Measuring Power Consumption of a Virtualized 5G Mobile Network Using OpenAirInterface

Yoran Dani Staal

17-2-2025

Supervisors:

Prof. Dr.ir. Geert J. Heijenk

Syllas Rangel C. Magalhães, Msc

Committee member:

Prof. Dr. ir. Ana Lucia Varbanescu

Design and Analysis of Communication Systems Group

Faculty for Electrical Engineering, Mathematics and Computer Science

University of Twente

Abstract

The transition to fifth-generation (5G) mobile networks brings unprecedented advancements in communication technologies, enabling higher data rates, ultra-low latency, and massive device connectivity. However, these innovations come with an increase in complexity and energy demands, particularly in the Radio Access Network (RAN). This thesis addresses the design and implementation of a 5G testbed for energy efficiency research, using OpenAirInterface (OAI) and Software-Defined Radios (SDRs). The testbed enables systematic and reproducible experiments to measure energy consumption in various network configurations. It supports both software-based and commercial-off-the-shelf (COTS) user equipment to evaluate performance and energy efficiency. Experiments are performed to investigate the impact of bandwidth, traffic load, and modulation and coding schemes on energy consumption of the testbed components, including the core network, RAN, and user equipment. Through comprehensive experiments, this work demonstrates the relationship between network parameters and power usage, providing information on energy usage of 5G networks. The findings lay the groundwork for further research on 5G power consumption at the University of Twente, serving as a foundational step toward the advancement of sustainable mobile networking technologies.

ACKNOWLEDGMENTS

Firstly, I would like to express my gratitude to my two daily supervisors, Prof. Dr. ir. Geert J. Heijenk and Syllas Rangel C. Magalhães, for their support throughout the entire duration of this thesis. I greatly appreciate all their input during all our meetings, and also the freedom they have given with this project. I would also like to thank Prof. Dr. ir. Ana Lucia Varbanescu for evaluating my work.

In addition to the committee, I would like to thanks the supporting staff of both DACS and CAES, Bernd and Dorus. They have both helped me out with acquiring all the necessary equipment for the testbed, as well as with their knowledge on using the equipment.

I also would like to thank Max from the adjacent IOT lab, for all the help and nice related and unrelated conversations, as well as the amazing IOT lab coffee.

Lastly, I would like to thanks my parents, brother, sister and housemates and friends for their support, as well as for the fun moments we had in the last year outside of working hours.

CONTENTS

List of Abbreviations

5G	Fifth Generation (Mobile Network)
AMF	Access and Mobility Management Function
CP-OFDM	Cyclic Prefix-Orthogonal Frequency Division Multiplexing
COTS	Commercial Off-The-Shelf
DL	Downlink
eMBB	Enhanced Mobile Broadband
FDD	Frequency Division Duplexing
FR1	Frequency Range 1
FR2	Frequency Range 2
ICT	Information and Communication Technology
IoT	Internet of Things
LTE	Long-Term Evolution
MIMO	Multiple Input, Multiple Output
mMTC	Massive Machine-Type Communications
NFV	Network Functions Virtualization
NR	New Radio (5G)
OAI	OpenAirInterface
O-CU	Open Centralized Unit
O-DU	Open Distributed Unit
O-RAN	Open Radio Access Network
O-RU	Open Radio Unit
PHY	Physical Layer
QoS	Quality of Service
RAN	Radio Access Network
SBA	Service-Based Architecture
SCS	Subcarrier Spacing
SMF	Session Management Function
SDR	Software-Defined Radio
TDD	Time Division Duplexing
UL	Uplink
URLLC	Ultra-Reliable Low-Latency Communications
UPF	User Plane Function
UE	User Equipment

1 INTRODUCTION

This chapter provides the context for the thesis and outlines its main objectives and research questions. It begins with an overview of the motivations behind studying the energy consumption in 5G networks. The chapter then details the research questions that guide the study, and concludes with a summary of the thesis structure, briefly describing the focus of each subsequent chapter.

1.1 Context

Throughout the evolution of mobile networks, each generation has made significant improvements over its predecessors. Fifth generation (5G) mobile networks, officially named ”New Radio” (NR), brings significant improvements in performance, supporting higher data rates, lower latency, and larger numbers of connected users to keep up with both the growing data traffic and connected devices. In addition to this, 5G is also designed to be used for a wider range of specific use cases, such as very high speeds, ultra-low latency, or massive device connectivity.

While mobile networks continue to increase performance, it is essential to recognize the accompanying challenges, such as in the realm of energy efficiency. As data rates increase and more devices are connected to MNs, the energy consumption of the infrastructure increases. The information and communication technology (ICT) sector is estimated to be responsible for 1.4% of global total CO2 emissions [1], and the share of wireless devices in ICT will increase [2]. With the development of mobile networks, there has been an exponential reduction in energy consumption per gigabyte (GB) [3]. However, despite these advances, mobile data usage has continued to increase and is projected to increase significantly in the future. According to Ericsson’s data forecast, the estimated total mobile data traffic is expected to grow from 160.1 EB in 2023 to 562.7 EB in 2029 [4]. Developing energy-efficient technologies is essential to reduce the environmental impact of mobile networks, ensuring sustainable growth as data demand continues to rise. As mobile networks evolve, the integration of advanced technologies and sustainable practices is crucial to addressing the growing demand for data while minimizing environmental impact.

In recent years, there has been a shift in the architecture of telecommunication networks, with the concept of network functions virtualization (NFV). Virtualization in the context of mobile networks involves the abstraction of network functions and services from dedicated hardware appliances to software-based implementations running on standard computing hardware. Through NFV, network functions such as routing, switching, and traffic management can be instantiated and orchestrated dynamically, allowing greater flexibility, scalability, and resource efficiency. Base station functions of the radio access network (RAN) up until the physical layer can now be done software-based instead of on specific hardware. NFV results in the shift of functions from dedicated edge hardware to white-box servers located in server centers [5].

The evolution from vendor-specific hardware in RAN technology to the advent of 5G and OpenRAN architecture has revolutionized the landscape of MN research. Before these advances, RAN technology was predominantly characterized by proprietary hardware solutions, which limited the flexibility and accessibility for scientific research. However, with the introduction of OpenRAN, the paradigm has shifted towards a more open and modular approach. The standardization of this flexible architecture network functions and interfaces is done by the O-RAN Alliance. General standards are set by 3GPP, the global organization that develops technical standards for mobile networks.

One project that implements these standards is OpenAirInterface (OAI). OAI is an open-source software project that provides a full-stack implementation of 4G LTE and 5G mobile networks, enabling researchers and developers to build and test mobile network technologies. Commercial off-the-shelf (COTS) hardware can be used to run the network, which includes the RAN and core network (CN). A software defined radio (SDR) can be used as the radio unit (RU), which handles the lower PHY functions and analog front-end.

1.2 Objectives

To contribute to energy efficiency 5G research, the EDGE Research Centre of the University of Twente is realizing a 5G testbed on which real-world experiments can be performed to gain insights into the energy usage of a single 5G network. Setting up this testbed will be the basis for this thesis. This work contains the basis for the establishment of advanced energy research on 5G networks at the University of Twente.

The testbed includes a core network, RAN, user equipment (UE), radios, and measurement hardware. A management machine is used to orchestrate the experiments. Multiple experiments will be done on the testbed to find the limitations of the testbed and demonstrate its functionality. Furthermore, experiments are performed to gain insight on the impact of multiple parameters on the energy usage of the testbed.

1.2.1 Main research question

How can a 5G testbed be designed to enable systematic and reproducible tests for 5G energy research?

The main goal of this thesis is to realize a fully functioning 5G testbed that can measure energy usage during a variety of experiments. Similar testbeds in literature are used as reference for the design.

For the design, emphasis is put on the fact that it has to be a platform that researchers can easily use for their experiments. Automating setup routines and easy configuration results in more time for actual research and less time on trying to realize experiments.

1.2.2 Sub research questions

1. **How can a fully functioning and widely configurable 5G network be realized using OpenAirInterface?**
The goal of this subject is to investigate the possibility of OpenAirInterface, and see how it can be easily configured for experiments.

2. **How can energy consumption data be measured on a multicomponent 5G testbed?**
The components of the testbed will be identified and measurement methods are researched to collect power usage data for these components.

3. **What is the impact of bandwidth, traffic load and the modulation and coding scheme on the energy usage of a 5G network?**
To demonstrate the testbed, multiple tests will be done, varying multiple parameters of the testbed: the bandwidth, traffic load, and modulation and coding scheme (MCS).

1.3 Thesis structure

This research project contains a background chapter, two core chapters, followed by the results and conclusion chapters. Chapter 2 provides background theory on the relevant topics of this thesis. The next two chapters contain the core of this research: Chapter 3 is dedicated to the realization of an 5G testbed using OpenAirInterface. It introduces OAI, after which the functional part (both implemented and planned) will be presented. After this, the implementation of the OAI testbed is presented. Finally, a section is dedicated to functionality that could be added to the testbed and guidelines on how to implement such features. Chapter 4 is about power measurements on 5G networks. It starts with a section on related work, followed by sections on the implementation. The experiments performed on the testbed, together with the results are presented in Chapter 5. The conclusion presents a summary of the work done in this thesis, followed by a section on how this work can be continued, and the final thoughts on this project.

2 BACKGROUND

This chapter provides the theoretical background for the research presented in this thesis. In the first section, an introduction to mobile networks is given on a functional level. Following this, the overall architecture of 5G networks is introduced, with a focus on both core and access network elements. The Radio Access Network (RAN) architecture is examined in greater detail in a subsequent section. OpenRAN and O-RAN are explained in their respective sections after this. The section after this explains the basic operation of SDRs. Finally, a section on the power usage of mobile networks is presented.

2.1 Mobile networks

2.1.1 History of mobile network generations

The evolution of mobile networks has seen a progression through the first four distinct generations, each demonstrating significant advances in wireless communication technology. The first generation (1G) introduced analog cellular networks in the early 1980s, enabling basic voice calls with limited coverage and low data transfer rates. The transition to second-generation (2G) systems in the 1990s brought digital voice transmission and introduced technologies like GSM (Global System for Mobile Communications) and CDMA (Code Division Multiple Access), laying the foundation for SMS messaging and rudimentary data services. With the advent of third-generation (3G) networks in the early 2000s, mobile communication entered the era of high-speed data transmission, enabling Internet access. The introduction of fourth-generation (4G) LTE (Long-Term Evolution) networks further revolutionized mobile connectivity, delivering even faster data speeds, lower latency, and enhanced support for bandwidth-intensive applications. Each generation has built upon the innovations of its predecessor, driving forward the capabilities and reach of mobile networks worldwide.

2.1.2 5th Generation (5G)

5G introduces a paradigm shift in mobile communication by addressing the diverse and specific requirements of modern applications, from enhanced broadband access to ultra-reliable and low-latency connections. Unlike previous generations, which focused primarily on improving data rates and connectivity, 5G incorporates a more flexible architecture capable of supporting a wider array of use cases. 5G is designed to deliver improved scalability, reliability, and efficiency across a wide range of scenarios like IoT, industrial automation, and real-time systems. To provide optimized services for this wide range of applications, three types of generic 5G services have been defined [6]:

1. Enhanced Mobile Broadband (eMBB): This is the most general 5G service, offering stable mobile connections with higher data speeds than previous generations. It supports ultra-high peak data rates and low latency, with moderate rates for cell-edge users.

2. Massive Machine-Type Communications (mMTC): This service is supporting high-density networks with a massive number of clients, making it optimal for internet-of-things (IoT) type of applications with sporadically active nodes that send small data payloads.
3. Ultra-Reliable Low-Latency Communications (URLLC): This service is designed to support applications that require ultra-low latency and high reliability, making it suitable for applications where real-time responsiveness and mission-critical reliability are essential.

There are two frequency ranges defined for NR: FR1 (450 MHz to 6 GHz) and FR2 (24.25 GHz to 52.6 GHz). FR1, often referred to as sub-6 GHz, is widely deployed in practical networks due to its broad coverage, good signal penetration, and the ability to support large areas, making it suitable for urban and rural environments. FR2, or millimeter wave (mmWave), offers much higher bandwidth and low latency, but its shorter range and limited penetration through obstacles restrict its practical deployment to dense urban areas, stadiums, and specific high-capacity use cases where ultra-fast data rates are essential. Most early 5G deployments focus on FR1, with FR2 being rolled out in targeted areas for enhanced capacity.

2.1.3 5G network architecture

The 5G network architecture represents a change from previous generations, emphasizing modularity, scalability, and service agility. While traditional mobile core networks relied on point-to-point (P2P) interfaces, the 5G core adopts a Service-Based Architecture (SBA) that aligns with cloud-native principles. This transformation enables greater functional and service flexibility.

In Figure 2.1, a high-level block diagram of the 5G network architecture is shown. The shown interfaces are functional, e.g. the UE only has an actual connection via the RAN. The User Equipment (UE) consists of the user hardware, e.g. Mobile phones. The RAN is the part of the network where UEs connect to. It contains the radio hardware and base stations. More details on the RAN are given in the next section. The UPF is the interface between the UE's data and the external data network (in most cases the Internet).

The network management is done by the 5G core network (CN), which is represented in the figure as the blue plane. The CN handles tasks such as session management, user authentication, and policy enforcement. The Service-Based Architecture (SBA) is fundamental to the CN, designed to decouple services from the underlying network infrastructure [7]. It introduces service-based interfaces (SBIs), allowing core network functions (NFs) to interact through standardized APIs rather than rigid predefined links. This approach is in contrast to P2P architectures, which rely on numerous unique connections.

With SBA, operators can add, remove, or upgrade individual NFs without disrupting the entire network. For example, a new session management function (SMF) instance can be introduced seamlessly. Custom service paths can be created dynamically to meet specific requirements, such as low-latency services for industrial automation or high-bandwidth services for augmented reality.

Key components of the Cores SBA, as defined in the 3GPP TS 23.501 specification [8], include:

- **Access and Mobility Management Function (AMF):** Manages signaling, mobility, and access control. Communicates with the RAN and UE.
- **Session Management Function (SMF):** Handles session establishment, QoS enforcement, and IP address allocation.
- **User Plane Function (UPF):** Routes and forwards user data.
- **Policy Control Function (PCF):** Implements policies for traffic management and QoS.

- **Unified Data Management (UDM):** Stores subscriber data and profiles.
- **Network Repository Function (NRF):** Facilitates NF discovery and registration, serving as the backbone of the service-based communication model.

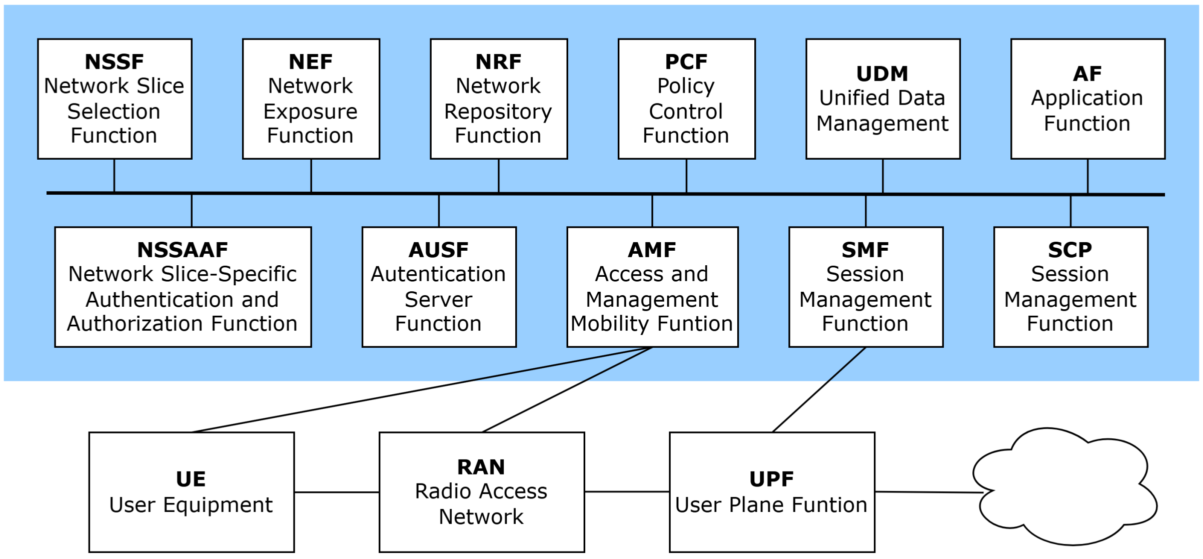
NSSF
Network Slice Selection Function
NEF
Network Exposure Function
NRF
Network Repository Function
PCF
Policy Control Function
UDM
Unified Data Management
AF
Application Function
NSSAAF
Network Slice-Specific Authentication and Authorization Function
AUSF
Autentication Server Function
AMF
Access and Management Mobility Funtion
SMF
Session Management Function
SCP
Session Management Function
UE
User Equipment
RAN
Radio Access Network
UPF
User Plane Funtion

Figure 2.1: 5G System architecture, adapted from [8]

2.2 RAN architecture

2.2.1 History of RAN architectures

Throughout the different generations of mobile network, the RAN architecture has changed significantly to keep up with the growing demand for performance. In this section, the architectures throughout the years will be discussed, followed by an analysis of the state-of-the-art mobile network architecture.

3G RAN consists of the remote radio head (RRH) and the baseband unit (BBU), which together form the traditional RAN or distributed RAN (D-RAN) architecture. The location of the RRH is close to the physical antenna in the cell site tower and contains the radio functions. Each RRH has a dedicated BBU, which is located in a central location and handles all baseband processing. The RRH and BBU communicate over a connection called the fronthaul. The interface used to implement the fronthaul is called the Common Public Radio Interface (CPRI), which is not an open protocol. Therefore, RRH and BBU must be supplied by the same manufacturer. The CPRI interface uses point-to-point fiber connections, creating complexity in large systems. A constant bitrate is required over CPRI no matter the load, and statistical multiplexing is not possible.

Centralized RAN (C-RAN) was introduced for 4G, where the BBUs of multiple cell sites are in a strategically good location. The BBU's in a BBU-pool are connected using a high-speed X2 interface. The combination of BBU's and the connection of them result in enhanced resource utilization, improved network scalability, improved network coordination, and load balancing [9]. This results in improved spectral efficiency, better conditions for interference coordination techniques, and faster handovers between cells in the same BBU-pool [10]. Similar to D-RAN, C-RAN still uses vendor-specific hardware and closed protocols, including CPRI.

C-RAN has multiple energy savings gains, which are identified in [11]. Firstly, stacking gains are achieved through optimized utilization of RAN hardware, resulting in reduced energy consumption. Secondly, the pooling gain is realized by using more powerful and energy-efficient

BBUs. Third, BBUs can be dynamically allocated to RRHs based on traffic demand, thus adjusting energy consumption in accordance with network load variations. Lastly, system cooling can be made more efficient by using centralized cooling systems.

The introduction of the virtual RAN (V-RAN) builds upon the centralized principles of C-RAN, advancing the idea by implementing RAN functions as software on general-purpose server hardware rather than dedicated infrastructure [12]. This evolution enables network functions to be deployed in cloud environments, allowing for greater flexibility and scalability. By virtualizing these functions, V-RAN supports dynamic allocation across different machines or data centers, facilitating efficient resource use as network demands change. Additionally, V-RAN adopts the Service-Based Architecture (SBA), which decouples functions into modular reusable services that can be easily updated or reconfigured.

With 5G, many more architectural options are available. Continuing with the concept of V-RAN, it is possible to allocate network functions in many ways. One possibility is splitting the BBU into two components, the centralized unit (CU) and distributed unit (DU), which together with the RRH form the disaggregated openRAN architecture. In the next section, OpenRAN will be explained in more detail.

The functional split (FS) refers to the division of network functions of the RAN. The authors of [10] present a survey on FS for 5G networks. Moving functions from the protocol stack towards the edge of the network allows for lower latency and reduced backhaul traffic. However, it may increase complexity and cost due to the need for additional hardware and management overhead. Moving network functions towards the CN in data centers enables easier management, more efficient resource utilization, and more flexible service deployment, at the cost of higher latency and reliance on backhaul connections, with increased vulnerability for network failures. Only a few split options are still seen as practical, and thus remain widely discussed in literature. In [13] an overview of the 3GPP functional split approach is given.

2.2.2 OpenRAN

OpenRAN is an initiative that aims to disaggregate traditional network elements and promote interoperability, flexibility, and cost-effectiveness. Unlike traditional RAN architectures, which rely on proprietary hardware and tightly integrated components from single vendors, OpenRAN is based on open standards and interfaces, allowing operators to mix and match hardware and software components from different vendors.

