UNIVERSITY OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

DEGREE PROGRAMME IN ELECTRONICS AND COMMUNICATIONS ENGINEERING

MASTER’S THESIS

UPLINK DATA MEASUREMENT AND ANALYSIS FOR 5G eCPRI RADIO UNIT

Author	Harri Hämäläinen
Supervisor	Ari Pouttu
Second Examiner	Ville Niemelä
(Technical Advisor	Keijo Nikula)

October 2020

Hämäläinen H. (2020) Uplink Data Measurement and Analysis for 5G eCPRI Radio Unit. University of Oulu, Faculty of Information Technology and Electrical Engineering, Degree Programme in Electronics and Communications Engineering. Master’s Thesis, 80 p.

ABSTRACT

The new 5G mobile network generation aims to enhance the performance of the cellular network in almost every possible aspect, offering higher data rates, lower latencies, and massive number of network connections. Arguably the most important change from LTE are the new RU-BBU split options for 5G promoted by 3GPP and other organizations. Another big conceptual shift introduced with 5G is the open RAN concept, pushed forward by organizations such as the O-RAN alliance. O-RAN aims to standardize the interfaces between different RAN elements in a way that promotes vendor interoperability and lowers the entry barrier for new equipment suppliers. Moreover, the 7-2x split option standardized by O-RAN has risen as the most important option within the different low layer split options. As the fronthaul interface, O-RAN has selected the packet-based eCPRI protocol, which has been designed to be more flexible and dynamic in terms of transport network and data-rates compared to its predecessor CPRI. Due to being a new interface, tools to analyse data from this interface are lacking.

In this thesis, a new, Python-based data analysis tool for UL eCPRI data was created for data quality validation purposes from any O-RAN 7-2x functional split based 5G eCPRI radio unit. The main goal for this was to provide concrete KPIs from captured data, including timing offset, signal power level and error vector magnitude. The tool produces visual and text-based outputs that can be used in both manual and automated testing. The tool has enhanced eCPRI UL datapath testing in radio unit integration teams by providing actual quality metrics and enabling test automation.

Key words: 5G, eCPRI, O-RAN

Hämäläinen H. (2020) Uplink datamittaukset ja -analyysi 5G eCPRI radiolla. Oulun yliopisto, tieto- ja sähkötekniikan tiedekunta, elektroniikan ja tietoliikennetekniikan tutkinto-ohjelma. Diplomityö, 80 p.

TIIVISTELMÄ

Uusi 5G mobiiliverkkogeneraatio tuo mukanaan parannuksia lähes kaikkiin mobiiliverkon ominaisuuksiin, tarjoten nopeamman datasiirron, pienemmät viiveet ja valtavat laiteverkostot. Luultavasti tärkein muutos LTE teknologiasta ovat 3GPP:n ja muiden organisaatioiden ehdottamat uudet radion ja systeemimoduulin väliset funktionaaliset jakovaihtoehdot. Toinen huomattava muutos 5G:ssä on O-RAN:in ajama avoimen RAN:in konsepti, jonka tarkoituksena on standardisoida verkkolaitteiden väliset rajapinnat niin, että RAN voidaan rakentaa eri valmistajien laitteista, laskien uusien laitevalmistajien kynnystä astua verkkolaitemarkkinoille. O-RAN:n standardisoima 7-2x funktionaalinen jako on noussut tärkeimmäksi alemman tason jakovaihtoehdoista. Fronthaul rajapinnan protokollaksi O-RAN on valinnut pakettitiedonsiirtoon perustuvan eCPRI:n, joka on suunniteltu dynaamisemmaksi ja joustavammaksi datanopeuksien ja lähetysverkon suhteen kuin edeltävä CPRI protokolla. Uutena protokollana, eCPRI rajapinnalle soveltuvia data-analyysityökaluja ei ole juurikaan saatavilla.

Tässä työssä luotiin uusi pythonpohjainen data-analyysityökalu UL suunnan eCPRI datalle, jotta datan laatu voidaan määrittää millä tahansa O-RAN 7-2x funktionaaliseen jakoon perustuvalla 5G eCPRI radiolla. Työkalun päätarkoitus on analysoida ja kuvata datan laatua laskemalla datan ajoitusoffsettia, tehotasoa, sekä EVM:ää. Työkalu tuottaa tulokset visuaalisena ja tekstipohjaisena, jotta analyysia voidaan tehdä niin manuaalisessa kuin automaattisessa testauksessa. Työkalun käyttöönotto on tehostanut UL suunnan dataputken testausta radio-integrointitiimeissä, tarjoten datan laatua kuvaavaa metriikkaa sekä mahdollistaen testauksen automatisoinnin.

Avainsanat: 5G, eCPRI, O-RAN.

TABLE OF CONTENTS

FOREWORD

This Master's thesis was done for Nokia at Rusko, Oulu. There are many people that have contributed in some way or another in the completion of this thesis and my degree, so firstly I would like to thank everyone that has supported or helped me during my studies.

My biggest gratitude goes to my managers at Nokia, Toni Siponkoski and Antti Moilanen, who took me under their wings in radio unit integration and provided me the topic of this thesis. Both have fully believed in me during my time working for them, and the technical and motivational support from them has been invaluable. I would also like to thank my technical supervisor Keijo Nikula for providing prompt feedback for the thesis. Moreover, I am thankful for all the colleagues who have helped me during my time at Nokia.

At the university of Oulu, I would like to thank Ari Pouttu for being the main supervisor of this thesis, and Ville Niemelä for providing extensive guidance and feedback. Also, I am thankful to all personnel at CWC who has guided and helped me during my studies.

I would like to thank all my friends, especially those who I have had the pleasure to study with. The countless hours we have put our heads together studying (and not studying) was the main reason I was able to finish my degree.

Finally, I would like to thank my family for continuously supporting me during my studies. Special gratitude goes to my girlfriend Jemina for her continuous love and support.

Oulu, October 11, 2020

Harri Hämäläinen

LIST OF ABBREVIATIONS AND SYMBOLS

3GPP	3rd Generation Partnership Project
5GC	5G Core
5GS	5G System
ADC	Analog-to-Digital Converter
AMF	Access and Mobility Management Function
API	Application Programming Interface
ARQ	Automatic Repeat Request
BBU	Baseband Unit
BFP	Block Floating Point
BW	Bandwidth
CAZAC	Constant Amplitude Zero Autocorrelation
CLI	Command Line Interface
CP	Cyclic Prefix
C-Plane	Control Plane
CPRI	Common Public Radio Interface
C-RAN	Cloud RAN
CUPS	Control and User Plane Seperation
DAC	Digital-to-Analog Conversion
DFT	Discrete Fourier Transform
DL	Downlink
DMRS	Demodulation Reference Symbol
D-RAN	Distributed RAN
eCPRI	Evolved CPRI
eMBB	Enhanced Mobile Broadband
eNB	Evolved NodeB
en-gNB	Enhanced Next-Generation NodeB
EPC	Evolved Packet Core
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
EVM	Error Vector Magnitude
FD	Frequency Domain
FFT	Fast Fourier Transform
FR1	Frequency Range 1
FR2	Frequency Range 2
gNB	Next-Generation NodeB
HARQ	Hybrid ARQ
ICI	Intercarrier Interference
iFFT	Inverse FFT
IoT	Internet of Things
IP	Internet Protocol
IQ	In-phase and Quadrature
ISI	Intersymbol Interface
ITU	International Telecommunication Union
IWF	Interworking Functions
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LAN	Local Area Network

