



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF ELECTRICAL, ELECTRONIC, AND INFORMATION ENGINEERING

"GUGLIELMO MARCONI"

DEI

SECOND CYCLE MASTER'S DEGREE

---

# TEST AND PERFORMANCE ANALYSIS OF AN O-RAN NETWORK

Dissertation in Mobile Radio Networks

**Supervisor**

**Prof. Ing. Roberto Verdone**

**Defended by**

**Weronika Maria Bachan**

---

**Graduation Session of March**

**Academic Year 2023/2024**



# Contents

<b>1</b>	<b>O-RAN: Context and Experimental Goals</b>	<b>1</b>
<b>2</b>	<b>Introduction</b>	<b>2</b>
<b>3</b>	<b>Evolution from Classical Radio Access Networks to Open RAN</b>	<b>4</b>
<b>4</b>	<b>OpenShift</b>	<b>10</b>
4.1	Understanding OpenShift . . . . .	10
4.2	How to deploy a POD using Pipelines . . . . .	11
<b>5</b>	<b>Hardware Setup</b>	<b>14</b>
5.1	Hardware Overview . . . . .	14
<b>6</b>	<b>Network Setup and Testing</b>	<b>17</b>
6.1	OAI Configuration File . . . . .	17
6.2	Signal Quality Indicators . . . . .	18
6.3	Sim Configuration . . . . .	21
6.4	Device Testing . . . . .	22
6.4.1	Testing with Motorola Edge 40 Neo . . . . .	23
6.4.2	Testing with iPhone 14 A2890 . . . . .	23
6.4.3	Testing with Samsung A55 and A25 . . . . .	24
6.4.4	Testing with Astreo device . . . . .	25
<b>7</b>	<b>Testing and Results: Single and Multi Device Scenarios</b>	<b>28</b>
7.1	Single Device in Different Locations . . . . .	31
7.2	Multi Device Performance Analysis in a Fixed Location . . . . .	38
7.3	Performance with an Increasing Number of Devices . . . . .	40
<b>8</b>	<b>Conclusions</b>	<b>42</b>
<b>A</b>	<b>Single Device in Different Location</b>	<b>43</b>
<b>B</b>	<b>Multi Device Performance Analysis in a Fixed Location</b>	<b>49</b>
<b>C</b>	<b>Performance with an Increasing Number of Devices</b>	<b>51</b>

# 1 O-RAN: Context and Experimental Goals

The evolution of Radio Access Network (RAN) has been driven by the increasing demand for high performance, reliability and cost efficient wireless communication systems, contributing to continuous advancements in telecommunications technologies.

Traditional RAN architecture relies completely on proprietary hardware and software, limiting the interoperability and increasing operational costs. Open Radio Access Network (O-RAN) aims to bypass these limitations by introducing an open, modular and virtualized architecture, allowing a multi vendor scenario. Integrating artificial intelligence (AI), cloud native technologies and standardized open interfaces, O-RAN allows also operators to optimize the network while reducing operating costs.

In this work the ORAN network is provided by the O-RAN alliance, which disaggregates the traditional base station into three elements: the Centralized Unit (CU), the Distributed Unit (DU) and the Radio Unit (RU). This approach enables virtualization, scalability and the integration of AI optimization.

The deployment of O-RAN on a cloud-native platform, OpenShift, enables orchestration, automation, and seamless integration with software-defined networking (SDN).

This thesis focuses on implementing and experimentally evaluating a 5G O-RAN network, using OpenAirInterface (OAI) for the gNB and Open5GS for the core network.

The experimental goals include assessing device compatibility by testing multiple commercial devices and evaluating connectivity and performance under different conditions. The tests aim to measure key signal indicators such as Reference Signal Received Power (RSRP), Signal-to-Noise Ratio and Block Error Rate (BLER) to analyze the performance of the O-RAN architecture.

Further analysis explores network behavior under single and multi device scenarios, with a focus on MAC (Medium Access Control) resource allocation and scheduling performance. The integration of OpenShift Pipelines enables automated deployment and testing, ensuring a scalable and efficient setup.

The findings demonstrate that O-RAN enables the connection of multiple devices while maintaining a stable 5G network.

## 2 Introduction

The complexity of the traditional Radio Access Network (RAN) is continuously increasing due to the rapid expansion of technologies and spectrum bands. Modern technologies, like Network Function Virtualization (NFV), Massive MIMO, Self-Organizing Networks (SON), and improvement in millimeter-wave and sub-terahertz communications are now part of the new RAN architecture. Each year brings new innovations layers of flexibility to the existing network infrastructure.

However, a significant drawback of the classical RAN architecture is the dependence on proprietary hardware and software. This tight coupling limits adaptability to varying network demands and complicates the seamless integration of solutions from multiple vendors, which consequently raises costs for network operators. For this reason, the Open Radio Access Network (O-RAN) offers a groundbreaking alternative.

Based on the O-RAN Alliance's specifications, O-RAN aims at modernizing the RAN infrastructure by introducing open, interoperable, and vendor-neutral systems.

Its architecture is divided into modular components: the distributed unit (DU), radio unit (RU), and centralized unit (CU). The CU is further split into Control Plane (CP) and User Plane (UP), enabling flexible deployment and virtualization of network functions. This disaggregated approach supports real-time data optimization, analytics, and closed-loop automation, enhancing overall network performance.

Unlike traditional RAN, which is dominated by a limited number of vendors that supply the entire system and limiting the potential for innovations, O-RAN enables interoperability with elements from various vendors. This flexibility allows the network to dynamically reconfigure based on real-time data and component configurations, with the result of a greater adaptability and efficiency. This approach simplifies upgrades and maintenance, giving network operators the freedom to select components to meet their specific needs without being locked into a single vendor's ecosystem.

O-RAN supports a wide range of use cases, from high-demand industrial environments with strict bandwidth and resource requirements to applications in academic institutions.

The first section focuses on the development of the O-RAN environment, starting from the hardware to software.

The second section focuses on experimental evaluation of the O-RAN network using several types of devices, including mobile phones and custom boards.

The experimental part aims to assess the compatibility and performance of O-RAN in a real scenario by analyzing key signal quality indicators. It is divided into experiments with a single device and multiple devices, investigating the network behavior under different load conditions.

In conclusion, this work provides an overview of the O-RAN network, from its development to real-world experimentation, highlighting its potential benefits and challenges.

### **3 Evolution from Classical Radio Access Networks to Open RAN**

The RAN (Radio Access Network) is a fundamental component of wireless communication systems, which connects end-devices to the core network. The end-devices include all the user equipment (e.g. computers, wireless module or IoT devices). It evolved over the years, starting from the first generation (1G) that marked the beginning of mobile communication, providing analog voice services but with limited capacity, security, and quality compared to modern networks. With the introduction of GSM (2G), network intelligence was centralized in the Core network, enabling digital voice and data services. Then with the fourth generation (LTE) part of the intelligence of the network was shifted back to the edge, within the RAN.

This shift enabled lower latency and faster data processing, and a lot of core functions were decentralized to the RAN. With 5G, distributed intelligence is at its peak, because the RAN includes capabilities like network-slicing, real-time analytics and different utilities to support for IoT and autonomous machines.

The increase of AI (Artificial Intelligence) applications and the high distribution of intelligence allow to process data closer to the user reducing the latency and increasing the network quality. The problem with traditional RAN is that its complexity has increased, to satisfy the rising demand: not only personal devices are connected to the RAN but also objects (IoT), so there is a growing necessity to manage the resources. Another problem is that the RAN (where main network function are present) is the opposite of an open system, instead, it is like a black box. RAN components are all-in-one solutions provided by different vendors, so there are no such possibilities to reorganize the network in the best way for the specific environment from the operator's point of view. Moreover, there is a problem with the coordination within all the nodes in a Network, also from the point of view of autonomous robots, it is really important to give the right number of resources between all the hosts.

All the reason gave above shows why an "Open" RAN is needed, allowing for enhanced performance analytics, process automation through a closed-loop system and programmability for greater flexibility.

The O-RAN Alliance initiative bridges the standardized front haul interface proposed by xRAN Forum and the centralized, cloud-based processing of C-RAN, enabling an open, interoperable RAN architecture that integrates external controllers and enhances network efficiency starting from 2018.[1]

O-RAN splits the traditional base station (gNB) into three different parts:

- O-CU (O-RAN Centralized Unit): manages the higher layers functions, like the QoS and the security. It is also divided into User-Plane (UP) and the Control-Plane (CP).
- O-RU (O-RAN Radio Unit): handles the transmission and reception of radio frequency signal.
- O-DU (O-RAN Distributed Unit): handles the lower layers function, performs scheduling and link adaptation.

In this way it is possible to have parts of it virtualized, vRAN (virtual RAN) and mix the components from different vendors, solving the main problem of the actual RAN.

A key component of the RAN architecture is the two RAN intelligent Controllers (RICs) that enable automation and intelligent radio network operations using AI algorithms. They are divided in two different type of time scale where each RIC works, the near-real-time (10 ms to 1s) and non-real-time (1s). These two controllers are based on hundreds of Key Performances Measurements (KPMs), metrics used to evaluate the performance and efficiency of a network, focusing on aspects like latency, throughput, and reliability.

The near real-time RIC (near-RT) operates through a control loop, with timing varying from 10 milliseconds to 1 second. It interacts with the DU and CU via the E2 interface. The non-RT instead focuses on longer term policy generation, Machine Learning (ML) model training and higher level of network analytics, interacting with the near-RT through the A1 interface. The near-RT is at the edge of the network since it is also connected to the eNBs (evolved Node Bases). Usually, the near-RT RIC is associated to more than one RAN nodes, serving thousand UEs (User Equipment's). The near-RT RIC is based on multiple software applications that support customization called xApps. They can be used for various purposes, such as resource management or interference mitigation.

These applications are based on RAN database, containing the number of users, the list of RAN's nodes, everything they can retrieve from the KPM.

The most important thing is to have continuous flow of data, to enhance the control on the RAN, because it is possible to have a lot of xApps in just one RAN. Each xApp receives data from the RAN and sends back the specific control using the E2 interface.

The near-RT RIC can control and optimize the RAN thanks to three different interfaces: A1, O1, and E2.

1. The A1 interface connects the near-RT RIC to the non-RT RIC enabling the non real-time control loops and enabling policy deployment through guidance and intelligent models.

The policies are based on a Json scheme, with an identification code for the non-RT RIC, because they can be used on a one User Equipment (UE), on a group of them, on a cell or in slices. They are then deployed to the near-RT RIC for QoS or KPI goals over HTTPS, allowing a secure connection based on SSL/TLS protocols. The non-RT RIC manages the O1 interfaces linking the RAN components for orchestration and control of network functionalities.

2. The O1 coordinates the life cycle of O-RAN elements, from initialization and fault tolerance to software and file management.
3. The E2 is an open interface interconnecting the near-RT RIC to RAN nodes, it allows to retrieve data periodically or after a trigger from the RAN and then to respond with the control commands from the near-RT RIC.

The E2 interface runs on top of the SCTP protocol, providing a reliable and ordered communication channel between the network components. Having a reliable connection is the first step in near-RT RIC and E2 node communication. The aim of this interface is to efficiently allocate the resources and the management functions from the perspective of network optimization.

Specifically, there are two endpoints of the E2 interface, one in the near-RT RIC and the other one in the gNB that allow the exchange of data between them.

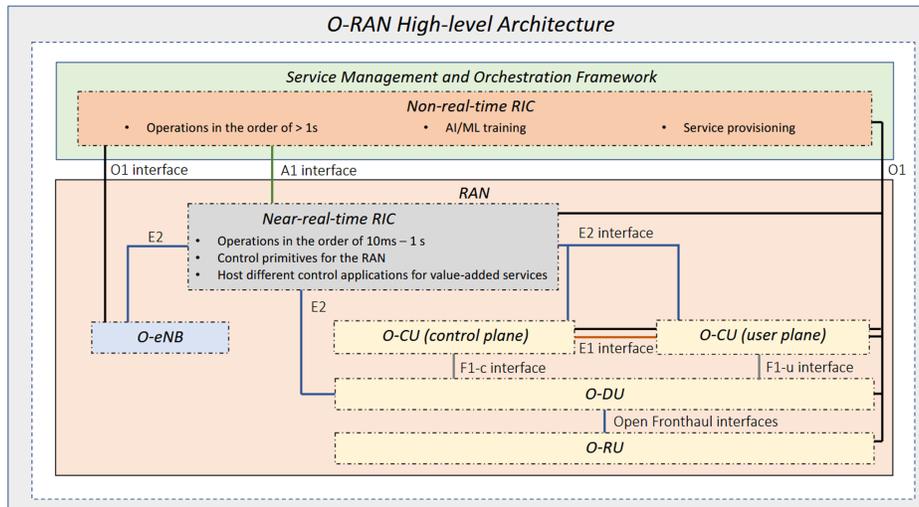


Figure 1: L. Bonati, M. Polese, S. D’Oro, S. Basagni, and T. Melodia, "Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead," Computer Networks, vol. 182, Dec 2020

E2 interface can be logically split into two parts: E2 Application Protocol (AP) and E2 Service Models (SM).

The first one dedicated to the communication between the near-RT RIC and the E2 interface. Instead, the E2 SMs are dedicated to the set of functions to control the RAN or just to report the metrics.

The E2AP (E2 Application Protocol) is a procedural protocol that coordinates the communication between E2 nodes, and the near-RT RIC embeds different E2 SMs.

The E2 interface uses a type of publish-subscribe communication: the E2 nodes publishes their data and then the xApp can subscribe to retrieve this information; this method allows the separation between all the nodes.

The E2AP allows four different services:

1. E2 Insert: notifies specific events from the E2 node to an xApp (inside the near-RT RIC), like the possibility of performing a handover. This type of service is associated with a RAN radio resource management procedure, which is paused when this message is sent. A timer is set, and upon its expiration, the procedure in the E2 node either resumes or terminates.

2. E2 Report: as the name suggests, it reports the E2 RIC Indication Messages containing telemetry from E2 node. The reporting has a specific timer in the E2 node, for example, based on a specific frequency. When this timer expires, a report is sent to the xApp.
3. E2 Control: activated by the near-RT RIC after receiving the E2 Insert message. This service is formed by two message procedures: RIC send a Ric Control Request to the E2 node and then the node responds with a RIC Control Acknowledgment (ACK). This service allows to modify the RAN parameters of the E2 node for real-time network adjustments.
4. E2 Policy: allows the E2 node to perform specific actions based on the rules that are predefined without a continuous input from the RIC.