Virtualization is the basis for OpenRAN, which involves separating the hardware and software layers of the RAN architecture. This enables operators to deploy software-based network functions on commercial off-the-shelf (COTS) hardware, such as standard servers and white-box radios, rather than relying on purpose-built equipment from specific vendors. By making network components software-implemented, OpenRAN reduces vendor lock-in, promotes competition, and lowers barriers for new vendors and the scientific community to contribute to the future of mobile networks.

2.2.3 O-RAN alliance

The O-RAN Alliance is a leading organization that implements and extends the OpenRAN concept. Although OpenRAN refers to the general idea of open and disaggregated RAN, the O-RAN Alliance develops specific technical standards and specifications to bring this vision into reality. O-RAN provides detailed specifications and standards for open interfaces, RAN controllers, and virtualization. In [14], a detailed tutorial on O-RAN is given. The goal of the O-RAN standard is to have a standardized and scaleable architecture to achieve flexibility, performance, and efficiency.

2.2.3.1 O-RAN architecture

The main building blocks of the standardized O-RAN architecture can be seen in Figure 2.2. Since the O-RAN standard is designed for virtualized and disaggregated network functions, its building blocks can be flexibly implemented. This means that components like the O-CU, O-DU, and O-RU can either run on separate physical machines or be co-located on the same hardware, depending on network requirements and available resources.

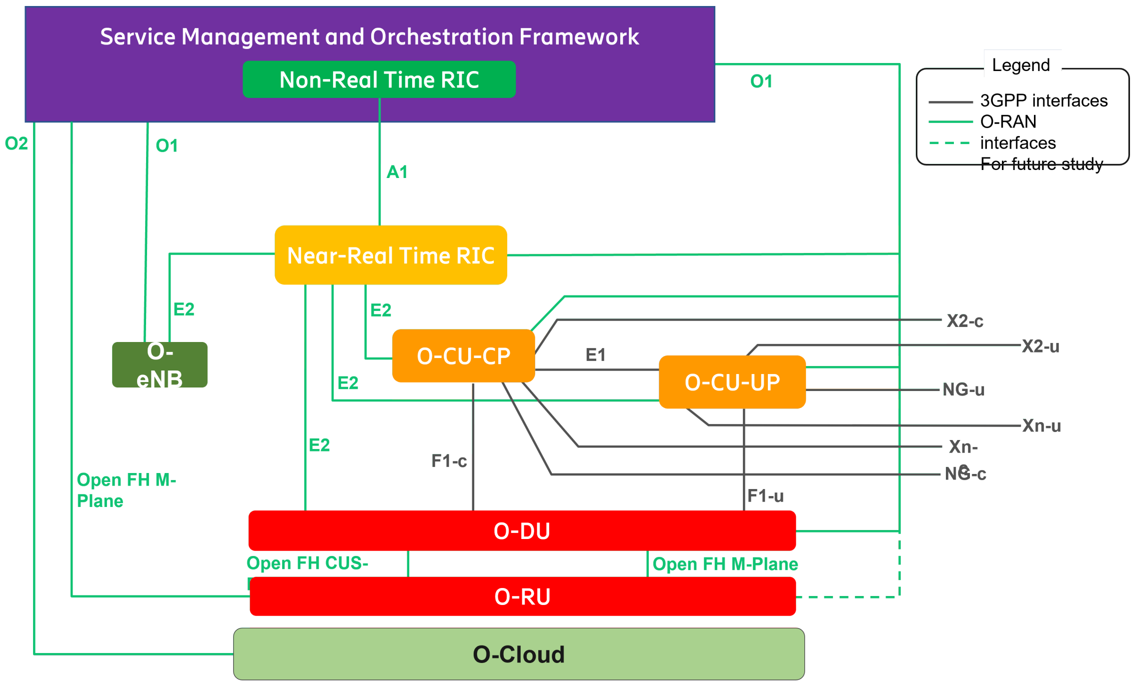

Figure 2.2: Block diagram of the O-RAN architecture and interfaces, adapted from [15]

- **The Open Centralized Unit (O-CU)** is split into two functional parts:
  - **O-CU Control Plane (O-CU-CP)** handles control signaling, including mobility management, session setup, and overall coordination between the RAN and the 5G Core via the NG-C interface. Ensures that control messages are routed correctly and that overall network performance is managed efficiently.
  - The **O-CU User Plane (O-CU-UP)** is responsible for handling user data traffic. It connects to the User Plane Function (UPF) in the 5G Core through the NG-U interface. The separation of user and control planes in the CU allows for independent scaling of control and data processing.
- **Open Distributed Unit (O-DU):** Handles the baseband processing tasks for the lower layers of the radio protocol stack. It processes both control and user plane traffic before it is transmitted over the radio interface. The O-DU is responsible for the direct scheduling of radio resources (according to the policies of the RICs).
- **Open Radio Unit (O-RU)**: Responsible for radio frequency (RF) processing, including modulation, demodulation, and transmission of data over the air. It is connected to the O-DU via the Open Fronthaul Interface (F1) and forms the final link in the RAN system before reaching the end-user devices.
- **Non-Real-Time RAN Intelligent Controller (Non-RT RIC):** Operates within the Service Management and Orchestration (SMO) framework and is responsible for handling non-real-time tasks, such as policy management, network analytics, and machine learning

model training. This is done with the help of rApps: applications that can be hosted in a flexible way on the Non-RT RIC.

- **Near-Real-Time RAN Intelligent Controller (Near-RT RIC):** Enables real-time optimization of RAN functions. It manages the resource allocation, interference management, and other time-sensitive tasks that require decisions in milliseconds. Like the Non-RT RIC, applications are used to execute these tasks, called xApps in the Near-RT domain.
- **O-Cloud:** Serves as the underlying infrastructure that makes flexible distribution possible. It includes both hardware (e.g. servers) and software (e.g. virtual machines or containers) that host the O-RAN functions.

2.3 5G NR physical layer

The resource allocation in 5G is much more flexible than in previous generations. However, this flexibility does add complexity. This section sheds light on the configuration options for the PHY layer of 5G NR. As explained in Subsection 2.1.2, two frequency ranges are defined for NR, FR1 and FR2. The work in this thesis focuses only on the FR1 frequency range.

2.3.1 Modulation

5G uses Cyclic Prefix-Orthogonal Frequency Division Multiplexing (CP-OFDM). OFDM is a multicarrier modulation technique that divides the available spectrum into multiple orthogonal subcarriers, each transmitting data independently. A cyclic prefix (CP) is added before every OFDM symbol to mitigate intersymbol interference caused by multi path propagation. By copying a portion of the end of the OFDM symbol and appending it to the start, the cyclic prefix ensures that delayed versions of the signal do not overlap with subsequent symbols, preserving orthogonality and enabling robust communication in noisy environments.

The modulation order Q_m determines the number of bits transmitted per OFDM symbol, directly influencing the spectral efficiency of the system. The number of distinct modulation symbols that must be represented on a subcarrier is 2^{Q_m}. For example, lower-order modulations, such as QPSK ($Q_m = 2$) encode 2 bits per symbol and are more robust in noisy or low-SNR conditions. Higher-order modulations such as 16-QAM ($Q_m = 4$), 64-QAM ($Q_m = 6$), and 256-QAM ($Q_m = 8$) pack more bits into each symbol, significantly increasing data rates but requiring higher signal quality for successful decoding.

In addition to multiple modulation orders, 5G uses channel coding to improve reliability. The coding rate (R_{max}) represents the fraction of useful bits (information) relative to the total transmitted bits, which includes redundancy added for error correction. A coding rate close to 1 means minimal redundancy and high throughput, suitable for good channel conditions. Conversely, a lower coding rate increases redundancy, improving error correction capability and reliability in poor channel conditions.

The Modulation and Coding Scheme (MCS) combines the modulation order and coding rate into a single parameter that defines the data transmission configuration. The MCS is dynamically adjusted based on real-time channel quality measurements, such as signal-to-noise ratio (SNR) and interference levels. For example, in high-quality channels, the system may select an MCS with high Q_m and R_{max}, enabling faster data transmission. In contrast, under poor channel conditions, the system uses an MCS with lower Q_m and R_{max} to maintain reliable communication. This adaptive mechanism ensures that 5G networks achieve a balance between spectral efficiency, throughput, and reliability, even in dynamically changing environments. The MCS index has a range from 0 to 28 for the UL and DL. The MCS table can be found in 3GPP Specification 23.214 [16].

The smallest physical resource in NR is a resource element (RE), which consists of a sub-carrier during one OFDM symbol. A physical resource block (PRB) is defined as 12 consecutive subcarriers in the frequency domain. While in LTE an RB is defined in both the frequency and time domains, it is not defined in the time domain in NR, which offers a lot more scheduling options in the time domain. There are multiple sub-carrier spacing (SCS) defined for NR. The SCS is defined by numerology. The different numerology settings are shown in 3GPP 38.211 Table 4.2-1 [17], and are shown in table 2.1.

μ	$\Delta f = 2^{\mu} \cdot 15$ [kHz]	Cyclic prefix
0	15	Normal
1	30	Normal
2	60	Normal, Extended
3	120	Normal
4	240	Normal
5	480	Normal
6	960	Normal

Table 2.1: 5G Numerology Table

In practice, an additional guard frequency band is needed to prevent interference between adjacent frequency channels [18]. The frequency range specification of 5G is described in 3GPP 38.101 [19]. The bandwidth used for a 5G frequency channel can be calculated using Equation 2.2.

$$\text{Bandwidth (MHz)} = N_{\text{PRB}} \cdot 2^{\mu} \cdot 15 \cdot 12 \cdot 10^{-3} + BW_{guard} \tag{2.1}$$

where

N_{PRB} is the number physical resource blocks

$2^{\mu} \cdot 15$ is the subcarrier spacing in kHz, defined by numerology μ.

12 is the number of subcarriers per resource block.

10^{-3} is the conversion factor from kHz to MHz.

BW_{guard} is the additional guard band in MHz.

2.3.2 5G frame structure and duplexing

The 5G frame structure is more flexible than LTE, catering to modern applications such as autonomous vehicles, IoT, and high-speed video streaming. Adapting to the specific needs of each application, it improves network efficiency. A key feature of this flexibility is the time-frequency resource allocation, which divides resources into blocks and grid elements. The 5G frame operates on a slot and symbol-based design.

The length of one 5G radio frame is 10 ms, and always contains 10 subframes of 1 ms. The amount of slots in a subframe depends on the subcarrier spacing: for example, if 30 KHz spacing is used, the same amount of information can be sent in half the time compared to a 15 KHz subcarrier spacing. Every subframe has 14 symbols.

For time division multiplexing (TDD) mode, the allocation of the slots and symbols depends on the periodicity (P): the periodicity determines the length in milliseconds of the allocation pattern. An overview of the existing P can be seen in the first column of table 2.2, which is taken from [20]. The table shows the number of slots that can be allocated in each period. Three types of slots are defined: uplink (UL), downlink (DL), and flex. With a flex slot, it is possible to allocate the individual 14 symbols within the slot (for UL, DL or as guard period).

P (ms)	Number of Slots in a P				
	0	1	2	3	4
0.5		1	2	4	8
0.625				5	10
1.25			5	10	20
2.5		5	10	20	40
5.0	5	10	20	40	80
10.0	10	20	40	80	160

Table 2.2: Number of slots in a period for each periodicity P and numerology μ

2.3.3 5G throughput

Getting an estimation for the maximum throughput for a set of PHY parameters in a NR connection is not a straightforward process. The increased flexibility of NR introduces greater complexity compared to LTE, where the PRB configurations are predefined in a large table.

In 3GPP 38.306 4.1.23 [21], a formula is given for the estimation of the data rate, shown here in Equation 2.2. An online calculator based on this equation can be found in [22]. The calculator uses information from the 3GPP tables, making it easy to use.

$$\text{Data rate (in Mbps)} = 10^{-6} \cdot \sum_{j=1}^{J} \left(v_{\text{Layers}}^{(j)} \cdot Q_m^{(j)} \cdot f^{(j)} \cdot R_{\text{max}} \cdot \frac{N_{\text{PRB}}^{\text{BW}(j),\mu} \cdot 12}{T_s^{\mu}} \cdot \left(1 - OH^{(j)}\right) \right) \tag{2.2}$$

where

J is the number of aggregated component carriers

$v_{\text{Layers}}^{(j)}$ is the number of MIMO layers

$Q_m^{(j)}$ is the modulation order

$f^{(j)}$ is the scaling factor, and can take the values 1, 0.8, 0.75, and 0.4.

R_{max} is the target code rate

$N_{\text{PRB}}^{\text{BW}(j),\mu}$ is the max number of PRBs

T_s^{μ} is the average OFDM symbol duration in a subframe for numerology μ, i.e. $T_s^{\mu} = \frac{10^{-3}}{14 \cdot 2^{\mu}}$, with the denominator being the max number of scheduled OFDM symbols per subframe

$OH^{(j)}$ Is the overhead for control channels. FR1 DL: 0.14, FR1 UL: 0.08

2.3.4 Example

To clarify Equation 2.1 and Equation 2.2, an example will be used. The parameter values of this example are within the capabilities of the testbed that is used in this thesis. For this example, we assume a DL configuration. 52 PRBs are used with numerology 1 (subcarrier spacing 30 KHz), and 1 MHz bandwidth is reserved for guard band. Using Equation 2.1, the bandwidth can be calculated:

$$\text{Bandwidth (MHz)} = 52 \cdot 2^1 \cdot 15 \cdot 12 \cdot 10^{-3} + 1 = 19.72 \text{ MHz} \tag{2.3}$$

A single frequency channel J is used, no MIMO (single layer). The scaling factor is default (1). MCS value 20 is used. From the MCS table in [16] it can be found that $R_{\text{max}} = 0.554$ and $Q_m = 6$ for this MCS value. The overhead for a FR1 DL channel is 0.14. Filling in the parameters in Equation 2.2 results in the following maximum theoretical throughput:

$$\text{Data rate (in Mbps)} = 10^{-6} \cdot 1 \cdot 6 \cdot 1 \cdot 0.554 \cdot \frac{52 \cdot 12}{\frac{10^{-3}}{14 \cdot 2^1}} \cdot (1 - 0.14) = 49.95 \text{ Mbps} \tag{2.4}$$

2.4 Software-defined radio

Radios are systems that enable wireless communication by transmitting and receiving information using electromagnetic waves within the radio frequency (RF) spectrum, typically ranging from 3 kHz to 300 GHz. A typical radio system consists of a transmitter which encodes and modulates the signal for transmission, and a receiver which captures the transmitted signal and reconstructs the original information.

Traditionally, radio communication systems are realized on dedicated electronic circuits to implement all their components. Software-defined radios (SDRs) represent a significant advancement in wireless communication technology, offering more flexibility and easy reconfigurability compared to traditional hardware-based radios. SDRs are radio communication systems in which a significant portion of the signal processing is implemented in software, rather than in dedicated hardware components. This allows for rapid development of radio systems, enabling the use of standardized platforms and protocols. This flexibility allows radio systems to be easily upgraded, modified, or repurposed without the need for costly hardware redesigns.

The general architecture of an SDR can be seen in Figure 2.3. The RF front-end connects to the antenna and handles RF-level tasks such as mixing, amplification, and filtering. The analog front-end interfaces with the digital part of the SDR via an ADC (RX) and a DAC (TX). The analog parts of an SDR are flexible, and parameters such as the carrier frequency and input gain, and output gain can be configured via the software.

The digital part of an SDR contains configurable digital hardware such as an FPGA. Generally, most digital baseband processing will be done on the host computer, also shown in Figure 2.3. The digital processing on the SDR is mostly used for collecting and forwarding IQ samples. IQ samples represent the in-phase (I) and quadrature (Q) components of a signal, which together describe both its amplitude and phase. These components are digitized and processed as complex-valued data, allowing flexible manipulation of signals in the digital domain. It is also possible to perform tasks such as filtering and compression, and other hardware optimizable tasks on the FPGA [23]. There are different interfaces for connecting the SDR to a host machine, with the most common being USB, Ethernet, and PCIe.

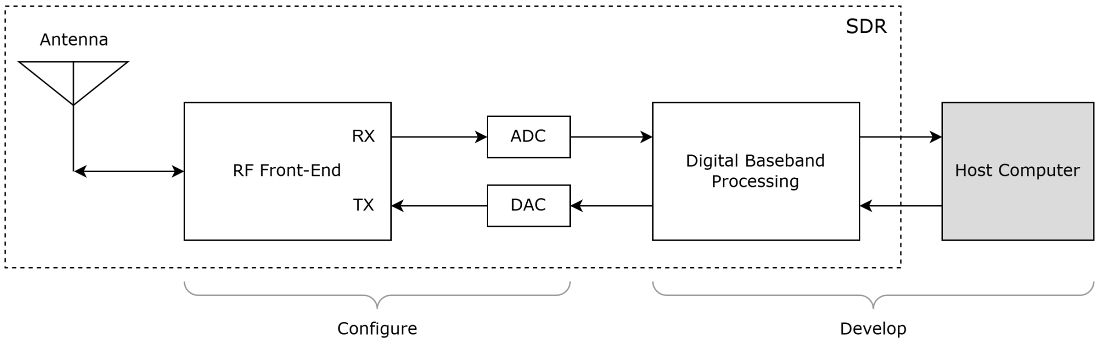

Figure 2.3: SDR tranceiver architecture [24]

A wide range of SDR's is available on the market, with big price differences. The most important parameters of SDR's include frequency range, bandwidth, MIMO capability (Multiple Input, Multiple Output), sampling rates, and on-board hardware acceleration.

2.5 5G energy usage

As has been noted in the introduction, the CO2 emission of mobile networks is significant. Allocating the energy usage in a large infrastructure such as global mobile network is a huge task, but attempts have been made to do this. The energy used for the realization of mobile networks can be divided into the following categories:

1. **Manufacturing and production:** The energy consumed in the creation of network hardware includes raw material extraction, processing, and the manufacturing of the hardware.
2. **Installation, deployment and maintenance:** Energy is required not only to physically install and deploy network infrastructure but also for regular maintenance and upgrades, which may include on-site visits.
3. **Operation:** The day-to-day running of the mobile network, including power for base stations and data centers.
4. **User equipment:** Mobile devices such as smartphones and IoT devices draw power from both the network and their own internal batteries, contributing to overall energy consumption.
5. **End-of-life disposal:** The disposal or recycling of outdated network equipment also incurs energy costs.

The work in this thesis focuses on the operational energy usage, which will be analyzed more in depth here. With the shift towards virtualized BSs, a new area of research on power usage for mobile networks arises. With legacy BSs, a large part of the network power usage was transmitting power. Because of network densification and smaller cell sizes in 5G, the BSs have become a more significant contributor to the overall power usage [25].

The research on mobile networking energy consumption can be divided in two approaches: The first approach covers macroscopic energy usage, for example, by analyzing statistics and data provided by vendors and industry reports. This macro-level perspective provides a broad understanding of energy consumption trends across various network deployments and operational scenarios. The second approach involves a more microscopic method, focusing on individual testbeds or experimental setups where detailed hardware measurements are taken. This microscopic approach is used to perform measurements on individual components of the network.

In [26], macroscopic energy usage of 5G is covered, and also presents a review on existing research on 5G energy usage. Their review aims to identify and address potential blind spots in current knowledge regarding the energy implications of 5G. According to the authors, most research on 5G energy usage has a narrow focus on single isolated components in the network, for example, a single component of the RAN. Three potentially significant gaps are identified. The first one is the lack of published research on the whole network assessments of the operational energy use of 5G networks. The authors also conclude that the embodied energy associated with network equipment and user devices is neither accounted for nor targeted in much of this literature. Lastly, the indirect energy use effects of 5G (enablement effects and rebound effects) have not been sufficiently evaluated.

Energy efficiency (EE) technologies are the subject in [9]. The authors define three EE technology categories for EE O-RAN networks. The first is the dynamic allocation of resources and network functions between CUs and DUs. The power consumption of the DU and CU is proportional to their active processing time, so by minimizing the processing time of the DUs and CUs while still satisfying QoS requirements has a positive effect on the EE of the network. The second approach considers the dynamic location of DUs and CUs on physical network nodes, as well as user association with the DUs. This approach focuses on completely shutting down the dormant nodes or physical machines due to more efficient use of computational resources via more centralized processing. The last category does not directly consider EE as the performance metric, but other metrics whose optimizations directly EE of O-RAN.