LMI	Local Management Interface
LO	Local Oscillator
LTE	Long Term Evolution
MAC	Media Access Control
MCR	MATLAB Compiler Runtime
MIMO	Multiple-Input Multiple-Output
mMIMO	Massive MIMO
mMTC	Massive Machine Type Communications
mmW	Millimeter Wave
M-Plane	Management Plane
MSB	Most Significant Bit
MU-MIMO	Multi-User MIMO
NAS	Non-Access Stratum
NF	Network Function
NG-RAN	Next Generation RAN
NR	New Radio
NSA	Non-Standalone
OAM	Operations, Administration and Maintenance
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
O-RAN	Open RAN
PAPR	Peak-to-Average Power Ratio
PCAP	Packet Capture
PDCP	Packet Data Convergence Protocol
PDU	Protocol Data Unit
PHY	Physical Layer
PNG	Portable Network Graphics
PRACH	Physical Random-Access Channel
PRB	Physical Resource Block
PTP	Precision Time Protocol
PTRS	Phase-tracking Reference signal
PUSCH	Physical Uplink Shared Channel
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technology
RB	Resource Block
RE	Resource Element
RF	Radio Frequency
RLC	Radio Link Control
RoE	Radio over Ethernet
RRC	Radio Resource Control
RU	Radio Unit
SA	Standalone
SBA	Service Based Architecture
SC-FDMA	Single-Carrier Frequency Division Multiple Access
SCS	Subcarrier Spacing
SDAP	Service Data Adaptation Protocol

SISO	Single-Input Single-Output
S-Plane	Synchronization Plane
SRS	Sounding Reference Signal
SU-MIMO	Single-User MIMO
TD	Time Domain
TDD	Time Division Duplexing
TDM	Time Division Multiplexing
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink
U-Plane	User Plane
URLLC	Ultra-Reliable and Low Latency Communications
VNF	Virtual NF
VR	Virtual Reality

bps	bits per second
dB	decibel
$dBFS$	decibels related to full scale
dBm	decibels with reference to a milliwatt

φ_{off}	carrier phase offset
BW_c	control data bandwidth
F_s	sampling frequency
G_{UL}	uplink power gain
$G_{corr(dB)}$	gain correction in dB
$G_{ref(dBm)}$	reference gain in dBm
I_{MAC}	amount of information required by MAC
L_C	component/cable loss
N_a	number of antenna elements
N_{err}	integer timing offset in samples
N_{frac}	fractional timing offset in samples
$N_{l,c}$	number of control data streams
N_l	number of data streams
N_{sc}	number of subcarriers
N_{sym}	number of symbols per sample
P_{dBFS}	power in dBFS
P_t	transmit power
R_b	bitrate
$R_{interface(dBFS)}$	smallest subcarrier power in dBFS
R_{peak}	peak data rate
$S_r(n)$	received constellation vector
$S_{ref}(n)$	reference constellation vector
X_k	k:th OFDM symbol
b_s	bits per sample
$s_r(n)$	received sequence
$s_{ref}(n)$	reference/transmitted sequence
x_n	n:th time domain sample

BW	bandwidth
OH	overhead
FS	full-scale reference power
sig	signalling

1 INTRODUCTION

Over the rather short 40-year history, mobile networks have undergone several advances and innovations. The evolution cycle of mobile network generations has traditionally been about 10 years, and each new generation has had immense expectations to improve performance of the network. The data rate progression of each generation is approximated in Figure 1 on a logarithmic scale. The first mobile network generation, 1G, built the base for what the mobile communications are today, offering the first wireless analogue voice services. 2G improved efficiency by making the jump from analogue to digital voice calls, while also introducing text-based messaging. With 3G came internet access and first multimedia applications, bringing in services such as email and video calling. 4G, conventionally known as Long Term Evolution (LTE), immensely enhanced the data services of 3G by reducing latency and improving data-rates, enabling easy access to services such as video streaming, online gaming, and instant messaging. The new 5G technology is expected to serve a growing number of users, services, and applications, ranging from self-driving cars to massive sensor networks. When just over 10 years ago the maximum data rate offered by the 3G technology was in order of megabits per second, 5G is expected to offer speeds over multiple Gbps. [1, 2, 3]

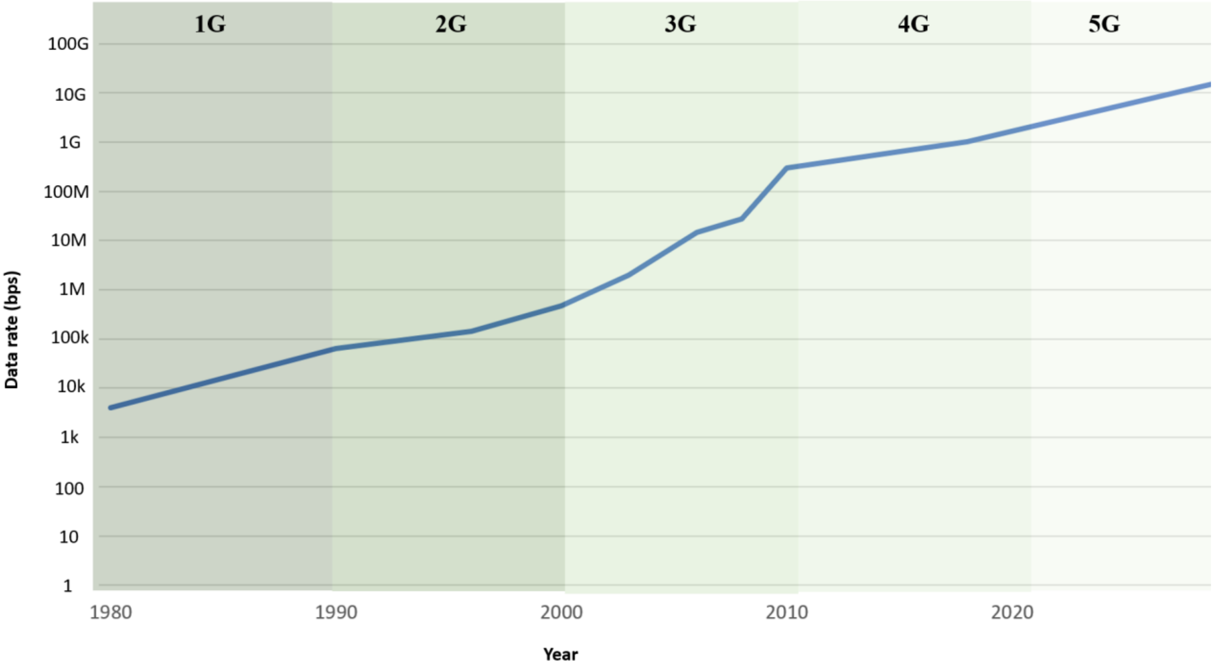


Figure 1. Data rate progression of current mobile network generations and future expectations for 5G on a logarithmic scale.

Today, mobile devices are more integral part of our daily lives than ever before. The advancements made in the cellular network technologies have enabled services, traditionally ran on PCs, to be ran directly on the mobile devices. Naturally, as the number of users and requirements of applications have risen, the evolution of mobile networks has followed to provide the needed data-rates and reliability. In addition to constantly growing human userbase in mobile networks, 5G is also expected to serve the fast-developing machine-to-machine market. It has been estimated that the mobile data traffic will grow about 300% between 2020 and 2025, as seen in Figure 2. By 2025, 5G networks are expected to handle about 45% of the total data traffic in cellular networks. [4]

