]. A deep learning-based approach that dynamically adapts uplink resource allocation to maximize throughput while considering the coupled binary constraint (which ensures that each PRB can be allocated to a single UE at most) is proposed in [54]. While ML-based approaches are highly adaptive and capable of managing complex, nonlinear relationships in network parameters, their performance is closely tied to the quality and diversity of the training dataset, as well as the availability of sufficient computational resources.

The key differences between these approaches lie in their complexity, adaptability, and system-wide optimality. Metric-based scheduling is fast and straightforward but may sacrifice performance for simplicity. Optimization-based scheduling provides optimal solutions but is computationally expensive and not practical for real-time applications. ML-based scheduling offers a balance, combining adaptability and intelligent decision-making, but relies on data and training, which can be resource-intensive.

One of the main objectives of scheduling algorithms is to maximize a rate-based utility function, making the choice of the rate function (that maps the SINR into an MCS) critical to their performance. A key distinction in this context lies between the theoretical and practical perspectives. The Shannon capacity formula for Gaussian channels that computes the rate as $B \log_2(1 + \text{SINR})$, where B is the bandwidth, is widely adopted in the literature as a benchmark due to its simplicity and nice properties. This formula assumes ideal conditions, such as infinite-length codewords generated from a normal distribution. However, practical systems, including open-source implementations of 5G RAN platforms, operate under different constraints, where achievable rates are limited to a discrete set defined by predefined MCS. Table 4.2 is an example of discrete MCS, from *srsRAN_Project*, alongside the corresponding SINR thresholds. Specifically, each MCS level is associated

with a unique SINR threshold, and an MCS level is selected only when the estimated SINR exceeds its threshold. This results in a piecewise constant rate function, as depicted in Fig. 4.2, which accurately reflects the constraints and realities of practical implementations.

Looking at Fig. 4.2, one key difference between the two rate functions is when the SINR falls below the minimum value required by the MCS table (e.g., -5.7 dB in the case of *srsRAN_Project*). In this situation, while the Shannon rate function still indicates a positive rate, meaning a non-zero rate is theoretically achievable, the UE will actually have no rate in practice because the MCS table does not support any rates below this SINR threshold. This key difference has an enormous impact on scheduling as will be discussed later.

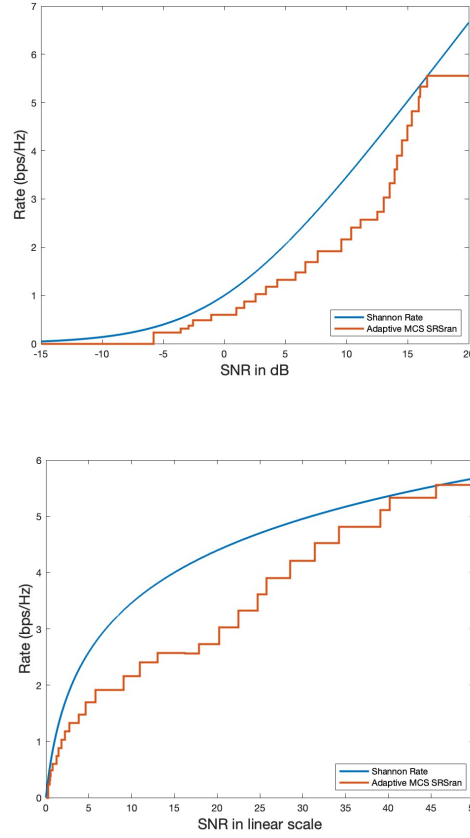


Figure 4.2: Achievable rate by Shannon vs. MCS function in linear and dB scales

4.6 UL Scheduling in srsRAN

srsRAN employs a simplified scheduling mechanism. For the UL, the scheduling process operates as follows:

1. First, the scheduling algorithm considers the buffer status reported by each UE.
2. Next, UEs with retransmissions get prioritized for an UL transmission over others, regardless of their buffer status. Then, based on the number of remaining UEs with UL traffic, and their reported buffer status and the scheduling policy, UEs are ranked. srsRAN supports two scheduling policies: round-robin (the default) and proportional fairness. The source code for the two scheduling policies, can be found at: `srsRAN_Project/lib/scheduler/policy` in the project’s Github page. Please note that, as we will explain later, some other parts of the srsRAN code also affect the output of the scheduler, regardless of the policy used, making understanding the scheduler challenging.
3. Finally, the scheduler allocates the remaining PRBs after allocating PRBs for re-transmissions, to UEs based on the ranking in the previous step.

It is important to note that srsRAN does not consider the channel condition, e.g., the SINR experienced by the UE in its scheduling RR algorithm (it does it with the Proportional Fairness (PF) algorithm, which will be discussed later).

4.6.1 Round Robin

Round Robin (RR) allocates equal number of PRBs to the UEs that are set to have an UL transmission for the next time slot. RR is the most basic scheduling strategy. It ensures that an equal share of PRBs is allocated to each scheduled UE. The RR code is implemented in “`lib/scheduler/policy/scheduler_time_rr.cpp`”, and Algorithm 5, is a pseudo-code of the RR implementation in srsRAN.

Overall, this code schedules resources in a time-domain RR fashion, supports multiple UEs while adhering to specific constraints and priorities in terms of the resources. It begins by defining a structure `scheduler_alloc_limits`, which sets a limit on the number of UEs that can be scheduled per time slot. This is based on a heuristic that has been defined by the srsRAN team. Then, the function `compute_max_nof_rbs_per_ue_per_slot`, calculates the maximum number of PRBs to allocate per UE, considering the current pending

Algorithm 5 High-level overview of srsRAN's Round Robin Scheduler, full-buffer scenario

```
1: Input: UE database, Resource Grid
2: Output: Scheduled UL grants
3: for all UE in UE database do
4:   if UE has pending UL data then
5:     Fetch UL HARQ candidates
6:     if HARQ candidate available then
7:       Allocate PUSCH grant to the UE
8:     else
9:       Skip UE
10:    end if
11:  end if
12: end for
13: Update the number of available PRBs
14: Round Robin Scheduling:
15: Fetch the number of UEs to schedule in a time slot based on this heuristic lookup= {
16:   {3_DU_UEs, 1_UE_tobe_Scheduled},
17:   {7_DU_UEs, 3_UEs_tobe_Scheduled},
18:   {15_DU_UEs, 4_UEs_tobe_Scheduled},
19:   {32_DU_UEs, 6_UEs_tobe_Scheduled},
20:   {MAX_NOF_DU_UEs, 8_UEs_tobe_Scheduled}
21: };
22: Number of allocated PRBs to scheduled UEs = Number of available PRBs ÷
    Number of UEs to be Scheduled
```

transmission states of the UEs. The code also includes functions to fetch candidates for hybrid automatic repeat requests (HARQ), ensuring that retransmissions are prioritized over new transmissions. Finally, the `alloc_ul_ue` function is responsible for granting resource allocations to UEs, checking their status and available HARQs.

This implementation is not a typical RR strategy. As mentioned, there is a table that limits the number of active users in each time slot. For instance, if there are four UEs with new transmissions waiting for a grant for UL, only three will be selected for the next time slot, and the remaining PRBs from the retransmissions will be divided equally among them. Moreover, UEs with retransmissions cannot receive more PRBs than what their initial transmissions require.

4.6.2 Proportional Fairness

In PF scheduling, after allocating the resources to UEs with retransmission, each active UE not involved in retransmission is assigned a weight directly proportional to the bit rate it can achieve in the next time slot (which is a function of its SINR), and inversely proportional to its recent data usage. The UEs are then scheduled based on a weighted RR on a time slot basis, i.e., there is at most one UE selected in a time slot that is not involved in retransmissions. The PF algorithm balances maximizing spectral efficiency with fairness, ensuring that users who were neglected in previous rounds receive higher priority, hence addressing resource starvation and improving overall data rate distribution. The PF code is implemented in “`lib/scheduler/policy/scheduler_time_pf.cpp`”.