In O-RAN Standard the service model message is inserted as payload in a E2AP messages. There are different services models from the O-RAN standard.

The KPM (Key Performance Management) service retrieves performance metrics, while the NI (Network Interface) sends back network interface data to the near-RT RIC. The CCC (Control and Configuration) manages RAN control, e.g. bandwidth. E2SM RC (RAN Control) enables fine-grained control over radio resources, management, and policies. These services allow dynamic RAN management through periodic, or trigger-based, reports and control actions.

To make the O-RAN network competitive with the speed of normal RAN the architecture uses different AI and ML algorithms to apply the control policies. This enables data-driven, closed-loop control to automatically optimize network slicing, load balancing, handovers, and scheduling policies in real-time.

The AI part is still a work in progress, but there are six structured steps for the workflow in O-RAN:

1. Creation of dataset: data collection from the O1, A1 and E2 interfaces. The dataset is then stored in repositories, the type of data depends on the interface's configurations. It is possible to change the time range within they are collected, or to choose which data to keep, e.g. throughput instead of latency, or Modulation and Coding Scheme. These data are then processed to be shaped for training and online inference. To reduce the dimensionality autoencoders can be used.

2. Training: AI model must be trained for O-RAN specifications, it is also possible to do it offline. This is essential to avoid network inefficiencies, inaccurate predictions or classifications.
3. Validation and Publishing: to increase the reliability of the decisions taken in the training step there is a validation test that assesses the performance of various AI solutions across different traffic patterns, user numbers and distribution, available bandwidth, and operational frequencies. If the models pass the validation step, they are stored on the SMO/non-RT RIC. But if it fails, they aren't just excluded, but they go through a retraining phase.
4. Deployment: models stored in the SMO/non-RT RIC can be downloaded and executed.
5. AI/ML Execution: the models are deployed on the host for inference and they start processing real-time data.

The main need of the AI/ML workflow is to continuously have new data to try different types of training and to update the models without creating delays in the network.

The non-real-time RIC complements the one real-time with all those operations that require a timing larger than 1 second, for the "intelligent" part of the RAN. So, it is part of the Service Management and Orchestration (SMO), it allows control all the Machine Learning models for the near-real-time control loop. This allows them to manage all the elements connected to the SMO, which significantly improves the scalability.

O-RAN open and intelligent architecture is a crucial advancement in the telecommunications world, improving the network efficiency, decreasing the operational costs while giving a greater flexibility. The O-RAN architecture makes the network adaptable to new use cases like IoT, future 6G networks and AI-driven applications without the need of new hardware, since the major elements of the architecture can be virtualized.

The next step will be the integration of real-time control loops, enabling operations under 10 milliseconds or even below; this will be particularly beneficial for tasks such as beam management, as well as the optimization and management of network devices. Further improvements will focus on improving the current architecture, including AI-driven applications and a seamless cloud-native infrastructure, ensuring greater efficiency and adaptability.

## 4 OpenShift

### 4.1 Understanding OpenShift

To efficiently manage a scalable and containerized O-RAN architecture, a solution supporting continuous integration (CI) and continuous deployment (CD) is essential [2].

OpenShift meets these requirements, as it is a Kubernetes-based platform that offers advanced orchestration and resource management capabilities. The core elements in OpenShift deployment include:

- Nodes: worker machines (virtual or physical) in an Open-Shift cluster running containerized workloads.
- Pods: smallest deployable units that contain one or more containers that share the same resources.
- Namespaces: virtual clusters that logically organize resources. In this O-RAN Network deployment, there are different namespaces, but the four main are:
  - ricplt: contains the near-RT RIC.
  - ricxapp: hosts Xapps.
  - default: holds the gNB
  - Open5gs: includes the core network components.

OpenShift uses operators to manage and automate configurations, reducing the need for direct manual intervention. This is one of the major differences compared to Kubernetes, where administrators need to manually configure and manage resources using custom scripts or third-party tools.

The deployment in OpenShift is centralized and its nodes are stateless, simplifying cluster management. Various operators, such as Argo CD, enable automated deployment, installation, and resource organization. With OpenShift Pipelines, users can build Docker images or deploy components such as OpenAirInterface. These pipelines also allow users to view and manage applications efficiently.

OpenShift includes an internal image registry, by default, enabling users to store and pull container images, similar to the Docker pull mechanism.

The Red Hat PaaS (Platform as a Service) provides administrators with a web-based dashboard to monitor, manage and configure all the deployments, pipelines and operators.

For this research, some key features are particularly useful. One of them is the Cluster Overview, a visual representation of the entire application and its dependencies.

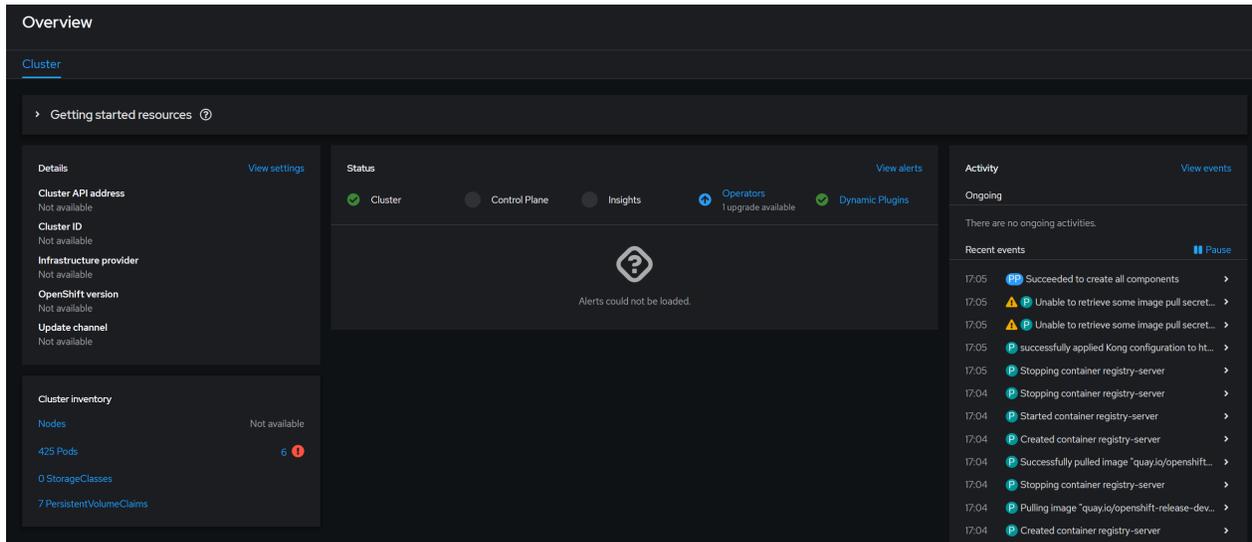


Figure 2: Overview of the architecture in Red Hat Dashboard

Another essential part of the dashboard is the Builds and Deployments section, which allows tracking of build process, deployments and container images. This built-in feature removes the need for external registry services, improving deployment efficiency.

## 4.2 How to deploy a POD using Pipelines

Thanks to OpenShift's flexibility, it is possible to easily create a new pod. For example, deploying another OpenAirInterface gNB with some custom settings. These customizations can include modifying parameters in the OAI configuration files, such as adjusting the output power of the USRP or changing the SNR threshold for power control.

To speed up these operations, OpenShift Pipelines are used. These pipelines enable the developer to define, automate and execute workflows, ensuring an efficient deployment.

They also simplify testing and scaling of applications within a cloud-native infrastructure.

To deploy a pod, it is necessary to define the following elements:

- Pipeline: structured workflow that defines how different tasks are executed. It also determines whether tasks should run sequentially or parallel.
- Tasks: individual processes, such as building an application, deploying code or running tests.
- TaskRun: each pipeline consists of multiple TaskRuns, which represent the individual execution steps needed to achieve the objective.
- Workspace: a shared storage space that allows data exchange through PVC (Persistent Volume Claim) and stores Secrets (e.g. password and authentication)

From the Red Hat Dashboard, it is possible to create a pipeline, in the Pipelines section. The first step is to add Tasks, which in this particular case include:

1. Fetching the source code from the OpenAirInterface GitHub repository.
2. Building the container image.
3. Running tests to validate the build.
4. Deployment of the final application.

All of these steps are defined in a YAML file, a human-readable data serialization format used for configuration files and data exchange. YAML allows users to define deployment settings in a clear and consistent way across different environments. Once the pipeline is configured, the next steps include:

- Filling out all the required sections, including the image tag and the release version of the deployment. (e.g. in this work the version of OAI gNB used is the release 2025w06)
- Selecting the appropriate worker node. In this specific case, there are two worker nodes: worker node 0 and worker node 1, each connected to one USRP.
- Specifying the correct namespace for deploying the POD.

- Configuring key parameters, such as: AMF-IP, E2TERM-IP and the USRP-IP.
- Defining a Persistent Volume Claim (PVC) to manage permanent storage.

Once all configurations are set, the pipeline executes the deployment process, ensuring that the pod is instantiated with the specified parameters. Through this automated approach, developers can efficiently deploy and manage O-RAN components in OpenShift, improving operational consistency.

## 5 Hardware Setup

### 5.1 Hardware Overview

The O-RAN (Open RAN) network is built on a cluster of five Dell PowerEdge R750xs servers, each equipped with 16 CPU cores and 200 GB of RAM. This number of cores and this amount of RAM are required to support the DU (Distributed Unit) and CU (Centralized Units) processing loads, ensuring smooth operation.

Each server includes a Mellanox ConnectX-6 NIC, enabling high-speed data transfer at 200 Gbps. These NICs are optimized for AI/ML workloads, making them ideal for handling computationally intensive tasks.

The transceiver setup consists of QSFP28 interfaces, which enable high-speed Ethernet communication. These interfaces are commonly used in cloud data centers and high-performance computing systems. In this setup, the platforms are connected through the A7 port, allowing them to operate with 100 Gbps connectivity.

This hardware configuration is highly scalable, making it adaptable for various applications, including O-RAN network deployment and cluster computing. Within the OpenShift logic, these five servers function as nodes and are programmed to handle different roles.

The workload is divided between the control plane nodes and worker nodes:

- **Three Control Plane Nodes:** responsible for managing the entire cluster, ensuring its health and orchestrating resources. These nodes also serve as the central interface for interaction between applications and users. They monitor cluster health, ensuring stability and resource availability while using a scheduler to allocate resources based on each node's workload
- **Two Worker Nodes:** dedicated to computational tasks. They handle tasks such as pod execution, network management and AI/ML model training when required.

Typically, OpenShift maintains strict separation between the control plane and the worker plane to ensure optimal cluster management. However, in this implementation, all five servers can be dynamically allocated to maximize computing power when demand is high.



Figure 3: O-RAN Server Rack: Five Dell PowerEdge R750xs Units

For over-the-air communication, the system integrates two NI Ettus USRP X410 devices, each connected to a worker node. The X410 model supports dual-channel transmission and reception, high-bandwidth operations and software-defined frequency control.



Figure 4: Two NI Ettus USRP X410 devices used in the O-RAN setup

Each USRP can operate at either 3900 MHz or 3940 MHz, supporting communication in the n78 and n77 bands. These bands ensure compliance with 5G frequency standards.

Key features of the USRP X410 that meet 5G requirements are [3]:

- Frequency Range: from 1 MHz to 7.2 GHz.
- Instantaneous Bandwidth: up to 400 MHz.
- Maximum Output Power: 23 dBm.

The connectivity between the servers and the USRP devices is established using QSFP28 to 4xSFP28 optical fiber cables, ensuring low latency and high-speed data transfer, as well as compatibility with the required frequency ranges for the standard.

For transmission, only the worker node are used, as each USRP is connected to a single worker server. However, a future implementation of a network switch will enable both USRPs to be connected to the same worker node, increasing flexibility. For tasks such as AI model training, all servers can be used to efficiently distribute the computational workload, ensuring optimal resource utilization.

## 6 Network Setup and Testing

### 6.1 OAI Configuration File

Before testing the setup, the fundamental step is to understand the OpenAirInterface (OAI) gNB configuration file. This YAML file defines the essential parameters for the operation of a 5G NR gNB within an O-RAN network. It configures every aspect, including network identification, radio parameters, network interfaces and logging preferences.

The configuration file is structured into the following sections:

1. General gNB Configuration, Identification, and Network Parameters
2. Physical Layer Configuration
3. Network Interfaces
4. MAC and RLC Layer Parameters
5. Radio Unit (RU) Configuration
6. Security and Logging Configuration

The first one describes the name and identification of the gNB, defining a unique gNB ID that identifies the area served by this gNB. Additionally, the PLMN (Public Land Mobile Network) is defined and used for the SIM card configuration and network connectivity. The PLMN list includes the Mobile Country Code (MCC) and the Mobile Network Code (MNC), which the gNB will broadcast for devices such as phones and modems to use.

The Physical Layer Configuration defines the radio parameters, including antenna settings (in this case one antenna for the uplink and one for the downlink), bandwidth allocation and frequency bands used. This part introduces an important parameter, pMax, which is the maximum transmit power for the USRP expressed in dBm. Increasing this value improves uplink coverage in weak signal conditions, while reducing it decreases interference.

The Network Interfaces section enables connectivity between the OAI gNB and the Open5GS core components, including the AMF (Access and Mobility Function) and the UPF (User Plane Function).

It defines the IP addresses of the AMF, as well as the IP and port of the POD through the chosen interface (e.g. eth0).

The MAC and RLC Layer Parameters Configuration section defines the settings that manage Hybrid Automatic Repeat Request (HARQ) retransmissions, scheduling, and RLC behavior. This section is fundamental, as it specifies the target Signal-to-Noise Ratio (SNR) for both downlink and uplink. Additionally, it determines the maximum Modulation and Coding Scheme (MCS) achievable in the uplink.

The following values are used:

```
MACRLCs = (  
{  
  num_cc = 1;  
  tr_s_preference = "local_L1";  
  tr_n_preference = "local_RRC";  
  pusch_TargetSNRx10 = 300;  
  pucch_TargetSNRx10 = 200;  
  pusch_FailureThres = 1000;  
  ul_max_mcs = 28;  
}
```

The Radio Unit section configures parameters related to the USRP, such as the number of transmitting and receiving antennas, the maximum receiving gain, and the Software-Defined Radio (SDR) address for the USRP connection. The fifth and final part focuses on integrity protection, ensuring that signaling messages and user data are protected against attacks.