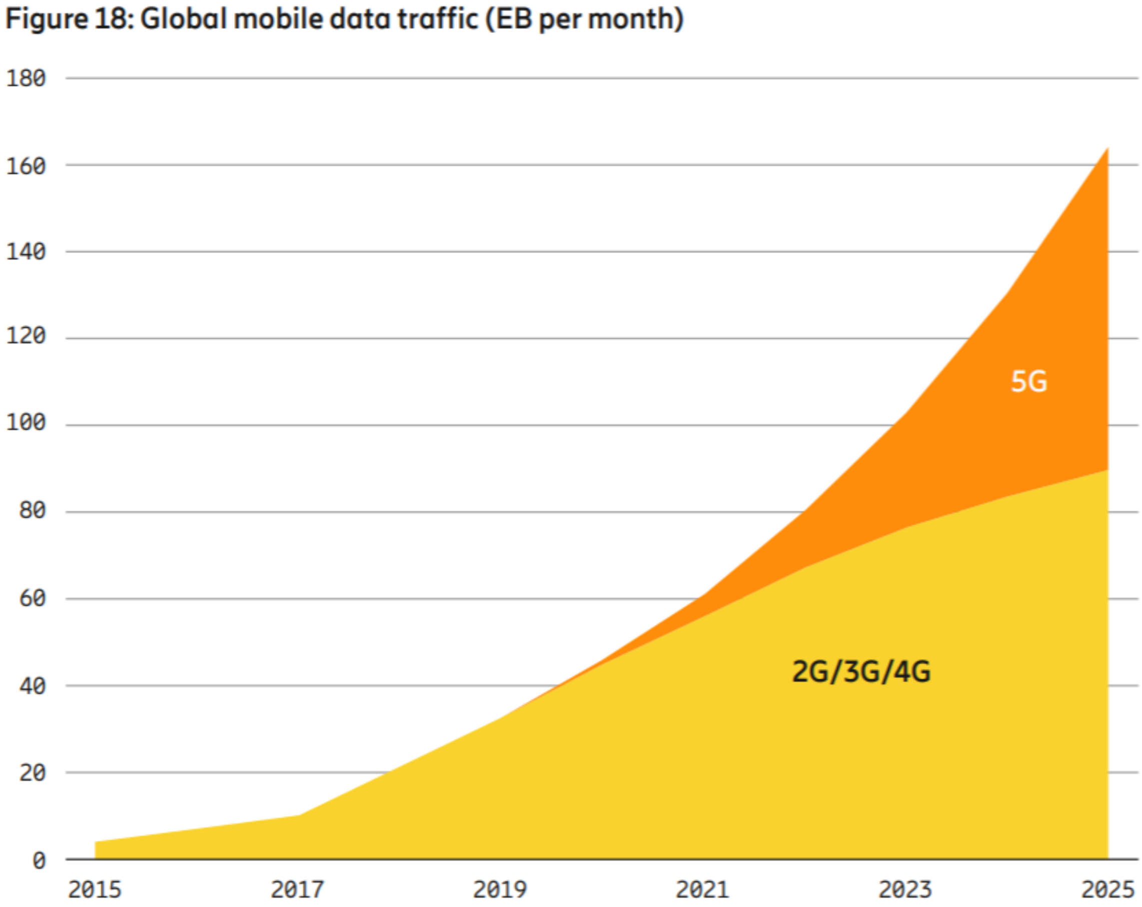


Figure 2. Mobile data traffic estimation for 5G and older technologies.

Developing new technology is never simple, especially for mobile networks where each new generation is expected to top the previous technology in all aspects imaginable. Advancements are expected to be made in both hardware and software level and taking new techniques and components into use brings challenges to all stages of development. Many times, proper tools to test these new, innovate features are not available straight away, further lingering the development process. This in turn increases the time-to-market, which is never desired in highly competitive field such as the telecommunication equipment market.

When developing cellular network equipment, the end-to-end testing of the devices is usually done in the integration teams. The main responsibility of integrators is to test the whole device under testing and validate that the different software and hardware components work as intended. Usually, the beginning stage of integration is the radio unit integration, in which the radio is tested and validated independently: other parts of the network, such as the baseband unit (BBU) and user equipment (UE), are not used in this part of the integration. Shortly put, the radio unit integration aims to confirm that the data, which is generated artificially using signal generators and baseband emulators, goes through the radio as intended in both downlink (DL) and uplink (UL) direction, DL being the data flow direction from the network to the user, and UL from the user to the network.

5G brings a whole lot of new technological advancements to the table that improve the performance of the network. One of these is the Evolved Common Public Radio Interface (eCPRI), a new packet-based protocol between the radio unit (RU) and the baseband unit. In radio unit integration perspective, eCPRI brings new challenges to the testing process. Since the testing is done without a baseband unit, the DL eCPRI traffic must be generated using baseband emulators. Validation of DL data quality is relatively simple task since it can be done just by plugging in a radio frequency (RF) spectrum analyser to the antenna ports of the radio. However, analysing the data in UL is not as straightforward. While the baseband emulators used to generate eCPRI traffic in DL are also capable to capture eCPRI formatted data in UL, proper analysis tools are still lacking for this new interface at unit integration level. Moreover, most of the existing tools are designed for manual testing, lacking the support for test automation. Automation is a very important concept in modern product development, and especially so in radio unit integration where the testing of the radio becomes a labour-intensive task when the number of testable features starts getting high. The process of testing existing features after a new software release is called regression testing and is one of the most important tasks of integrators.

In this thesis, a new eCPRI UL data analysis tool is developed for both manual and automated data analysis and regression testing purposes. The main motivation for the tool is to provide analysis metrics for captured data, so that the UL performance of any Open RAN (O-RAN) based eCPRI radio can be evaluated clearly by means of manual and automated testing. Existing tools and their shortcomings are introduced, and the new tool is designed and implemented to overcome them, while introducing additional analysis features. To understand how the tool parses, handles and analyses eCPRI data, a necessary theoretical background for 5G is also given in this thesis.

The first part of the thesis gives a comprehensive overlook of the key technology and architectural enablers of 5G, especially focusing on the fronthaul and radio parts of the network. In the second part, a deeper dive is taken into the architecture of an O-RAN based eCPRI radio unit, and UL test setup and key performance and functional parameters are presented. The third chapter focuses on the existing eCPRI analysis tools for UL data, their pros and cons, and presents the arguments and requirements for the new tool. Finally, the tool is designed and implemented based on the given requirements.

2 5G ARCHITECHTURE AND TECHNOLOGY

Every new mobile network generation has had its own set of requirements to fulfil, based on the future user, application, and market trends. For 5G, these requirements are more demanding than ever. While the number of devices connected to mobile networks grows by hundreds of millions annually, areas such as data rate, latency, network coverage and mobility are expected to satisfy the growing needs of applications such as media streaming, Internet of Things (IoT), virtual reality (VR), and self-driving cars [5]. These requirements have been commonly generalized into three main use-cases by the International Telecommunication Union (ITU), as shown in Figure 3 [1]:

- The Enhanced Mobile Broadband (eMBB): Focuses on high data-rate and wide-area coverage. Includes applications such as 4K video streaming and VR.
- Massive Machine Type Communications (mMTC): Concentrates on connecting a large number of low data volume devices to the network. IoT applications like sensor networks of smart factories and smart cities fall into this use-case.
- Ultra-Reliable and Low Latency Communications (URLLC): Includes applications requiring low network latency, high reliability, and constant availability, i.e. self-driving cars and remote control of industrial devices and surgery equipment.

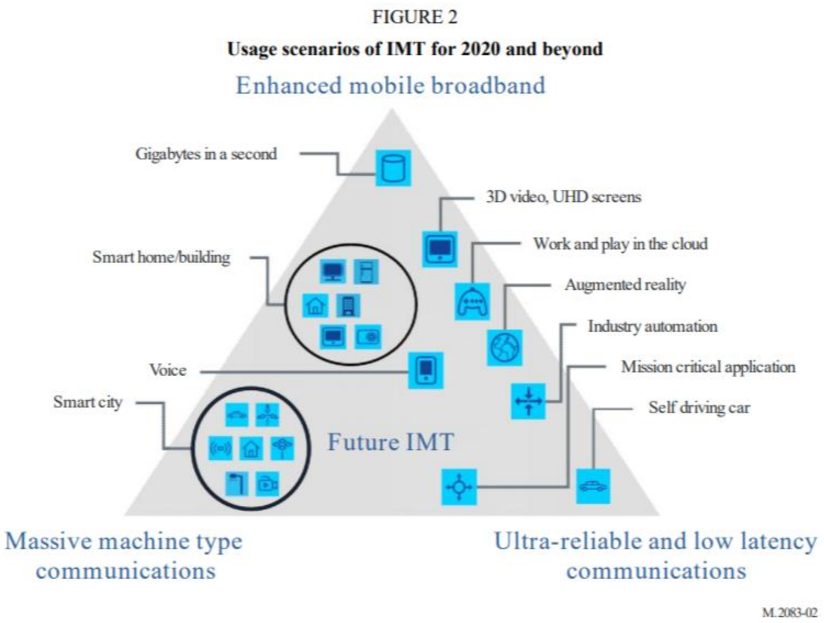


FIGURE 2
Usage scenarios of IMT for 2020 and beyond
Enhanced mobile broadband
Gigabytes in a second
3D video, UHD screens
Work and play in the cloud
Smart home/building
Augmented reality
Industry automation
Mission critical application
Voice
Self driving car
Smart city
Future IMT
Massive machine type communications
Ultra-reliable and low latency communications
M.2083-02

Figure 3. Use cases of 5G grouped into three main categories.