It is important to mention that although both scheduler strategies we just discussed are implemented in srsRAN, RR is set to be the default scheduler. However, adding the following lines to the configuration file of the gNB makes it possible to change the scheduler to PF:

```
cell_cfg:
  cell_cfg:
    sched_expert_cfg:
      policy_sched_cfg:
        pf_sched:
          pf_sched_fairness_coeff: 2.0
```

We decided to work with the RR scheduler due to its simpler implementation compared to the PF scheduler. Moreover, on a single-UE setup, the two schedulers (RR, and PF),

yield the same output in terms of the number of PRBs allocated to the UE. Meaning that, if the UE has UL traffic (full buffer scenario), both schedulers will allocate the entire 106 PRBs to it. Next, for the multi-UE scenarios, srsRAN's implementation of PF does not allow simultaneous UL transmissions from multiple UEs in a single time slot. Instead, every time slot the UE with the highest priority will be allocated the entire set of PRBs. On the other hand, srsRAN's RR scheduler does support multiple simultaneous UL transmissions, using a simple heuristic for resource allocation. For this reason, we decided to use RR as our benchmark.

4.7 Single-UE Case: Allocating the right number of PRBs per time slot

Now that we have laid the groundwork, we can introduce our approach to enhancing the UL scheduler implemented in srsRAN.

A crucial aspect of the default scheduling strategy of srsRAN, is that, assuming a full buffer scenario, it aims to allocate as many PRBs as possible to each UE, based on the set policy. For example, if a single UEs has a new UL transmission, under RR, in each time slot, it will be allocated the entire available set of PRBs.

Our goal is to refine this allocation process to enhance overall system performance, in terms of UE's achieved data rate. **We argue that UEs are not greedy in the number of PRBs** [63], i.e., they do not necessarily want as many PRBs as the gNB could allocate to them.

Specifically, we answer the following important question, *“What is the number of PRBs to allocate to an individual UE, based on its SINR, to offer it the highest data rate?”*

Let us assume that the average channel gain in a time slot for user u , is given by G_u and P_u is the transmit power budget used by user u . Moreover, let σ^2 be the noise plus interference at BS.

Let l be the MCS index (based on Table 4.2, l can take integer values from 0 to 28). Given l , the number of PRBs to allocate to user u is $m_l^u = \min \left(C', \left\lfloor \frac{P_u G_u}{\beta_l \sigma^2} \right\rfloor \right)$. Where C' is the maximum number of PRBs based on the bandwidth, available to the PUSCH for uplink data, excluding the PRBs reserved for the PUCCH for uplink control signaling, and numerology (as defined in Table 4.1), and β_l is the SINR threshold that corresponds to MCS level l , and is given in Table 4.2. This is because $m_l^u \leq C'$, and $\frac{P_u G_u}{m_l^u \sigma^2} \geq \beta_l$.

The rate seen by user u can be defined as $m_l^u r_l$. Recall that we defined the measured SINR at the UE to be $\frac{P_u G_u}{\sigma^2 m_l}$, assuming equal power allocation. Please recall that power control mechanism is not supported by srsRAN.

Then the number of PRBs which maximizes the total UL rate of user u , is then given by solving,

$$m_u^* = \underset{m_l^u}{\text{Arg max}} \{m_l^u r_l\} \quad (4.5)$$

along with the MCS index:

$$l_u^* = \underset{l}{\text{Arg max}} \{m_l^u r_l\} \quad (4.6)$$

In general, for practical settings $m_u^* < C'$, meaning that UE benefits from using less PRBs than C' .

Given P_u , G_u , and σ^2 , we can calculate m_u^* . For example, if $P_u = 0.02W$, $G_u = 2dBi$, and $\sigma^2 = 39.13W$, then $m_u^* = 80 < C' = 103$.

We plan to change srsRAN's scheduler code in a way that instead of allocating C' to the single UE, it allocates a number of PRBs to it equal to X . We will try different values of X and see if the best value equals m_u^* . After recompiling the code, we can perform experimental validation on the testbed. This is the first step towards the development of a more efficient scheduler.

4.8 Multi-UE Case: Allocating the right number of PRBs to multiple UEs per time slot

Having established how to compute m_u^* for each UE u individually, we can now discuss a possible mechanism for efficiently sharing the available PRBs among multiple UEs. Given C' , the number of available PRBs to PUSCH, the set U of active UEs, we have:

1. The resource blocks required for the retransmissions are allocated first.

2. Let the number of available PRBs after step 1, be C'' , and the set of active UEs with new transmissions be U' .
3. Compute m_u^* , $\forall u \in U'$.
4. Rank $u \in U' / m_1^* \leq m_2^* \leq \dots \leq m_{|U'|}^*$
5. $i = 1$, Do:
 - A) if $m_i \leq \frac{C''}{|U'|}$, then:
 - Allocate $m_i = m_i^*$ PRBs to UE i .
 - $C'' = C'' - m_i$.
 - $U' = U' - \{i\}$.
 - if $i \neq |U'|$, $i = i + 1$.
 - Go to A.
 - else, STOP.
 - B) else:
 - Allocate $m_i = \frac{C''}{|U'|}$ PRBs to all UEs in set U' (i.e., $\forall u \in U', m_u = \frac{C''}{|U'|}$)
 - STOP.

To implement this, we have to change the code to compute m_u for all active users and to assign m_u PRBs to each UE individually in the code, based on each UE's *Cell Radio Network Temporary Identifier (crnti)*. After recompiling the code, we can proceed with experimental validation on the testbed. Laying the foundation for our method to enhance srsRAN's uplink scheduler in both single and multi-UE cases, we can now delve into the experimental aspects and validation.

4.9 Experimental 5G-SA Setup

In this section we will introduce the testbed that was utilized for assessing our proposal. It is testbed $\mathcal{T}_{36}$, as defined in Table 3.2. Table 4.3 provides the details on the elements/sub-elements of $\mathcal{T}_{36}$. To assess our proposal, we needed to set up a testbed, which allows us to create scenarios, where the UE can experience varying levels of SINR. Therefore, we opted for PC #2 (as introduced in Chapter 3), which is a mini PC connected to USRP B210. This PC will host the 5GC, and RAN software and hardware elements. Moreover,

it allows the portability of the testbed. As we see later in Section 4.11, creating scenarios with specific SINR values, turned out to be extremely challenging and we had to conduct experiments in many locations to find just a few, which met our scenario needs.

Table 4.3: Elements and sub-elements of testbed $\mathcal{T}_{36}$

Software Elements			Hardware Elements	
S_{5GC}	Open5GS	H_{5GC}	Intel® Core™ i7-10700 CPU @ 2.90GHz (both S_{5GC} and S_{RAN} run on the same host PC)	
	OS: Ubuntu 20.04.6 LTS			
	Kernel: 5.15.0-xx-lowlatency			
S_{RAN}	srsRAN	H_{RAN}	Intel® Core™ i7-10700 CPU @ 2.90GHz	
	OS: Ubuntu 20.04.6 LTS			
	Kernel: 5.15.0-xx-lowlatency		USRP B210	
	Driver: UHD v4.5.0		Antenna: Molex 5GHz compatible	
S_{UE_2}	OS: Ubuntu 20.04.6 LTS	H_{UE_2}	Host PC: Intel® Core™ i7-3770 CPU @ 3.4GHz	
	Kernel: 5.14.0-051400-generic		5G Modem: Quectel RM502Q-AE	
			SIM Card: sysmoISIM-SJA2	
S_{UE_3}	OS: Windows 11	H_{UE_3}	Host PC: Intel® Core i7-1255U CPU @ 1.70 GHz	
			5G Modem: Quectel RM502Q-AE	
			SIM Card: sysmoISIM-SJA2	

We have configured $\mathcal{T}_{36}$ to operate on $b = n77$. Please note that, based on the Canadian Radio Spectrum Chart, which shows the public map of frequency allocations, band $n77$ is used by Canadian operators [3]. Therefore, although we were extremely careful to conduct our tests at times (primarily very early in the morning or very late at night in our office in UWaterloo’s main campus) to avoid disturbing potential devices around us, there may still have been other activities occurring on this band during our tests.