More information on microscopic 5G energy research is presented in the beginning of Chapter 4, where existing experimental mobile networking setups will be analyzed.

3 OPENAIRINTERFACE TESTBED

This chapter presents the OpenAirInterface project. The purpose of this chapter is to introduce the usage of OAI, and therefore, this chapter has a more practical orientation with a personal perspective. First, an overview of the OAI project is given. The second section presents a more detailed overview of the features of the project. The third section of this chapter looks at the testbed that has been realized for this thesis. In the final section, the performance of the testbed is evaluated.

3.1 Introduction to OpenAirInterface

The OpenAirInterface (OAI) is an open-source, standards-compliant implementation of a 3GPP 5G NR stack that runs on general x86 hardware, and makes use of SDRs as radio units. The development was initiated by Eurecom, which is a research institute in France. The OAI project is currently managed by the OpenAirInterface Software Alliance (OSA).

OAI is one of the largest open source projects for 5G development. The project enables researchers to build, test, and modify complete 5G networks, having access to all parts of the network. This makes it possible to develop and test new protocols, optimize performance, and simulate real-world scenarios.

OpenAirInterface 4G LTE is the predecessor of the OAI 5G NR project. It might not always be immediately clear in published papers, for example, from the title, which version is used. The 5G NR project is under very active development, and has implemented a lot of 3GPP NR (38 series specifications), however, many features are still under development. OSA has published road maps for their planning to add new features. The 5G version is currently in the $develop$ branch of the project's repository.

Open, standardized interfaces are the fundamental characteristics of OAI. Therefore, external projects can be used in combination with OAI components to deploy different types of network setups.

OAI works with COTS hardware, as well as their own ue-softmodem software. This can work with an SDR as radio, but is also possible to visualize the wireless channel using the OAI phy-simulator, where IQ samples are sent over IP.

3.1.1 OpenAirInterface components

OSA maintains multiple 5G software projects that together can be used to realize a complete 5G network. In the following subsections, the RAN, core, and UE projects will be introduced.

3.1.1.1 RAN software

The RAN software is maintained in the openairinterface5g repository [27]. The software is referred to as *softmodem* within the OAI ecosystem. It contains the code to compile and run a vRAN softmodem. The RAN uses the Ettus USRP uhd library to interface with SDRs.

The configuration of the RAN is done using a configuration file that has configuration parameters based on 3GPP standard definitions. The repository provides example configurations covering a wide range of supported features. The parameters in this file can be overwritten from the command line when starting the RAN application.

The repository also contains a selection of additional useful tools that will be explained in Section 3.2.

3.1.1.2 Core network

The core network is a separate project [28]. The OAI core does not only work with OAI gNBs: it has also been tested with commercial gNBs and other open-source projects. The OAI core network project currently supports basic functionality: connection, registration (UE registration, de-registration, and service requests) and session management. Additional features have been implemented and can be found in the component repositories [28].

The core network components (presented in Subsection 2.1.3) have their own repository in the OAI Gitlab [28]. Docker Compose can be used to create containers for every component, as well as virtual network interfaces. All communication between these components, as well as between the core and the RAN, is done over IP.

3.1.1.3 User equipment

For UE hardware, there are two options for an OAI testbed using the ue-softmodem that is included in the OAI project, or use COTS hardware. In this subsection, both options will be presented, and the advantages and disadvantages are explained:

- **ue-softmodem** The first option is to use the ue-softmodem software, which is located within the RAN repository [27] and is compiled together with the RAN softmodem. The ue-softmodem can connect to the RAN softmodem in two ways. The first is to use physical SDRs for both the RAN and UE. The SDRs can then connect wirelessly using antennas or using a coaxial cable with attenuator. A second virtual method that can be used is to simulate the channel and send IP packets with the I/Q samples. With this method, it is also possible to use only one machine for both the RAN and UE softmodems, and use the local loop-back network interface to transfer the packets.

  The biggest advantage is that the softmodem is fully implemented in software. This is useful for UE-side debugging and development. ue-softmodem can also be used in PHY mode, where no core network is needed to provide a network interface over the 5G connection. This mode can be used to perform PHY level testing without the need for a core network.

- **COTS Hardware** OpenAirInterface has been tested with a selection of COTS 5G devices. These devices include both 5G phones and M.2 5G modems that can be installed on laptops. Evaluation boards are also available for these M.2 modems, so that they can be tested without being installed internally. This also allows for more flexible antenna configurations as well as wired connections. COTS devices will require a programmable SIM card to work. Open Cells Project [29] works in partnership with the OAI Alliance, they sell programmable sim cards and radio hardware, and they have blogs with information on how to run OAI with COTS hardware (including a list of tested devices).

  A big advantage of using COTS hardware is the performance. COTS UE is designed optimally to modulate and demodulate signals and can achieve much higher data rates than the software-based ue-softmodem. Therefore, when RAN capabilities need to be tested at high rates, COTS hardware will be better. Using COTS hardware also gives a more realistic deployment scenario.

3.2 OpenAirInterface features

A detailed set of features can be found in [30]. Some of the most important parameters taken from this list are:

- Static TDD or FDD
- Subcarrier spacings 15 and 30 kHz (FR1) and 120 kHz (FR2)
- Bandwidths: 10, 20, 40, 60, 80, and 100 MHz
- Slot format: 14 OFDM symbols in UL or DL
- Procedures for 2-layer DL and UL MIMO
- Support of up to 16 UEs (can be increased to 32)
- Support of 7.2 CU/DU split

3.2.1 Operating modes

The OAI 5G NR software stack supports the 5G standalone (SA) mode, which operates independently with a 5G core, and the non-standalone (NSA) mode, which relies on existing 4G infrastructure. In addition, there are two more operating modes: phy-test and noS1. In the phy-test mode, the gNB and UE have a hard-coded connection with static schedule. This mode can be used for PHY level testing without Core. The noS1 mode is an extension of the phy-test mode. When using noS1 mode, a virtual network interface is created that can be used to inject user-plane traffic over the 5G connection. These modes can only be used with the ue-softmodem: The static connection parameters are defined by the RAN and stored in two custom files, which have to be read by the ue-softmodem to get a working connection.

3.2.2 OpenAirInterface tools

OAI has some software tools that are useful for things like adding functionality, debugging, and monitoring:

- **T Tracer:** The T tracer is a tool that is used to debug and monitor the softmodem [31]. It can be configured to display and collect real-time data while the softmodem is running. The tracer can be configured to listen to a wide range of values in every layer of the softmodem. Logging high-rate signals, such as raw IQ samples, will result in performance loss. The tracer can be used to display information at runtime in a dedicated graphical interface, or it can be used to store data in a raw file format. This file can then be replayed, or the data can be extracted to a readable format.
- **Wireshark:** OAI has integrations for Wireshark, which can be used in real time or with tracer recordings [32]. The first use case for Wireshark is to monitor the information between the testbed components, which is already done over IP. An important use case of this is the monitoring of the Next-Generation Application Protocol (NGAP). NGAP is an application layer protocol that is used for communication between the RAN and the AMF in the core. It carries information on session management and other signaling-related functions and is therefore useful for monitoring and debugging the initial connection process of the UE with the network.

  A second use case for Wireshark is to analyze MAC Protocol Data Units (PDUs), which are the formatted units of data exchanged between protocol layers. Specifically, PDUs in the MAC layer encapsulate user and control data for transmission over the PHY layer. In this scenario, the tracer enables these messages to be made accessible via the local loopback network interface for detailed inspection in Wireshark.

- **Telnet server:** The built-in Telnet can be used to monitor the softmodem remotely at runtime [33]. Unfortunately, not all features that are implemented for the LTE version of OAI are available for the NR application. For example, the Telnet server has the option to show the resource usage per process for the softmodem [34]: This functionality would be useful for power analysis of the realized testbed.
- **FlexRIC:** FlexRIC [35] is a near-RT-RIC framework that can be used to deploy xApps in an OAI testbed. It is a standalone project developed by the same organization as OAI.

3.2.3 Documentation and help

All documentation and tutorials can be found in OAI repository. Trying to find information here can be quite challenging. The Wiki page in the repository has a lot of outdated information (shown as a warning on the top of the pages). The most up-to-date information can instead be found within the doc folder of the repository. Guides on installation and building the source code can be found in this folder. In addition, the documentation mainly shares information through tutorials. These will get the examples within the repository up and running, but detailed information on how to make changes, for example, does not exist.

In [36], it is emphasized that it can be very challenging to setup an OAI instance. The authors goal is to connect the theoretical background of 5G with the OAI configuration process. Having a good background knowledge on O-RAN and 5G in general will make the process of learning how to use OAI much easier. The work in [37] helps connect the practical usage of OAI with the underlying theory.

There is an active and growing community of developers and users that use OAI, and therefore getting personal help is becoming easier. OAI does not have a forum. Instead, it uses multiple mailing lists that are used to ask questions. The use of a mailing list has some disadvantages. Firstly, it is hard to search back in time to see if someone else already asked the question that you want to ask. Secondly, the threshold for asking questions can be a bit higher.

3.3 Testbed design and realization

This section presents the design and realization of the OAI part of the Sustainability Lab 5G testbed. Details on the exact steps on the installation process and Linux configuration are included in the readme of the testbed's repository. Section 4.4 will present a more in-depth explanation on the testbed hardware, software, and used tools. The testbed supports both COTS UE and softmodem testing.

3.3.1 Testbed hardware

A block diagram of the system can be seen in Figure 3.1. In Table 3.1, a list of the hardware used is given. The Core and RAN run on two desktop computers and the UE runs on a laptop. All three machines have an Intel i7-13700 processors, which meets the minimal requirements set by OAI with 8 performance cores. Linux 22.04 LTS is installed on each machine, and the RAN and UE run a low-latency kernel (to optimize for real-time processes like SDR communication). For the testing in this chapter, the CPU settings were set according to the OAI documentation: The CPUs are in constant C0 state and use the $performance$ governor, which sets the CPU frequency statically to the highest base clock. A Windows laptop is used to manage the testbed and perform experiments.

All machines in the testbed are interconnected through a dedicated control network, shown in Figure 3.1 in red. This network is used to control the testbed machines from the management laptop. The control network also has Internet access. The RAN and Core machines have a second network port, which is used for a separate connection for communication between the two machines for the 5G operational network (RAN to Core components communication).

Both the RAN and the softmodem UE use an Ettus Research USRP B210 SDR as the radio unit. In addition to the SDR, the UE laptop can also be used with a COTS 5G modem. The Quectel RM500U M.2 modem, in combination with a Quectel 5G-M2 EVB kit, is used for this. The testbed uses RF cables as opposed to antenna's, to create a fully controlled environment for consistent testing. Currently, only one of the two UE options can be connected during an experiment.

The SDRs and COTS hardware are connected to the computing hardware via USB 3.0. The radios can be powered via USB; however, it is possible to power them via a DC input. The DC connection is used for the testbed, since a lab power supply with power meters can then be used to get measurements of the power usage of the radios. More about this and the rest of the power measurement side of the testbed is explained in Chapter 4.

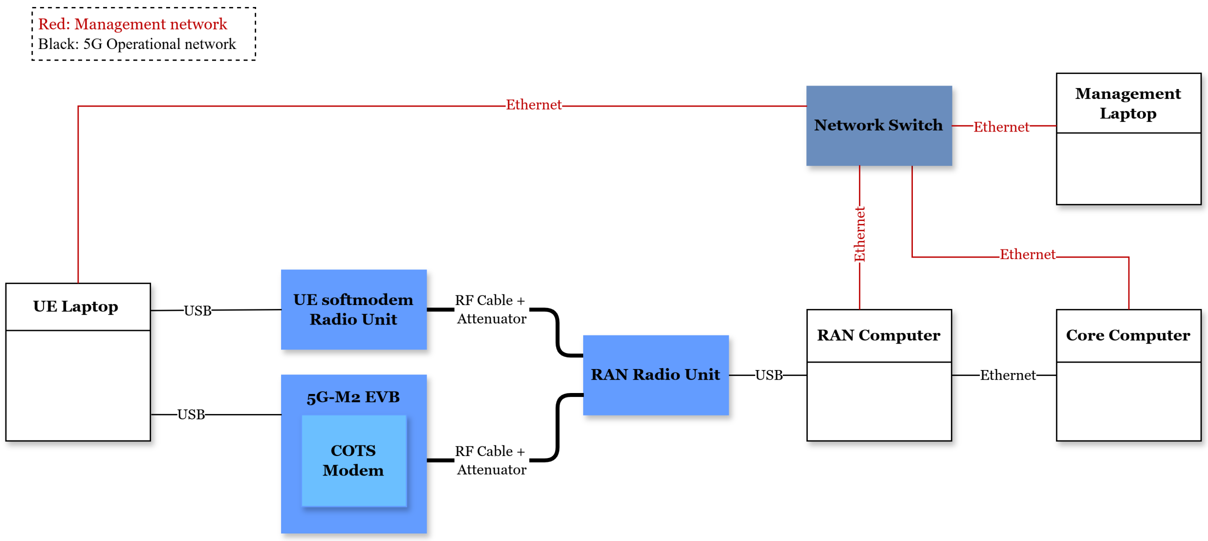

Figure 3.1: Architecture diagram of the OAI testbed

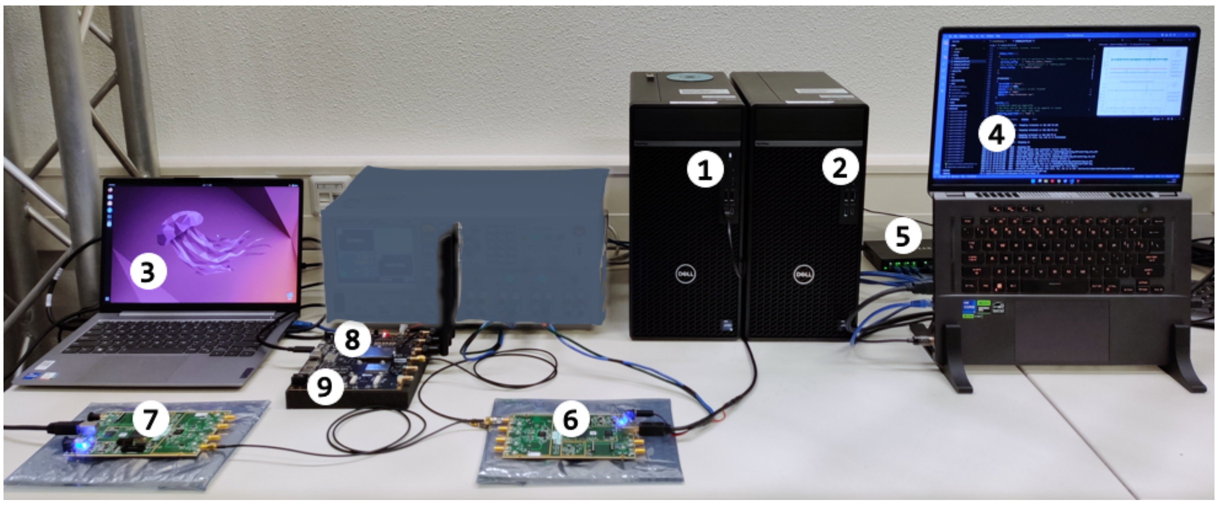

Figure 3.2: Photo of the OAI testbed

Table 3.1: List of used equipment for the OAI testbed

	Name	Hardware
1	RAN	Dell PC i7-13700
2	Core	Dell PC i7-13700
3	UE	Lenovo ThinkBook 14 G6 IRL i7-13700
4	Management	Generic laptop
5	Network Switch	Netgear GS308E
6	RAN SDR	Ettus Research USRP B210
7	UE SDR	Ettus Research USRP B210
8	UE COTS 5G Modem	Quectel RM500U
9	UE 5G Evaluation Board	Quectel 5G-M2 EVB

3.3.2 Testbed configuration

Separate configuration files were made for each supported bandwidth, as changing the bandwidth in a configuration requires changing a lot of parameters. Smaller changes to single parameters can easily be made via the command line. The available configurations are for a bandwidth of 10, 20 and 40 MHz. All configurations use 30 KHz subcarrier spacing, and use band 78. The configurations all have the same static TDD, of 7 DL slots, 2 UL slots, and a flex slot with 6 DL symbols and 4 UL symbols.

To let the RAN function properly, the argument $--continuous-tx$ needs to be added. This arguments prevents the transmitter of the radio from turning off when no data has to be transmitted. In the merge request for this function [38], the following is said about why this feature is needed: ”In case of TDD we usually disable TX during RX. Unfortunately some USRPs create self-interference after disabling TX. To be able to work-around this issue, we add the flag ”–continuous-tx”.” This applies to the B210 as well.

3.4 Testbed performance evaluation

In this section, the performance of the 5G testbed is evaluated in its current state. The performance metrics of the connection are latency and throughput. The same tests have been performed for both the ue-softmodem and the ue COTS modem.

3.4.1 Throughput

The expected throughput has been calculated based on the formula given in Subsection 2.3.3. Throughput testing was done using $iperf3$ as traffic generator, with the server running on the Core machine and the client running on the UE machine. Figure 3.3 shows the throughput of the testbed for different bandwidths, both in the UL and DL directions. It can be seen that the UL performance of the testbed is not 100% compared to the theoretical maximum. The COTS modem performs slightly better on the UL compared to the softmodem.

Two potential reasons, which might also be related to each other, are causing the relatively poor performance of the UL. The first reason is that it is also possible that the UL decoding in OAI is simply not optimally implemented for maximum data rates. There is no official documentation for OAI where the maximum data rates are shown. The second reason is that the UL requires more computations: in [25], it is claimed that the UL requires 2.5 times more computations compared to the DL.

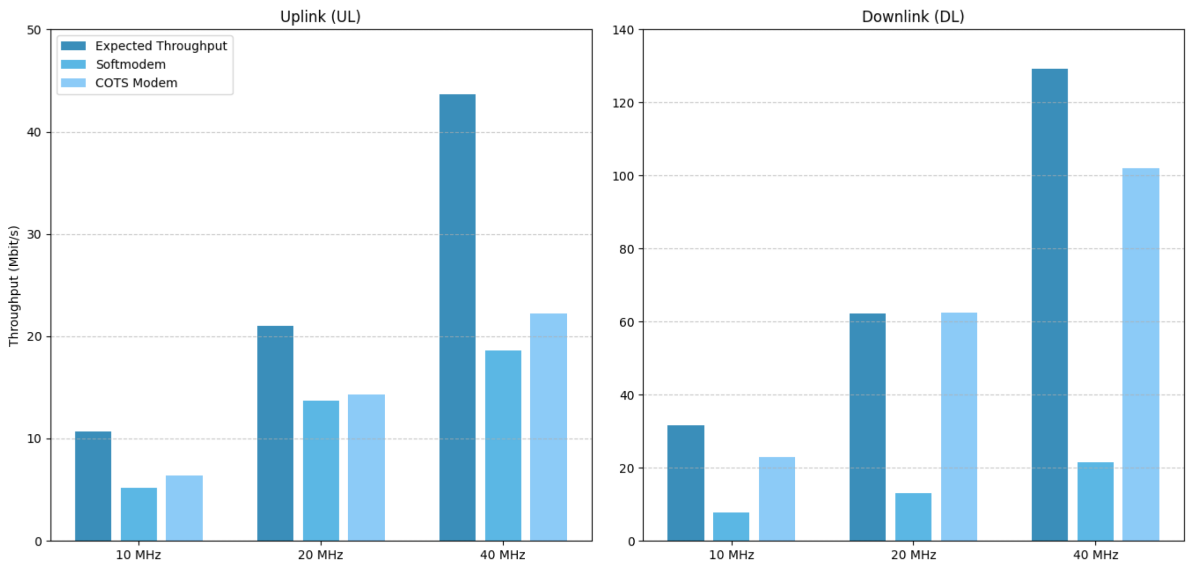

Figure 3.3: Maximum throughput for three OAI bandwidth configurations

3.4.2 Latency

The latency is tested using $ping$. The COTS UE is used for this test. A ping command from the UE laptop to the core was used, with a duration of 60 seconds, interval 1 second. This test was performed for each of the three bandwidth configurations. The latency is measured by taking the round trip time (RTT), which is the time it takes for a ping echo request to be sent from a host to a remote machine and back to the host.

The results are presented in Table 3.2. The average round-trip time was the lowest for the 20 MHz configuration, which was also the best performing configuration for the throughput test. The 10 MHz and 40 MHz test have quite similar results. What is notable is that these two configurations have some outliers, especially on the lower side. The average RTT for every configuration is within the typical range of actual 5G networks [39].