In early-5G, the most focused use-case of the technology has been eMBB since the speed of the network is the most obvious upgrade from previous technologies for the general public, making it the most profitable marketing point for operators and 5G network equipment vendors early on.

To achieve the requirements put up by the wide range of use-cases, multiple innovative technologies and improved architectural solutions have been taken into use with 5G. In this chapter, the key technologies and their enablers are discussed.

2.1 5G System Architecture

The basic structure of a traditional mobile network can be grouped roughly into two main components: The core network provides users access to different services, and the radio access network (RAN) provides the wireless connection between the core network and the UE. These networks are huge, consisting of hundreds of devices, and it makes no practical sense for operators to completely replace their existing networks with a new technology overnight. This is also true with the first 5G deployments, especially since most of the mobile devices don't even have 5G support yet. To allow 5G system (5GS) deployments happen alongside current Long Term Evolution (LTE) networks, the 3rd Generation Partnership Project (3GPP) has defined both standalone (SA) and non-standalone (NSA) deployment options for 5G. In completely Standalone (SA) deployment, the 5G network is built from ground-up to function without the need of any element of LTE network, consisting of the 5G Core Network (5GC) and the Next Generation Radio Access Network (NG-RAN) with 5G radio nodes (next-generation NodeB, gNB). In early phases of 5G, the SA deployment model is most suited for areas which have no current LTE coverage, due not needing any existing network structure. [6, 7]

The Non-Standalone (NSA) deployment models utilize new 5G network components and technology in conjunction with current elements of LTE. The earliest commercial 5G deployments will still utilize LTE's core network Evolved Packet Core (EPC) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN). The role of 5G capable radio nodes (enhanced next-generation NodeB, en-gNB) is to act as additional "capacity boost" alongside the existing LTE radio nodes (evolved NodeB, eNB). This NSA deployment model is shown in Figure 4. In general, the non-standalone deployment models enable operators to launch 5G on top of their current LTE networks, smoothing the transition to completely standalone 5G. [6, 7]

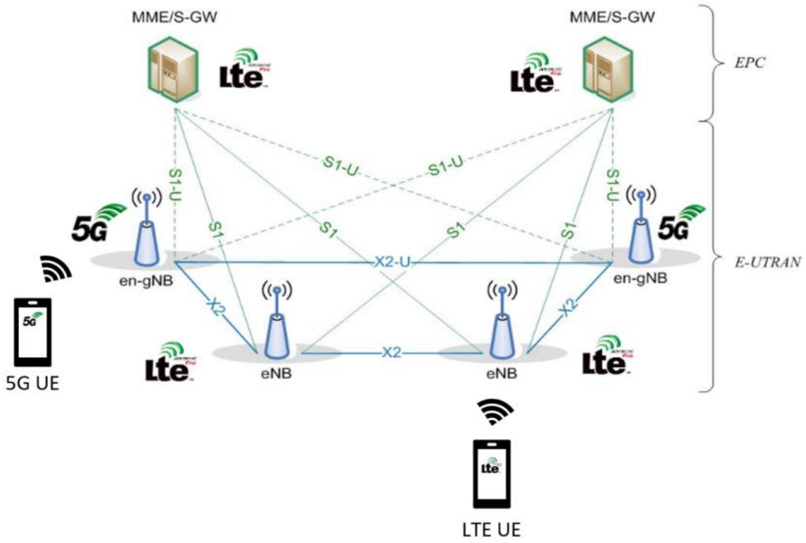


Figure 4. Non-standalone 5G system architecture (5G radio nodes in LTE network).

The mobile networks before 5G were mostly built statically, and the same network was expected to serve all users and use-cases equally. This "one-fits-all" design philosophy is obviously not sufficient for the diverse needs of different 5G use cases: the network needs to be designed with flexibility in mind. Therefore. 5GC is built based on the Service Based Architecture (SBA), which emphasizes modularity, interoperability, and reusability of Network Functions (NFs) by introducing common interfaces for service providers, as well as enabling virtualization and localized services by cloud native design. These are drastic improvements on scalability compared to LTE's EPC, which uses more fixed network entities. Another key concept of 5GC is the control plane and user plane processing separation. While the concept of Control and User Plane separation (CUPS) was already introduced in LTE as an optional extension, it is an integral part of 5G networks from the get-go. By separating UP functions from the core and moving them closer to the users, backhaul data traffic is reduced, thus making the user data processing faster and more efficient. SBA and CUPS also play a huge role in enabling network slicing, a method in which the physical and virtual elements of the end-to-end 5G network are grouped into sets (or slices), each fulfilling specialized needs of a certain service category. [6, 8]

2.2 Next Generation Radio Access Network

To establish internet or network connection in mobile networks, the UE must first connect to the radio access network (RAN). RAN communicates directly with the users using radio waves and acts as the gateway between the UE and network core. Traditionally, the RAN has consisted of two main nodes, the radio unit (RU) and the baseband unit (BBU), both located at the radio site. This sort of RAN architecture is known as distributed RAN (D-RAN), shown in Figure 5. For example, the first LTE deployments were done using D-RAN architecture. The connection between the RU and BBU is commonly known as the fronthaul connection, and the BBU is connected to the core network via the backhaul connection. [9]

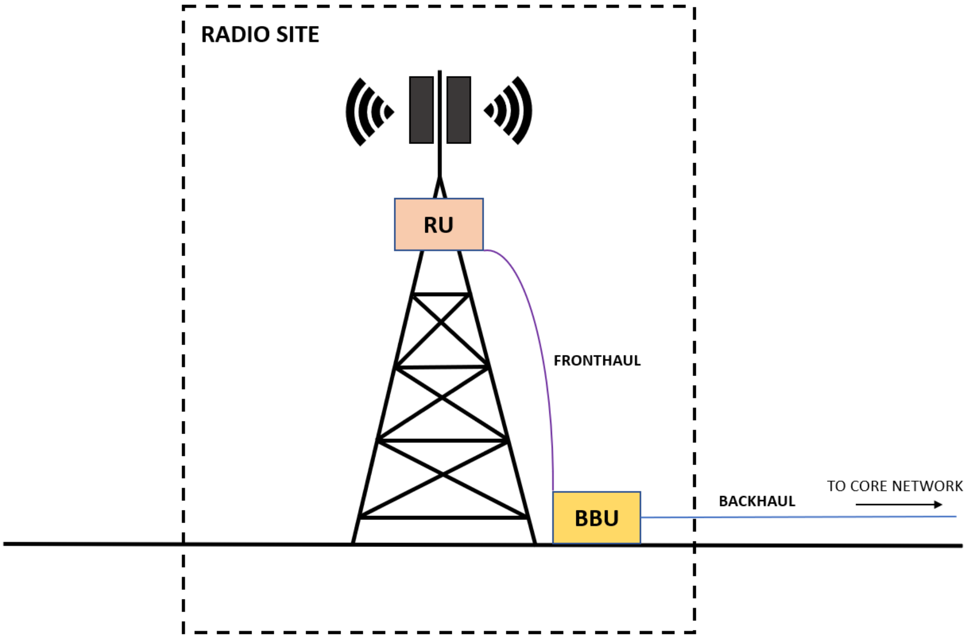


Figure 5. Distributed RAN architecture.

While D-RAN architecture has satisfied the needs of previous mobile network generations, it will not be fit for 5G’s strict list of requirements due to its lack of centralized coordination of radio resources. In centralized RAN architecture, the BBUs of different radio cells are located in a centralized location, also known as a BBU pool. Centralized RAN was already introduced for LTE in 2011 and is a pivotal concept of 5G RAN [10]. However, 5G has also adapted a concept called cloud RAN which can be considered as an evolutionary step from LTE’s centralized RAN architecture. In addition to centralized coordination, cloud RAN architecture aims to maximize RAN efficiency by bringing in cloud computing and virtualization concepts to the RAN. The main benefits of cloud RAN include:

- Enhances spectral efficiency by enabling efficient cell coordination.
- Lowers maintenance and upgrade costs by centralizing hardware.
- Reduces energy consumption by enabling efficient computing resource sharing.
- Enables the separation of general network functions from the BBU to general-purpose hardware by using virtual network functions (VNF)
- Enables the BBUs to be scaled down to do the designated signal processing tasks.