## 6.2 Signal Quality Indicators

The assessment of signal quality in 5G OpenAirInterface is based on various parameters extracted from the gNB scheduler, specifically from the NR MAC (Medium Access Control) layer logs.

The MAC scheduler in OAI operates based on a Proportional Fair (PF) scheduling policy, dynamically allocating Resource Blocks (RBs) to User Equipment (UE) according to their respective Proportional Fair coefficient. This coefficient is influenced by the historical Modulation and Coding Scheme (MCS) and directly affects the current MCS selection.

The scheduling process prioritizes UEs with the highest PF coefficient, assigns RBs in descending order until all available resources are allocated.

Once the highest-priority UE has been scheduled, the process continues with the next UE in decreasing order of the PF coefficient, ensuring an efficient distribution of network resources.

The Block Error Rate (BLER) and the MCS selection are strongly correlated. The scheduler evaluates the BLER threshold by monitoring first-round retransmissions within a 50-millisecond observation window. There is a dual-threshold mechanism that allows the scheduler to optimize the balance between channel robustness and data rate. If the retransmission remains below the predefined lower BLER threshold, the channel is classified as good and the MCS is increased by one level. Conversely, if the retransmission ratio exceeds the upper BLER limit threshold, the MCS is decreased by one level to maintain transmission reliability.

To differentiate downlink and uplink transmissions, the scheduler maintains two distinct BLER metrics:

- BLER ULSCH (Uplink Shared Channel BLER): uplink transmission quality assessment.
- BLER DLSCCH (Downlink Shared Channel BLER): downlink transmission quality assessment.

The scheduler logs different metrics in the "nrMAC-stats.log" file, which is generated upon the initiation of the gNB's frame slot print process and is updated every second. This log provides insights on radio resource allocation, transmission reliability, and network efficiency, enabling a detailed evaluation of scheduling performance in an OAI-based 5G O-RAN environment [4].

The figure below provides an example of a log file:

```
[NR_MAC] Frame.Slot 256.0
UE RNTI 2212 CU-UE-ID 1 in-sync PH 29 dB PCMAX 21 dBm, average RSRP -88 (16 meas)
UE 2212: dlsch_rounds 586/30/16/12, dlsch_errors 10, pucch_DTX 65, BLER 0.16161 MCS (1) 6
UE 2212: ulsch_rounds 30028/253/69/51, ulsch_errors 44, ulsch_DTX 24, BLER 0.15671 MCS (1) 22 (qm 8 deltaMCS 0 dB) NPRB 5 SNR 38.0 dB
UE 2212: MAC: TX 279728 RX 4919395 bytes
UE 2212: LCID 1: TX 812 RX 3065 bytes
UE 2212: LCID 2: TX 0 RX 0 bytes
UE 2212: LCID 4: TX 178918 RX 106204 bytes
```

Figure 5: OAI nrMAC-stats.log file showing MAC-level scheduling parameters

The log entries are structured into multiple lines, each corresponding to a specific parameter category. The first line contains UE-related information, such as the UE ID and whether it is actively responding and scheduled. Other relevant parameters include the Power Headroom (PH), which is the additional power available for the uplink transmission (from the UE) and the PCMAX, that represents the maximum uplink transmit power available to the UE in this channel.

The last parameter, RSRP, expresses the downlink signal strength, based on the reference gNB signal measured at the UE. Values below -95 dBm indicate an unreliable connection.

The next two rows are particularly relevant for this work as they contain the Channel State Information (CSI), which can be enabled or disabled based on the OAI configuration file. These parameters provide information about channel quality, rank indication and precoding configurations for both uplink and downlink transmissions.

The following is a list of the key parameters of the rows [4]:

- `dlsch_rounds A/B/C/D`: number of downlink transmissions from the gNB for each HARQ round. A = First Round, B = Second Round, C = Third Round, D = Fourth Round. The same applies to `ulsch_rounds A/B/C/D` for uplink.
- `dlsch_errors`: number of failed transmissions after four HARQ rounds.
- `DLSCH BLER`: Block Error Rate in the downlink, evaluated from  $\frac{A}{B}$ . A reference level of 10% - 30% is considered optimal for higher throughput [4]. `ULSCH BLER` same concept as `DLSCH BLER`, but applied to uplink transmissions.
- `MCS (Q) M`: Q indicates which of the three official 5G MCS tables is used while M represents the modulation scheme used. The same metrics are shown for both uplink and downlink.
- `SNR`: Signal-to-Noise Ratio (SNR) measured at the gNB based on the received UE signal quality.

The last three lines in the log include the Logical Channel (LCID), which indicates whether the UE is actively connected to the gNB. Specifically, LCID 4 refers to bytes transmitted and received over the Internet connection.

These logs are crucial in the evaluation of channel quality in real-time, allowing continuous

monitoring of network conditions between the gNB and the UE.

In this study, these logs are systematically analyzed to assess network performance across all experiments.

### 6.3 Sim Configuration

The SIM cards used to connect the UE to the O-RAN network are provided by the Open Cells Project [5]. These SIMs can be configured for GSM, WCDMA, LTE and 5G technologies, supporting both Standalone (SA) and Non-Standalone (NSA) modes.

They follow the 3GPP standard for subscriber identification, meeting the requirements of the custom O-RAN deployment. These SIM cards are programmable using a dedicated card reader/writer, which works as a serial interface over USB.



Figure 6: Open Cells SIM cards and SIM card reader/writer

Using a Linux system and specialized UICC programming software (provided by Open Cells), it is possible to configure the SIM cards by specifying different parameters including:

- IMSI (International Mobile Subscriber Identity), uniquely identifies the subscriber.
- ISDN (Integrated Services Digital Network), essentially the phone number associated with the SIM.
- Key (128-bit cryptographic key), used by the Operator Authentication Center for security.
- OPc (Operator Code), a security parameter for authentication.

- ACC (Access Control Class), defines priority levels in network access.
- SPN (Service Provider Name), the provider's name displayed on the device.

These programmable SIM cards enable subscriber authentication within the O-RAN environment, allowing the UE to connect to the OAI gNB and to the core network components, such as AMF. Their flexibility and programmability make them an optimal choice for research and testing in private 5G deployments.

## 6.4 Device Testing

After deploying the network, multiple devices were tested to assess their compatibility and performance in a 5G O-RAN environment.

The tested devices include:

- Three smartphones: Motorola Edge 40 Neo, iPhone 14 (model A2890), Samsung A25 and Samsung A55.
- An embedded module: a 5G SIMCOM SIM8380G-M2 module on a development board.

As the O-RAN network operates as a 5G SA mode, all devices were tested under the same conditions to ensure consistency in the experimental procedure.

The first step was to power on the desired USRP and access the OAI OpenShift pod shell. The OAI gNB was then initialized using a dedicated script. This Bash script applies the configuration file to the gNB, generating logs related to USRP configuration, core network status, and successful gNB attachment to the core network, as well as deployment details.

Once this initial logging phase is complete, the system generates a series of "Frame Slot" messages, indicating that the network is ready for UE connections.

#### **6.4.1 Testing with Motorola Edge 40 Neo**

After inserting the dedicated SIM card into the smartphone, it was necessary to configure the APN (Access Point Name) settings. This involved navigating to device settings and adding a new APN named “Internet”.

After disabling Airplane Mode, the device successfully connected to the O-RAN network.

To verify the network performance, an Ookla SpeedTest was performed [6]. The device initiated multiple parallel TCP connections to an Internet server, transmitting data chunks and computing the downlink throughput as the ratio of total downloaded bits to the duration of the transfer.

This test reported a downlink speed of 90 Mbps, confirming successful network performance.

Unfortunately, the uplink test could not be performed due to the short distance between the USRP and the device. In this short range scenario, the power control algorithm could not compensate correctly, with the result of an immediate disconnection of the UE from the 5G network.

#### **6.4.2 Testing with iPhone 14 A2890**

The network configuration remained unchanged across all devices, but some device-specific settings were required. On the iPhone 14 A2890, it was necessary to enable the 5G Stand-Alone (5G SA) mode, which allows the device to connect to a 5G network without relying on the 4G infrastructure (5G SA).

After switching the SIM card and configuring the APN, the device initially connected to the network but disconnected after multiple attempts due to network slicing issues.

Network slicing, a key feature of 5G, creates multiple virtual networks, each optimized for specific applications and services. In this case, an incompatibility in the slicing configuration blocked the device from maintaining a stable connection. One troubleshooting step was to remove the APN from the MMS configuration. But then another problem appeared: the IMS services blocked the connection. Since this is a private 5G network, cellular manufacturers impose IMS confirmation requirements, preventing devices from connecting to networks that do not provide IMS validation. As a result, the iPhone failed to maintain a stable connection.

### **6.4.3 Testing with Samsung A55 and A25**

For both the Samsung A55 and the Samsung A25, the configuration remained unchanged, as they share the same manufacturer.

After inserting the private SIM card, the Samsung A55 required the installation of an application from the Play Store called "NetMonster". This application allows users to access a hidden menu, making it possible to change the network type from the default NR/LTE/GSM/WCDMA to NR (New Radio).

After configuring the APN with the name "Internet" and setting up the network profile, the device connects to the network. To verify the O-RAN performance speed tests were conducted using the SpeedTest application by Ookla, with the downlink speed reaching 88.18 Mbps, demonstrating the network's adequate performance and similar performance to the Motorola Edge 40 Neo.

Similarly to the iPhone, the Samsung Devices also had disconnection issues. This was due to the fact that the O-RAN 5G is not certified for IMS services, which causes the devices to reject the connection after a few minutes.

But in this case, since they are based on an Android system, they offer more customization options compared to the Iphone. Previously, it was possible to disable and enable IMS services from a secret menu on the dialer, but with recent security patch releases, this menu is no longer available. To solve this issue, a computer was used to disable IMS services, through two applications installed on the mobile device. The first one uses APIs with elevated privileges via debug USB while the second tunes the different parameters then opening the main menu of the IMS services, where there are all the IMS provider search, the IMS services were disabled.

This did not cause any problem, as calls and messages over the 5G O-RAN network are still in the development phase. This solution perfectly matches the main objective of this thesis, which is to ensure continuous network connectivity for testing purposes.

#### 6.4.4 Testing with Astreo device

The Astreo device is a custom board, composed of two components: Raspberry Pi Compute Module and 5G module SIMCOM SIM8380G-M2.

The access to the system is managed via SSH (Secure Shell), allowing remote configuration and monitoring. SSH connection is a protocol that enables a secure and encrypted link between a client and a remote server to execute commands or transfer files.

After powering on the core and running the O-RAN gNB, and inserting the SIM card into the board, it was necessary to update the current APN in the configuration files and change the band to 77 from 78.

By checking the logs of the ppp0, a virtual interface of the Astreo device, it was possible to verify the connection from the device to the O-RAN network since this interface is automatically created once the 5G SIMCOM module successfully set up the connection.

From an Astreo perspective, the process involves modifying the configuration using AT commands. The log output in Figure 7 confirms the establishment of a 5G connection after applying these modifications.

```
Jan 10 14:33:54 astreo5G chat[1789]: send (at+csysssel="lte_band",7^M)
Jan 10 14:33:54 astreo5G chat[1789]: expect (OK)
Jan 10 14:33:54 astreo5G chat[1789]: ^M
Jan 10 14:33:54 astreo5G chat[1789]: at+csysssel="lte_band",7^M^M
Jan 10 14:33:54 astreo5G chat[1789]: OK
Jan 10 14:33:54 astreo5G chat[1789]: -- got it
Jan 10 14:33:54 astreo5G chat[1789]: send (at+csysssel="nr5g_band",77^M)
Jan 10 14:33:54 astreo5G chat[1789]: expect (OK)
Jan 10 14:33:54 astreo5G chat[1789]: ^M
Jan 10 14:33:54 astreo5G chat[1789]: at+csysssel="nr5g_band",77^M^M
Jan 10 14:33:54 astreo5G chat[1789]: OK
Jan 10 14:33:54 astreo5G chat[1789]: -- got it
Jan 10 14:33:54 astreo5G chat[1789]: send (AT+CGDCONT=1,"IP","internet","",0,0^M)
Jan 10 14:33:54 astreo5G chat[1789]: expect (OK)
Jan 10 14:33:54 astreo5G chat[1789]: ^M
Jan 10 14:33:54 astreo5G chat[1789]: AT+CGDCONT=1,"IP","internet","",0,0^M^M
Jan 10 14:33:54 astreo5G chat[1789]: OK
Jan 10 14:33:54 astreo5G chat[1789]: -- got it
Jan 10 14:33:54 astreo5G chat[1789]: send (ATD*99#^M)
Jan 10 14:33:55 astreo5G chat[1789]: timeout set to 22 seconds
Jan 10 14:33:55 astreo5G chat[1789]: expect (CONNECT)
Jan 10 14:33:55 astreo5G chat[1789]: ^M
Jan 10 14:33:55 astreo5G chat[1789]: ATD*99#^M^M
Jan 10 14:33:55 astreo5G chat[1789]: CONNECT
Jan 10 14:33:55 astreo5G chat[1789]: -- got it
Jan 10 14:33:55 astreo5G chat[1789]: send (^M)
Jan 10 14:33:55 astreo5G pppd[1493]: Serial connection established.
Jan 10 14:33:55 astreo5G pppd[1493]: Using interface ppp0
Jan 10 14:33:55 astreo5G pppd[1493]: Connect: ppp0 <-> /dev/ttyUSB3
```

Figure 7: Log output from the Astreo Board, showing 5G connection status

To test the connection between the 5G module and the O-RAN network, the IPERF tool was used. IPERF is a software application that can act as both a server and a client, allowing measurement of performance metrics such as bandwidth and throughput, removing fluctuations due to Internet connection, and measuring the effective speed.

In this case the device is able to transmit also in uplink, since it is a 5G module for experimental purposes, it is not designed to connect to a gNB located kilometers away.

First, the IPERF server was started on the appropriate pod in the O-RAN, specifically the User Plane Function (UPF) pod. This pod is the first component of the PDU tunnel established when the connection starts.

Before starting the client in Astreo, a new route had to be configured to connect to the network's IP and associate it with the ppp0 interface, since the board includes other interfaces such as Wi-Fi.