We set the bandwidth to be 40 MHz. We used Numerology 1, with an SCS value of 30 kHz, which means that the entire number of available PRBs per time slot is $C = 106$, with $C' = 103$ dedicated to PUSCH. Moreover, the TDD frame structure is set as: “DDDFUUUUUU”, which results in 6 UL slots, 3 DL slots, and 1 flexible slot. Note that, though srsRAN was set to use 3 DL slots, 6 UL slots, and 1 flexible slot, after multiple tests and a careful (and time-consuming) investigation of the logs with this configuration, we observed that only 3 time slots were allocated to the UL direction, while the remaining 3 were left unused.

SINR Measurement: srsRAN provides measured SINR for UE’s data transferred over PUSCH. There are three types of SINR that are reported for PUSCH [21].

- *sinr_ch_est:* The SINR estimated at the channel estimator, calculated as the ratio be-

```

2024-10-26T23:54:56.660720 [PHY      ] [D] [ 943.4] PUSCH: rnti=0x4601 h_id=0 prb=[3, 9) symb=[0, 14)
mod=64QAM rv=0 tbs=544 crc=OK iter=2.0 sinr=28.9dB t=147.4us uci_t=0.0us ret_t=0.0us
rnti=0x4601
h_id=0
bwp=[0, 106)
prb=[3, 9)
symb=[0, 14)
oack=0
ocsi1=0
part2=entries=[]
alpha=0.0
betas=[0.0, 0.0, 0.0]
mod=64QAM
tcr=0.92578125
rv=0
bg=1
new_data=true
n_id=1
dmrs_mask=00100001000100
n_scr_id=1
n_scid=false
n_cdm_g_wd=2
dmrs_type=1
tbs_lbrm=159749bytes
slot=943.4
cp=normal
nof_layers=1
ports=0
dc_position=636
crc=OK
iter=2.0
max_iter=2
min_iter=2
nof_cb=1
sinr_ch_est=31.9dB
sinr_eq[sel]=28.9dB
sinr_evm=26.2dB
evm=0.03
epre=-21.4dB
rsrp=-21.3dB
sinr=+28.9dB
t_align=0.32us
cfo=-22.2Hz
2024-10-26T23:54:56.660739 [MAC      ] [I] [ 943.13] UL rnti=0x4601 ue=0 subPDUs: [lcid=1: len=3, lcid
=1: len=10, lcid=4: len=66, lcid=4: len=448, LBSR: lcg=2 bs=71 , SE_PHR: ph=[9..10)dB p_cmax=[17..18)d
Bm, PAD: len=0]
2024-10-26T23:54:56.660750 [RRC      ] [D] ue=0 c-rnti=0x4601: RX SRB1 PDCP PDU
4647,16 0%

```

srsRAN's PHY logs with the three types of sinr measurements in PUSCH

Figure 4.3: Sample srsRAN log file excerpt

tween the estimated received signal power and noise power. To estimate “sinr_ch_est”, pilot signals are used.

- *sinr_eq*: The SINR estimated after the equalizer, when the effects of fading have been removed and after integrating the signals received from all available antenna ports.

- *sinr_evm*: A derived metric from the Error Vector Magnitude (EVM) estimator.

To report the SINR measurements in srsRAN, the PHY level logs should be activated in the gNB configuration file as follows:

```
log:
  filename: /tmp/gnb.log
  phy_level: debug
```

This will activate logging of all PHY-level messages to the specified file. Fig. 4.3 shows an example of a log file containing a PUSCH message and its content.

srsRAN logs all PUSCH messages every time slot. The MCS index is then selected based on the adjusted *sinr_eq* after Outer Link Layer Adaptation (OLLA). OLLA is a scheme used to cope with SINR estimation inaccuracies to minimize BLER and adjust the MCS selection accordingly. Based on the srsRAN configuration file, OLLA can adjust the SINR value by up to 5 dB before the MCS calculation. Unfortunately, srsRAN does not record the value of $\{\text{SINR} + \text{OLLA}\}$, which will be the basis of MCS selection, nor does it record the OLLA value in the logs, so we added some lines of code to the scheduler to log: (i) the OLLA value and (ii) $\text{SINR} + \text{OLLA}$. This allowed us to access the input value fed to the MCS function, i.e., an adjusted SINR that will result in an MCS index.

Data Rate Measurement: We measure the average UL data rate in Mbps using the *SpeedTest* application by *Ookla* [17]. Given the high granularity of our SINR measurements (every 0.5 ms) we selected *SpeedTest* for its ability to quickly assess data rates, completing the process in approximately 10 seconds. We perform three *SpeedTest* trials using the 5G modems.

4.10 Methodology

For the assessment of our proposal, after many careful trials, we selected two locations for testing. The reason for choosing two locations was to show that if our UE is in a location very close to the SDR device, and hence is experiencing a high SINR, it benefits from being allocated all C' PRBs. However, when it is based further away from the gNB, it achieves a better data rate with fewer PRBs rather than being allocated the full PRBs limit.

Fig. 4.4 illustrates the map of the fourth floor of the *Centre for Environmental and Information Technology* building on the main campus of University of Waterloo where we

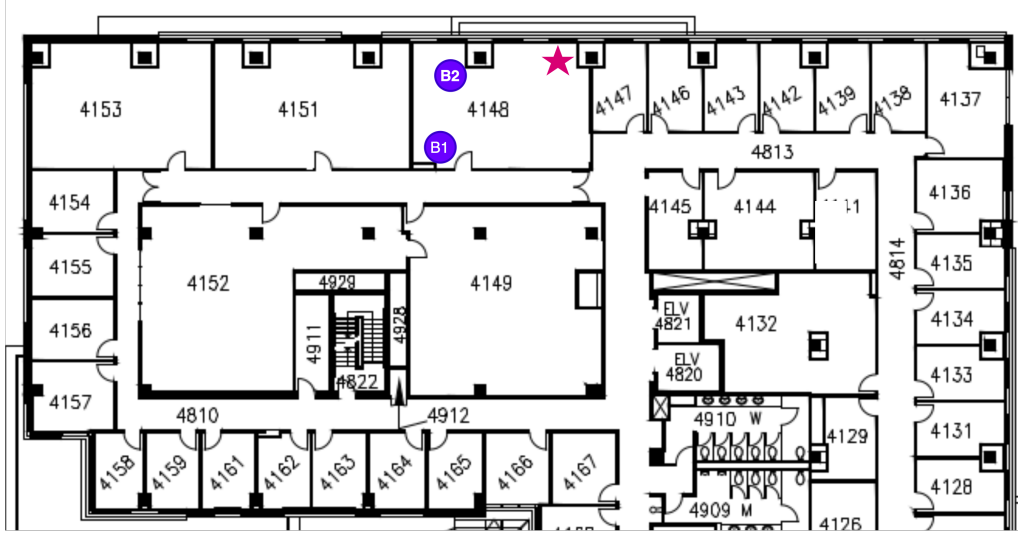


Figure 4.4: Map indicating the positions where scheduling enhancement assessments tests were conducted

conducted our tests. Our lab is in room 4148, and we have indicated the location of the SDR by a star sign on the map. The two locations selected for this round of tests are also marked in the figure.