In many publications, the term C-RAN is used interchangeably to describe both centralized and cloud-based RAN. In this thesis, C-RAN refers to cloud RAN. The difference between D-RAN, centralized RAN and C-RAN is illustrated in Figure 6. [11, 12, 13]

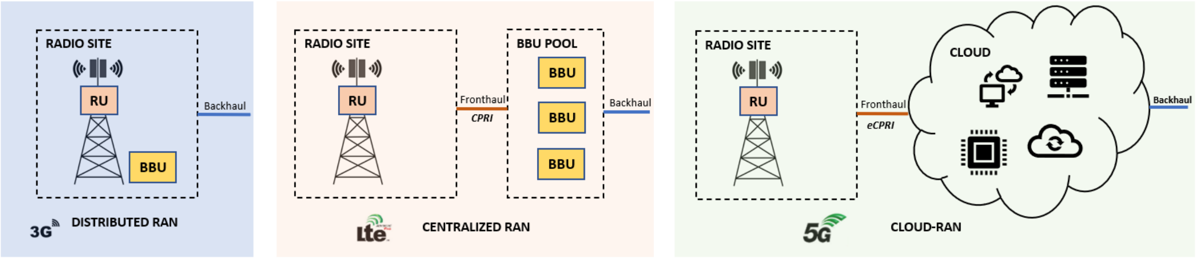


Figure 6. Distributed RAN, Centralized RAN and Cloud RAN.

Another common aspect of previous RAN generations has been equipment vendor lock down: operators have been tied to buy all the hardware and software from the same vendor to have a functional RAN. To move the industry into more open and interoperable direction, operators, vendors, and technology providers have formed alliances between them. Most notably, the Open RAN (O-RAN) alliance was founded in 2018 by merging two existing RAN alliances: the xRAN forum and the C-RAN forum. One of the big goals of O-RAN is to standardize the interface between RUs and BBUs in a non-proprietary way by providing common specifications for equipment vendors. The implementation of the fronthaul interface has traditionally been vendor specific, making vendor interoperability impossible. O-RAN also aims to make RAN more intelligent and flexible by promoting the cloud-RAN concept, machine learning and artificial intelligent.

2.2.1 Protocol stack overview

The data traffic in wireless networks is often categorized in two separate planes: the user plane, which carries the user traffic, and the control plane, which handles the control signalling between UE and gNB. The NG-RAN protocol stacks for user and control plane, as defined by 3GPP, are shown in Figure 7 [14]. 3GPP has divided the functional blocks into three main layers: physical layer (also known as layer 1), layer 2, and Radio Resource Control (RRC) layer. Closest to the air interface is the physical layer (PHY), which transforms the digital bits into radio waves in downlink, and vice versa in uplink, and hosts functions such as signal sampling, modulation, Fast Fourier Transform (FFT), and beamforming. [14, 15]

Above the physical layer lies layer 2, which is divided into four sublayers [15, 16, 17]:

- Media Access Control (MAC): Responsible for resource scheduling for UEs and forward error correction with the Hybrid Automatic Repeat Request (HARQ) protocol.
- Radio Link Control (RLC): Responsible for reformatting and reordering out-of-sequence protocol data units (PDUs) received from PDCP and forwarding them to MAC, as well as packet retransmission Automatic Repeat Request (ARQ).
- Packet Data Convergence Protocol (PDCP): Responsible for data ciphering/deciphering and integrity protection, along with header compression.
- Service Data Adaptation Protocol (SDAP): Responsible for mapping Quality of Service (QoS) flows to radio bearers. QoS flows correspond to different applications with different set of latency and capacity requirements.

The highest layer of NG-RAN is the RRC layer, which handles functions such as system information broadcasting, paging of UEs, and mobility handling between different Radio Access Technologies (RATs). In RRC perspective, each UE is in one of three states based on UE's activity: connected, inactive, or idle. RRC uses the state information to assign resources for each UE. In connected mode, the UE has established an active connection to the NG-RAN and the 5G core. In both inactive and idle states, the UE is in a power-saving state. However, whilst being inactive, the connection between the network and the UE is retained. Inactive mode can be seen as an intermediate step before the UE is completely disconnected from the network, and it ensures quick connection re-establishment if the UE starts being active again. The Non-Access Stratum (NAS) protocol is a part of the 5G core's Access and Mobility Management Function (AMF), and is used for core network functionality such as user registration and authentication. [18]

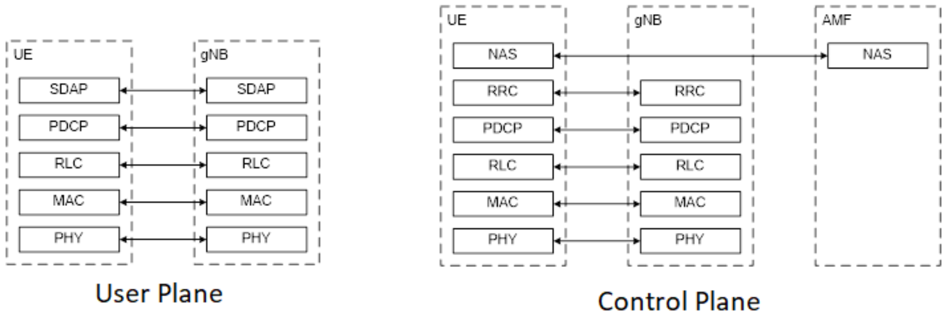


Figure 7. NG-RAN protocol stacks for user plane and control plane.

2.2.2 Evolved Common Public Radio Interface

To transport data over the fronthaul, a common transport protocol between the RU and BBU needs to be used. In LTE, the most widely used protocol has been the Common Public Radio Interface (CPRI) protocol. It was specified by the CPRI cooperation, a joint effort between many of the biggest vendors of mobile network equipment, with a goal of defining a publicly available specification for the fronthaul interface. CPRI was also one of the enablers of the centralized RAN concept, since it allowed to replace high-loss coaxial cables with more efficient fibre cables as the fronthaul connection. [19]

CPRI is a constant bitrate serial interface in which the data is multiplexed into the transmission line with time division multiplexing (TDM). The protocol is defined in physical layer and data-link layer, and consists of three different data flows:

- User plane: Carries the quantized In-phase and Quadrature (IQ) data.
- Control and Management plane: Carries control signalling between RU and BBU.
- Synchronization plane: Carries timing related information to keep the equipment in-sync.

CPRI is mainly used as a point-to-point interface, meaning that it requires a direct link between the RU and BBU, although it does support chained topologies as well. Figure 8 shows the basic architecture of a CPRI based RAN. Most commonly, the role of the BBU is to act as the master, while the RU is assigned as the slave. The master hosts the master port and is responsible for actions such as initializing the line-rate negotiation and physical layer synchronization with the slave port. [20]

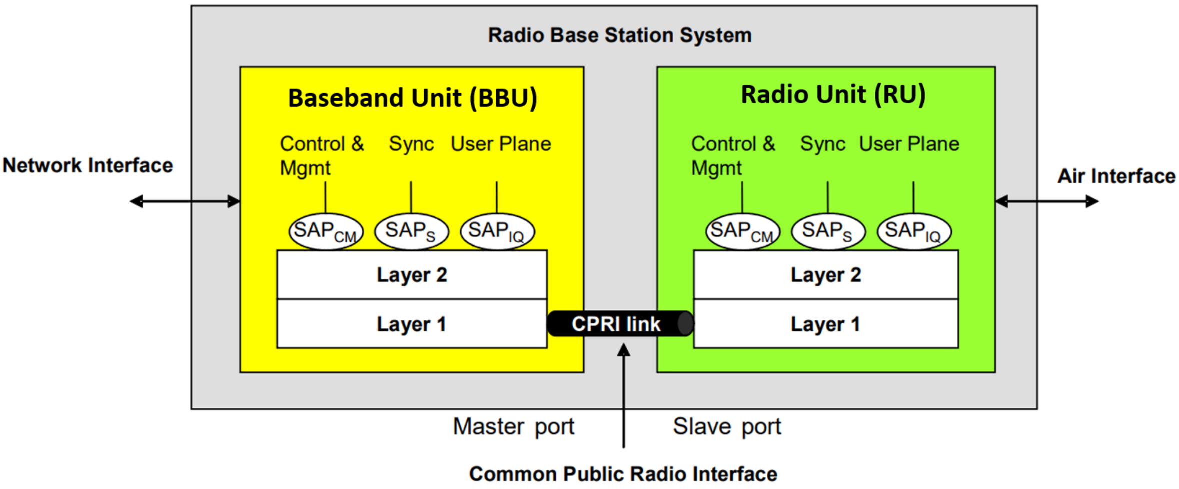


Figure 8. Basic CPRI link structure between a BBU and a RU.