Initially a basic ping was used to test the connectivity:

```
root@astreo5G:~# ping 10.45.0.1
PING 10.45.0.1 (10.45.0.1) 56(84) bytes of data.
64 bytes from 10.45.0.1: icmp_seq=1 ttl=64 time=15.1 ms
64 bytes from 10.45.0.1: icmp_seq=2 ttl=64 time=61.7 ms
64 bytes from 10.45.0.1: icmp_seq=3 ttl=64 time=9.77 ms
64 bytes from 10.45.0.1: icmp_seq=4 ttl=64 time=17.2 ms
64 bytes from 10.45.0.1: icmp_seq=5 ttl=64 time=26.0 ms
```

Figure 8: Ping test from the Astreo Board to the UPF POD within the O-RAN 5G Core

Then, the IPERF client was started, first using the TCP (Transmission Control Protocol) without specifying the bandwidth.

```
local 10.45.0.22 port 45320 connected to 10.45.0.1 port 5201
Interval      Transfer      Bitrate      Retr      Cwnd
0.00-1.00    sec  445 KBytes   3.65 Mbits/sec  0    45.2 KBytes
1.00-2.00    sec  255 KBytes   2.08 Mbits/sec  0    58.0 KBytes
2.00-3.00    sec  445 KBytes   3.65 Mbits/sec  0    72.1 KBytes
3.00-4.00    sec  382 KBytes   3.13 Mbits/sec  0    87.7 KBytes
4.00-5.00    sec  191 KBytes   1.56 Mbits/sec  0    100 KBytes
5.00-6.00    sec  509 KBytes   4.17 Mbits/sec  0    129 KBytes
```

Figure 9: IPERF client running TCP to measure network throughput

The second test using IPERF was conducted in reverse mode, where the server transmitted data to the client, using UDP (User Datagram Protocol) with a fixed bandwidth of 100 Mbit/s.

UDP was chosen for this test because, unlike TCP, it does not require control messages such as acknowledgments (ACK), with the result of reducing the transmission overhead.

```
root@astreo5G:~# iperf3 -c 10.45.0.1 -R -u -b 100M
Connecting to host 10.45.0.1, port 5201
Reverse mode, remote host 10.45.0.1 is sending
[ 5] local 10.45.0.22 port 49435 connected to 10.45.0.1 port 5201
[ ID] Interval      Transfer      Bitrate      Jitter      Lost/Total Datagrams
[ 5] 0.00-1.00    sec 5.38 MBytes 45.1 Mbits/sec 0.274 ms 1690/5587 (30%)
[ 5] 1.00-2.00    sec 6.45 MBytes 54.1 Mbits/sec 3.150 ms 4424/9093 (49%)
[ 5] 2.00-3.00    sec 3.75 MBytes 31.5 Mbits/sec 0.232 ms 6223/8942 (70%)
[ 5] 3.00-4.00    sec 5.14 MBytes 43.1 Mbits/sec 0.365 ms 2803/6525 (43%)
[ 5] 4.00-5.00    sec 4.28 MBytes 35.9 Mbits/sec 0.334 ms 84/3183 (2.6%)
[ 5] 5.00-6.00    sec 1.93 MBytes 16.2 Mbits/sec 1.195 ms 150/1548 (9.7%)
[ 5] 6.00-7.00    sec 1.70 MBytes 14.2 Mbits/sec 1.140 ms 38/1266 (3%)
[ 5] 7.00-8.00    sec 1.30 MBytes 10.9 Mbits/sec 0.806 ms 0/938 (0%)
[ 5] 8.00-9.00    sec 854 KBytes 7.00 Mbits/sec 0.538 ms 0/604 (0%)
```

Figure 10: IPERF Client performing UDP test in reverse mode (server to client transmission)

The final test involved maintaining the previous settings but switching to forward mode, where the client receives data from the server.

```
root@astreo5G:~# iperf3 -c 10.45.0.1 -u -b 10M
Connecting to host 10.45.0.1, port 5201
[ 5] local 10.45.0.23 port 43289 connected to 10.45.0.1 port 5201
[ ID] Interval      Transfer      Bitrate      Total Datagrams
[ 5] 0.00-1.00    sec 1.19 MBytes 10.0 Mbits/sec 863
[ 5] 1.00-2.00    sec 1.19 MBytes 10.0 Mbits/sec 863
[ 5] 2.00-3.00    sec 1.19 MBytes 10.0 Mbits/sec 864
[ 5] 3.00-4.00    sec 1.19 MBytes 10.0 Mbits/sec 863
[ 5] 4.00-5.00    sec 1.19 MBytes 10.0 Mbits/sec 863
[ 5] 5.00-6.00    sec 1.19 MBytes 10.0 Mbits/sec 863
[ 5] 6.00-7.00    sec 1.19 MBytes 10.0 Mbits/sec 864
[ 5] 7.00-8.00    sec 1.19 MBytes 10.0 Mbits/sec 863
[ 5] 8.00-9.00    sec 1.19 MBytes 10.0 Mbits/sec 863
```

Figure 11: IPERF client performing UDP test in forward mode (client to server transmission)

The results confirmed that different types of devices, including smartphones and embedded modules, can have varying behaviors and problems.

After these tests, it is possible to conclude the following:

- Astreo board is optimal for performing uplink tests.
- Smartphones are better suited for mobility tests due to their portability; whereas the Astreo requires a constant power source.

## 7 Testing and Results: Single and Multi Device Scenarios

Various tests were performed to evaluate the performance of the O-RAN network using four devices: two Samsung A25, Samsung A55 and the Astreo device. These devices were selected for their stable connection performance, which ensures reliable evaluations.

For the tests, a graphical representation of the signal quality indicators at the MAC level, showing their behavior across different scenarios, is included.

The key parameters in the OAI configuration file are initialized as follows:

<b>Parameter</b>	<b>Value</b>
Frequency band	3900 MHz (Band 77)
Channel Bandwidth	40 MHz
Output Power	20 dBm
Target SNR in PUSCH	30 dB
Target SNR in PUCCH	20 dB
Subcarrier Spacing	30 kHz
SSB power	-25 dBm
Max MCS uplink	28
Downlink slots	5 ms

The key metrics were extracted from the MAC scheduler log file, which is updated every second to minimize excessive data accumulation.

To retrieve these metrics, a Python script was developed, inside the OAI POD, which reads the log file every second and exports the data to a CSV file containing timestamps, UE IDs, and all relevant metrics (detailed in Chapter 6.2).

The dynamic assignment of UE Unique Identifiers was not linear, as they changed even for the same device due to intermittent disconnections, complicating post-processing.

To deal with this issue, the UE IDs were manually reassigned in post-processing as shown in the following list:

- UE ID 1: Samsung A25.
- UE ID 2: Samsung A25 (Second Device).
- UE ID 3: Astreo Device.
- UE ID 4: Samsung A55.

To validate the RSRP (Reference Signal Received Power) values, additional metrics were collected directly from mobile devices. Real-time signal strength readings were obtained by accessing the Service Mode menu through the dialer (\*#0011#) and selecting STACK 1.

```
Band: 77, ARFCN: 659424
PCI: 0
CellId: 0x0000000000BC614E
CP: 0
CHBW: 40, BWP: 0, CHUIBW: 40
SSB Index:0
SSB RSRP:-80
SCS: 30
RSRP:-77, RSRQ:-10
RSSI: -49
NR AvgRSRP: -74, AvgRSRQ: -10
Act ANT:0b1111, RSRP Diff:-5
SINR: 27, RI: 0, CQI: 0
Tx Pwr: -16
TP DL: 96Mbps
TP UL: 1Mbps
MCS DL/UL : 27/17
```

Figure 12: Samsung A25 Service Mode Menu

From this secret menu, it was possible to confirm the operating frequency (3900 MHz, Band 77) and the channel bandwidth (40 MHz), confirming the alignment between the log values and the observed data. Additionally, the MCS reference table did not change and remained the 5G standard Table 1 (256-QAM) in all tests.

For further evaluation, the NetMonster application was used to measure the SNR at the device level, which showed values between 20 dBm and 22 dBm that matched the target SNR (20 dB) in the OAI configuration file.

The post-processing involved three dedicated Python scripts, each tailored for a specific test type.

The Pandas library was used for data analysis, while Matplotlib was used for plot realization.

A crucial pre-processing step was filtering out meaningless data from the CSV file.

Instead of the MCS scheduler BLER  $\frac{B}{A}$ , a refined approach was used:

$$\text{BLER} = \frac{B + 2C + 3D}{A + 2B + 3C + 4D}$$

For practical purposes, assuming  $C = D = 0$ , the BLER was approximated as:

$$\text{BLER} \approx \frac{B}{A + 2B}$$

Where A,B,C,D were the downlink transmission rounds from the gNB per HARQ process defined as follows:

$$A = \text{First Round}, \quad B = \text{Second Round}, \quad C = \text{Third Round}, \quad D = \text{Fourth Round}.$$

This method applied to both uplink and downlink calculations. It's worth noting that this definition differs from the one used by the MAC scheduler to evaluate BLER, as this formulation also takes retransmissions into account, which are not required for MCS adaptation by the MAC scheduler.

## 7.1 Single Device in Different Locations

The first single-device test employed the Samsung A25. The device was placed in 13 fixed positions, 90 centimeters apart, from closer to furthest from the USRP, maintaining the line of sight. The device height was fixed at 40 cm.

During testing, a 500 MB file was continuously downloaded to generate realistic network load, rather than relying only on control messages.

In this initial test, all the metrics were collected within a single established connection without restarting the gNB.

The phone was connected to the O-RAN network and data recorded for 30 seconds at each position.

In post-processing, the following steps were carried out:

1. Verification of UE ID consistency: given that certain metrics values reset when the UE ID changes (e.g. total transmitted bytes), a check was conducted to identify such occurrences. The UE ID was then standardized to 1, corresponding to the Samsung A25.
2. Segmentation of the dataset: the CSV file was divided into 30-second time intervals, with each segment assigned the corresponding distance measurement.
3. Computation of average metrics: the collected metrics were averaged for each distance. Since the values were originally expressed in dB and dBm, they were first converted to a linear scale, averaged, and subsequently reconverted to dBm for representation.
4. BLER estimation: BLER was computed using the least approximated methodology to ensure a more precise evaluation of network performance.

One key observation can be made on the work of the power control, as depicted in fig. 13, the SNR is maintained at approximately 30 dB (which was the target SNR):

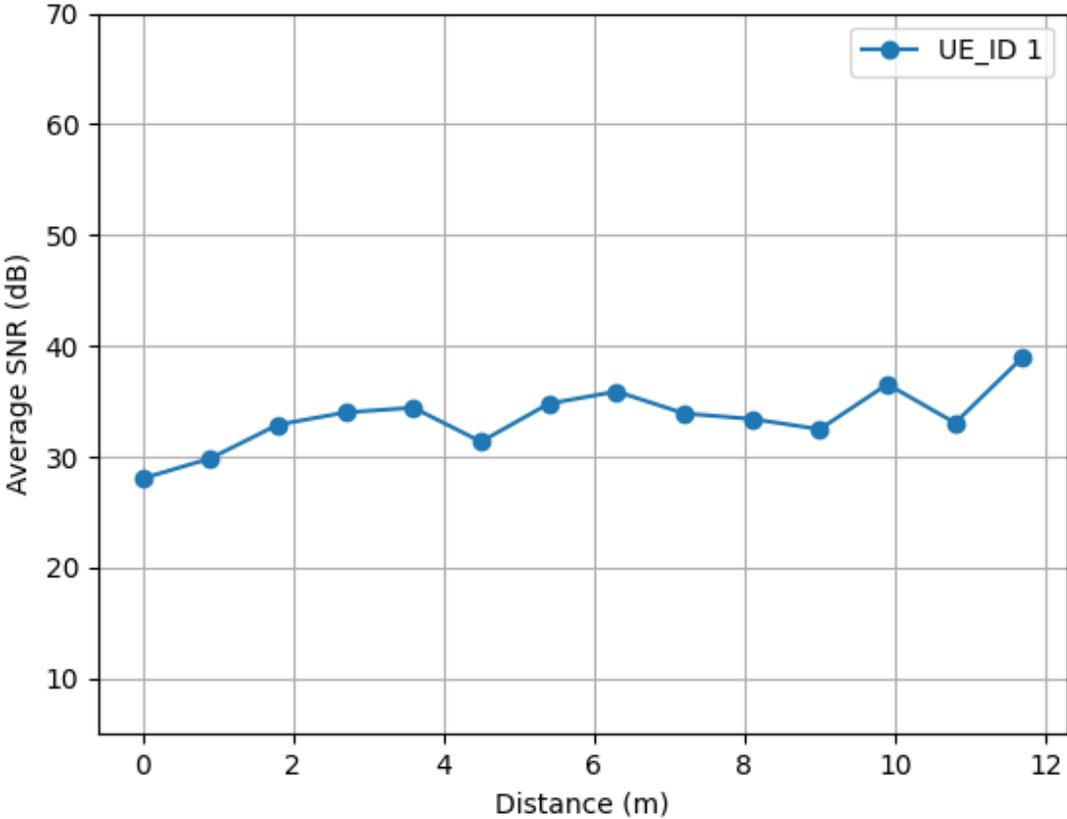


Figure 13: Average SNR over 13 positions for Samsung A25

This trend can be appreciated in the downlink BLER, that hovers around  $10^{-1}$ , which in a conventional 5G network would be considered high. However, in the O-RAN framework, such BLER values are optimized for maximizing throughput, even at the cost of network reliability.