For each location, we conducted multiple tests with different values for the number of PRB allocated to the UE. Those tests required a change of code in the scheduler, so that instead of the benchmark srsRAN's RR policy, specific values of PRBs be allocated to the single UE (moreover, for the case of two-UEs, we changed the benchmark RR code in a way that the two different UEs, each get allocated their specific value of PRBs, though, as we will see later, the two-UE case was not successful). After setting the PRB value for each test, we ran the testbed with the UE placed on a static cart at the designated location. Once the UE was attached to the testbed, we conducted three *Speedtests*. The testbed logged all PHY data for the test. As a result, we collected both the data rate measurements from the UE and the SINR measurements from srsRAN. By the end of each test, when we have collected all the data from the gNB and the rates achieved on the UE side, we shut down the testbed, change the number of available PRBs in the RR scheduler's code, recompile srsRAN, and start running it again for the next test.

4.11 Implementation Challenges

In this section we discuss the main challenges and obstacles we faced during our experimental validation phase.

4.11.1 Challenges in conducting UL tests with steady SINR measurements

We spent a considerable amount of time on finding locations that allowed us to demonstrate the further the UE's location, the fewer PRBs are required for the UE to achieve a higher data rate, compared to srsRAN benchmark scheduler. Our plan was to confidently move to the multi-UE case after successfully conducting the single-UE case, and validating our approach. Our experimental validation relies heavily on the assumption of *having steady and stable SINR measurements*, since we were keeping the number of PRBs constant for the duration of a test. Unfortunately, as we will see shortly, this assumption did not hold in many test locations, posing significant challenges to fully validating the single-UE case of our proposal. It took considerable effort and time to identify locations and configurations that allowed us to successfully validate this case.

Fig. 4.5, shows a series of SINR measurements that are steady and not show abrupt variations in the measurements. It shows the evolution in time of the 3 SINRs reported in the log presented in Section 4.9, and highlights the three speedtests. Except for a few outliers that can be observed in the measurements, where the SINR values have dropped significantly, the SINRs are quasi constant. We call this measurement a “good” one, because the values are very consistent. Therefore, when it comes to averaging these values, it is easy to remove the outliers, and still be sure that the average accurately represents the SINR experienced by the UE at any time slot.

On the contrary, we have a sample of a “bad” measurement, shown in Fig. 4.6. We call this sample a bad one, as the variations in the SINR measurements are extreme, sometimes up to 40 dB. We note that, the high level of variance will not necessarily create issues in the connectivity of the UE. The UE can maintain its connection to the testbed and we would successfully conduct a speed test assessment. However, the data rates achieved for each test will be drastically different from one another. For instance, in a location with “bad” SINR measurements, where the UE is allocated 60 PRBs, *Speedtests* might yield rates of {16, 3, 12} Mbps. Simply put, there is no certainty in the rate achieved by the UE in such measurement conditions. Although we could not pinpoint the root cause of this behavior, we identified several configuration methods, explained shortly, that helped

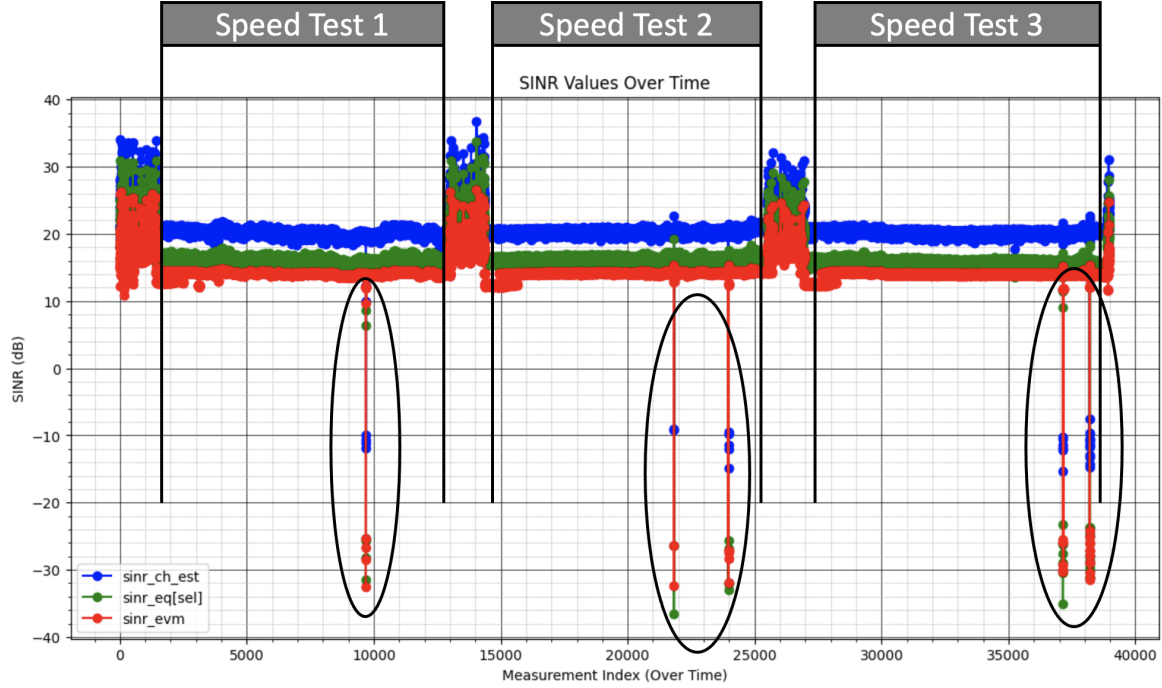


Figure 4.5: Sample of a good SINR measurement

us create an environment with steady SINR for the single-UE case. Please note that this behavior was observed when the number of allocated PRBs was low, especially when the UE was located far from the SDR device.

Unfortunately, during the initial phase of our measurements, we encountered many bad measurements with drastic SINR fluctuations. At this stage, the TDD slot configuration was set to “DDDDDDFUUU”, which corresponds to 6 DL slots, 3 UL slots, and 1 flexible slot. However, by using “DDDFUUUUUU” as the TDD slot configuration for the testbed, we observed a reduction in the measurement variations (please recall that this specific configuration in srsRAN does not lead to 6 UL slots, but 3 UL slots, and 3 unused slots).

Reconfiguring the TDD slot pattern reduced the variation in SINR measurements, but it was not sufficient. Though, our next reconfiguration significantly improved the situation. We observed that configuring srsRAN with a lower *Rx gain* and keeping the UE relatively close to the SDR yielded much better results compared to using the previous Rx gain configuration and positioning the UE farther away from the SDR device. Lowering the *Rx gain* allowed the UE to achieve lower SINR values without needing to move far from