The CPRI specifications determine several pre-determined fronthaul bitrates for the CPRI transmission line, ranging from 614 Mbps to 24 Gbps [20]. Due to being a constant bitrate protocol, the CPRI transmission line must maintain the selected bitrate regardless of the actual data traffic in the radio cell. This causes obvious efficiency problems with 5G, where the expected fronthaul data rates can exceed multiple hundreds of Gbps [21]. Moreover, even though the goal of CPRI was to create a commonly available specification for the fronthaul interface, some parts of the protocol were left vendor specific. As a result, the implementations of CPRI by different vendors have been more or less proprietary, making interoperability between equipment from different vendors unguaranteed. This is against the open RAN concept promoted by O-RAN.

As a solution on the vast fronthaul requirements of 5G, the CPRI cooperation has defined a new packet-based fronthaul transmission interface called evolved CPRI (eCPRI). Instead of requiring a point-to-point physical link, eCPRI is not defined in the physical layer, and instead works in a higher layer ethernet-switched or IP-routed transport network, as shown in Figure 9. The protocols used to form the transport network are not constrained by eCPRI, meaning that eCPRI packets can move very flexibly in any conventional ethernet or Internet Protocol (IP) based network. This allows eCPRI packets to flow between any eCPRI capable device within the transport network. While the interface is not backwards compatible with CPRI, they can be bridged together using Interworking Functions (IWF). [22]

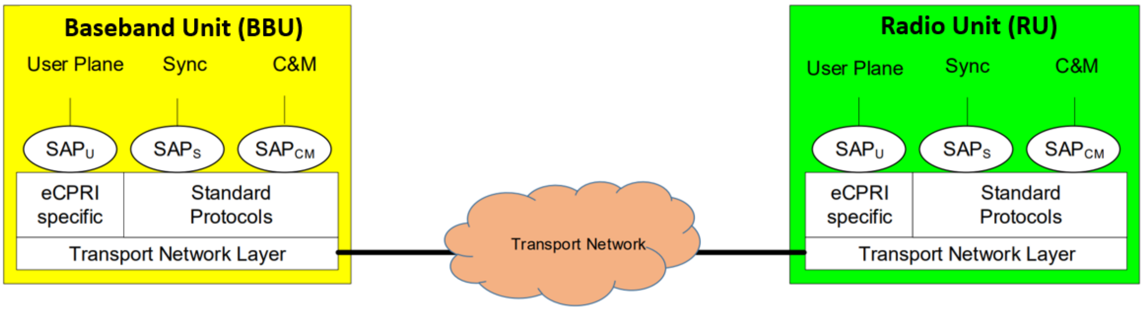


Figure 9. Basic eCPRI link structure between a BBU and a RU.

An overview of the eCPRI protocol stack is seen in Figure 10. Optional layers and header fields are marked with dashed lines. It is important to note that eCPRI only defines the protocol used to transfer the user plane data (eCPRI protocol layer). Other included data flows, such as Control & Management data, synchronization data and connection OAM (operations, administration, and maintenance) are represented with in-use protocols. For these flows, the use of other protocols is not preluded. [22]

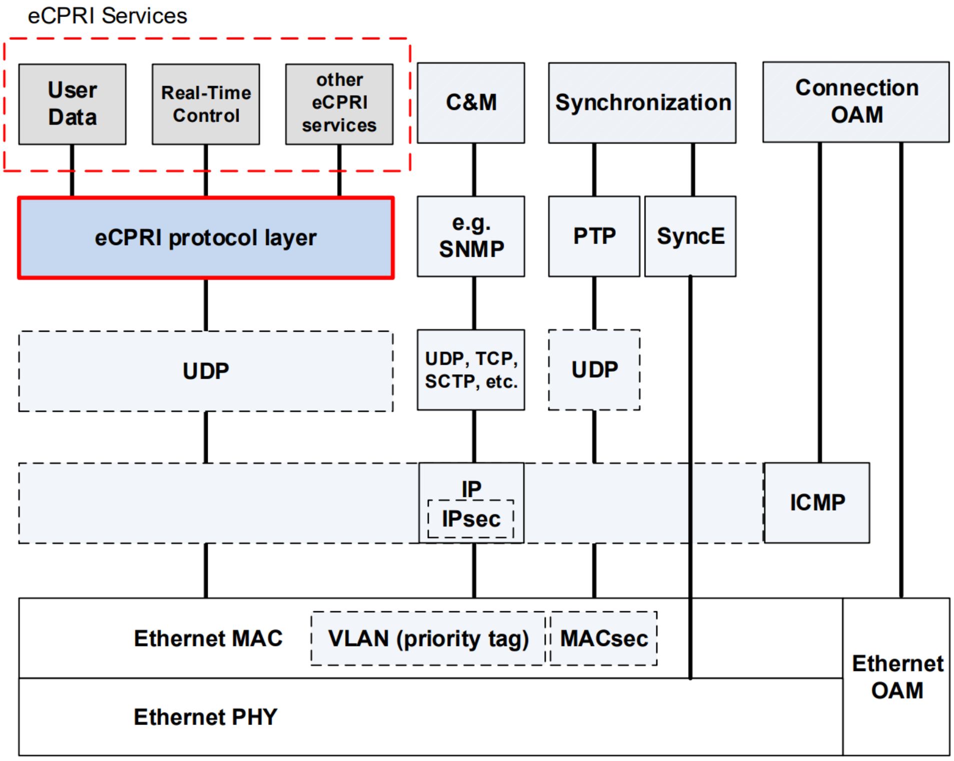


Figure 10. eCPRI protocol stack as defined by the CPRI cooperation.

The physical layer of eCPRI is based completely on the ethernet standard. This enables the device vendor to select an electrical or optical ethernet standard based on the needed line bitrate, link length and costs. Since eCPRI is a packet-based interface, it does not require a constant bitrate like CPRI, stripping the need of pre-defined line-rates. Instead, the fronthaul bitrate scales with the user traffic, and is only constrained by the maximum line rate supported by the selected ethernet interface, which can be as high as 100 *Gbps*. [22]

Depending on whether the underlaying fronthaul network is ethernet-switched or IP-routed, the eCPRI data can be chosen to be encapsulated in the data-link layer within an ethernet frame or in the higher layers within User Datagram Protocol (UDP). Figure 11 shows these two options for eCPRI data encapsulation.