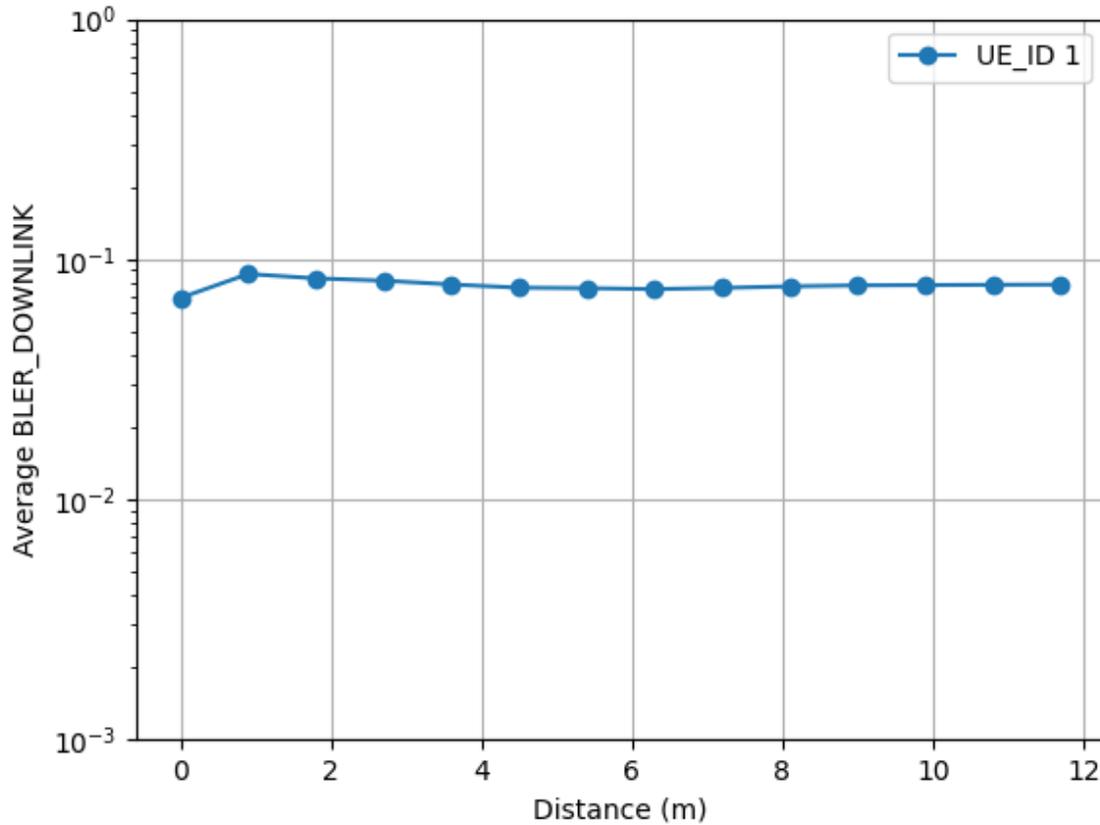


Figure 14: Average Downlink BLER over 13 positions for Samsung A25

Another key parameter analyzed is Power Headroom. As the device moves farther from the base station, the available uplink transmission power margin decreases. Initially, the Power Headroom is 45 dB and declines to a minimum of 20 dB, with minor fluctuations.

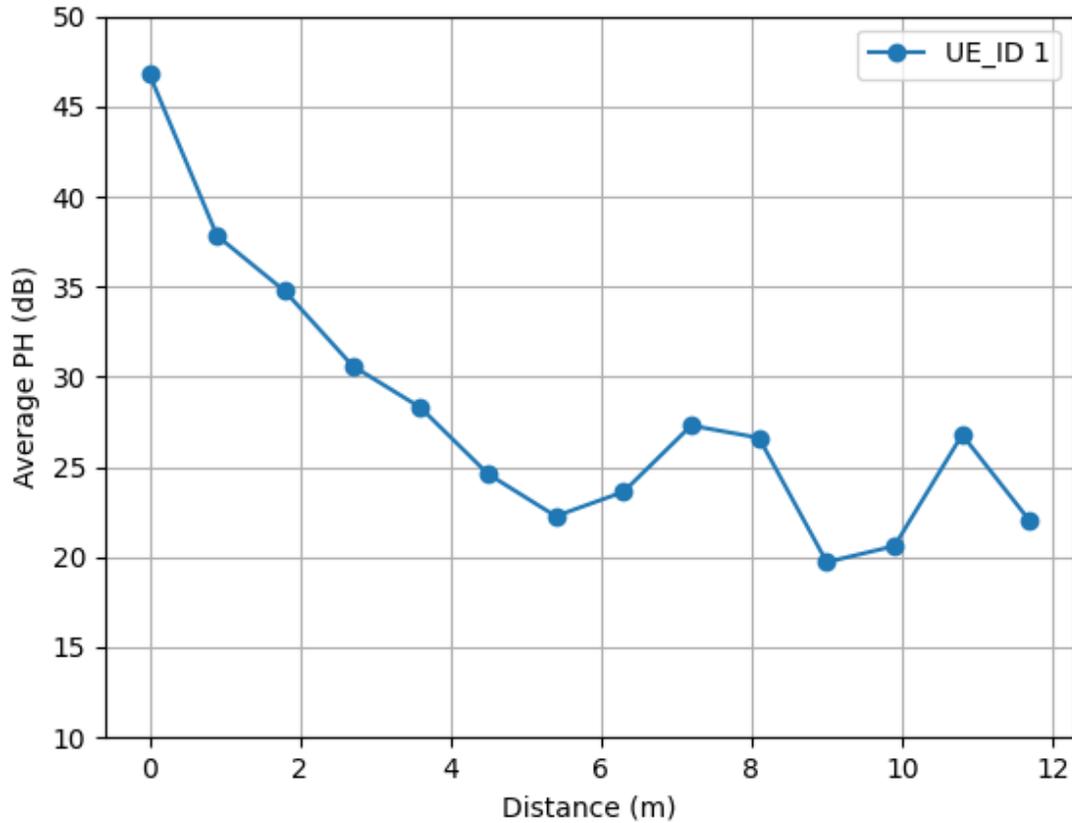


Figure 15: Average Power Headroom over 13 positions for Samsung A25

PCMAX, which represents the maximum transmission power of the UE, remains constant across all positions, as expected. This value is reported by the UE to the gNB and in this case is equal to 21 dBm.

To evaluate the network behavior for an increased gNB output power, another experiment was conducted. In this case, at each position, the gNB was restarted, and the connection was maintained for 126 seconds. This approach provided a more accurate representation, as each position had 126 samples instead of just 30. To highlight the difference, both power levels are compared in the same graph.

The first notable result is that the downlink BLER is slightly lower with higher power, with some expected fluctuations. Additionally, as the device moves away from the USRP, RSRP decreases, as shown in the figure below, since the power control mechanism does not influence RSRP that it is a passive measurement taken by the UE.

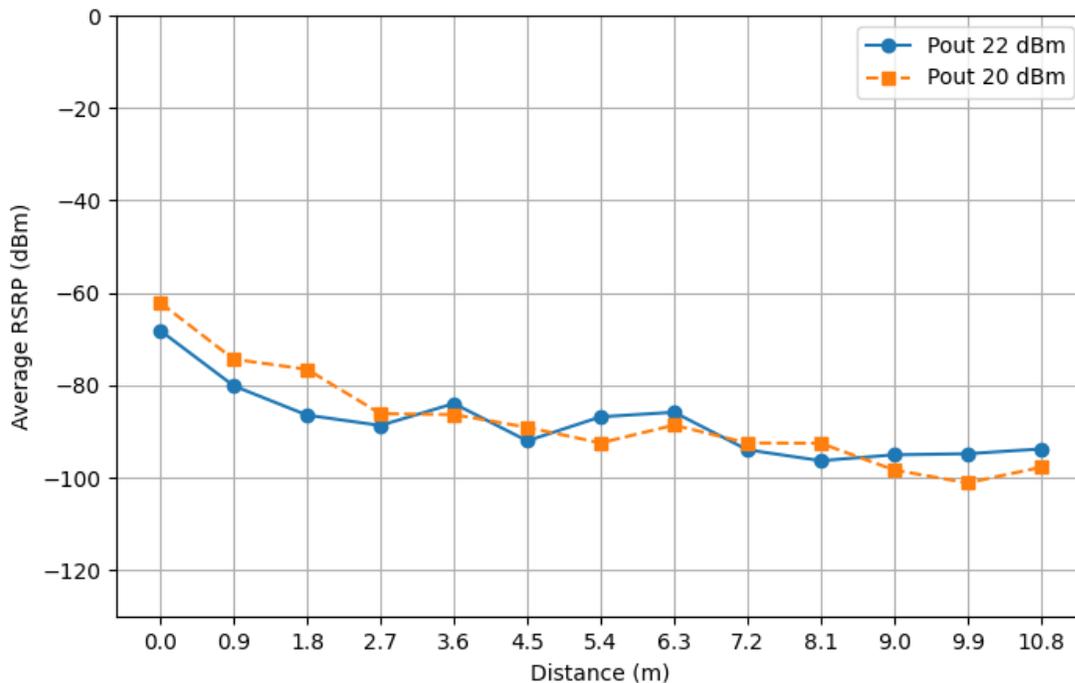


Figure 16: Average RSRP over distances for two different gNB output power levels

The BLER\_DLSCH of the MAC scheduler was also reported, as it influences the MCS selection. The system maintained  $Q=1$ , meaning that it continued to use the 256-QAM table.

The following graphs illustrate that when BLER\_DLSCH increases, the downlink MCS decreases to maintain signal quality.

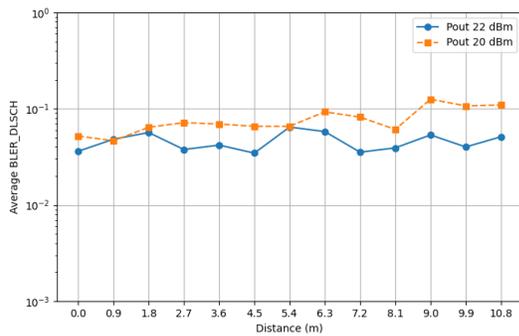


Figure 17: BLER\_DLSCH over 13 positions for different power levels

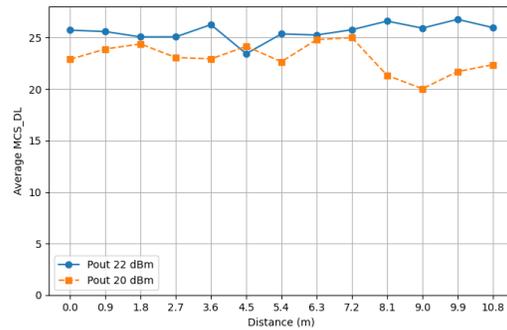


Figure 18: Downlink MCS over 13 positions for different power levels

The final comparison evaluates the downlink BLER in both power settings, showing that the downlink BLER is lower in the 22 dBm scenario, as expected.

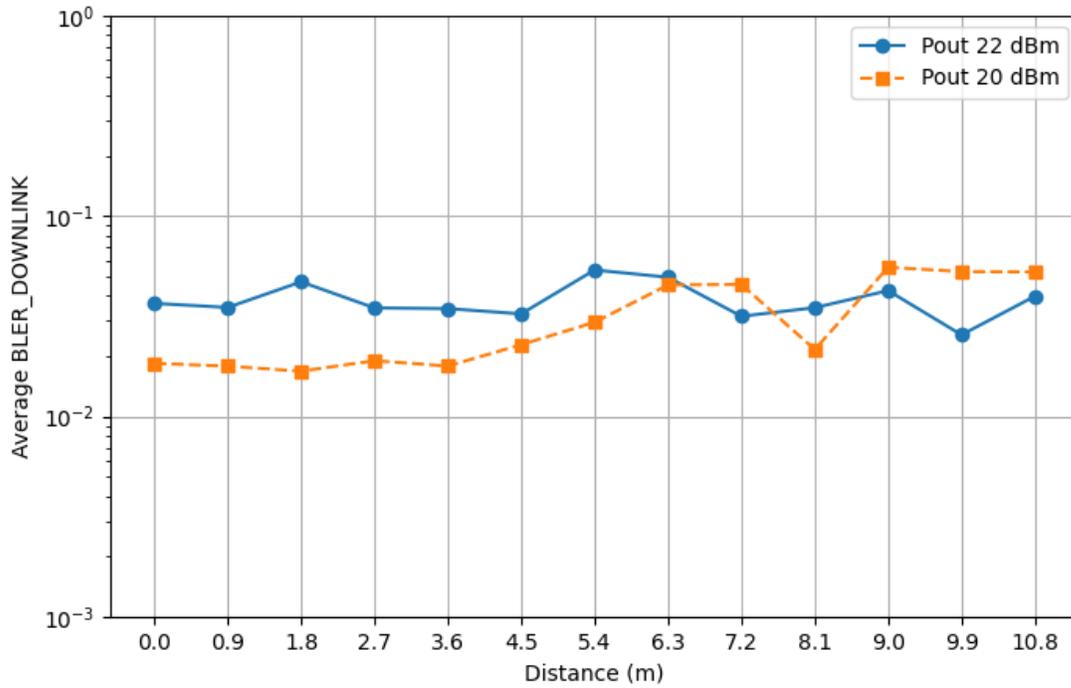


Figure 19: Downlink BLER over distance for two gNB output power levels

## 7.2 Multi Device Performance Analysis in a Fixed Location

To evaluate device performance under identical network conditions, tests were conducted in a fixed location at 40 centimeters.

The performance of Astreo Device and of the Samsung A25 were measured over a 150 seconds which results in 150 samples. The Samsung A25 downloaded files of 500 MB, while the Astreo device fetched data from a test server.

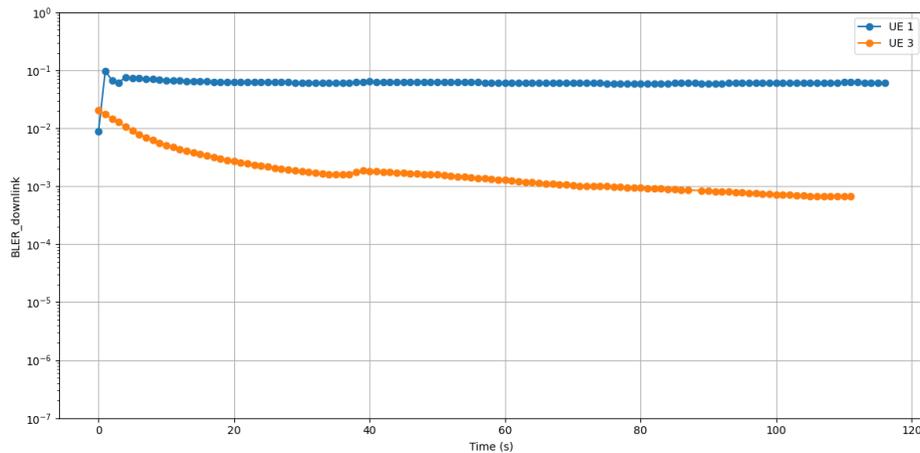


Figure 20: Downlink BLER: Astreo vs. Samsung A25

Despite having a lower BLER, the Astreo device achieved a lower throughput than the Samsung A25. The downlink MCS values indicate that Astreo consistently achieved higher MCS values compared to the Samsung A25.