Table 3.2: RTT for each bandwidth configuration

Bandwidth (MHz)	**Min RTT (ms)**	**Avg RTT (ms)**	**Max RTT (ms)**
10	9.906	30.748	33.884
20	19.257	24.517	31.442
40	11.808	30.628	37.098

4 POWER MEASUREMENTS

This chapter focuses on the methodology and implementation of power measurement techniques in the 5G testbed environment. Accurate power consumption data is crucial for understanding the energy demands of each component in the network. The chapter begins by examining the primary contributors to 5G power consumption and provides an overview of related testbeds used in similar research. Next, the hardware and software configurations employed to measure the power within the testbed are explained. After this, a discussion of the Linux CPU governors is presented. The final section of this chapter presents the design decisions and implementation of the power measurement elements on the OAI testbed.

4.1 Background

4.1.1 5G Power consumption components

In [9], the power consumption components of O-RAN are divided into two categories: The Radio Network and the Transport Network. In the context of this thesis, the Radio Network is divided into two categories: The computing hardware and the radio hardware. This decision is made because in the to-be-designed testbed, the approach for measuring the power consumption of these components is different. The three categories will be explained below:

(a) **Computing hardware:** As explained in Section 2.2, the implementation of mobile networks has shifted from specific hardware to software implementation on COTS server hardware. This means that the main contribution to the power consumption now comes from the power used by general-purpose processors (GPPs), which run the DUs and CUs. Also included in this category is all the involved hardware to run the GPP servers: cooling hardware, monitoring hardware, power supplies and conversion systems.

(b) **Radio hardware:** This category contains all hardware used to transmit and receive radio signals, such as RF transceivers and power amplifiers.

(c) **Transport network:** This component contains all the networking hardware, like switches, that are used for the backhaul, midhaul, and fronthaul transport. In [40], the contribution of the transport network on the total power consumption is researched for three split options 6, 7 and 8. The contribution of transport network for these splits is concluded to be about 2%, 30%, and 60%, respectively. This shows that the functional split has a large influence on the ratio of the energy used by this component.

4.2 Existing power testbeds

Many publications on the realization of 5G testbeds are available, as it has become more accessible with the virtualization and availability of open-source software. However, most of the work that uses these testbeds focuses on performance measurements. In the literature review, three publications on experimental setups for energy consumption measurements were found, which contain a lot of useful information for this thesis. They share many similarities, as will be shown later. In this section, the research goals and conclusions of the papers found will be presented first, followed by subsections that explain the methods that were used to do the measurements.

The first paper is published by Ayala-Romero et al. [25]. This work will be referred to as testbed A. Testbed A uses srsRAN [41] as RAN software. The work measures and compares the total power consumption of a COTS computer and the part that is used by the CPU alone. The experiments are carried out with varying traffic load, channel quality, modulation selection, and bandwidth with uplink transmissions. The authors justify the choice for exploiting only the uplink because the power-load relation is non-linear and involves 2.5 times more computations. The collected data is used to design two linear mixed-effect power usage models. In addition to these models, the authors also published the data set that was collected from their research.

Testbed A evaluated the impact of the COTS hardware by using a total of 4 different machines for the RAN computer. An increase in power consumption of 460.86% was found when changing from small NUC pc to general-purpose server. The portion of power consumed by the CPU with respect to the total power ranges from 28.96% to 49.69%.

The second paper using an experimental power measurement testbed is published by Salvat et al. [42], and will be called testbed B. The setup in the paper is larger, utilizing not one but five radio units connected point-to-point with five UE laptops. The radios are controlled by a single vRAN desktop computer. SrsRAN is also used as a software platform for this testbed.

In addition to power measurements, testbed B also collects performance data on different radio metrics and data flow metrics. The authors have collected three datasets: a computing, energy, and application datasets. The latter is to use an edge server application to run an image-processing algorithm for UEs. The data sets have been made available for further research.

The third paper is published by Pawar et al. [43]. This testbed will be referred to as testbed C. The research focuses on tuning the computing resources of the COTS hardware on which an OAI LTE RAN is running. The energy consumption is profiled, and the CPU frequency is optimized for different parameters. The researchers of testbed C investigated the Linux CPU governor settings in more detail (refer to Section 4.3 for a detailed explanation on this). One of the tuning methods included manually setting the CPU frequency and finding the lower clock speed for which there was no reduced performance on the testbed. They also investigated the $conservative$ governor, which adjusts CPU frequency according to load, but in a much more conservative way than the normal $ondemand$ governor. This governor performed slightly less efficient than the manual tuning method, but did not influence testbed performance.

The Ettus USRP B210 is used as RRH in all three setups. The papers have in common that 4G versions of srsRAN and OAI are used. No published paper on 5G power testbeds was found.

4.2.1 Hardware power measurements

Both testbed A and B use a GW-INSTEK GPM-8213 power meter in combination with a GPM-001 measurement adapter. The power cords of each machine connect to this adapter. Therefore, all power consumed by the machines is measured (power supply, CPU, RAM, cooling, etc.). Furthermore, the power consumed by the RRH (the B210) is also included in these measurements, since it is powered by the USB 3.0 cable which connects to the RAN computer. Testbed B measures the power consumption of both the vRAN and the mobile core machine.

4.2.2 Software power measurements

The second method to measure the power of the computers is using the Intel Running Average Power Limit (RAPL). RAPL is a hardware feature of Intel processors which provides several counters for power and load of the CPU. It does not actually measure the power usage of the CPU, but instead uses hardware performance counters and I/O models to calculate an estimate. Turbostat [44] is a Linux utility program that reads these RAPL registers and that can log the data to the terminal or log files. Turbostat is used by all three testbeds to get the CPU power usage. A correlation coefficient of 0.99 is found in [45] between the power outlet measurements and the RAPL values. Performance overhead was tested for a range of sampling frequencies and test algorithms and was found to be ranging from 0.70% to 1.20% for all tested algorithms at a frequency range of 1000 Hz to 1100 Hz. In [46], it is concluded that the difference between the power outlet measurements and the RAPL measurements is not constant, which is caused by the other components of the computer, such as the power supply unit and the fans.

4.2.3 Experiment parameters

Testbed A configures the following parameters for each experiment: transmission mode (TM), uplink traffic load, transmission gain at the UE (which directly impacts the SNR), MCS, and Airtime. Airtime is represented by $a \in [0, 1]$ where $a = 1$ indicates that all subframes are used, and $a = 0$ that no frames are used. Testbed B executed experiments with a wide selection of configurable parameters: DL and UL MCS, UL and DL traffic load, bandwidth, TX gain and airtime allocation. In addition to experiments with random data injection, the authors also execute an edge application dataset. Testbed C used a constant BW of 10 MHz (50 PRBs). In their experiments, they changed the MCS, traffic load, and the percentage of PRBs used.

4.2.4 Data collection

The testbed measurement data must be collected in a central place to process and store them. Testbed A uses a Python script that selects a configuration and gathers measurement data in a centralized way.

Testbed A uses a statistical measurement method: It runs a certain test configuration for one minute, while continuously sampling the software and hardware power consumption during this period, from which the average and variance are calculated.

Testbed B takes another approach for their dataset and instead generates a time-based dataset. In addition to power measurements, the monitoring metrics from the vRAN platform and O-Cloud are also stored. The near-RT RIC receives the monitoring metrics through the E2 interface, after which the data is passed to the non-RT RIC via the A1 interface. A custom rAPP pushes the data to the time-based database.

Testbed C uses the same static measuring method as testbed A: It runs an iteration of an experiment for 120 seconds and uses the average power during this period.

4.3 Linux CPU governors

In Chapter 3, the CPU settings were applied according to the OAI documentation. Linux CPU governors are mechanisms within the operating system that manage the CPU clock frequency to balance performance and power consumption. Thus, the power consumption of the RAN depends on the chosen CPU governor. These governors, part of the Linux kernel CPUFreq subsystem, dynamically adjust the CPU frequency based on the system's workload. There are several types of governors [47]:

- **Performance:** Sets the CPU statically to the highest frequency.
- **Powersave:** Sets the CPU statically to the lowest frequency.

- **Userspace:** Allows user-space programs to set the CPU frequency, providing manual control.
- **Ondemand:** Dynamically adjusts the CPU frequency based on the current system load, increasing the frequency when demand increases, and decreasing it during idle periods. The system load is checked every sampling_period, default 100 Hz.
- **Conservative:** Similar to the Ondemand governor but increases and decreases the CPU frequency more gradually, aiming for smoother transitions.
- **Schedutil:** Integrates with the kernel scheduler to make frequency scaling decisions based on task scheduling information, in order to achieve more efficient and responsive adjustments.

OAI recommends setting the RAN CPU governor to $performance$. This permanently sets the CPU at the highest base clock speed. This guarantees maximum performance and constant clock; however, the CPU power usage will also be constantly high. Since the goal of this testbed is to get insight on the power usage of 5G networks, a dynamic CPU governor is needed. In Subsection 5.2.1, an experiment is presented that compares the $ondemand$ and $conservative$ governors with the $performance$ governor.

4.4 Power measurement testbed realization

In this section, the design and implementation of the power measurement testbed for 5G energy research are presented. The section begins by outlining the key requirements and design considerations that guided the development of the testbed. After this, the testbed hardware will be presented, followed by a section on the software of the testbed.

4.4.1 Testbed requirements

Below, a list of requirements is presented that the testbed must meet.

1. The testbed must have configurable PHY parameters: Bandwidth, MCS and transmission power
2. The testbed must support UL and DL traffic generation at configurable data rates
3. The testbed must be stable for extended operation without interruptions.
4. The testbed has to support both the OAI ue-softmodem and COTS UE hardware.
5. Experiments must be performed in a controlled environment using an interference-free transmission medium.
6. The CPU governor settings of the RAN computer must be configurable.
7. The testbed must accurately measure the power consumption for the computer and radio hardware.
8. The testbed must have good precision for all measurements.
9. Measurements should be collected with a resolution sufficient for detecting variations in power usage during experiments.
10. The testbed software must be easy to use.

4.4.2 Design decisions

This section presents the higher-level choices that were made for the testbed design. Details on the design and implementation are found in the last two sections of this chapter.

The main objectives of the testbed design are configurability, repeatability, and the use of widely available COTS hardware. OAI is used for the RAN and the core. Two different UE are used for the testbed: a software-implemented UE (OAI with SDR) and a COTS 5G modem.

The design decisions for the testbed design have been categorized and are presented below.

- **Hardware:** COTS hardware is used for flexibility and cost efficiency, and SDRs are used as radios. A COTS UE modem is used for higher performance and additional measurement capabilities. The RF connection is performed using antenna cables and attenuators. A lab power meter is used for measurements of SDR and COTS UE power consumption. The power consumption of the RAN, core, and UE computers is measured using a software-based method using Turbostat.
- **Software:** Python scripts are used to fully automate the testbed and improve repeatability. A separate management computer is used to run this script, which uses SSH connections to run commands on each testbed machine. The script has an argument list that makes it possible to easily change the experiment parameters.
- **Measurement methods:** During the measurement phase, the testbed operates in a steady state, ensuring that the duration of the experiments does not influence the statistical values obtained. Only the precision of the measurements may be affected by the duration, as longer experiments typically reduce random variations. The testbed currently provides *relative measurements*, meaning the results are most useful for comparing different configurations or parameter changes within the testbed itself, rather than directly reflecting real-world deployment scenarios. As such, accuracy is not the primary focus, but rather the consistency and comparability of the results under controlled conditions.
- **Data handling:** All measurements are collected and processed in a single file. The Turbostat and lab power meters measurements are combined into a single dataset which is used for statistical analysis. The results are stored in organized directories and automated plots are generated to visualize power consumption patterns.

4.4.3 Testbed hardware

The architecture of the functional parts of the testbed to run the network is presented in Section 3.3. In this section, the power measurement parts that have been added to the testbed on hardware and software level will be explained. Figure 4.1 shows a block diagram of the testbed. Figure 4.2 shows a photo of the testbed and Table 4.1 lists the hardware used.

Turbostat has been installed on the core, RAN and core machines to measure CPU power usage. The PkgWatt value is used to have the total energy usage of the CPU. No power meters are used to measure the power consumption of the computers.

A laboratory power meter is used to measure the power usage of the SDRs: It deliverers power for the board and measures the voltage and current of the port. The SDRs can be powered over USB, but they also have a DC barrel jack input for connecting a dedicated power supply. Using a USB power measurement tool, it was found that when this external power supply port is used, the B210 will still draw current from the USB port of the computer. In a first attempt to eliminate this, the USB cable was carefully stripped and the 5V line was cut. It was then discovered from the schematics that the voltage regulation circuit of the external power supply input needed the 5V from the USB to be enabled: A second cable was made, to split the 5V from the power meter to both the USB input and the barrel jack.

The 5G-M2 EVB kit has functionality built into the board to measure the power usage of the installed M.2 card. When the EVB is configured in this mode, the M.2 board must be powered by an external source using a dedicated connector. These pads are connected to the power meter. Since the UE laptop does not process any data with the COTS hardware, it functions more as a control machine to inject and receive traffic. The power usage of the UE laptop is therefore not interesting when using the COTS UE modem.

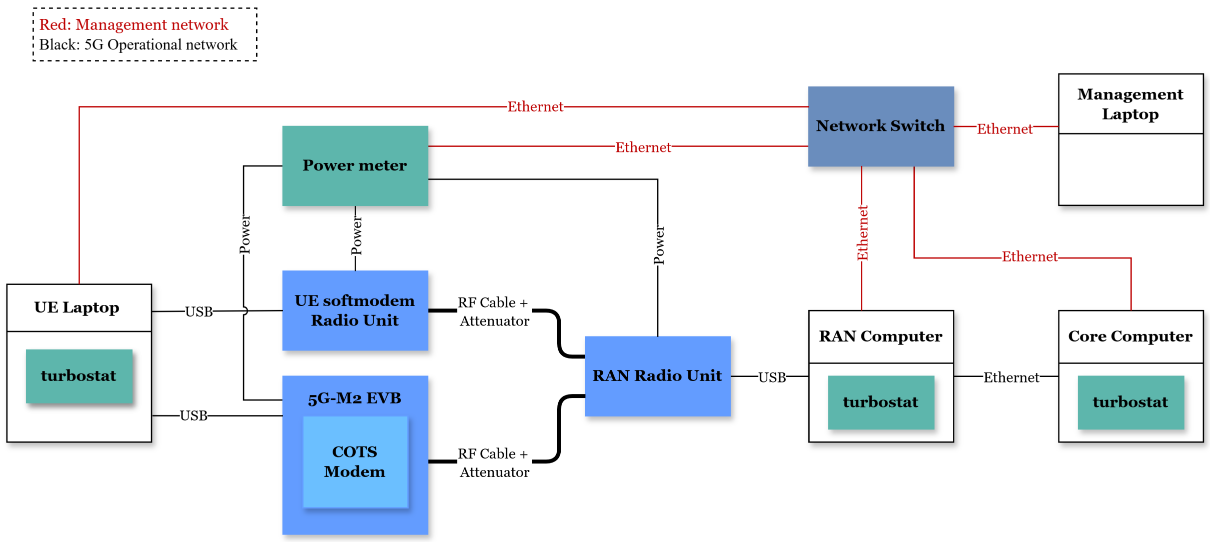

Figure 4.1: Architecture diagram of the OAI testbed (power measurement elements are shown in green)

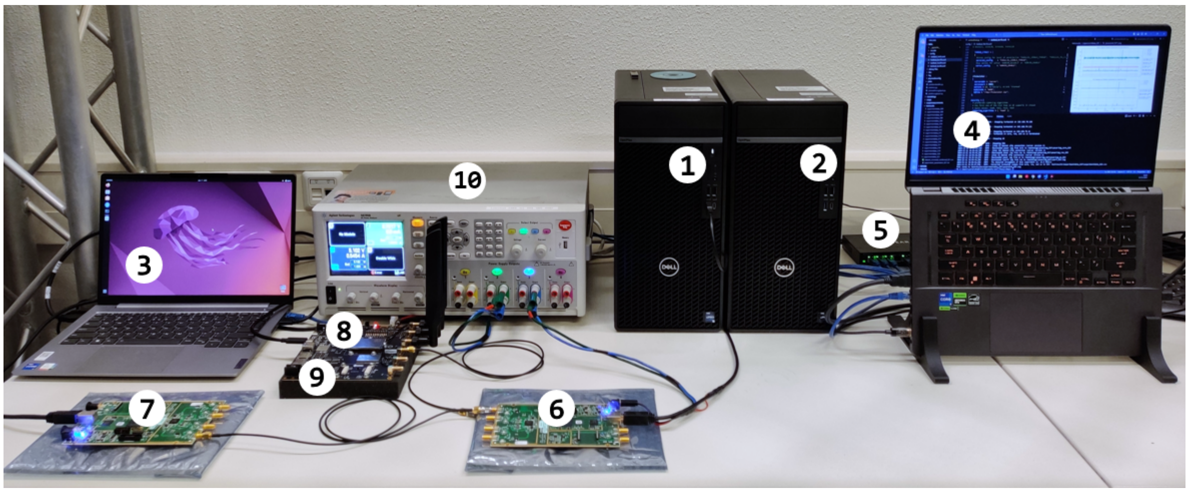

Figure 4.2: Photo of the OAI testbed with power measurement hardware

Table 4.1: List of used equipment

	Name	Hardware
1	RAN	Dell PC i7-13700
2	Core	Dell PC i7-13700
3	UE	Lenovo ThinkBook 14 G6 IRL i7-13700
4	Management	Generic laptop
5	Network Switch	Netgear GS308E
6	RAN SDR	Ettus Research USRP B210
7	UE SDR	Ettus Research USRP B210
8	UE COTS 5G Modem	Quectel RM500U
9	UE 5G Evaluation Board	Quectel 5G-M2 EVB
10	Power meter	Agilent Technologies N6705B

4.4.4 Testbed software

The testbed is controlled from the experiment management machine, which runs a Python script to completely automate the experiments. All code used for the management machine can be found in the repository [48].

4.4.4.1 Software Tools

Every Linux machine on the network is controlled using SSH. The Python library Paramiko [49] is used to execute individual CLI commands. The OAI applications for the RAN and UE run in the foreground: This means that when they are started, they will prevent the management script from continuing. By running the SSH commands in 'asynchronous' mode, the main script will not be stopped. To validate that the applications are running, the script will wait a while and then check if the applications started by using a separate command.

Data recording with the scope is also handled by the script: the Virtual instrument software architecture (VISA) API from the scope is used to send commands and data from and to the scope via the testbeds control network. The PyVISA library [50] is used to do this from the Python script.

4.4.4.2 Software structure

The main script that is used to run the testbed is called $run\_testbed.py$. Running an experiment from this script in the $normalmode$ involves four stages: initialization, execution, termination, and data processing. The four stages will be briefly explained below.

- *Initialization:* This includes all setup tasks for both the 5G network and the measurement tools. First, the scope output is initialized to enable power for the radio units. After this, the 5G machine applications are started in order: Core, RAN, UE. The software measurements are then started on each machine, and finally the scope measurement is started. Lastly, the CPU settings of the RAN are applied.
- *Execution:* In this stage, the experiment is performed. The default test is an iPerf3 traffic injection, of which the runtime and direction can be set in command line. Custom tests can also be added.

- *Termination:* The measurements and OAI components are terminated here. The UE will be stopped first, and after this the RAN is terminated. The core does not have to be killed after every experiment.
- *Data processing:* Since the software and hardware measurements are stored locally, they have to be collected after the experiment is executed. The log files on the OAI machines are collected, trimmed, and combined into a single CSV file together with the hardware measurement data. A plotting script then creates a plot which also shows the mean and standard deviation of each measurement.

In addition, there is the option to run the test in $quickmode$. In this mode, the script does not start and stop the RAN, core and UE. This mode is time efficient, when the only parameters that are changed during experiments do not require the RAN to restart, like a throughput test.

An additional script $batchtest.py$ is available to automatically run the $run\_testbed.py$ script multiple time, with the option to change parameters for each run.

4.4.4.3 Configurability

The $run\_testbed.py$ has an argument list that can be used to configure the testbed for an experiment. An overview of the different configuration parameters is given here.