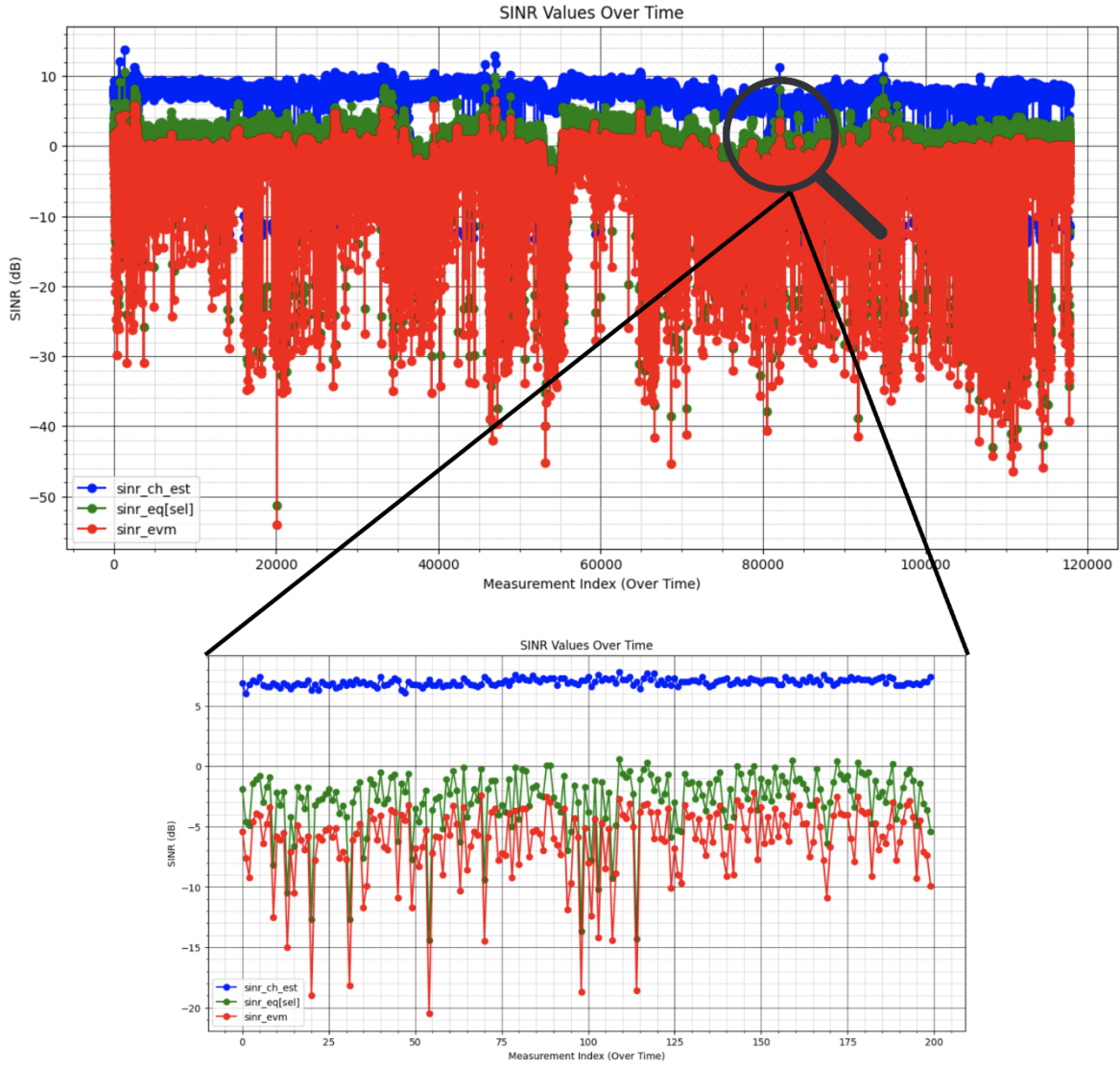


Figure 4.6: Sample of a bad SINR measurement

the SDR, resulting in more consistent SINR measurements. As a result, we decreased the *Rx gain* from 35 dB (the setting used in Chapter 3) to 4 dB for all the tests done in this chapter.

4.11.2 Challenges in PRB Allocation

During our initial stages in this project, while examining the RR scheduler code, we found that we could not change the number of allocated PRBs to any value in the range of {51, 52, 53, ..., 102}. More specifically, when we set the number of allocated PRBs to any number in the said range in the scheduler, we later observed that the UE was actually allocated 103 PRBs. It is important to note that this behavior was not observed when setting the PRBs to any number in the range of {1, 2, 3, ..., 50}.

After more investigation, we finally found out that the developers have set some limits to the PRBs allocated in another file:

```
srsRAN_Project/lib/scheduler/ue_scheduling/ue_cell_grid_allocator.cpp
```

Basically, this piece of code checks whether to allocate all the remaining PRBs (out of the PRBs allocated through the scheduling) to UE in a given time slot. To do this, if the number of available PRBs is less than twice the amount the UE needs, the code will allocate all the remaining PRBs to that user:

```
if (twice_grant_crbs_length < (mcs_prbs.n_prbs * 2)) {  
    mcs_prbs.n_prbs = twice_grant_crbs_length;}  
}
```

After commenting on this piece, this issue was resolved.

4.12 Test Scenarios, and Results for single-UE Case

We first outline our original plan and the steps required to accurately validate our approach on testbed $\mathcal{T}_{36}$. Our single-UE test scenario consists of the following:

1. Fix the location.
2. Change the code to manually allocate the number m of PRBs to use for the duration of the test and collect the rate. Each test is done three times.
3. Validate: “UE does perform better with less than the 103 benchmark PRBs.”
4. Change location.

Results on tests conducted at the two designated locations

Tables 4.4 and 4.5 show that the UE is better off not being allocated too many PRBs. This is evident when comparing the highlighted rows in both tables and examining the results using the benchmark PRB values at the two locations. At location *B1*, the SINR experienced by the UE in the benchmark scenario is quite high, and thus the UE benefits from being allocated a higher number of the resources, i.e., 83 PRBs. We can see that, on average, by allocating 20 PRBs fewer than the benchmark, the rate is increased by 27.7% within location *B1*. On the other hand, at location *B2*, the UE achieves the best UL rate with only 30 PRBs. Which means that on average, by saving 73 PRBs in network resources, the UE is achieving more than double the rate in comparison to the benchmark.

For both locations, we see that decreasing the number of PRBs is not affecting the BLER, as it remains below 10%. Additionally, the impact of the number of PRBs on the measured SINR is apparent: fewer PRBs result in a higher SINR.

Another important observation in the results below is the measurement of SINR values that fall below the minimum value required by srsRAN (based on Table 4.2) to achieve any UL rate. For instance, in the first row of Table 4.5, SINR values as low as -13.1 dB are observed. The rate is non-zero because the SINR values reported in the table are averaged over the duration of a *Speedtest*, and there are many time slots where the measured SINR exceeds the minimum value for a non-zero rate. Moreover, when SINR values fall below the minimum threshold of -5.7 dB (please refer to the MCS table in Section 4.2), srsRAN's MCS calculation function, which is available at: `{lib/scheduler/support/mcs_calculator.cpp}` in the source code, returns the minimum SINR value. As a result, -13.1 dB is adjusted to -5.7 dB, which is then passed to the MCS calculation function, returning the lowest MCS (MCS 0) as the corresponding MCS. Please note that the same position yields different SINR values when different numbers of PRBs are allocated because the power per PRB is changing.

Table 4.4: Measurements results done in location B1

Number of PRBs	MCS Level Used	UL Rate (Mbps)			BLER (%)			SINR (dB)		
		ST1	ST2	ST3	ST1	ST2	ST3	ST1	ST2	ST3
Benchmark	Dynamic MCS (11)	11.96	12.45	12.18	4.8	3.5	4.3	5.8	6.4	5.3
90	Dynamic MCS (12)	11.25	12.65	11.42	3.6	4.7	4.4	6.5	8	6.5
85	Dynamic MCS (13)	12.63	12.95	15	4.9	4.4	5.4	9.3	9.6	10.7
83	Dynamic MCS (15)	15.68	15.82	15.05	3.9	5.1	4.6	10.9	11	10.1
80	Dynamic MCS (14)	13.85	15.27	15.13	4.3	4.2	4.6	9.3	11	10.4
75	Dynamic MCS (14)	13.81	13.92	13.95	4.3	3.75	4.1	10.2	10.5	10.7

Table 4.5: Measurements results done at location B2

Number of PRBs	MCS Level Used	UL Rate (Mbps)			BLER (%)			SINR (dB)		
		ST1	ST2	ST3	ST1	ST2	ST3	ST1	ST2	ST3
Benchmark	Dynamic MCS (0)	1.91	1.79	1.74	5.3	5.8	0.8	-8.9	-9.7	-13.1
20	Dynamic MCS (12)	2.3	3.16	2.62	8.38	7.05	9.29	6.9	8.75	7.1
28	Dynamic MCS (9)	3	3	3	9.1	10	5.5	5.8	5.3	7
30	Dynamic MCS (13)	4.35	4.36	4.93	5.72	7.8	7.4	8.6	8.3	8.9
35	Dynamic MCS (9)	3.47	1.65	3.48	7.1	8.6	7.5	6.2	2.3	6.1

4.13 Test Scenarios, and Results for Two-UE Case

After validating that a single UE with poor SINR can significantly benefit from being allocated fewer PRBs, we proceed to the second step of our experimental validation: the case of multi-UE. For an effective multi-UE test scenario, we needed locations that will result in m_u^* values as low as 20, or 30. Therefore, we could validate our approach on multi-UE scenarios. Hence, we selected the two locations $\{B1, B2\}$, in the previous section. Moreover, in addition to the code changes mentioned earlier, we had to make another modification to the RR code. Recall that in the default RR policy implemented by srsRAN, the minimum number of UEs required for simultaneous uplink transmission was 7 (see Algorithm 5). Since we only had two UEs available for these tests (as the Android phone did not support band $n77$), we had to change the heuristic table defined in the code to allow two simultaneous uplink transmissions under the RR policy.