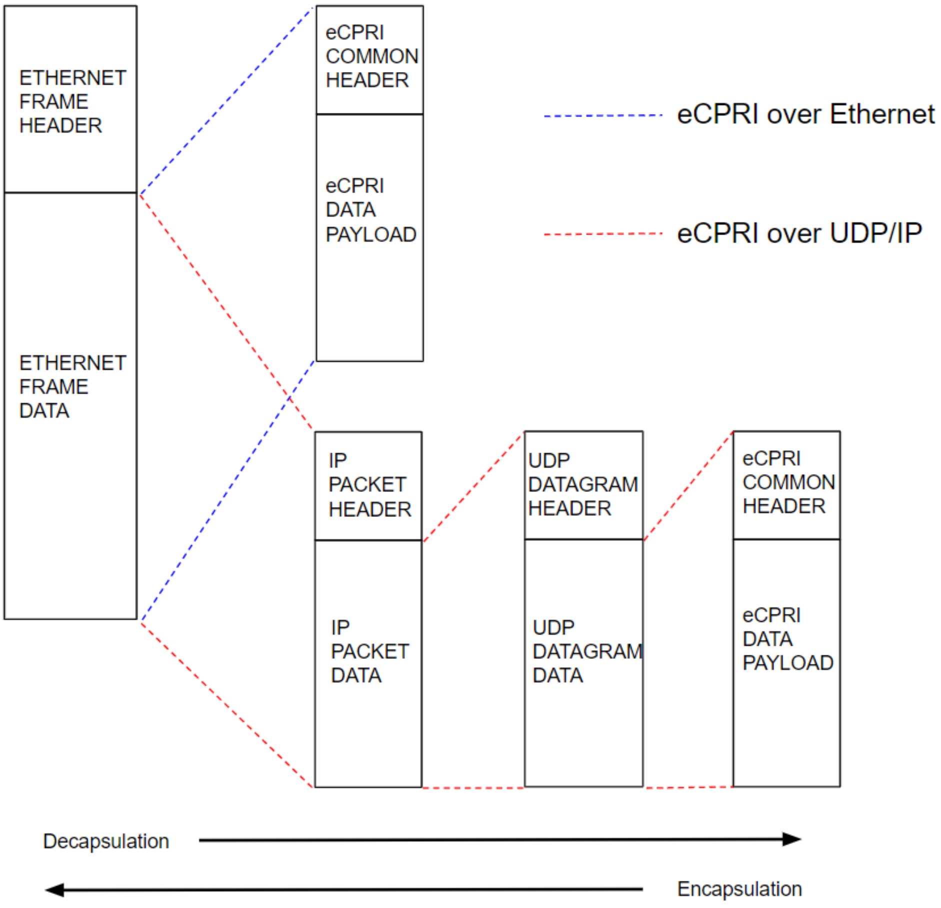


Figure 11. Ethernet and IP/UDP encapsulation for eCPRI data.

2.2.3 Fronthaul functional splits

In traditional D-RAN architecture, as well as in LTE's Centralized RAN architecture, the role of RU has been relatively simple. Pretty much all the functions from the RAN protocol stack have been allocated to the BBU, leaving only the basic RF functions to the RU, such as transmit and receive functionality and power amplification. While in centralization perspective this traditional RU-BBU functional split supports features requiring inter-cell coordination very efficiently, it has high requirements for the fronthaul delay and capacity. Since 5G is expected to deliver 10-100 times bigger data rates than LTE, even tens of *Gbps*, the traditional functional split is not going to cut it. Therefore, multiple alternative functional splits have been suggested for 5G's Cloud-RAN. [23]

3GPP has proposed eight initial functional split options for RU-BBU split based on the protocol stack of LTE, as shown in Figure 12 [21]. Along with the main suggestions, split proposals made by other relevant organizations are also presented. Making the split more towards the left side means less functions are dedicated to the BBU, meaning more of the data processing functions are done in the RU, easing the fronthaul bandwidth (BW) requirements. On the other hand, splitting the functionality on the right side means less complex radio unit. Split options within the PHY layer and between MAC and PHY are commonly known as low-layer splits, whilst split options within layer 2 and between RRC and layer 2 functions are called high-layer splits.

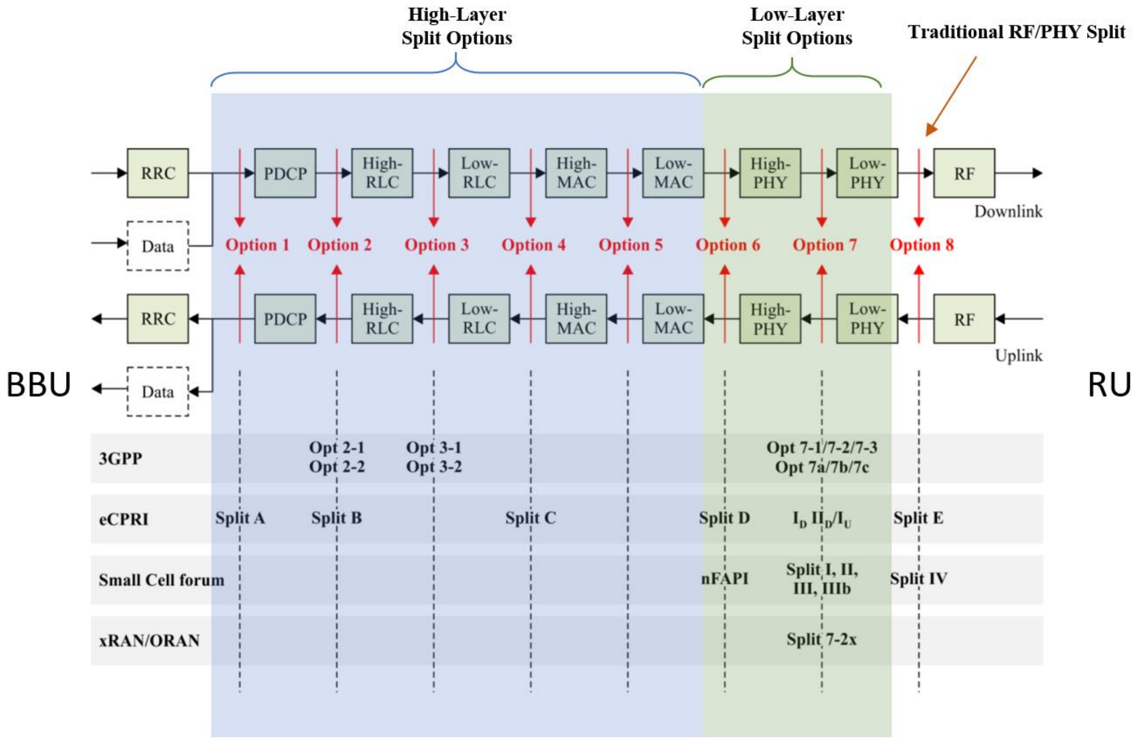
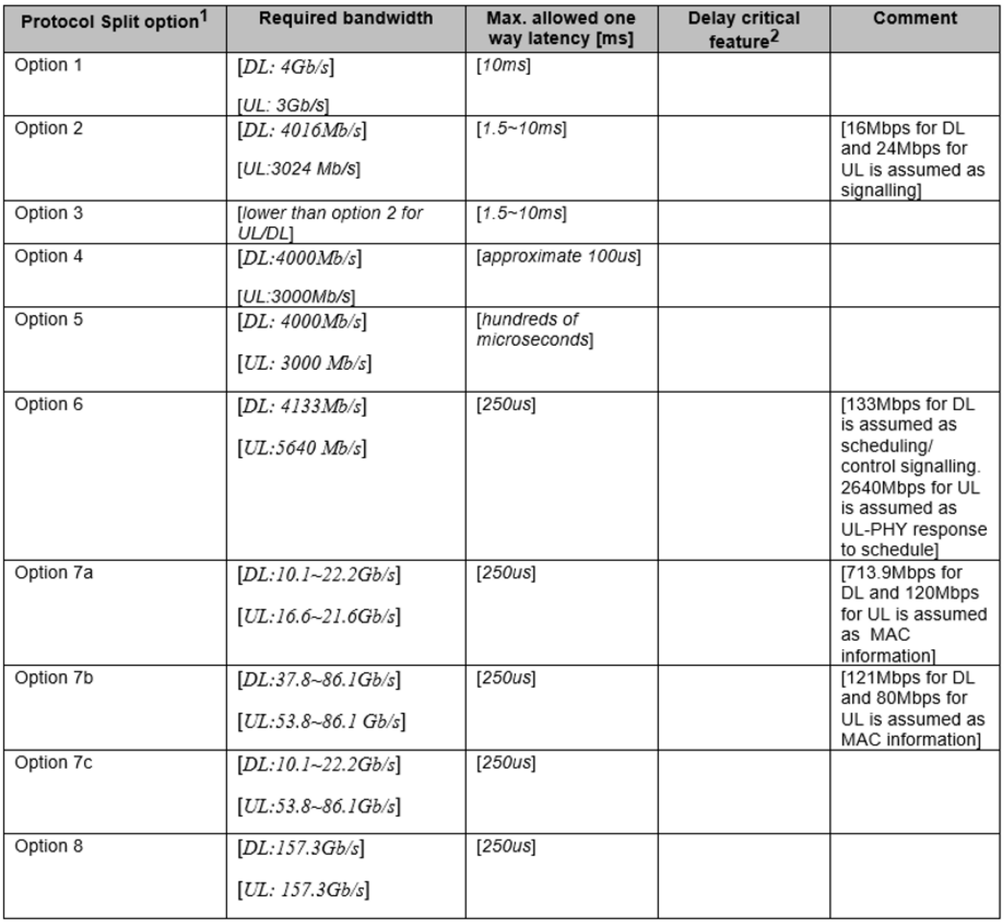


Figure 12. Overview of RU-BBU functional splits proposed by different parties.