- **General parameters:** Testing with the ue-softmodem or the COTS UE hardware can be selected via the command line.
- **PHY/radio parameters:** The bandwidth can be set to 10, 20 or 40 MHz. The MCS can be set from 0-28. Radio attenuation of the analog front end of the SDR can be set for rx/tx in steps of 3 dB.
- **Data traffic parameters:** The direction of the test can be set to UL or DL, setting the direction of the iPerf3 traffic. The experiment time and datarate in Mbit/s can be set as well.
- **CPU Settings:** The CPU governor of the RAN can be set to $performance$, $ondemand$, $conservative$, or $userspace$. With the $userspace$ governor, the CPU frequency has also to be added to the argument list.

4.4.4.4 Testbed data

Two folders are created for each experiment, with a unique experiment ID: One for the raw logs and one for the processed results. All experiment parameters are stored in a text file within the results folder.

The sampling rate for both hardware and software measurements is 50 Hz: Although both the turbostat and the power meter can support higher rates (at least 1 KHz), it was found that this makes the RAN computer instable when using the $ondemand$ and $conservative$ governor. The testbed in its current configuration supports statistical analysis, providing aggregate metrics across complete measurement sets. Data collected from each machine and the power meter is combined into a single file without synchronization. Measurements are initiated and terminated as close to the same moment as possible. Despite this, slight variations in data length (particularly in Turbostat files) can occur due to latencies in the execution of the SSH command. Hardware measurements are also not synchronized with the Turbostat measurements.

Data trimming is applied to the beginning and end of each dataset, ensuring that only the stable, steady-state measurements are retained for analysis. This approach works well for static tests, which the testbed presently performs, where synchronized timing is less critical.

An automatic plot of the data set is generated to visualize the power consumption patterns over the duration of the test. Since these signals are not synchronized, the plot cannot be used to correlate signals.

5 EXPERIMENTS AND RESULTS

In this chapter, a series of experiments is presented, divided into three main sections: the general methodology, testbed evaluation, and research into the influence of network parameters on power usage.

The chapter begins with a section detailing the methodology and outlining the setup and procedures applicable to all experiments. Following this, the second section focuses on the evaluation of the testbed, which contains two experiments. The first experiment examines the performance of various CPU governors on the RAN computer, identifying the most suitable configuration. The second experiment evaluates the precision of the measurement system by repeatedly running the same test to analyze the consistency and reliability of the results.

The next section examines how network parameters affect power consumption. The first experiment explores the impact of bandwidth usage, the second focuses on MCS, and the final experiment investigates traffic load on power consumption.

Finally, some general key insights that were found while analyzing all the results will be discussed in the last section of this chapter.

5.1 General setup

- In Section 3.4, the performance of the testbed was evaluated with both the ue-softmodem and the COTS UE. It was found that the throughput performance of the system is better with COTS UE, especially on the DL. Therefore, the COTS UE will is used for all experiments in this chapter. It allows the RAN to operate at a higher maximum capacity, ensuring that the RAN operates closer to its full potential, thus providing more representative data to analyze the impact of network parameters on power consumption.
- Since the UE laptop does not perform processing related to the 5G network with the COTS hardware, it functions more as a control machine to inject and receive traffic. Therefore, the power consumption of the UE computer is not considered in the experiments in this chapter.
- Except in throughput-varying experiments, user-generated traffic is injected into the 5G connection at more than 100% of the maximum throughput for each setup. This approach ensures that all transmission slots are fully utilized, maximizing resource usage and providing consistent conditions.
- In the testbed evaluation experiments, the $ondemand$ and $conservative$ CPU governors were evaluated for the RAN computer. Following the conclusions of these experiments, the $ondemand$ governor was selected as the default governor for the power consumption experiments.
- All measurements were performed while the testbed was in a stable state. The average power and throughput over the time that the experiments run are calculated by the $runtestbed.py$ script and are shown in the results.

5.2 Testbed evaluation

5.2.1 CPU governor

Introduction

This experiment investigates the Linux CPU governors that have been listed in Section 4.3 on the RAN machine. The governors of interest are the $performance$, $conservative$ and $ondemand$ governors. As mentioned above, the OAI specifies the $performance$ governor as the governor that should be used when running OAI. However, this will result in constant, high power draw. The $ondemand$ and $conservative$ governors allow dynamic CPU scaling, making power measurements possible. Understanding the trade-offs between throughput performance and power consumption for these governors is essential for running experiments on the testbed.

Methodology

The RAN application was found to crash when the governor was set to $conservative$ or $ondemand$ before the application was started. However, once the connection with the UE was established, it was possible to change the governor to one of these two. This workaround has been implemented in the testbed script: the script starts at a constant 2100 MHz in the $userspace$ governor and then switches to the governor that has been selected in the argument list.

For this experiment, the 20 MHz DL configuration is used, since it has the most stable performance. This configuration is the only one capable of reaching the theoretical throughput. This minimizes the randomness caused by imperfections in the RF part of the network. Traffic generation is 100% of the theoretical throughput. Each governor has been tested for both the UL and DL directions, for a period of 180 seconds.

Results

In Table 5.1, an overview is shown of the average throughput and power usage of the RAN CPU for the $performance$, $conservative$ and $ondemand$ governors. Figure 5.1 and Figure 5.2 show the processor power draw over time during the experiment for the UL and DL direction, respectively.

Table 5.1: Performance comparison of different CPU governors for the RAN

	UL		**DL**	
CPU Governor	**Power (W)**	**Throughput (Mbit/s)**	**Power (W)**	**Throughput (Mbit/s)**
$performance$	65.00	10.20	65.00	61.00
$conservative$	18.46	8.83	17.98	60.10
$ondemand$	16.73	8.81	16.70	60.90

Analysis

From the measurements, it is clear that both dynamic governors draw significantly less power compared to the $performance$ governor. This is expected as the performance governor is tuned to a constant high clock frequency. The $conservative$ governor draws more power than the $ondemand$ governor. This could be because the $conservative$ governor maintains higher frequencies, whereas the $ondemand$ governor adjusts more aggressively to demand, using less power overall. As has been pointed out in Subsection 3.4.1, the UL does need more resources compared to DL, and it might not yet be optimally implemented in OAI.

Performance wise, it can be seen that the UL tests do suffer from having non-static CPU governors. For the DL, the differences are minimal; however, throughput is also slightly lower. The decrease in performance is suspected to be caused by the processor not being able to react quickly to increasing processing demand, causing missed deadlines in the real-time processing.

Based on these results, the $ondemand$ governor is selected to be suitable for general power experiments on the testbed. It uses the least power, is able to respond the fastest, and has only a limited performance decrease compared to the $conservative$ governor.

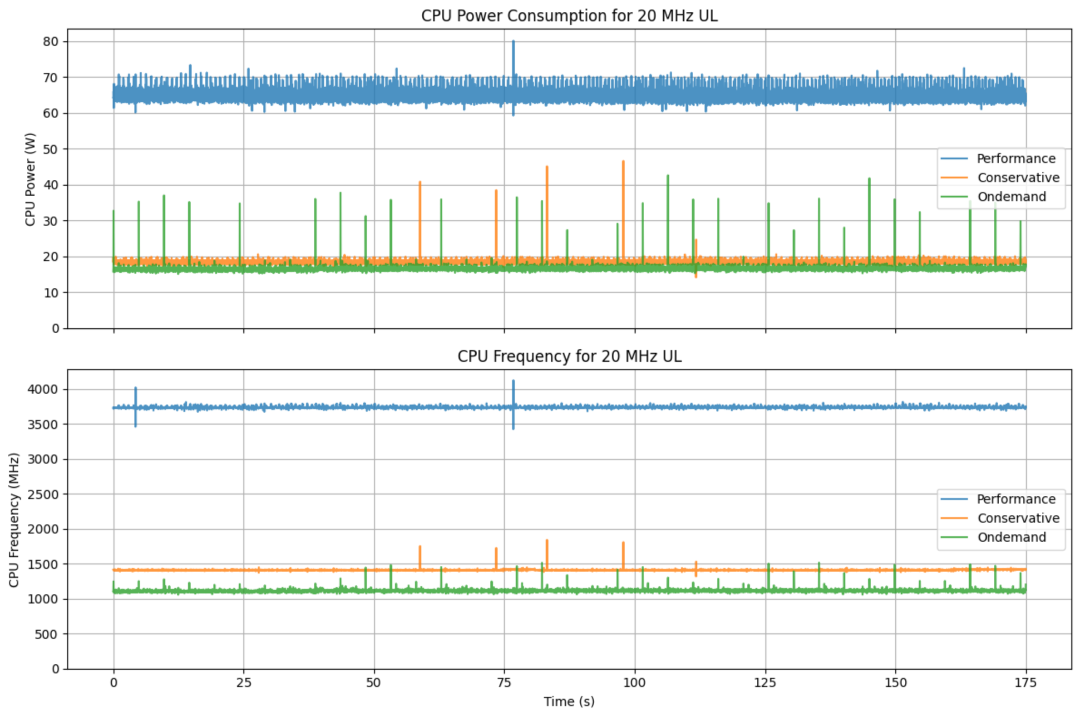

Figure 5.1: RAN CPU power and frequency during a 20 MHz UL test for three CPU governors

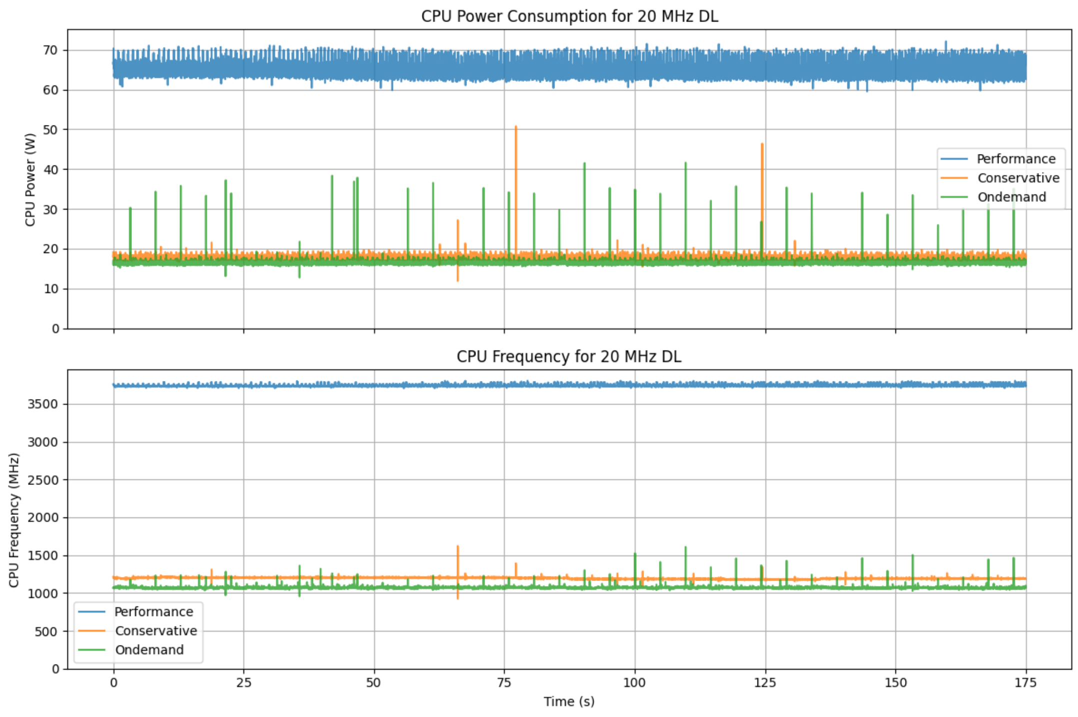

Figure 5.2: RAN CPU power and frequency during a 20 MHz DL test for three CPU governors

5.2.2 Precision evaluation

Introduction

An essential requirement of the testbed is repeatability. *Precision* refers to the degree of consistency or repeatability in the results of measurements or experiments. In other words, it indicates how close multiple measurements are to each other under the same conditions. In the context of this research, precision can be affected both by the power measurements themselves and also by inconsistent behavior by components of the testbed.

In this experiment, the precision of the measurements is evaluated by repeatedly running a single experiment and comparing the resulting measurements. The experiment will be carried out for both the $conservative$ and $ondemand$ governor for the RAN CPU. For the RAN CPU, both quickmode (where the testbed does not reset) and normal mode (where the testbed is reset after each experiment) of the testbed script will be evaluated.

The repeatability of the testbed is analyzed by evaluating the distribution of a data set from a single test. The *range* of the measurements is defined as the difference between the minimum and maximum values in this data set. The distribution of the measurements will be further analyzed using *box plots*. A box plot is a graphical representation of the distribution of a data set that summarizes its central tendency, variability, and potential outliers. It consists of a rectangular box that shows the interquartile range (IQR), which is the range that covers the middle 50% of the data (calculated as the difference between the third quartile, Q3, and the first quartile, Q1). The line inside the box represents the median. Whiskers extend from the edges of the box to the smallest and largest values within 1.5 times the IQR, while points beyond the whiskers are considered outliers and are plotted individually. Finally, the relative precision is a measure used in this experiment and is defined by Equation 5.1. Note that a lower value is better.

$$\text{Relative Precision (\%)} = \frac{\text{Range}}{\text{Mean}} \times 100 \tag{5.1}$$

Methodology

The 20 MHz configuration with DL traffic is used for this experiment. This configuration was chosen for its proven stability, which minimizes the impact of network imperfections or randomness. By isolating these factors, the experiment focuses solely on power measurement variations and any inherent randomness caused by the testbed hardware. It must be emphasized that the CPU governor was only changed for the RAN computer, leaving the core computer CPU in the default governor.

Four data sets were collected: normal ondemand, quickmode ondemand, normal conservative, quickmode conservative. For each dataset, the same routine was used to try to represent the atypical usage of the testbed. During testing, power cycles were performed in which computers and radios were fully powered down and restarted. This was done to introduce randomness into the testbed state, ensuring that measurements captured as wide a range of possible conditions as feasible.

The power cycles should not be confused with experiment restarts in normal mode. In normal mode, the core and RAN are restarted by closing the software and starting it again with the new parameters. The computer hosts remain powered on.

The $batchtest$ script was used to run four consecutive experiments between each power cycle, with each individual experiment lasting 180 seconds. This routine was repeated three times, resulting in a total of 12 experiments per data set.

Results

The results are presented using box plots in Figure 5.3 and Figure 5.4. Furthermore, the data sets are visualized using a scatter plot that plots the power measurements against the throughput in Figure 5.5. A table showing the relative precision for each measurement for each data set can be found in Table 5.2.

The plots presented in this experiment use a truncated Y-axis to maximize resolution. Although this might give the impression that precision is poor, it actually provides the clearest possible insight into the precision of the measurements by emphasizing even small variations in the data.

Table 5.2: Relative precision comparison for each of the four measurement data sets

Component	Normal Ondemand	Quickmode Ondemand	Normal Conservative	Quickmode Conservative
Core CPU	4.01%	5.14%	0.75%	0.66%
RAN CPU	2.81%	1.59%	8.40%	8.54%
RAN Radio	0.00%	0.59%	0.00%	0.29%
Throughput	7.79%	2.86%	8.58%	3.26%
UE Modem	2.61%	4.14%	4.16%	7.46%

Analysis

Analysis is performed for each testbed component, after which the complete experiment is concluded.

- **Core CPU:** The core CPU governor was not changed during the experiments; however, the results between the data sets are different. The total range of all experiments is much smaller as the RAN CPU measurements: the range of all measurements is 0.70 w. Excluding the outliers, the range of all measurements is 0.32 W. The results were more precise when the $conservative$ governor was used for the RAN CPU.
- **RAN CPU:** For the RAN CPU, the most noticeable observation is the big difference in range between the two CPU governors. The ondemand governor has a much better relative precision range for both normal mode (2.81%) and quickmode (1.59%). The conservative tests also have a higher average. The quickmode ondemand test has the best precision.
- **RAN Radio:** The RAN radio has very consistent measurements, with almost all power measurements steady at 3.39 to 3.40 W. The range of the measurements is 0.03 W.
- **UE Modem:** The UE modem shows quite similar results for three of the four data sets. An anomaly is the quickmode conservative test, where the range of measurements is much wider (0.15 W), with a significantly higher average. This test also has the lowest average throughput; this could be related, because the UE has to process more retransmissions and has to adapt more to a less stable RAN performance, which makes the link less stable. The range of the UE modem measurements is small, but average power consumption is also low, making the relative precision lower.
- **Throughput:** The throughput has a combined range of 6.0. Three outliers match the approximate theoretical maximum throughput. This shows that even with the most stable RAN configuration, throughput is now always exactly the same when using the dynamic CPU governors. Normal conservative has the highest throughput variability (5.10 Mbps), while quickmode ondemand is the most stable (1.70 Mbps). In the scatter plots, no clear overall trends can be detected for the relation between the throughput and power usage of the testbed components.

An general observation from the experiment was that the sets of four measurements taken between power cycles showed no correlation with each other, or with other sets of four measurements. This indicates that the power cycles and the order of the measurements do not bias the results.

Core CPU power is largely unaffected by the mode and scaling governors. The ondemand governor results in more consistent power usage and generally lower power consumption for the RAN CPU compared to the $conservative$ governor. RAN radio measurements are very similar across all tests. The UE modem unexpectedly has higher measurements for the quickmode conservative test. The UE modem measurements all have a small distribution range, with the normal mode experiments being the most precise.

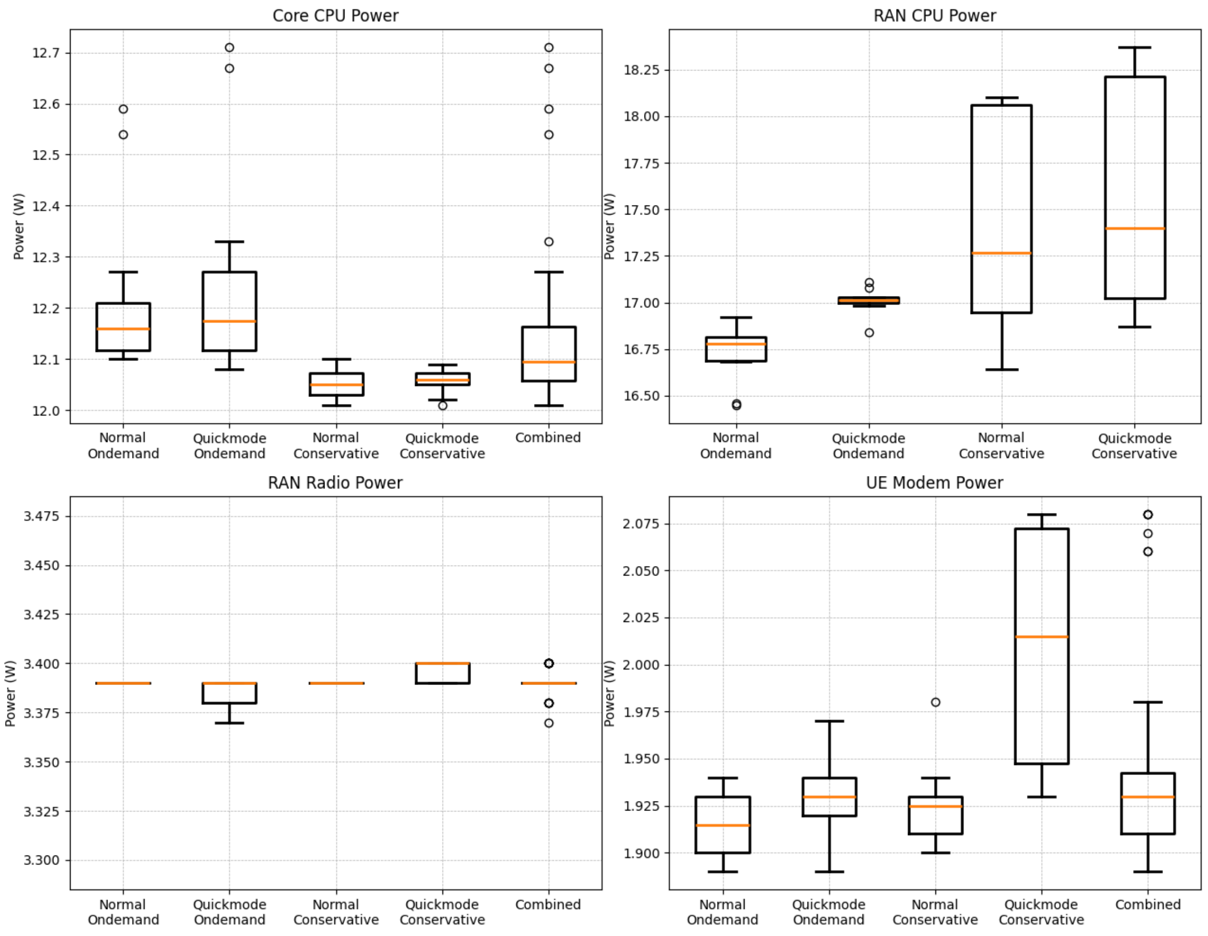

Figure 5.3: Distribution of the power measurements of the repeated experiment

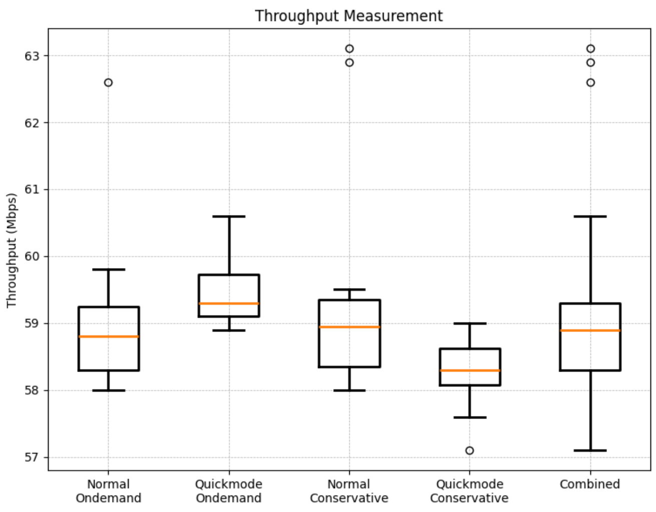

Figure 5.4: Distribution of the throughput measurement of the repeated experiment