The outline of the two-UE test scenario is as follows:

1. Use the locations B1 for UE₂ & B2 for UE₃.
2. Change the code so that two UEs are allowed to have simultaneous uplink transmissions in a time slot.
3. Change the code to manually allocated 80 PRBs to UE₂ at location B1, and 30 to UE₃ at location B2, for the duration of the test and collect the rates reported by each UE.
4. Validate: *“UEs perform better compared to the default Round Robin scheduling policy, i.e., one being allocated 51, and the other being allocated 52 PRBs.”*

Results on tests conducted at the two designated locations

Unfortunately, this step was ultimately unsuccessful. Even though we had successfully established single-UE scenarios, and observed consistent SINR measurements, the measured SINR for the two-UE scenario exhibited significant inconsistencies, which left us without a clear explanation for this behavior. Please note that we suspect this inconsistency is related to changing the number of PRBs compared to the benchmark resource allocation implemented in srsRAN. The main reason for this hypothesis is that we did not observe such inconsistencies in the tests conducted in Chapter 3.

4.14 Conclusion

In this chapter, we studied the scheduler implemented in the popular RAN software platform, srsRAN. We proposed a more efficient approach to PRB allocation, suggesting that UEs, particularly those located farther from the gNB, should not be allocated an excessive number of PRBs. Instead, by concentrating their transmit power on a smaller set of PRBs, distant UEs can achieve a higher SINR on those specific PRBs, leading to a higher data rate. We then partially validated our hypothesis, we conducted experiments on an experimental testbed that utilizes srsRAN. Our results on a single-UE confirmed our approach.

For ease of implementation, we decided to keep the number of PRBs allocated to a UE constant for the duration of a test. However, unfortunately, due to the high variance observed in the measured SINR per time slot in multi-UE, we could not validate our proposal for multi-UE scenarios.

Chapter 5

Conclusions

In this thesis, we implemented, analyzed, and compared 35 open 5G-SA experimental testbeds. We first established a precise nomenclature to characterize a 5G-SA testbed, and then introduced the components of an operational 5G-SA testbed and explore various options for each element. In Chapter 2, we studied the evolution of the cellular networks, and provided a review of the literature in the context of open and full-fledged testbeds.

In Chapter 3, we first focused on the challenges faced during the implementation and our solutions. We also addressed interoperability by assessing how different open-source 5GC and RAN software from various developers work together. We observed that the performance of a UE connected to testbeds using two different 5GC software platforms, with all other elements being fixed, does not exhibit a significant dependency on the specific open-source 5GC software.

To ensure a systematic approach to our studies, in all chapters we maintained this structured approach for our tests, which included introducing the assessment target, providing a clear methodology, and detailing our test scenarios, followed by the presentation of results. We then offered our commentary on these results. This rigorous structure was intentionally designed to ensure that our work is reproducible for other research groups.

We then move on to the comparative analysis of different elements of a 5G-SA testbed. In this regard, we have selected prominent open-source solutions for the 5GC and the RAN. We also picked SDR options on two extremes, i.e., a more budget-friendly SDR with basic functionality and a very high-end, high-performance SDR to investigate their effect on the performance of the testbed. Moreover, on the UE front, we have multiple types of UEs.

A key aspect of this thesis is the investigation of multi-UE scenarios across various locations with respect to the SDR device. By defining three groups of locations, “*good, fair, bad*”, for the connected UEs, we first analyze how each 5G open-source RAN platform performs, given good UE positions. We then scatter the UEs in different locations and observe the performance achieved in adverse conditions. Our findings indicate that if *srsRAN* is executed on a powerful host PC, its performance can be superior given good UE positions. However, the table turns when UEs from further locations seek connection. In these scenarios, *srsRAN* not only lacks the automatic power adjustment feature of *OAI-RAN*, resulting in lower data rates for UEs connected from distant locations, but also exhibits a discrepancy in the achieved performance based on the type of connected UE. In this regard, *OAI-RAN* being UE-type agnostic and more robust, wins. By analyzing the coverage support of four different testbeds, we revealed that the choice of S_{RAN} is more influential than the choice of the SDR device on coverage. Our results also showed that *OAI-RAN* outperforms *srsRAN*, in the E2E latency.

Finally, in Chapter 4, we examined the UL schedulers implemented in *srsRAN*. We proposed an enhancement to the current schedulers based on the PRB allocation, arguing that UEs, especially those positioned farther from the gNB, are better off being allocated fewer PRBs. This way, by focusing their transmit power on a smaller set of PRBs, these UEs can achieve higher data rates. To validate our hypothesis, we conducted experiments on a testbed utilizing *srsRAN*, and our results for a single-UE scenario supported our approach. Unfortunately, we could not validate our approach with multi-UE scenarios.

5.1 Future Research Directions

The transition to open cellular networks, particularly open RANs, is at its peak now. O-RAN, an industry-initiated alliance and a significant player in the open RAN domain, has introduced a RAN architecture that extends the one proposed by 3GPP with additional elements and interfaces. In their proposed architecture, they introduce three elements to split the 5G protocol stack: the Radio Unit (RU), Distributed unit (DU), and the Central Unit (CU), each responsible for running parts of the protocol stack. Currently, there is a growing interest in the research community not only to build 5G-SA experimental testbeds but also to develop testbeds that comply with the O-RAN standard. At the time of this writing, very few O-RAN Radio Unit devices are available (e.g., the *FlexFi O-RU* from LITE-ON Technology and Foxconn *RPQN*). The main challenge in deploying an O-RAN compliant 5G-SA testbed is maintaining synchronization between the RU and DU, a process that requires precise timing and extensive communication. These challenges

should be thoroughly addressed in future studies on O-RAN compliant, 5G-SA testbeds.

Furthermore, our study on UL scheduling can be further extended. While we experimentally showed that UEs with poor channel conditions benefit more from being allocated fewer resource blocks, a comprehensive validation of this claim in multi-UE scenarios is still required. A promising next step for this project would be to extend the study to other testbeds, to determine whether the unexpected behavior observed in the multi-UE case is due to the srsRAN implementation.