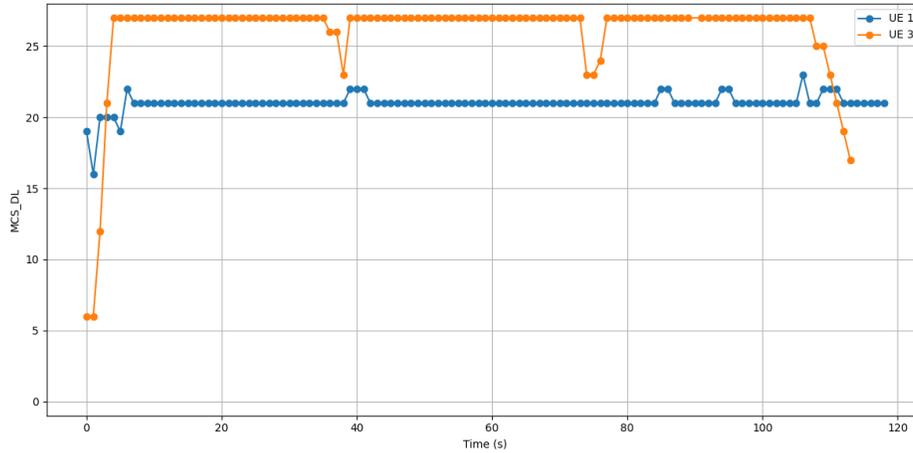


Figure 21: Downlink MCS: Astreo vs. Samsung A25

Although Astreo demonstrated better signal quality, the Samsung A25 achieved twice the download speed.

The throughput was computed using the following formula:

$$T = \frac{\text{Bits Downloaded}}{\text{Time (s)}}$$

Since 1 byte = 8 bits, the throughput in bits per second (bps) is:

$$T = \frac{\text{Bytes Downloaded} \times 8}{\text{Time (s)}}$$

If measured in Megabits per second (Mbps):

$$T = \frac{\text{Bytes Downloaded} \times 8}{\text{Time (s)} \times 10^6} \text{ Mbps}$$

Results:

- Samsung A25: 615.34 MB over 150 seconds → 32 Mbps.
- Astreo Device: 333 MB over 150 seconds → 17.76 Mbps.

### 7.3 Performance with an Increasing Number of Devices

The final test involved comparing the results when multiple devices were connected, ranging from one to four. This experiment enabled an evaluation of how the Proportional Fair scheduler distributes resources under varying device loads.

The tested devices included:

- Samsung A25 (ID 1, ID 2).
- Samsung A55 (ID 4).
- Astreo Device (ID 3).

Each device attempted to download as much data as possible over a 126 seconds window. The CSV data was processed using a moving average window of 10 samples to smooth fluctuations. The test was conducted in line-of-sight conditions at a distance of 5 meters.

The following figure demonstrates that power control struggled to maintain the target SNR, resulting in fluctuations around 30 dB even when four devices were connected.

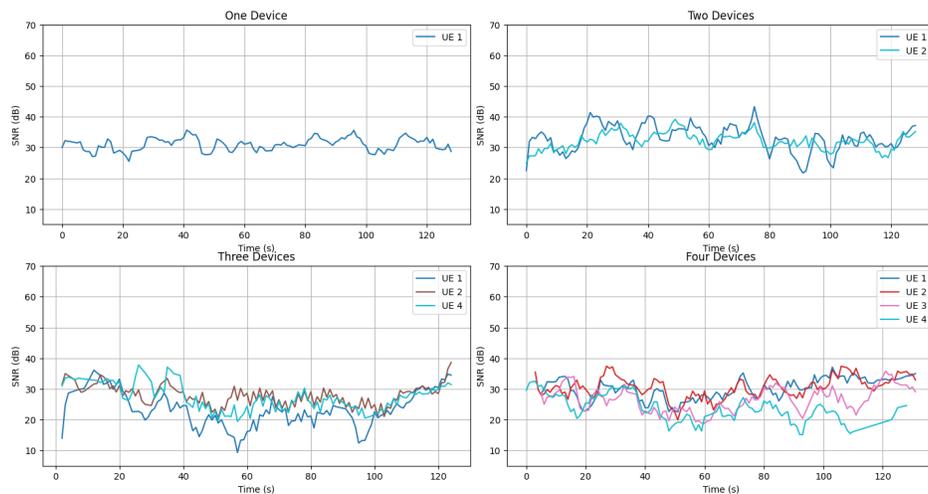


Figure 22: Moving Average SNR over 4 devices

As a result of the SNR stability, the BLER in downlink remained relatively stable across all four test scenarios.

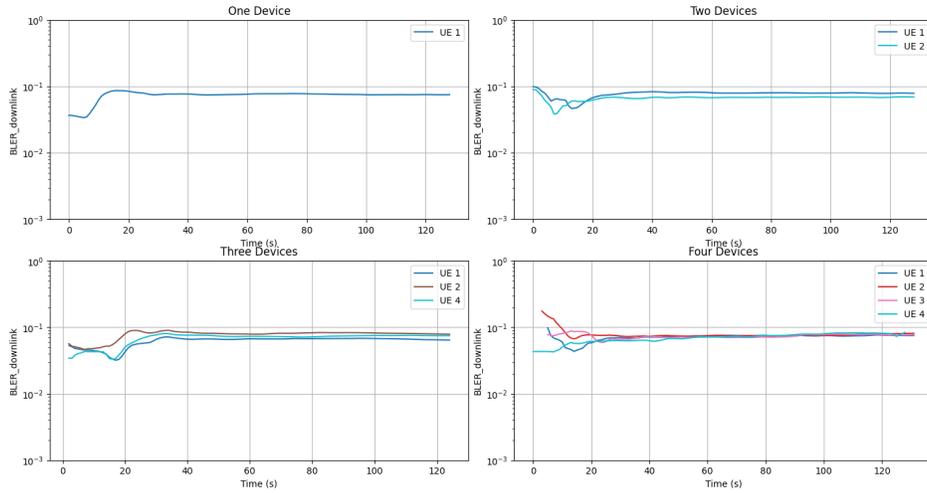


Figure 23: Moving Average downlink BLER over 4 devices

The scheduler dynamically adjusted the MCS levels based on network conditions, aiming to maintain the DLSCH\_BLER within a target range of 10-30%. However, with four devices, the MCS selection became more unstable, leading to greater fluctuations.

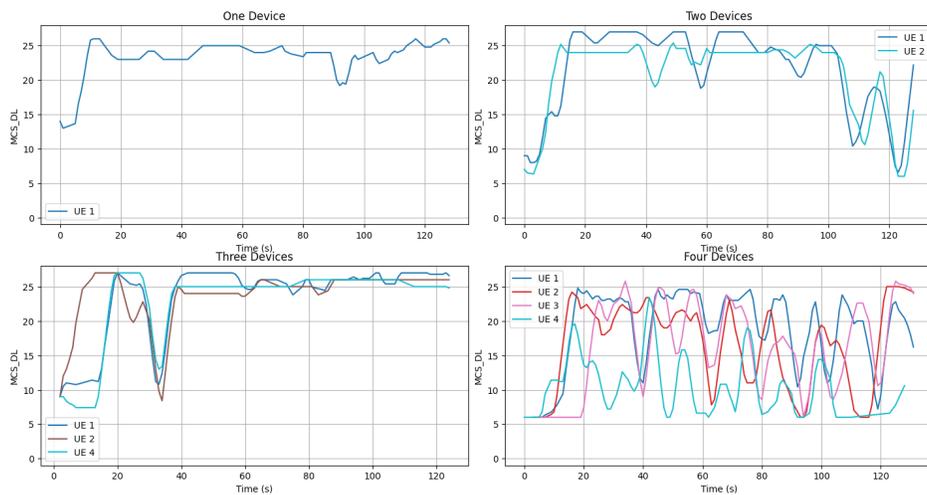


Figure 24: Moving Average of the downlink MCS over four devices

These results provide insights into how the O-RAN scheduler distributes network resources when handling multiple devices under varying load conditions.

## 8 Conclusions

The goal of this thesis was to analyze the performance of a 5G O-RAN network developed using OpenAirInterface for the gNB and Open5GS for the core network, both deployed with OpenShift. Metrics such as RSRP, SNR, BLER, and MCS were collected across different tests to assess how the network behaves with various types of devices. To perform the analysis different scenarios were tested, where network metrics were collected using single to multiple devices deployed in various locations, observing how the O-RAN network behaved differently as more users connected. Results confirmed the presence of power control, which successfully maintained the SNR around the targeted values, although some fluctuations occurred during higher network load situations. The downlink BLER ranged between 10% and 30%. These values would be considered high in a traditional 5G network since up to one-tenth of the packets could be lost. However, in O-RAN, optimization focuses on maximizing throughput rather than maintaining strict reliability. Some devices, particularly the iPhone 14, showed compatibility issues, mainly due to the lack of network slicing, preventing successful connection. Some tests, such as using two USRPs instead of one, could not be executed due to the lack of instrumentation. Additionally, interpreting logs from the MAC scheduler was challenging due to the dynamic assignment of user's IDs, increasing the complexity and time needed for data processing. The absence of an official guide for OAI configuration files significantly increased results comprehension. In future tests, additional parameters in the OAI configuration file could be adjusted to enhance performance and reduce the currently high BLER. Further research should also include various device types beyond mobile phones and different environmental conditions. In this study, tests were limited to a maximum line-of-sight of 13 meters and no test were conducted in outdoor condition; upcoming experiments should incorporate non-line-of-sight scenarios, including obstacles such as walls and other physical barriers. However, several technical challenges remain, such as improving connection stability and compatibility with commercial devices.

In this section, additional visualizations generated from other parameters are included. Although they do not contribute to key evaluations, they are included for reference and further observation.