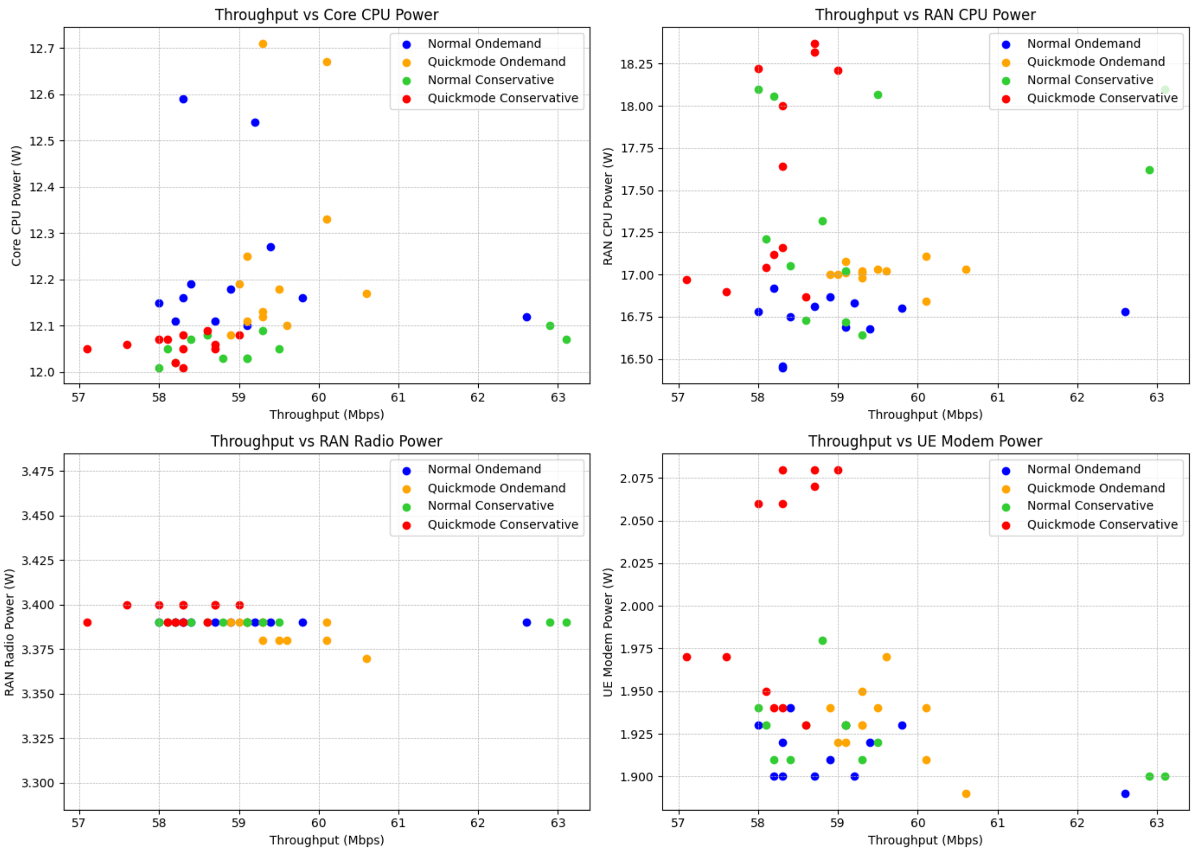

Figure 5.5: Scatter plots of the throughput against power measurements for the repeated experiment

5.3 Power consumption research

In this section, a three experiments are presented that investigate the power consumption of the testbed components. An important goal of these tests is to demonstrate the functionality of the testbed. Three experiments are done to investigate the influence of bandwidth, throughput, and MCS on the power consumption of the core CPU, RAN CPU, RAN radio and UE modem.

5.3.1 Bandwidth

Introduction

This experiment explores the relationship between channel width and power usage. The bandwidth expansion allows more data to be transmitted in the same time frame, increasing throughput. However, a larger bandwidth requires higher processing power.

Methodology

Six experiments were conducted: Both an UL and a DL test for the 10, 20 and 40 MHz configurations. Each experiment has a runtime of 180 seconds. 100% of the theoretical maximum data rate was injected, however, as has been found in Section 3.4, this 100% throughput performance is not achieved with most configurations. Therefore, it is expected that the results will be different from those of a network that performs at full capacity. In addition to these measurements, the idle state is also measured. This is a measurement where the OAI software was not running and the RAN radio and UE were powered on without connection.

Results

The average power consumption of each component of the 5G network is shown in Figure 5.6 for every configuration. Furthermore, the throughput obtained during these experiments is shown in Table 5.3.

Analysis

It can be seen that the general trend for the RAN CPU and RAN radio is that more power is drawn for higher-bandwidth configurations, both for the UL and DL. The RAN CPU has slightly less power consumption for DL tests. However, the UL TDD slots are already less, and the performance is less than that of the DL. Therefore, it is expected that UL will be more CPU intensive than DL.

The core CPU seems to slightly follow this trend in most measurements. An exception to this is the 20 MHz downlink measurement. What makes this configuration unique is that, in contrast to the other configurations, it can run at 100% of the expected throughput performance. These two things could be related. It is also notable that the core CPU only has slightly higher power usage compared to the idle measurement.

The UE modem does not follow the trend of using more power for higher bandwidth configurations. It uses more power for the UL than the DL. Most of the power usage of the UE modem is expected to come from the radio part. Processing is designed in an optimized way for this modem, and the modem is capable of running at much larger bandwidth and data rates.

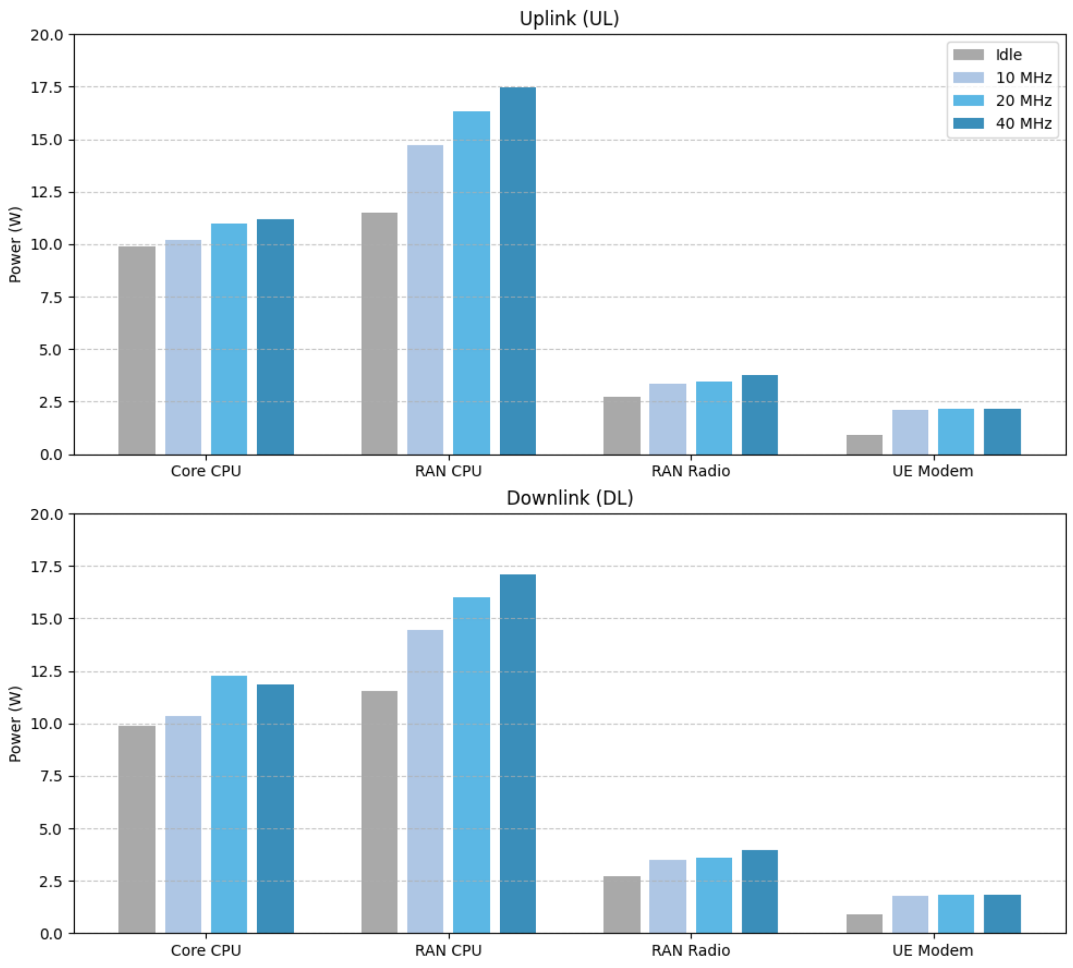

Figure 5.6: Power consumption of the 5G network components for idle state, 10 MHz, 20 MHz, and 40 MHz bandwidth

Table 5.3: Throughput of the bandwidth experiments

Bandwidth (MHz)	**UL Throughput (Mbps)**	**DL Throughput (Mbps)**
10	6.13	22.5
20	10.1	60.0
40	27.4	113.0

5.3.2 Throughput

Introduction

This experiment investigates the influence of the percentage of user traffic on the 5G connection in relation to the maximum theoretical throughput. The power consumption of the system components is expected to increase as more user data need to be processed.

Methodology

Tests are done for both UL and DL, for 10, 20 and 40 MHz. Data rates from 0-100% in steps of 10% are being used. The values have to be given in Mbit/s for the testbed script and are calculated manually by having 11 values between 0 and the maximum theoretical throughput of each configuration. These values were put into the $batchtest$ script. Quickmode was used to run all throughput values for a single bandwidth and data direction. The duration of a single measurement is 60 seconds.

Results

Figure 5.7, Figure 5.8 and Figure 5.9 show the results for the UL tests. The DL results can be found in Figure 5.7, Figure 5.8 and Figure 5.9.

Each figure has four subplots for the Core CPU, RAN CPU, RAN radio and UE modem power usage. The x-axis show the injected traffic percentage compared to the theoretical maximum. However, not every configuration is able to achieve the full 100%. The maximum achieved throughput is shown by a black striped line: This means that actual throughput on the right of this line is capped at the value where the line is placed.

The plots presented in this experiment use a truncated Y-axis to maximize resolution, providing the best insight into the power consumption trends.

Analysis

The analysis is done separately for the four components of the testbed, after which the analysis is summarized.

- **Core CPU:** While there are trends visible for the individual figures, not many relations between those trends over all configuration experiments can be found. In the 10 MHz DL test, the core CPU is almost perfectly constant. For the 20 MHz and 40 MHz DL tests, the Core CPU power is varying more. After the maximum theoretical throughput threshold, there is a bigger jump visible for both tests. However, with the 20 MHz DL test it goes down by 0.4 W, while for the 40 MHz DL test it goes up 0.4 W.

  The 10 MHz DL test is also quite constant, like the UL test with the same BW. The 20 MHz test also seems constant for the whole range except for when no traffic is send over the network. The 40 MHz DL test seems to increase exponentially until the max reachable throughput, and then drops. The resolution of this curve is so small however, that this could also be caused by randomness.

  In general, there does not seem to be a clear relation between the amount of user traffic over the 5G connection and the core network power usage. The tasks performed on the core network are primarily related to user-plane packet forwarding, control-plane signaling, session management, and other tasks that will not heavily be influenced by the amount of data sent over a single link.

- **RAN CPU:** For the RAN CPU, a clearer relation can be seen between throughput and power. For most configurations, the step from 0% to 10% is large, after which there is a stable increase in power. The only exception is the 40 MHz DL, where the power increase between 0% to 10% is about the same as the rest of the steps. At the throughput cap, the power consumption stays more or less the same for every configuration.

- **RAN radio:** Overall, the RAN radio has almost constant power usage throughout the whole range of throughput for every test. For the UL tests, the radio is mostly receiving, and no increase in power is detected for higher throughput. The DL tests have a very slight increase in power consumption for higher data rates, since the radio is transmitting more. For the 40 MHz UL test, the power is significantly lower when no data is being sent. This is not true for the other configurations.

  The consistency in the RAN radio power usage is expected to be caused by the OAI $continuous - tx$ argument, which is needed for the B210 leaves the transmitter of the radio always on (explained in Subsection 3.3.2). The transmitter is the main contributor to the power consumption of the SDR, and used a significantly higher amount of power compared to the receiver and processing hardware.

- **UE modem:** The UE modem mostly shows similar behavior as the RAN CPU: a steady increase in power as throughput increases, up to the threshold, after which the power stays more or less constant. With the 40 MHz UL configuration, power seems to keep increasing slightly by 0.2 W. The power consumption goes up slightly at 0% throughput when a higher bandwidth configuration is used.

Overall, each of the individual four testbed components show behavior that can be explained logically. The power core does not show a clear relation to the throughput. The RAN CPU and UE modem show an increase in power for increasing throughput, until the throughput cap is reached. A small variation in their power consumptions can still be observed for the values after the throughput cap: however, the range of these variations falls within the precision range that was found in Subsection 5.2.2. The RAN radio only shows a small increase in power for DL tests, but remains quite constant since the TX is continuously on.