References

- [1] 5G Switch - Force 5G Only. <https://play.google.com/store/apps/details?id=com.sladjan.sava.petg&hl=en&gl=US>. [Online; accessed 20-October-2024].
- [2] ArchLinux. <https://archlinux.org>. [Online; accessed 20-October-2024].
- [3] Canadian Radio Spectrum Chart. https://ised-isde.canada.ca/site/gestion-spectre-telecommunications/sites/default/files/attachments/2022/2018_Canadian_Radio_Spectrum_Chart.PDF. [Online; accessed 8-January-2025].
- [4] Ettus Research. <https://www.ettus.com>. [Online; accessed 20-October-2024].
- [5] Fedora. <https://fedoraproject.org>. [Online; accessed 20-October-2024].
- [6] free5GC. <https://free5Gc.org>. [Online; accessed 20-October-2024].
- [7] he.net-network tools. https://play.google.com/store/apps/details?id=net.he.networktools&hl=en_CA&gl=US. [Online; accessed 20-October-2024].
- [8] iperf3. <https://iperf.fr/>. [Online; accessed 20-October-2024].
- [9] Lime microsystems. <https://limemicro.com>. [Online; accessed 20-October-2024].
- [10] Network Signal Guru. https://play.google.com/store/apps/details?id=com.qtrun.QuickTest&hl=en_CA. [Online; accessed 14-October-2024].
- [11] Nuand. <https://www.nuand.com>. [Online; accessed 20-October-2024].
- [12] O-RAN Alliance. <https://www.o-ran.org>. [Online; accessed 3-October-2024].
- [13] OAI GitLab. <https://gitlab.eurecom.fr/oai/openairinterface5G>. [Online; accessed 20-Oct-2024].

- [14] OAI List of Tested COTS UEs. https://gitlab.eurecom.fr/oai/openairinterface5G/-/blob/develop/doc/TESTING_GNB_W_COTS_UE.md. [Online; accessed 3-October-2024].
- [15] OPEN AIR INTERFACE. <https://openairinterface.org>. [Online; accessed 20-October-2024].
- [16] Open5GS. <https://open5Gs.org>. [Online; accessed 20-October-2024].
- [17] SPEEDTEST. <https://www.speedtest.net>. [Online; accessed 20-October-2024].
- [18] srsRAN List of Tested COTS UEs. https://docs.srsran.com/projects/project/en/latest/knowledge_base/source/cots_ues/source/index.html#cots-ues. [Online; accessed 3-October-2024].
- [19] srsRAN MCS Calculation. https://github.com/srsran/srsRAN_Project/blob/9d5dd742a70e82c0813c34f57982f9507f1b6d5d/lib/scheduler/support/mcs_calculator.cpp#L23. [Online; accessed 3-October-2024].
- [20] srsRAN Project. <https://www.srsran.com>. [Online; accessed 20-October-2024].
- [21] srsRAN-SINR Values. https://github.com/srsran/srsRAN_Project/issues/792. [Online; accessed 3-October-2024].
- [22] sysmoISIM-SJA2 programmable SIM/USIM/ISIM cards. <https://sysmocom.de/products/sim/sysmousim/index.html>. [Online; accessed 20-October-2024].
- [23] Ubuntu. <https://ubuntu.com>. [Online; accessed 20-October-2024].
- [24] ZeroMQ. <https://zeromq.org>. [Online; accessed 3-October-2024].
- [25] xRAN Forum approves Fronthaul Interface Specification for NexGen Open RAN architecture. <https://techblog.comsoc.org/2018/04/13/xran-forum-approves-fronthaul-interface-spec-for-nexgen-open-ran-architecture/>, 2018.
- [26] 3GPP. NG-RAN; NG Application Protocol (NGAP). Technical Specification (TS) 38.413, 3rd Generation Partnership Project (3GPP), 9 2023. Version 17.6.0.
- [27] 3GPP. NR; Physical Layer; General Description. Technical Specification (TS) 38.201, 3rd Generation Partnership Project (3GPP), 9 2023. Version 18.0.0.

- [28] 3GPP. System Architecture for the 5G System (5GS). Technical Specification (TS) 23.501, 3rd Generation Partnership Project (3GPP), 9 2023. Version 18.3.0.
- [29] 3GPP. NG-RAN; Architecture Description. Technical Specification (TS) 38.401, 3rd Generation Partnership Project (3GPP), 1 2024. Version 18.0.0.
- [30] 3GPP. NR; NR and NG-RAN Overall Description; Stage-2. Technical Specification (TS) 38.300, 3rd Generation Partnership Project (3GPP), 9 2024. Version 18.3.0.
- [31] 3GPP. NR; Physical Channels and Modulation. Technical Specification (TS) 38.211, 3rd Generation Partnership Project (3GPP), 9 2024. Version 18.4.0.
- [32] 3GPP. NR; Physical Layer Procedures for Control. Technical Specification (TS) 38.213, 3rd Generation Partnership Project (3GPP), 9 2024. Version 18.4.0.
- [33] 3GPP. NR; Physical Layer Procedures for Data. Technical Specification (TS) 38.214, 3rd Generation Partnership Project (3GPP), 9 2024. Version 18.4.0.
- [34] 3GPP. NR; User Equipment (UE) Radio Transmission and Reception; Part 1: Range 1 Standalone. Technical Specification (TS) 38.101-1, 3rd Generation Partnership Project (3GPP), 10 2024. Version 18.7.0.
- [35] Renato B Abreu, Guillermo Pocovi, Thomas H Jacobsen, Marco Centenaro, Klaus I Pedersen, and Troels E Kolding. Scheduling Enhancements and Performance Evaluation of Downlink 5G Time-sensitive Communications. *IEEE Access*, 8:128106–128115, 2020.
- [36] Ruan P Alves, João Guilherme A da S Alves, Mikael R Camelo, Wilker O de Feitosa, Victor F Monteiro, and Fco Rodrigo P Cavalcanti. Experimental Comparison of 5G SDR Platforms: srsRAN x OpenAirInterface.
- [37] Maryam Amini, Ahmed El-Ashmawy, and Catherine Rosenberg. Implementing an Open 5G Standalone Testbed: Challenges and Lessons Learnt. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–2. IEEE, 2023.
- [38] Maryam Amini, Ahmed El-Ashmawy, Catherine Rosenberg, and Khandani Amir. 5G DIY: Impact of Different Elements on the Performance of an E2E 5G Standalone Testbed. In *2023 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2023.

- [39] Maryam Amini and Catherine Rosenberg. A Comparative Analysis of Open-source Software in an E2E 5G Standalone Platform. In *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2024.
- [40] Maryam Amini and Catherine Rosenberg. Performance Analysis and Comparison of Full-Fledged 5G Standalone Experimental TDD Testbeds in Single & Multi-UE Scenarios, 2024.
- [41] D Bhavana and Shilpa Chaudhari. Performance Evaluation of Openairinterface’s Scheduling Algorithms for 5G Networks. In *2023 4th International Conference for Emerging Technology (INCET)*, pages 1–4. IEEE, 2023.
- [42] Emmanouil-Zafeirios G Bozis, Nikos C Sagias, Michael C Batistatos, Michail-Alexandros Kourtis, George K Xilouris, and Anastasios Kourtis. A Versatile 5G Standalone Testbed Based On Commodity Hardware. In *2024 Panhellenic Conference on Electronics & Telecommunications (PACET)*, pages 1–4. IEEE, 2024.
- [43] Emmanouil-Zafeirios G Bozis, Nikos C Sagias, Michael C Batistatos, Michail-Alexandros Kourtis, George K Xilouris, and Anastasios Kourtis. Enhancing 5G Performance: A Standalone System Platform with Customizable Features. *AEU-International Journal of Electronics and Communications*, page 155515, 2024.
- [44] Maurine Chepkoech, Elena-Ramona Modroiu, Joyce Mwangama, Marius Corici, and Thomas Magedanz. Evaluation of OSS-Enabled OpenRAN Compliant 5G StandAlone Campus Networks. In *2023 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages 1–7. IEEE, 2023.
- [45] Maurine Chepkoech, Ngonidzashe Mombeshora, Bessie Malila, and Joyce Mwangama. Evaluation of Open-Source Mobile Network Software Stacks: A Guide to Low-cost Deployment of 5G Testbeds. In *2023 18th Wireless On-Demand Network Systems and Services Conference (WONS)*, pages 56–63. IEEE, 2023.
- [46] Alvaro Gabilondo, Zaloa Fernandez, Ángel Martín, Roberto Viola, Mikel Zorrilla, Pablo Angueira, and Jon Montalbán. 5G SA Multi-vendor Network Interoperability Assessment. In *2021 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–6. IEEE, 2021.
- [47] Long Gao and Shuguang Cui. Efficient Subcarrier, Power, and Rate Allocation With Fairness Consideration for OFDMA Uplink. *IEEE Transactions on wireless communications*, 7(5):1507–1511, 2008.