## A Single Device in Different Location

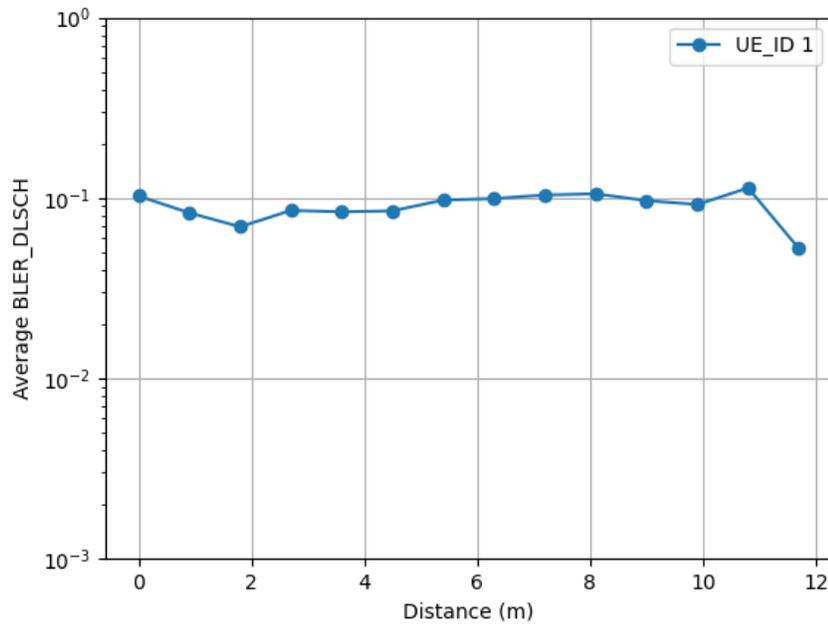


Figure 25: Average BLER DLSCH over 13 positions for Samsung A25

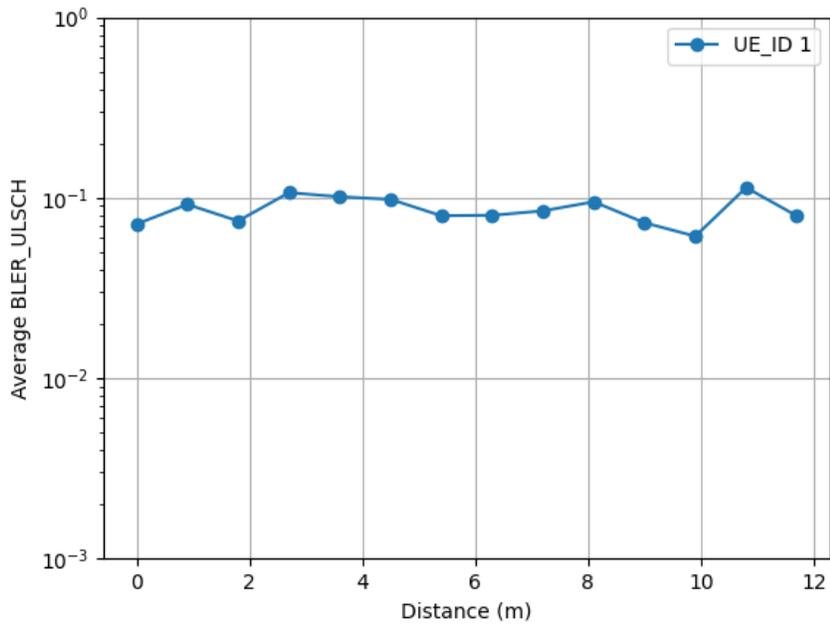


Figure 26: Average ULSCH BLER over 13 positions for Samsung A25

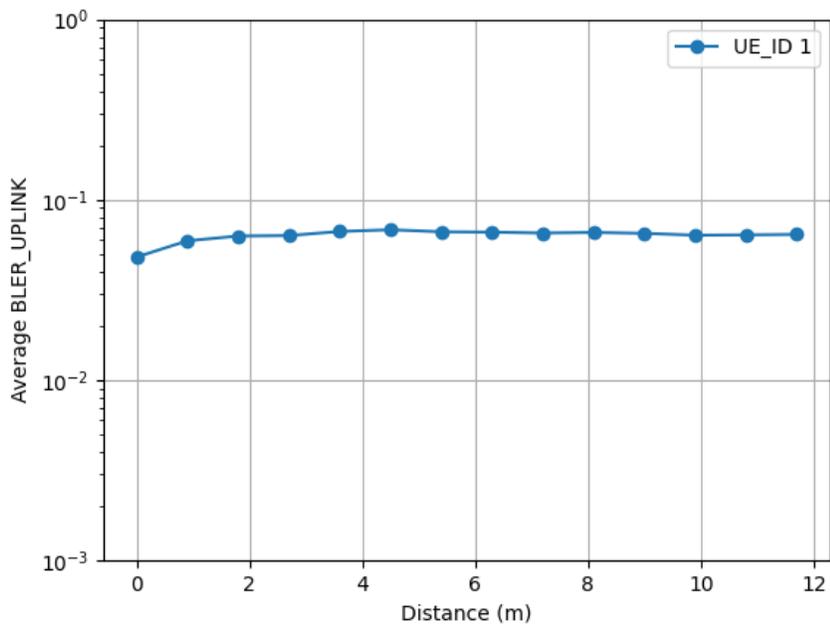


Figure 27: Average Uplink BLER over 13 positions for Samsung A25

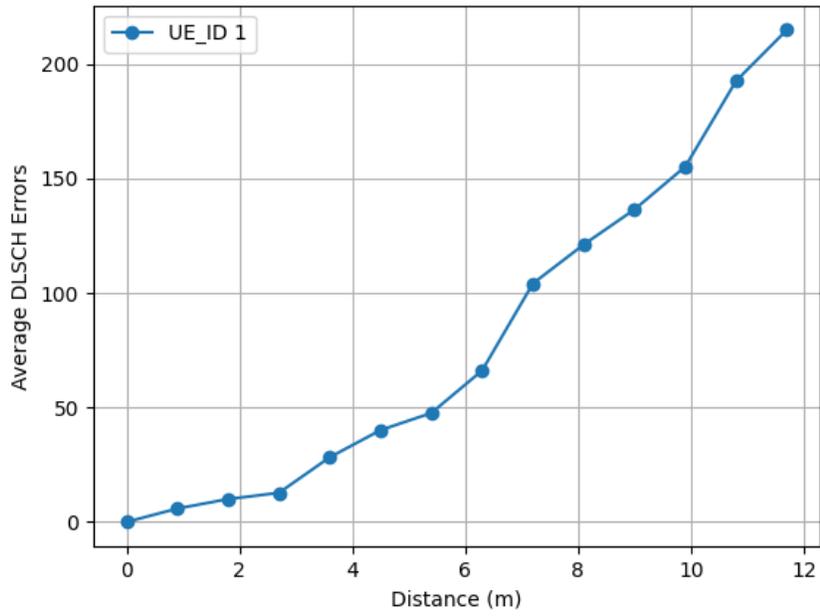


Figure 28: Average Downlink Errors over 13 positions for Samsung A25

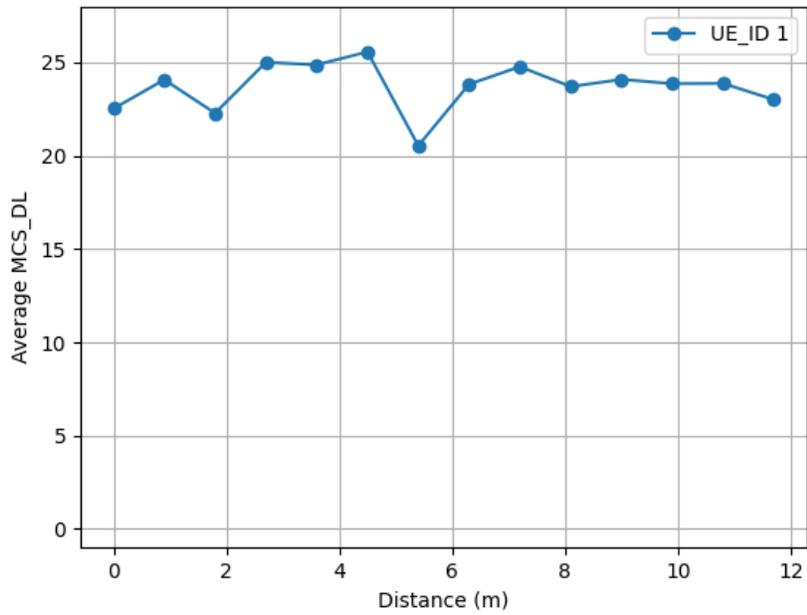


Figure 29: Average Downlink MCS over 13 positions for Samsung A25

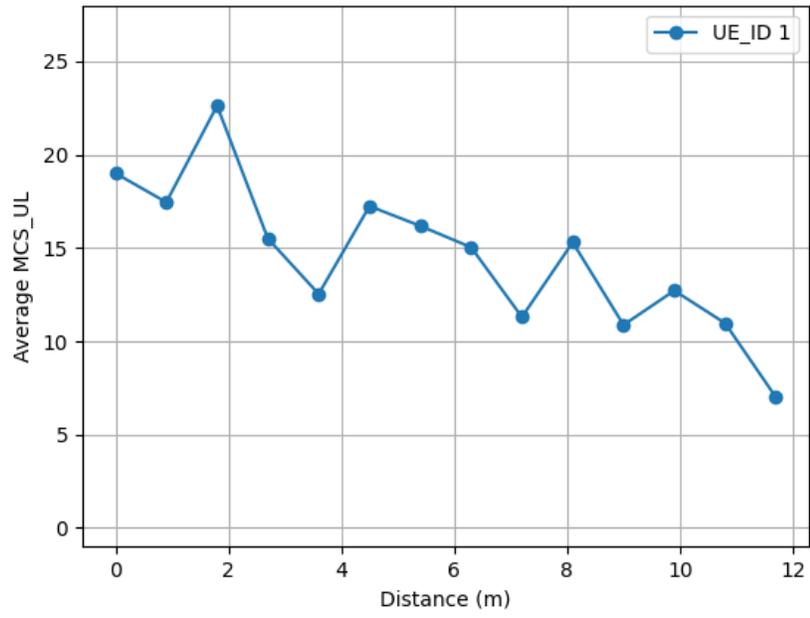


Figure 30: Average Uplink MCS over 13 positions for Samsung A25

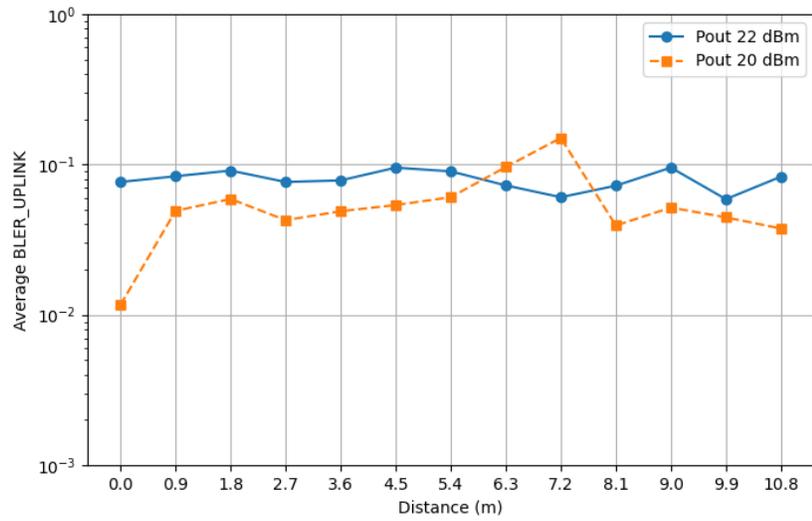


Figure 31: Average Uplink BLER over distance for two gNB output power levels

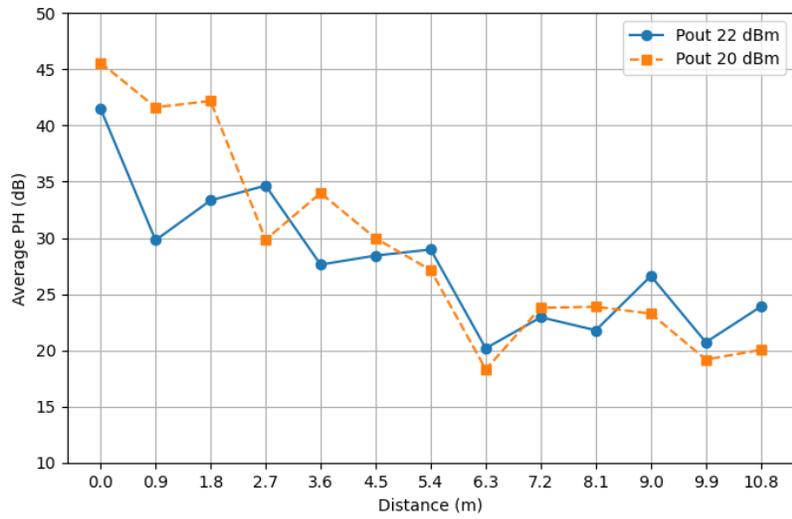


Figure 32: Average Power Headroom distance for two gNB output power levels

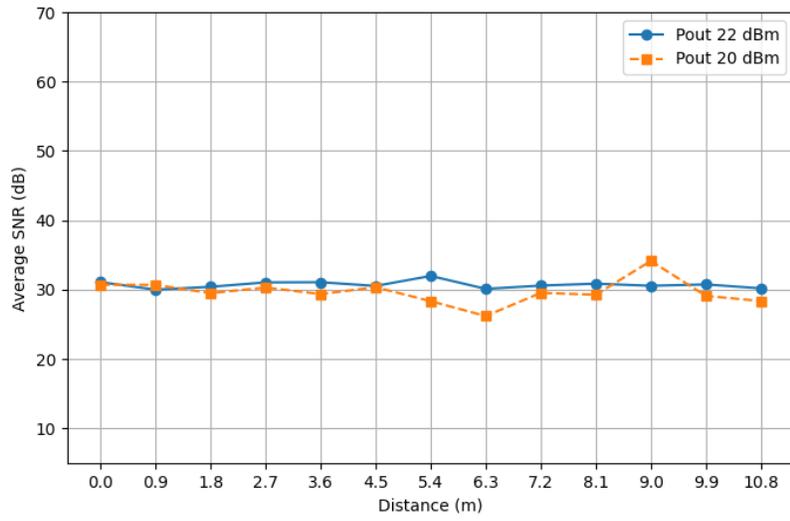


Figure 33: Average SNR over distance for two gNB output power levels

# B Multi Device Performance Analysis in a Fixed Location

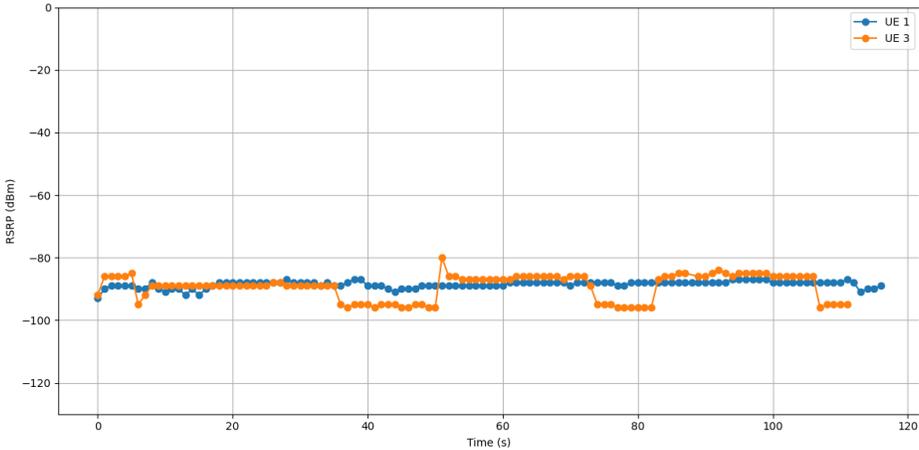


Figure 34: RSRP: Astreo vs. Samsung A25

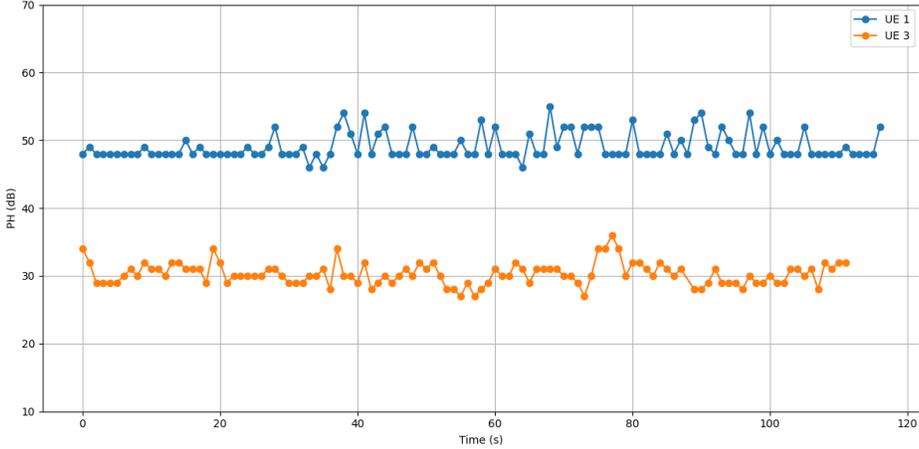


Figure 35: Power Headroom: Astreo vs. Samsung A25

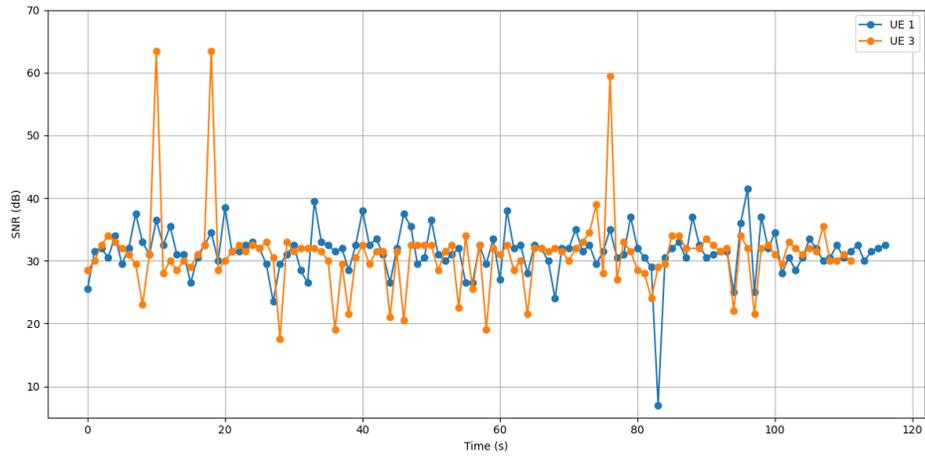


Figure 36: SNR: Astreo vs. Samsung A25

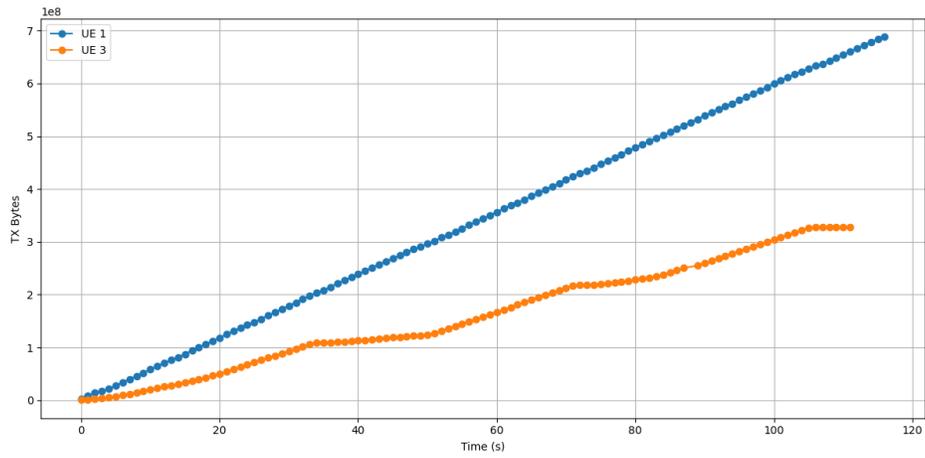


Figure 37: Transmitted Bytes: Astreo vs. Samsung A25

## C Performance with an Increasing Number of Devices

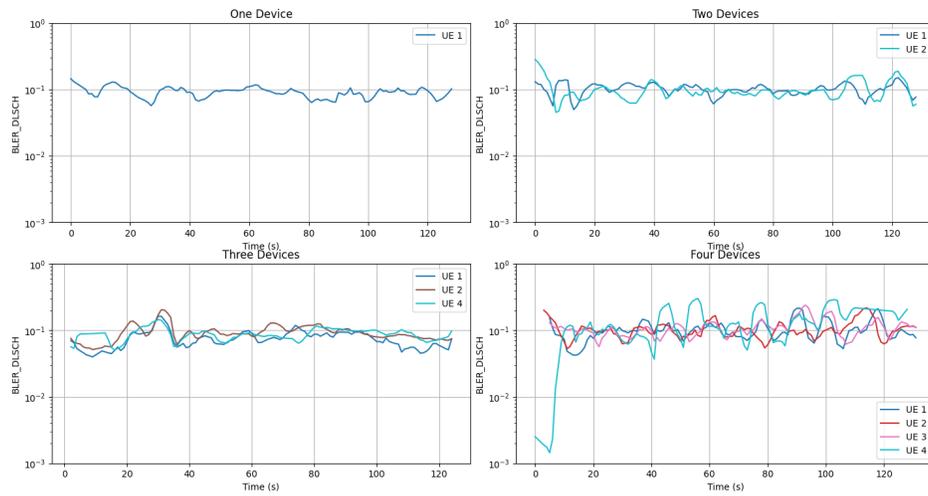


Figure 38: Moving Average DLSCH BLER over 4 devices

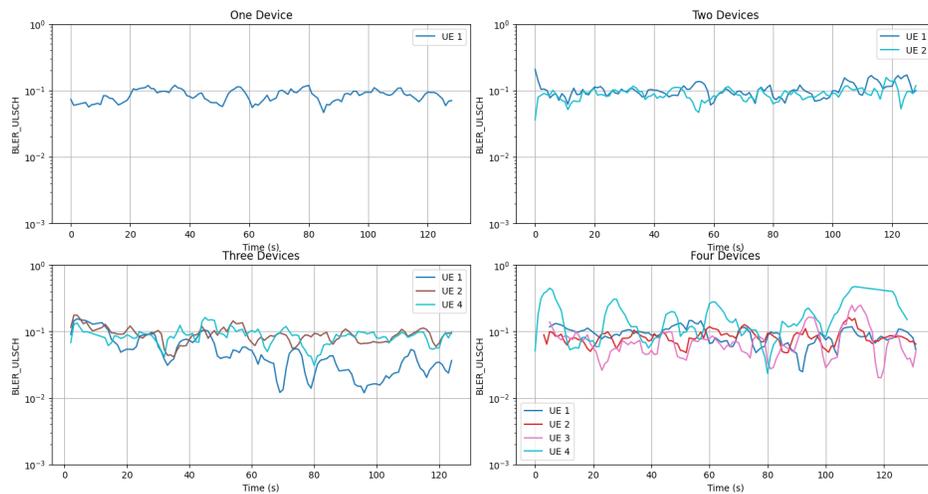


Figure 39: Moving Average ULSCH BLER over 4 devices

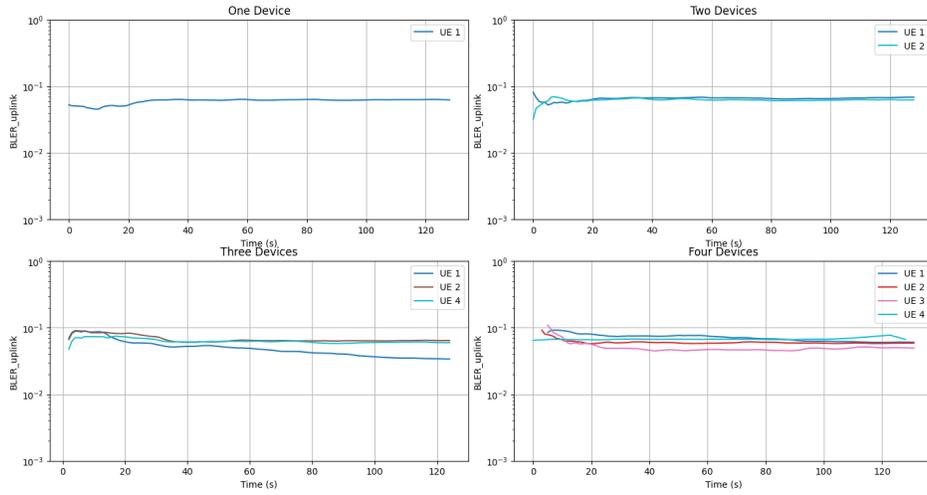


Figure 40: Moving Average Uplink BLER over 4 devices

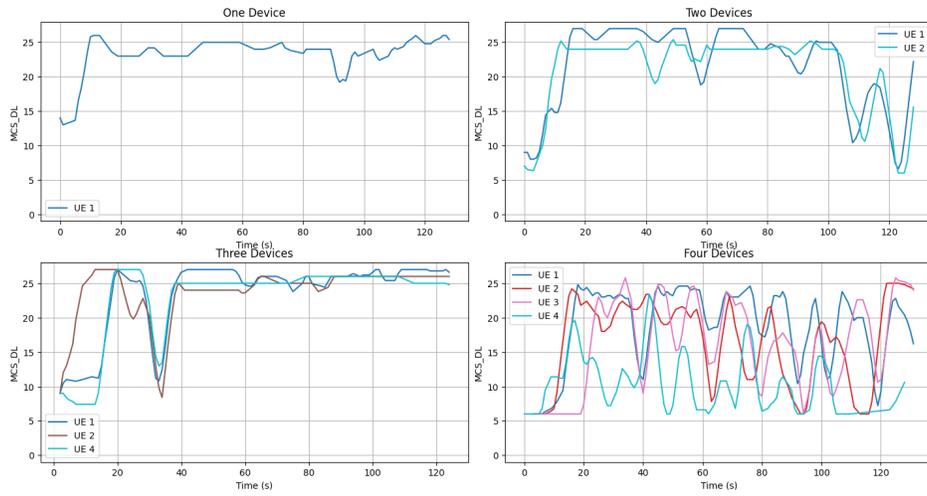


Figure 41: Moving Average Downlink MCS over 4 devices

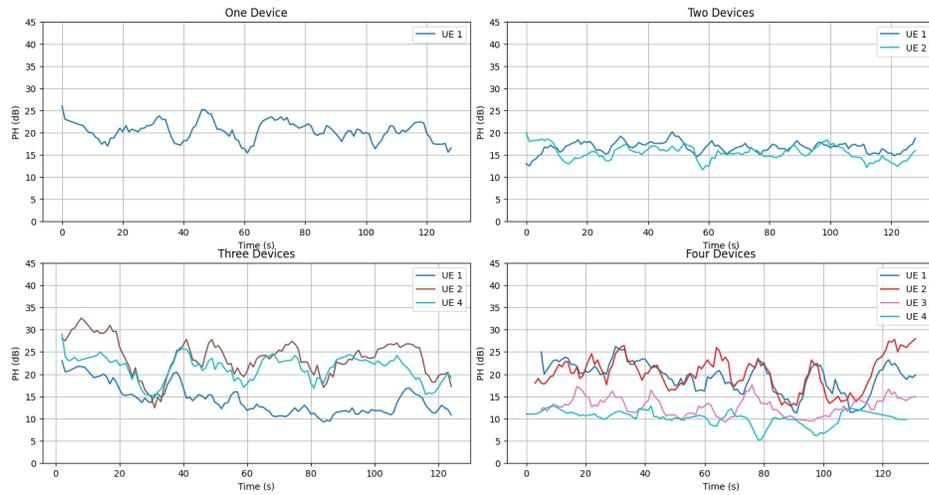


Figure 42: Moving Average Power Headroom over 4 devices

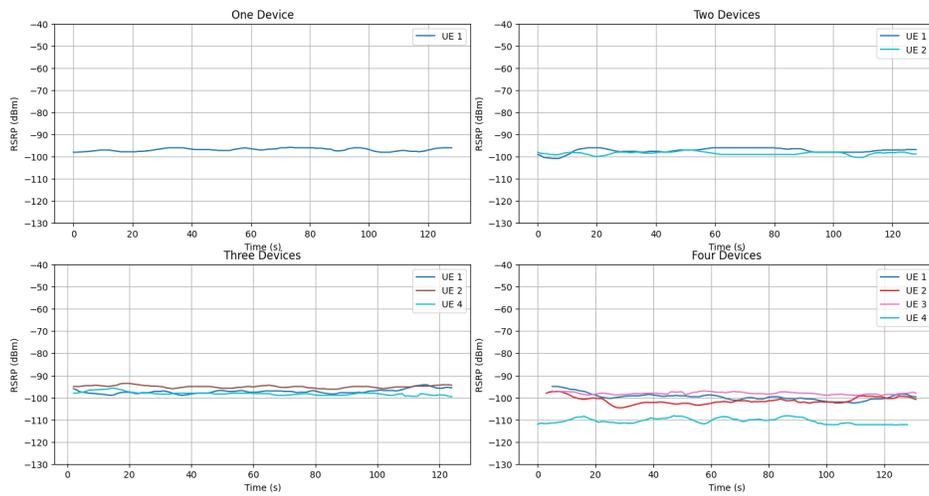


Figure 43: Moving Average RSRP over 4 devices

## Bibliography

- [1] M. Polese, L. Bonati, S. D’Oro, S. Basagni, and T. Melodia, “Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges,” IEEE Communications Surveys & Tutorials, 2020.
- [2] PraxiPaaS, “PraxiPaaS: A Decomposable Machine Learning System for Efficient Container Package Discovery.”
- [3] Ettus Research, “USRP X410.” Available: <https://www.ettus.com/X410/>.
- [4] OpenAirInterface, “MAC Layer Usage Guide.” Available: [github.com/OPENAIRINTERFACE/.../mac-usage.md](https://github.com/OPENAIRINTERFACE/.../mac-usage.md).
- [5] Ookla, “Speedtest by Ookla.” Available: <https://www.speedtest.net>.

# List of Figures

1	L. Bonati, M. Polese, S. D’Oro, S. Basagni, and T. Melodia, ”Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead,” Computer Networks, vol. 182, Dec 2020 . . . . .	7