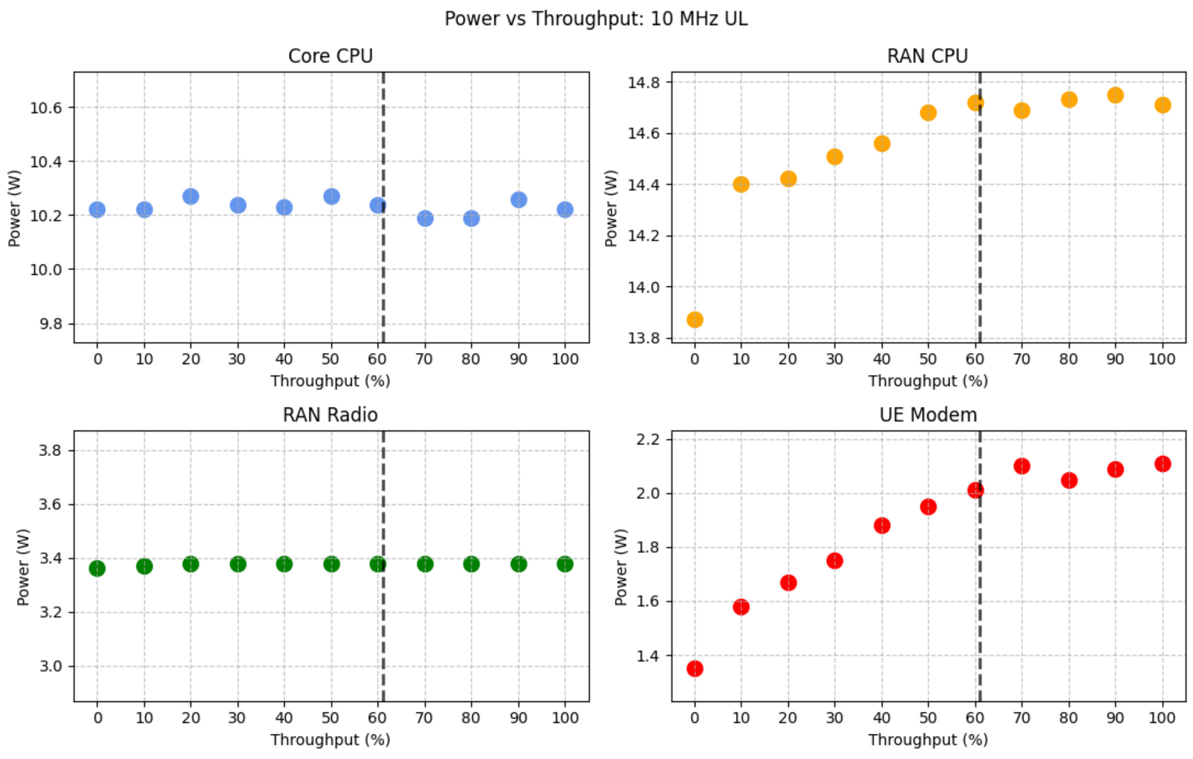

Figure 5.7: Power consumption against UL throughput for 10 MHz configuration

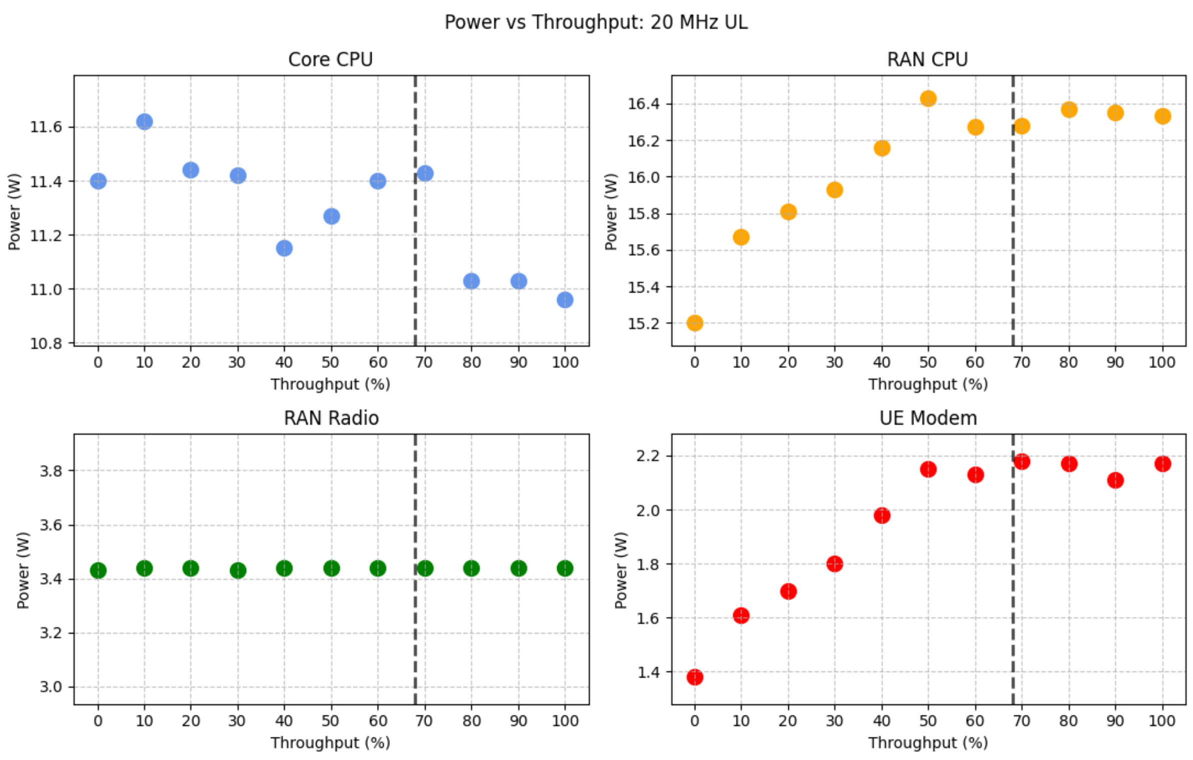

Figure 5.8: Power consumption against UL throughput for 20 MHz configuration

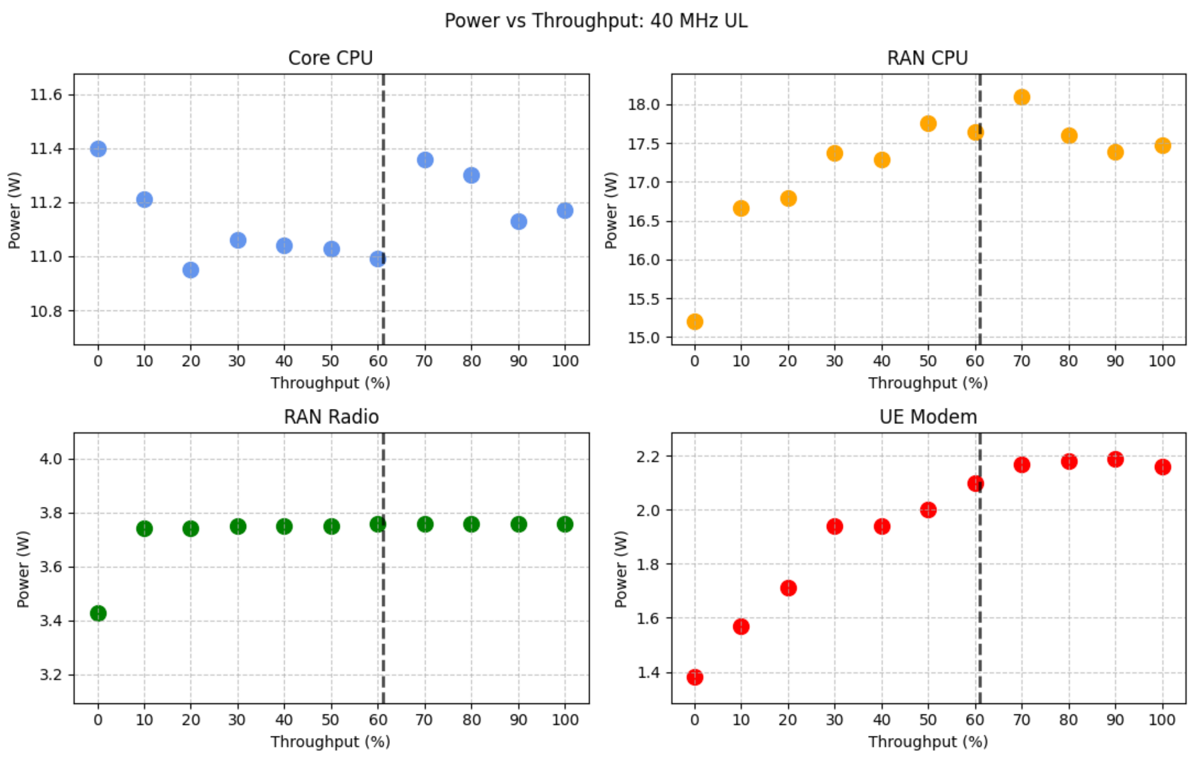

Figure 5.9: Power consumption against UL throughput for 40 MHz configuration

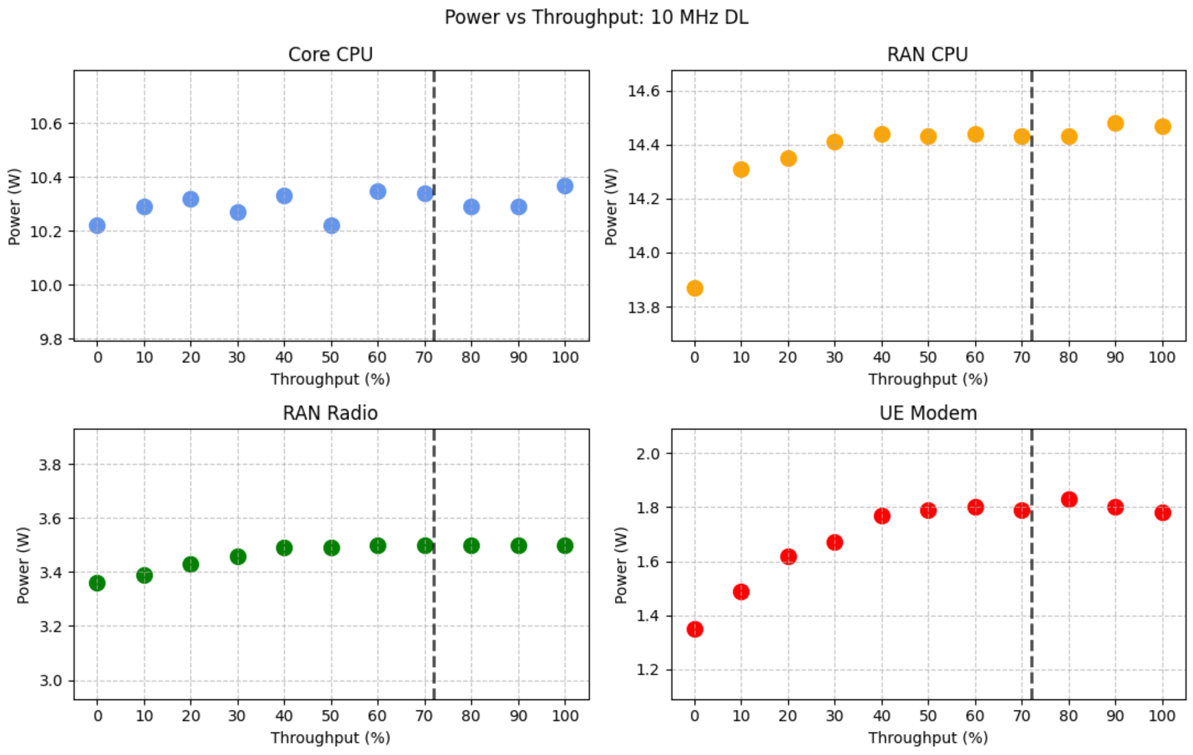

Figure 5.10: Power consumption against DL throughput for 20 MHz configuration

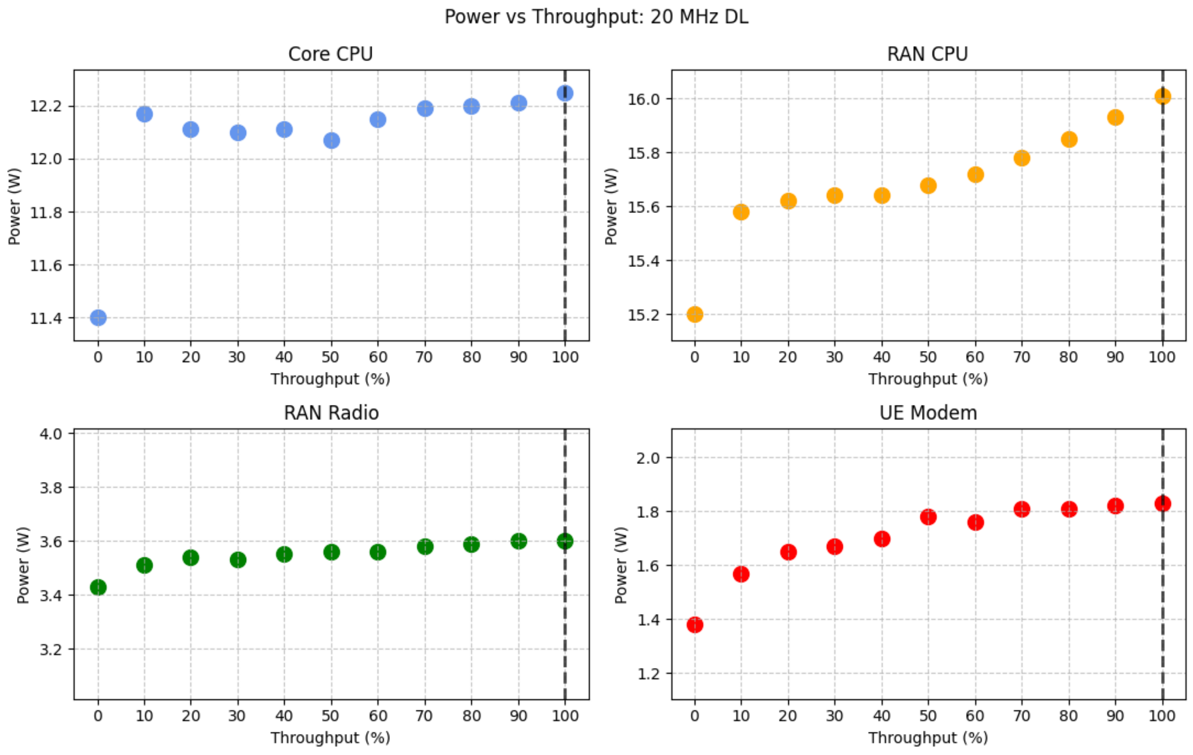

Figure 5.11: Power consumption against DL throughput for 20 MHz configuration

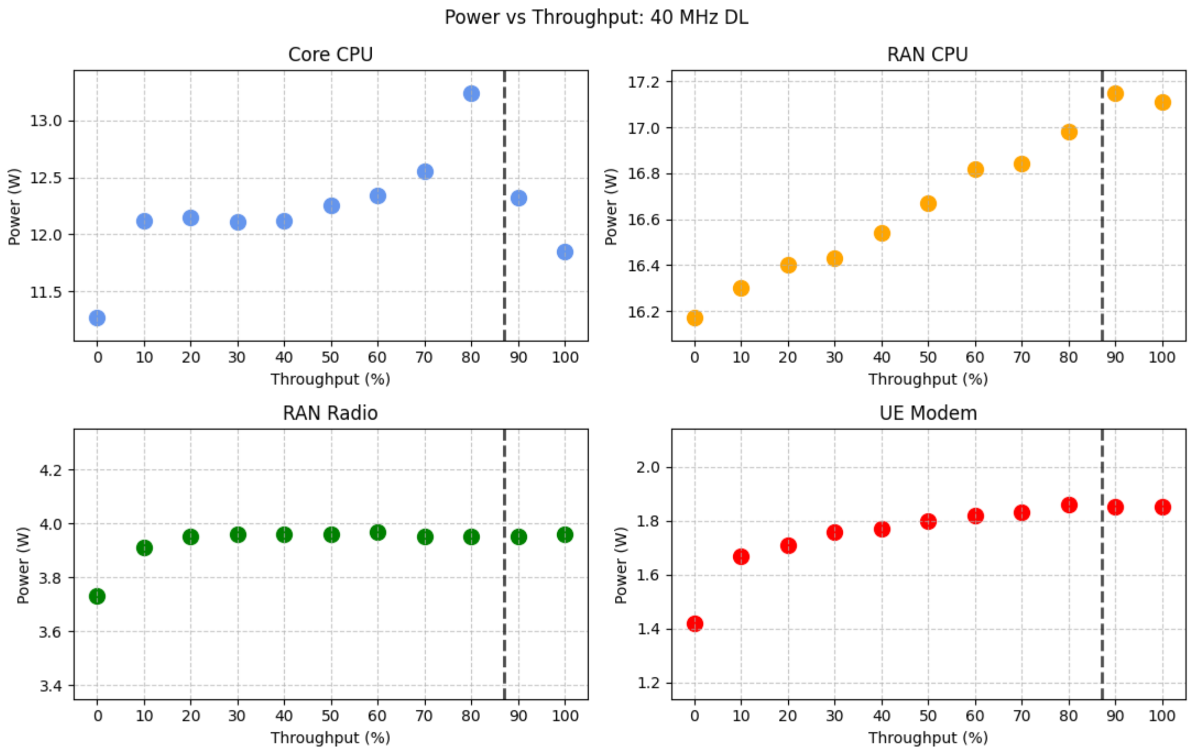

Figure 5.12: Power consumption against DL throughput for 40 MHz configuration

5.3.3 MCS

Introduction

In this experiment, the relationship of the maximum MCS and the power consumption of the 5G network is tested. Higher MCS requires more processing and possible higher TX power to maintain a high enough SNR. Since MCS directly influences throughput, it is expected that the plots will look similar to those of the throughput test.

The normal mode of the testbed script has to be used, in contrast to the throughput test, where quickmode could be used. This is because with the current setup it is not possible to change the MCS parameter while the RAN software is running. This may make the results less precise compared to the throughput tests.

Methodology

The $batchtest$ script is used to run a single configuration with the different MCS values. The MCS starts at 3, since the lower MCS did not allow for successful iperf3 tests. Steps of 5 are used, which results in a total of 6 tests per configuration. As has been said in the introduction of this experiment, normal mode has to be used for the test script to allow the MCS to be changed. A single test has a duration of 60 seconds.

Results

The results of the experiment are shown in Figure 5.13, Figure 5.14 and Figure 5.15. Each figure has four subplots for the Core CPU, RAN CPU, RAN Radio, and UE mode power usage. The plots presented in this experiment use a truncated Y-axis to maximize resolution.

Analysis

Analysis is performed per testbed component:

- **Core CPU:** The core does not show clear trends and remains quite constant for 10 MHz and 20 MHz BW. The 40 MHz test has some bigger jumps, with a range of 0.5 W.
- **RAN CPU:** For the 10 MHz test, the power remains withing a small range of 0.29 W, and no clear trend is visible. The 20 MHz test exhibits a climbing trend, but the measured values deviate noticeably from the expected trend line, indicating potential inconsistencies. The 40 MHz shows are clearer trend line, with only the power at 28 MCS not following the line exactly, possibly caused by the system not being able to function correctly with this highest MCS value. From the precision experiment in Subsection 5.2.2, it was found that the precision of the RAN CPU measurements is much lower for the normal mode than the for the quickmode method. This is expected to be the reason that the RAN CPU measurements do not show a clear relation.
- **RAN Radio:** Like the throughput experiments, the RAN radio measurements remain very constant for all measurements.
- **UE Radio:** The UE radio has a small increase in power consumption for higher MCS, however, since the tests are DL and the modem is mostly receiving, there is only a small increase in power.

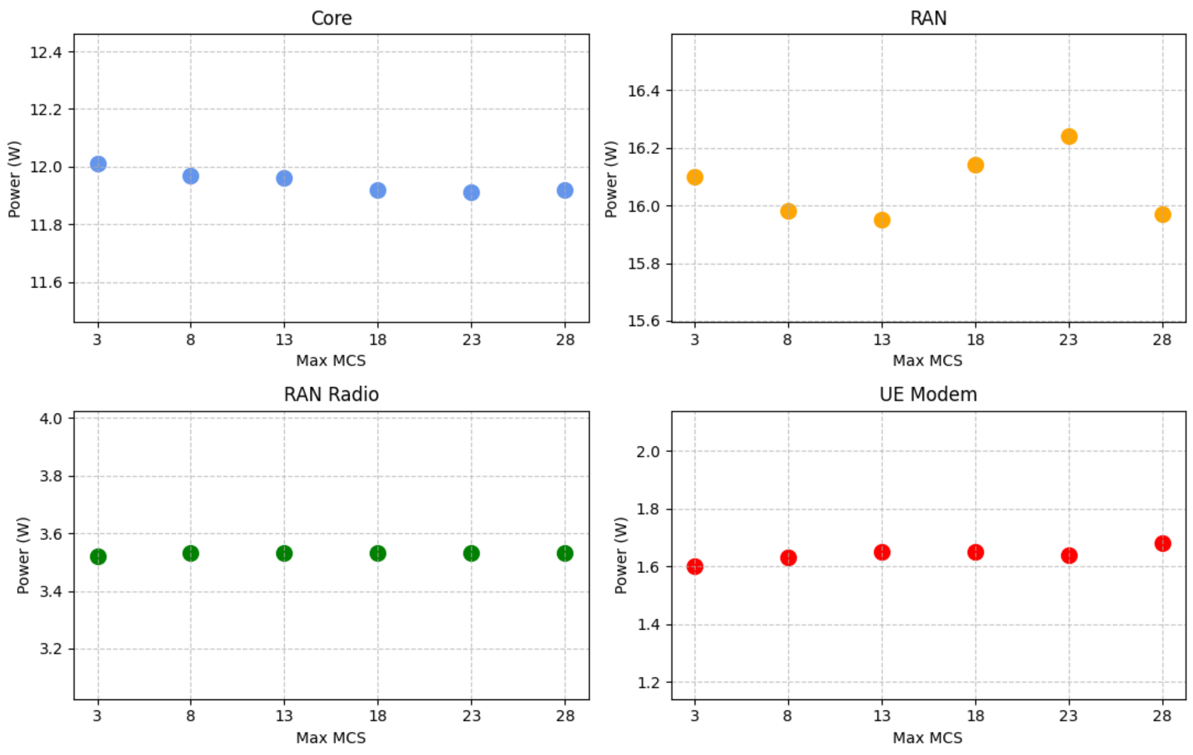

Figure 5.13: Power usage of the testbeds components for different MCS, 10 MHz DL

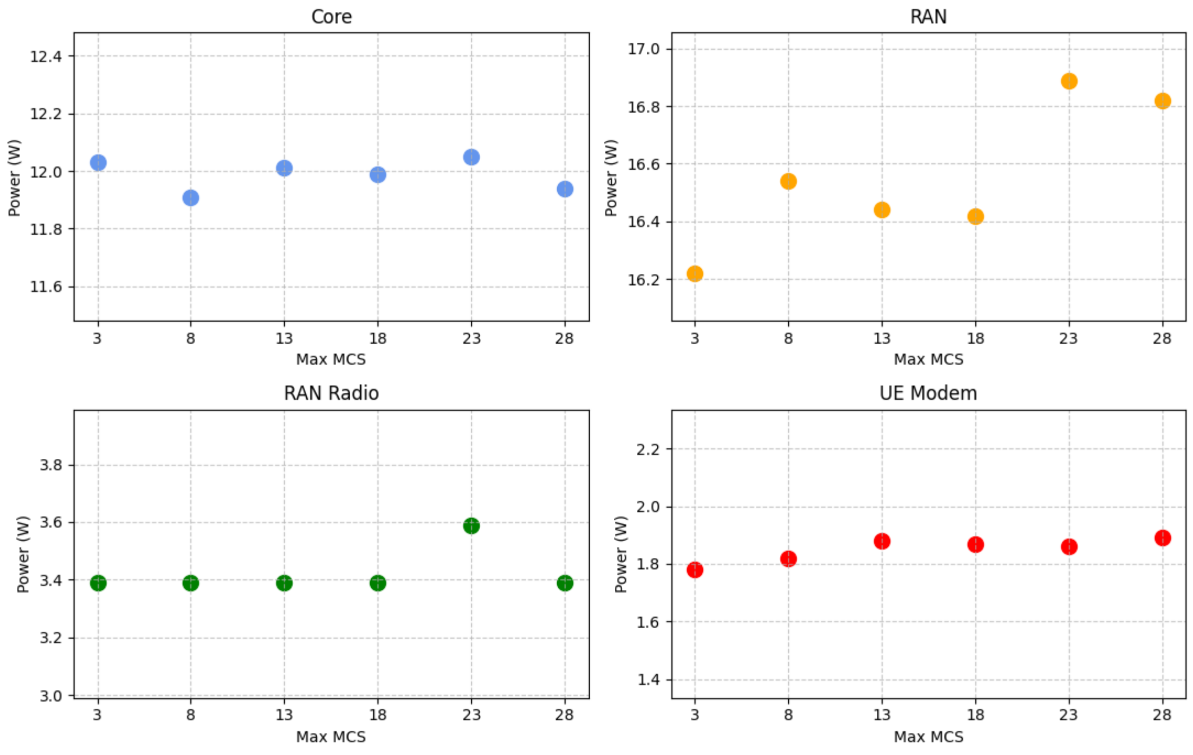

Figure 5.14: Power usage of the testbeds components for different MCS, 20 MHz DL

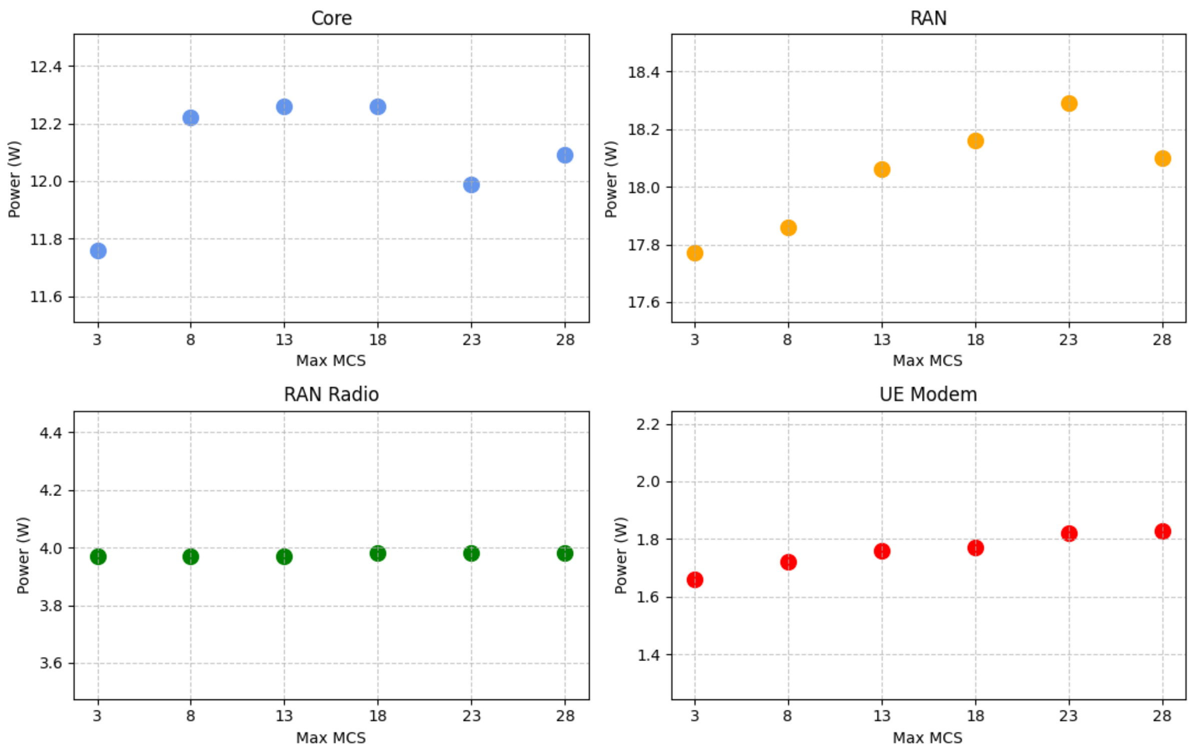

Figure 5.15: Power usage of the testbeds components for different MCS, 40 MHz DL

5.4 Summary and key insights

The experiments conducted in this chapter provide insights into the factors influencing power consumption in a 5G testbed environment. Below is a summary of the findings categorized by the key parameters investigated.