- [48] Jan Erik Håkegård, Henrik Lundkvist, Ashish Rauniyar, and Peter Morris. Performance Evaluation of an Open Source Implementation of a 5G Standalone Platform. *IEEE Access*, 12:25809–25819, 2024.
- [49] Md Emdadul Haque, Faisal Tariq, Muhammad RA Khandaker, Kai-Kit Wong, and Yangyang Zhang. A Survey of Scheduling in 5G URLLC and Outlook for Emerging 6G Systems. *IEEE access*, 11:34372–34396, 2023.
- [50] Hu Jinling. TD-SCDMA/TD-LTE Evolution—Go Green. In *2010 IEEE International Conference on Communication Systems*, pages 301–305. IEEE, 2010.
- [51] Juha Korhonen, 3GPP MCC. User Scheduling in 3GPP Technologies. <https://www.3gpp.org/technologies/scheduling>, 2023.
- [52] Keunyoung Kim, Youngnam Han, and Seong-Lyun Kim. Joint Subcarrier and Power Allocation in Uplink OFDMA Systems. *IEEE Communications Letters*, 9(6):526–528, 2005.
- [53] Ki Hoon Kwon, Younggoo Han, and Sehun Kim. Efficient Subcarrier and Power Allocation Algorithm in OFDMA Uplink System. *IEICE transactions on communications*, 90(2):368–371, 2007.
- [54] Yinghan Li, Shengqian Han, and Chenyang Yang. User scheduling for uplink ofdma systems by deep learning. In *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–5. IEEE, 2021.
- [55] Alberto Lometti, Claudio Colombo, Silvano Frigerio, and Vincenzo Sestito. Network Architectures for CPRI Backhauling. In *2012 14th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2012.
- [56] Yao Ma and Dong In Kim. Rate-maximization Scheduling Schemes for Uplink OFDMA. *IEEE Transactions on Wireless Communications*, 8(6):3193–3205, 2009.
- [57] Asmae Mamane, Mohammed Fattah, Mohammed El Ghazi, Moulhime El Bekkali, Younes Balboul, and Said Mazer. Scheduling Algorithms for 5G Networks and Beyond: Classification and Survey. *IEEE Access*, 10:51643–51661, 2022.
- [58] Lusani Mamushiane, Albert Lysko, Hlabishi Kobo, and Joyce Mwangama. Deploying a Stable 5G SA Testbed Using srsRAN and Open5GS: UE Integration and Troubleshooting Towards Network Slicing. In *2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pages 1–10. IEEE, 2023.

- [59] Mehaseb Ahmed Mehaseb, Yasser Gadallah, Ahmed Elhamy, and Hadia Elhennawy. Classification of LTE Uplink Scheduling Techniques: An M2M Perspective. *IEEE Communications surveys & tutorials*, 18(2):1310–1335, 2015.
- [60] Kamar Mubasier, Frank Y Li, Jon Anders S Øgaard, and Marius-Constantin Vochin. Campus-Based Full-Scale and Portable Open-Source 5G SA Networks: Prototyping and Experiments. In *2023 26th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 1–6. IEEE, 2023.
- [61] Cho Yiu Ng and Chi Wan Sung. Low Complexity Subcarrier and Power Allocation for Utility Maximization in Uplink OFDMA Systems. *IEEE Transactions on Wireless Communications*, 7(5):1667–1675, 2008.
- [62] Mohamed W Nomeir, Yasser Gadallah, and Karim G Seddik. Machine learning-based uplink scheduling approaches for mixed traffic in cellular systems. *IEEE Access*, 11:10238–10253, 2023.
- [63] Yiğit Özcan and Catherine Rosenberg. Uplink Scheduling in Multi-cell OFDMA Networks: A Comprehensive Study. *IEEE Transactions on Mobile Computing*, 20(10):3081–3098, 2020.
- [64] Ivan Palamà, Stefania Bartoletti, Giuseppe Bianchi, and Nicola Blefari Melazzi. Experimental Assessment of SDR-based 5G Positioning: Methodologies and Insights. *Annals of Telecommunications*, 79(5):301–313, 2024.
- [65] TBLELR Ramjee, T Bu, and LE Li. Generalized Proportional Fair Scheduling in Third Generation Wireless Data Networks. In *IEEE INFOCOM*, pages 1–12, 2006.
- [66] Arash Sahbafard, Robert Schmidt, Florian Kaltenberger, Andreas Springer, and Hans-Peter Bernhard. On the Performance of an Indoor Open-Source 5G Standalone Deployment. In *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2023.
- [67] Stefan Schwarz, Christian Mehlführer, and Markus Rupp. Low Complexity Approximate Maximum Throughput Scheduling for LTE. In *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, pages 1563–1569. IEEE, 2010.
- [68] Mauri Seidel, Andreas Ingo Grohmann, Peter Sossalla, Florian Kaltenberger, and Frank HP Fitzek. How to Get Away with OpenAirInterface: A practical Guide to 5G

- RAN Configuration. In *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 1–6. IEEE, 2023.
- [69] Lucas BD Silveira, Henrique C de Resende, Cristiano B Both, Johann M Marquez-Barja, Bruno Silvestre, and Kleber V Cardoso. Tutorial on Communication Between Access Networks and the 5G Core. *Computer Networks*, 216:109301, 2022.
 - [70] Alexander L Stolyar and Kavita Ramanan. Largest Weighted Delay First Scheduling: Large Deviations and Optimality. *Annals of Applied Probability*, pages 1–48, 2001.
 - [71] Fengxiao Tang, Yibo Zhou, and Nei Kato. Deep reinforcement learning for dynamic uplink/downlink resource allocation in high mobility 5g hetnet. *IEEE Journal on selected areas in communications*, 38(12):2773–2782, 2020.
 - [72] Boris S Tsybakov. File Transmission Over Wireless Fast Fading Downlink. *IEEE Transactions on Information Theory*, 48(8):2323–2337, 2002.
 - [73] Răzvan-Mihai Ursu, Arled Papa, and Wolfgang Kellerer. Experimental Evaluation of Downlink Scheduling Algorithms using Openairinterface. In *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 84–89. IEEE, 2022.
 - [74] Pablo Vázquez, Wilder Peña, Walter Piastri, Lucas Inglés, and Claudina Rattaro. MAQ5G: Deployment of a Complete 5G Standalone Network Testbed for Testing and Development. In *2024 XVI Congreso de Tecnología, Aprendizaje y Enseñanza de la Electrónica (TAAE)*, pages 1–7. IEEE, 2024.
 - [75] Pablo Andrés Vázquez, Walter Piastri, Wilder Pena, Lucas Ingles, and Claudina Rattaro. Development of New Radio Schedulers in OpenAirInterface. In *Proceedings of the 2024 Latin America Networking Conference*, pages 47–51, 2024.
 - [76] Elias Yaacoub and Zaher Dawy. A Survey on Uplink Resource Allocation in OFDMA Wireless Networks. *IEEE Communications Surveys & Tutorials*, 14(2):322–337, 2011.
 - [77] Hao Yin, Lyutianyang Zhang, and Sumit Roy. Multiplexing URLLC Traffic within eMBB Services in 5G NR: Fair Scheduling. *IEEE Transactions on Communications*, 69(2):1080–1093, 2020.
 - [78] Xin Yuan and Zhenhai Duan. Fair Round-robin: A Low Complexity Packet Scheduler With Proportional and Worst-case Fairness. *IEEE Transactions on Computers*, 58(3):365–379, 2008.