2	Overview of the architecture in Red Hat Dashboard . . . . .	11
3	O-RAN Server Rack: Five Dell PowerEdge R750xs Units . . . . .	15
4	Two NI Ettus USRP X410 devices used in the O-RAN setup . . . . .	15
5	OAI nrMAC-stats.log file showing MAC-level scheduling parameters . . . . .	19
6	Open Cells SIM cards and SIM card reader/writer . . . . .	21
7	Log output from the Astreo Board, showing 5G connection status . . . . .	25
8	Ping test from the Astreo Board to the UPF POD within the O-RAN 5G Core . . . . .	26
9	IPERF client running TCP to measure network throughput . . . . .	26
10	IPERF Client performing UDP test in reverse mode (server to client transmission) . . . . .	27
11	IPERF client performing UDP test in forward mode (client to server transmission) . . . . .	27
12	Samsung A25 Service Mode Menu . . . . .	29
13	Average SNR over 13 positions for Samsung A25 . . . . .	32
14	Average Downlink BLER over 13 positions for Samsung A25 . . . . .	33
15	Average Power Headroom over 13 positions for Samsung A25 . . . . .	34
16	Average RSRP over distances for two different gNB output power levels . . . . .	35
17	BLER_DLSCHE over 13 positions for different power levels . . . . .	36
18	Downlink MCS over 13 positions for different power levels . . . . .	36
19	Downlink BLER over distance for two gNB output power levels . . . . .	37
20	Downlink BLER: Astreo vs. Samsung A25 . . . . .	38
21	Downlink MCS: Astreo vs. Samsung A25 . . . . .	39
22	Moving Average SNR over 4 devices . . . . .	40
23	Moving Average downlink BLER over 4 devices . . . . .	41
24	Moving Average of the downlink MCS over four devices . . . . .	41
25	Average BLER DLSCHE over 13 positions for Samsung A25 . . . . .	43
26	Average ULSCHE BLER over 13 positions for Samsung A25 . . . . .	44

27	Average Uplink BLER over 13 positions for Samsung A25 . . . . .	44
28	Average Downlink Errors over 13 positions for Samsung A25 . . . . .	45
29	Average Downlink MCS over 13 positions for Samsung A25 . . . . .	45
30	Average Uplink MCS over 13 positions for Samsung A25 . . . . .	46
31	Average Uplink BLER over distance for two gNB output power levels . . . . .	47
32	Average Power Headroom distance for two gNB output power levels . . . . .	47
33	Average SNR over distance for two gNB output power levels . . . . .	48
34	RSRP: Astreo vs. Samsung A25 . . . . .	49
35	Power Headroom: Astreo vs. Samsung A25 . . . . .	49
36	SNR: Astreo vs. Samsung A25 . . . . .	50
37	Transmitted Bytes: Astreo vs. Samsung A25 . . . . .	50
38	Moving Average DLSCCH BLER over 4 devices . . . . .	51
39	Moving Average ULSCCH BLER over 4 devices . . . . .	51
40	Moving Average Uplink BLER over 4 devices . . . . .	52
41	Moving Average Downlink MCS over 4 devices . . . . .	52
42	Moving Average Power Headroom over 4 devices . . . . .	53
43	Moving Average RSRP over 4 devices . . . . .	53