- **Influence of CPU governor:** Selection of the CPU governor significantly impacts the power consumption of the RAN CPU. The $ondemand$ governor demonstrated the most efficient power usage while maintaining acceptable throughput performance. Compared to the $performance$ governor, it reduced power consumption by up to 74% in the DL direction and 74.3% in the UL, with minimal degradation in throughput performance. The $conservative$ governor, while less power efficient than $ondemand$, also showed notable power savings compared to $performance$. However, the variability and precision of measurements suggest that $ondemand$ is the optimal choice for experiments involving dynamic scaling.
- **Impact of bandwidth:** The bandwidth experiments revealed a general trend of increasing power consumption for the RAN CPU and RAN radio with higher channel widths. The UE modem exhibited a slight increase in power usage for higher bandwidths, particularly in the UL tests, while the core CPU showed minimal changes across configurations. These results suggest that bandwidth expansion imposes higher processing demands on the RAN and UE, although the core network remains relatively unaffected.
- **Impact of throughput:** Throughput experiments highlighted a steady increase in power consumption for the RAN CPU and UE modem as user traffic increased, up to the maximum achievable throughput. Beyond this point, power consumption plateaued, indicating that components reach a steady-state power draw once fully utilized. The RAN radio maintained nearly constant power due to its continuously enabled transmitter, regardless of throughput, while the core CPU exhibited no consistent relationship with traffic levels.
- **Impact of MCS:** Higher MCS values, which increase the spectral efficiency of data transmission, resulted in a corresponding increase in power consumption for the RAN CPU and UE modem. However, the precision of measurements in these experiments was lower because of the need for normal mode operation in the testbed script. The RAN radio and core CPU showed negligible changes, reinforcing that MCS primarily affects components involved in direct data processing and transmission.

The experiments demonstrate the nuanced interaction between network parameters and power consumption. A summary of the key insights gained from the experiments:

- The $ondemand$ governor is ideal for balancing power efficiency and performance in testbed experiments.
- Bandwidth and throughput have the most significant impact on power consumption for components involved in data processing, particularly the RAN CPU and UE modem.
- The power consumption of the RAN radio remains largely constant due to continuous transmission settings, emphasizing the need for alternative configurations to study dynamic transmitter power behavior.
- The core network exhibits minimal sensitivity to PHY-layer variations, indicating that its power consumption is based on consistent management tasks.

6 CONCLUSION

This chapter concludes the work that was done for this thesis. A summary of the work is given, followed by a discussion of the results. The limitations of the current testbed will then be discussed, and future work to extend the possibilities and insights of the testbed is discussed.

6.1 Summary

This thesis represents the first research conducted on the newly established 5G energy efficiency testbed of the EDGE Research Centre of the University of Twente. The research goal is summarized by the main research question: *“How can a 5G testbed be designed to enable systematic and reproducible tests for 5G energy research?”*. To answer this question, first background research was done on the relevant topics.

After this, the OAI project was investigated and a testbed was set up to answer subquestion (1): *“How can a fully functioning and widely configurable 5G network be realized using OpenAirInterface?”*. 10, 20 and 40 MHz bandwidths were realized on the testbed. Both an SDR-based UE and COTS UE are supported, with the COTS UE having better overall performance.

To answer subquestion (2): *“How can energy consumption data be measured on a multi-component 5G testbed?”* literature research on other, similar setups was done. It was decided to use software-based CPU power measurements for the computers and a lab power meter for the radios. To ensure consistent and repeatable testing, the entire process was automated using Python scripts to achieve easy data collection and analysis. Experiments were performed to validate the precision of the testbed, as well as to find the best RAN CPU governor settings.

Through a series of experiments, this work demonstrated the current capabilities of the testbed for investigating energy consumption across all main components of a 5G network. This was done to address subquestion (3): *“What is the impact of bandwidth, traffic load, and the modulation and coding scheme on the energy usage of a 5G network?”*

In summary, this thesis developed a robust 5G energy testbed that enables systematic and reproducible testing. The work contributes to sustainable mobile networking research by providing a flexible platform for further research on energy efficiency in 5G networks at the University of Twente.

6.2 Discussion of results

For the RAN computer, the recommended *performance* governor for OAI uses constant 65.0 W power and is not usable for power research, so the *ondemand* and *conservative* governor were both tested. The dynamic scheduler causes a decrease in throughput performance, especially on the UL.

The repeatability support of the testbed was tested by conducting a precision experiment. It was found that the range of the measurements done on a single repeated test was very dependent on the CPU governor used for the RAN CPU. The most precise results for the RAN CPU measurements were found to be with the *ondemand* governor. Using the quickmode of the test script further improved the precision. However, this mode cannot be used with all tests, as the RAN needs to be restarted to change certain parameters.

The effect of this can be seen when comparing two of the power experiments: the throughput experiment and the MCS experiment. The throughput experiment could be performed in quickmode, giving clear results in which the measurements followed the expected increase in power as throughput increased. For the MCS experiments, normal mode was used. The results had a much higher variance from the expected trend line.

In general, the individual results confirm the anticipated relationships between the variable parameter and the power consumption of the testbed components. The RAN CPU measurements show clear relations between the amount of bandwidth and the traffic used. The same can be said about the UE modem. The core CPU measurements did not exhibit significant variations. This is likely due to the experiments mainly influencing the PHY layer, where the core network does not have different task loads depending on the tested parameters. The RAN radio shows no or very little increase in power because the radio TX is always on.

One of the requirements is that the testbed should be easy to use. The testbed script makes it possible to run a script and see the statistics on the measurements immediately after the experiment. The script did not fail a single time during the experiment phase.

6.3 Limitations

Firstly, the biggest current limitation of the testbed is the limited performance of the 5G connection. The theoretical throughput, especially that of the UL, is not achieved by most configurations: even after long extensive manual tuning of radio parameters, the scheduler did not manage to use maximum MCS for these configurations. The limited performance has influenced the measurements, as it is expected that an actual 100% throughput is expected to have resulted in higher power. The dynamic CPU governors also negatively influence throughput performance.

Although the testbed is precise enough to detect trends, especially in quickmode, the precision is not very high when the range of measurements is compared to the range of all measurements.

On a higher level, the testbed provides only a limited representation of a real 5G network, as it simplifies certain aspects such as the number of connected devices, real-world traffic patterns, mobility, the wireless medium and the variety of network configurations. This limitation is due to the use of OAI, which is still a software-based implementation under active development.

The testbed script currently does a lot of automating for a single experiment, however, the data storing and processing could be increased. The data from each individual test still have to be manually copied to a plotting script to plot data from multiple experiments, and individual tests are not grouped into one folder.

6.4 Future work

- Performance can still be improved, especially on the UL. Fine-tuning the radio power parameters seemed to have a significant influence on throughput. Research on the radio link, for example, by investigating the radio power levels more in-dept, could shine a light on why the throughput is not ideal.

- If the range of power measurements was larger, the proportional precision of the measurements would improve, enabling more accurate assessments of power trends and variations. This could be achieved by increasing the range of experiments, such as adding more UEs to the testbed or conducting tests with larger bandwidth configurations. These adjustments would enhance the overall usefulness of the testbed.

- The efficiency of the RAN is very dependent on the implementation of the OAI RAN software. In future research, the focus could be more on the implementation of certain aspects of the OAI code. In addition to this, in the LTE version of OAI it is possible to analyze the resource usage of individual processes of the OAI RAN. If this feature would be implemented for the 5G version, it could be a great tool for a implementation-level research on software-defined RAN.
- Core measurements will be more interesting when performing higher-level experiments where the network management plays a role. Extending the network to have more UEs, RANs, or operators will make power research on the Core more interesting, since it will need to perform more management tasks.
- OAI has interfaces to use FlexRIC, which is an open source RAN intelligent controller. This software can be used to deploy xApps in the network, which could be a great way to add features to the testbed, such as runtime control of the RAN parameters and logging features.
- Currently, the data of each experiment is stored in a separate folder. Storing measurements in a database could greatly improve workflow, as the data can be more easily grouped and retrieved for processing after the experiment has already been executed.

6.5 Closing remark

Throughout my research, a recurring theme emerged: for every challenge that needed to be researched, two new questions would arise. 5G mobile networks are very complex, and while I tried to keep the research as focused and manageable as possible, it was often necessary to learn about other aspects of 5G to fully understand what was going on. Although I wanted to continue investigating more features and learning more, I needed to make choices due to the available time.

OAI has a lot of functionality but is still in very active development to expand its features. It will be very interesting to follow the releases, and add more functionality to the testbed as OAI evolves.

I am proud to have been the first person to work on realizing functioning 5G network available at the University of Twente, and I believe this project holds a lot of potential for both functional and power-related 5G research.

REFERENCES

[1] J. Malmodin, N. Lövehagen, P. Bergmark, and D. Lundén, “ICT Sector Electricity Consumption and Greenhouse Gas Emissions – 2020 Outcome,” *SSRN Electronic Journal*, 4 2023. [Online]. Available: https://papers.ssrn.com/abstract=4424264

[2] A. Fehske, G. Fettweis, J. Malmodin, and G. Biczok, “The global footprint of mobile communications: The ecological and economic perspective,” *IEEE Communications Magazine*, vol. 49, no. 8, pp. 55–62, 8 2011.

[3] H. Pihkola, M. Hongisto, O. Apilo, and M. Lasanen, “Evaluating the Energy Consumption of Mobile Data Transfer—From Technology Development to Consumer Behaviour and Life Cycle Thinking,” *Sustainability 2018, Vol. 10, Page 2494*, vol. 10, no. 7, p. 2494, 7 2018. [Online]. Available: https://www.mdpi.com/2071-1050/10/7/2494/htmhttps://www.mdpi.com/2071-1050/10/7/2494

[4] “Mobile data traffic forecast – Mobility Report - Ericsson.” [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast

[5] P. S. Upadhyaya, A. S. Abdalla, V. Marojevic, J. H. Reed, and V. K. Shah, “Prototyping Next-Generation O-RAN Research Testbeds with SDRs,” 5 2022.

[6] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, “5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view,” *IEEE Access*, vol. 6, pp. 55 765–55 779, 2018.

[7] G. Brown, “Service-based architecture for 5g core networks,” Heavy Reading, Tech. Rep., November 2017, produced for Huawei Technologies Co. Ltd. [Online]. Available: https://www-file.huawei.com/-/media/corporate/pdf/white%20paper/heavy-reading-whitepaper--service-oriented-5g-core-networks.pdf

[8] 3GPP, “Specification 23.501.” [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144

[9] A. I. Abubakar, O. Onireti, Y. Sambo, L. Zhang, G. K. Ragesh, and M. Ali Imran, “Energy Efficiency of Open Radio Access Network: A Survey,” *IEEE Vehicular Technology Conference*, vol. 2023-June, 2023.

[10] L. M. Larsen, A. Checko, and H. L. Christiansen, “A survey of the functional splits proposed for 5G mobile crosshaul networks,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 1, pp. 146–172, 1 2019. [Online]. Available: https://www.researchgate.net/publication/328033623_A_Survey_of_the_Functional_Splits_Proposed_for_5G_Mobile_Crosshaul_Networks

[11] M. Fiorani, S. Tombaz, J. Mårtensson, B. Skubic, L. Wosinska, and P. Monti, “Modeling energy performance of C-RAN with optical transport in 5G network scenarios,” *Journal of Optical Communications and Networking*, vol. 8, no. 11, 2016.

[12] R. Mijumbi, J. Serrat, J.-L. Gorricho, J. Rubio-Loyola, and S. Davy, “Server placement and assignment in virtualized radio access networks,” in *2015 11th International Conference on Network and Service Management (CNSM)*. IEEE, 11 2015, pp. 398–401.

[13] Eugina Jordan, “Open RAN functional splits, explained - 5G Technology World.” [Online]. Available: https://www.5gtechnologyworld.com/open-ran-functional-splits-explained/

[14] M. Polese, L. Bonati, S. D’Oro, S. Basagni, and T. Melodia, “Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges,” *IEEE Communications Surveys and Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[15] O-RAN Alliance, “O-RAN documentation.” [Online]. Available: https://docs.o-ran-sc.org/en/latest/architecture/architecture.html

[16] 3GPP, “Specification 23.214.” [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3216

[17] 3GPP, “Specification 38.211.” [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3213

[18] D. Demmer, R. Gerzaguet, J. B. Dore, and D. Le Ruyet, “Analytical study of 5G NR eMBB co-existence,” *2018 25th International Conference on Telecommunications, ICT 2018*, pp. 186–190, 9 2018.

[19] 3GPP, “Specification 38.101.” [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3283

[20] Jaeku Ryu, “5G/NR - tdd UL/DL Common Configuration.” [Online]. Available: https://www.sharetechnote.com/html/5G/5G_tdd_UL_DL_configurationCommon.html

[21] 3GPP, “Specification 38.306.” [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3193

[22] 5G-tools.com. [Online]. Available: https://5g-tools.com/5g-nr-throughput-calculator/

[23] Z. Wang, “Slow wireless communication testbed based on software-defined radio,” Master’s Thesis, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science, Enschede, The Netherlands, November 2017.

[24] F. K. Jondral, “Software-defined radio - Basics and evolution to cognitive radio,” *Eurasip Journal on Wireless Communications and Networking*, vol. 2005, no. 3, pp. 275–283, 8 2005. [Online]. Available: https://jwcn-eurasipjournals.springeropen.com/articles/10.1155/WCN.2005.275

[25] J. A. Ayala-Romero, I. Khalid, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, “Experimental Evaluation of Power Consumption in Virtualized Base Stations,” *IEEE International Conference on Communications*, 6 2021.

[26] L. Williams, B. K. Sovacool, and T. J. Foxon, “The energy use implications of 5G: Reviewing whole network operational energy, embodied energy, and indirect effects,” *Renewable and Sustainable Energy Reviews*, vol. 157, p. 112033, 4 2022.

[27] OpenAirInterface Software Alliance, “Openairinterface 5g ran repository.” [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g

[28] OpenAirInterface Software Alliance , “OpenAirInterface 5G Core Network repository.” [Online]. Available: https://gitlab.eurecom.fr/oai/cn5g

[29] Open Cells. [Online]. Available: https://open-cells.com/

[30] OpenAirInterface Software Alliance, “OpenAirInterface Feature Set.” [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/doc/FEATURE_SET.md

[31] OpenAirInterface Software Alliance , “OpenAirInterface T Tracer.” [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/common/utils/T/DOC/T.md

[32] OpenAirInterface Software Alliance , “Wireshark with OpenAirInterface.” [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/common/utils/T/DOC/T/wireshark.md

[33] OpenAirInterface Software Alliance, “Openairinterface telnet server.” [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/common/utils/telnetsrv/DOC/telnetsrv.md

[34] OpenAirInterface, “.” [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/common/utils/telnetsrv/DOC/telnetmeasur.md

[35] Mosaic5G, “FlexRIC.” [Online]. Available: https://gitlab.eurecom.fr/mosaic5g/flexric

[36] M. Seidel, A. I. Grohmann, P. Sossalla, F. Kaltenberger, and F. H. 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)*. IEEE, 7 2023, pp. 1–6.

[37] Q. Douarre, E.-M. Djelloul, P. Berthou, D. Dragomirescu, and P. Owezarski, “Design of a 5G experimental platform based on OpenAirInterface,” p. 3855794, 2022. [Online]. Available: https://laas.hal.science/hal-03855794

[38] OpenAirInterface Software Alliance, “Openairinterface continuous-tx merge request.” [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/-/merge_requests/1439

[39] Reiner Ludwig, “Who cares about latency in 5G?” [Online]. Available: https://www.ericsson.com/en/blog/2022/8/who-cares-about-latency-in-5g

[40] D. Lopez-Perez, A. De Domenico, N. Piovesan, G. Xinli, H. Bao, S. Qitao, and M. Debbah, “A Survey on 5G Radio Access Network Energy Efficiency: Massive MIMO, Lean Carrier Design, Sleep Modes, and Machine Learning,” *IEEE Communications Surveys and Tutorials*, vol. 24, no. 1, pp. 653–697, 2022.

[41] “srsRAN Project website.” [Online]. Available: https://www.srslte.com/

[42] J. X. Salvat, J. A. Ayala-Romero, L. Zanzi, A. Garcia-Saavedra, and X. Costa-Perez, “Open Radio Access Networks (O-RAN) Experimentation Platform: Design and Datasets,” *IEEE Communications Magazine*, vol. 61, no. 9, pp. 138–144, 9 2023.

[43] U. Pawar, B. R. Tamma, and F. A. Antony, “Traffic-Aware Compute Resource Tuning for Energy Efficient Cloud RANs,” *Proceedings - IEEE Global Communications Conference, GLOBECOM*, 2021.

[44] L. Brown, “Turbostat.” [Online]. Available: https://www.linux.org/docs/man8/turbostat.html

[45] K. Nizam Khan, M. Hirki, T. Niemi, J. K. Nurminen, Z. Ou, M. Hirki, T. Niemi, J. K. Nurminen, Z. Ou, and K. N. Khan, “RAPL in Action: Experiences in Using RAPL for Power Measurements,” *ACM Trans. Model. Perform. Eval. Comput. Syst*, vol. 3, no. 9, 2018. [Online]. Available: https://doi.org/10.1145/3177754

[46] M. Jay, V. Ostapenco, L. Lefèvre, D. Trystram, A.-C. Orgerie, and B. Fichel, “An experimental comparison of software-based power meters: focus on CPU and GPU.” [Online]. Available: https://inria.hal.science/hal-04030223v2

[47] Linux Kernel Organization, “Linux CPU Governor Documentation.” [Online]. Available: https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt

[48] Y. D. Staal, “5G Power Testbed Repository.” [Online]. Available: https://gitlab.utwente.nl/s2185849/5g-power-testbed

[49] J. Forcier, “Paramiko.” [Online]. Available: https://www.paramiko.org/

[50] G. T. Torsten Bronger, “Pyvisa.” [Online]. Available: https://pyvisa.readthedocs.io/en/latest/