The fronthaul data-rate and latency requirements for each split option, estimated by 3GPP, are shown in Figure 13 [24]. The choice of functional split is dependent on the use-case requirements, since it always results in a trade-off between transport capacity, latency, complexity of RU, centralization gains, deployment costs and the choice fronthaul interface protocol. The common endeavour of the industry seems to be to standardize at least one low layer split and one high layer split.



Protocol Split option[1]	Required bandwidth	Max. allowed one way latency [ms]	Delay critical feature[2]	Comment
Option 1	[DL: 4Gb/s] [UL: 3Gb/s]	[10ms]		
Option 2	[DL: 4016Mb/s] [UL:3024 Mb/s]	[1.5~10ms]		[16Mbps for DL and 24Mbps for UL is assumed as signalling]
Option 3	[lower than option 2 for UL/DL]	[1.5~10ms]		
Option 4	[DL:4000Mb/s] [UL:3000Mb/s]	[approximate 100us]		
Option 5	[DL: 4000Mb/s] [UL: 3000 Mb/s]	[hundreds of microseconds]		
Option 6	[DL: 4133Mb/s] [UL:5640 Mb/s]	[250us]		[133Mbps for DL is assumed as scheduling/ control signalling. 2640Mbps for UL is assumed as UL-PHY response to schedule]
Option 7a	[DL:10.1~22.2Gb/s] [UL:16.6~21.6Gb/s]	[250us]		[713.9Mbps for DL and 120Mbps for UL is assumed as MAC information]
Option 7b	[DL:37.8~86.1Gb/s] [UL:53.8~86.1 Gb/s]	[250us]		[121Mbps for DL and 80Mbps for UL is assumed as MAC information]
Option 7c	[DL:10.1~22.2Gb/s] [UL:53.8~86.1Gb/s]	[250us]		
Option 8	[DL:157.3Gb/s] [UL: 157.3Gb/s]	[250us]		

Figure 13. Fronthaul data-rate and latency requirements for each functional split, estimated by 3GPP.

Split option 8 represents the traditional RF/PHY split that is used in LTE and legacy cellular networks. In this split option, only the RF functions such as signal sampling and baseband up/down conversion are left to the radio unit. This leads to a relatively simple and cheap RU design and maximises the gains of centralization by situating most of the baseband functions in the BBU. For example, the amount of processing resources for each radio cell can be efficiently scaled throughout the whole protocol stack based on user traffic. However, this "all-in" centralization also brings up notable challenges for the fronthaul connection. Firstly, the timing-sensitive packet re-transmission protocols such as HARQ are located in the BBU, making latency requirements very strict between RU and BBU. 3GPP has specified the maximum one-way fronthaul latency of split option 8 to be 250 µs [25]. Secondly, the data is conventionally transported as quantized time-domain IQ-samples using the CPRI fronthaul interface, which requires a very high constant fronthaul transport rate that scales linearly as the number of antennas increase. Theoretically required maximum bitrate for split option 8 fronthaul ($R_{b,8}$) can be calculated with Equation 1 [25]:

$$R_{b,8} = F_s * N_a * b_s * (\frac{BW}{20\ MHz}) + OH \tag{1}$$

where F_s is the sampling frequency, N_a the number of antenna elements, b_s bits per sample, BW the total system bandwidth and OH the transport protocol-specific overhead. For example, an RU with a single antenna port, 100 MHz system bandwidth, 30.72 *MHz* sampling frequency and 32-bit sample length would end up requiring a bit rate of 4.92 Gbps for the fronthaul connection (assuming no overhead). In contrast, a similar carrier configuration with 64 antennas requires 315 Gbps link capacity. Since 5G is expected to support antenna arrays consisting up to 256 antenna elements, the traditional RF/PHY split is not scalable enough for it. [15]

The high-layer functional split options aim to massively relax the fronthaul link requirements by de-centralizing the real-time functions of the protocol stack. 3GPP has been promoting the split option 2 as the official high layer split and standardized the F1 interface as its fronthaul transport protocol [25]. This split separates BBU-RU functionalities between RLC and PDCP, moving timing-sensitive protocols such as ARQ and HARQ to the RU, along with MAC and the whole physical layer. This makes the timing requirements much more tolerant than in the RF/PHY split, maximum allowed one-way latency being up to 10 ms [25]. In option 2, required fronthaul capacity ($R_{b,2,DL}$ for downlink, $R_{b,2,UL}$ for uplink) scales based on the number of independent data streams instead of number antenna ports, as seen in Equations 2 and 3:

$$R_{b,2,DL} = R_{peak} * (BW/BW_c) * (N_l/N_{l,c}) * (8/6) + sig \qquad (2)$$

$$R_{b,2,UL} = R_{peak} * (BW/BW_c) * (N_l/N_{l,c}) * (6/4) + sig \qquad (3)$$

where R_{peak} is the peak data rate, BW the channel bandwidth, BW_c the bandwidth required for control data, N_l the number of data streams, $N_{l,c}$ the number of layers required for control data and sig the amount of control signalling. In the case of a system using 256 Quadrature Amplitude modulation (QAM) with $R_{peak} = 150$ Mbps, $BW = 100$ MHz, $BW_c = 20$ MHz, $N_l = 8$, $N_{l,c} = 2$ and no control signalling, the maximum required bitrates for DL and UL would add up to about 4 Gbps and 3 Gbps, respectively. [15, 25]

The flip side of choosing a high layers split is that it makes the RU much more complex than traditionally. Complexity increases power consumption, weight and size of the RU, which in turn leads to higher operating costs. Additionally, high layer splits have less to gain from C-RAN architecture since less of the RAN functions are centralized. Low layer split options aim to provide a well-balanced option between the traditional RF/PHY split and the high layer RLC/PDCP split. In particular, splits within the physical layer have mostly been under consideration. While 3GPP has not promoted any particular split as the official low layer split, other groups such as the CPRI cooperation, Small Cell forum and O-RAN alliance (previously xRAN), have taken a bigger role on the standardization work. The relation between the low layer split options suggested by different parties can be seen in Figure 14 [21]. Since the physical layer splits do not include the real-time functions in the RU, the fronthaul requirements are latency-wise similar to option 8 used in LTE. On the other hand, fronthaul capacity requirements scale either with the number of antennas or number of data streams, depending if port expansion (digital beamforming) from antenna streams to data streams is allocated to RU ($R_{b,7,pe}$) or not ($R_{b,7}$), and can be approximated with Equations 4 and 5 [26]:

$$R_{b,7,pe} = N_{sc} * b_s * N_a * N_{sym} * 1000 + I_{MAC} \qquad (4)$$

$$R_{b,7} = N_{sc} * b_s * N_l * N_{sym} * 1000 + I_{MAC} \qquad (5)$$

In the equations, N_{sc} is the number of subcarriers, b_s bits per sample, N_a the number of antenna ports, N_l the number of layers, N_{sym} number of symbols per sample, and I_{MAC} the amount of information required by the MAC layer, such as the resource block (RB) assignment information and antenna configuration. For a 100 MHz LTE band with 5*1220=6000 total subcarriers, 32-bit sample length, 32 antennas, 8 data streams and 14 symbols per sample, and assuming the MAC information to be 120 Mbps, the fronthaul bitrates for intra-PHY splits would be about 86 Gbps with port expansion in RU and 21,6 Gbps without it.

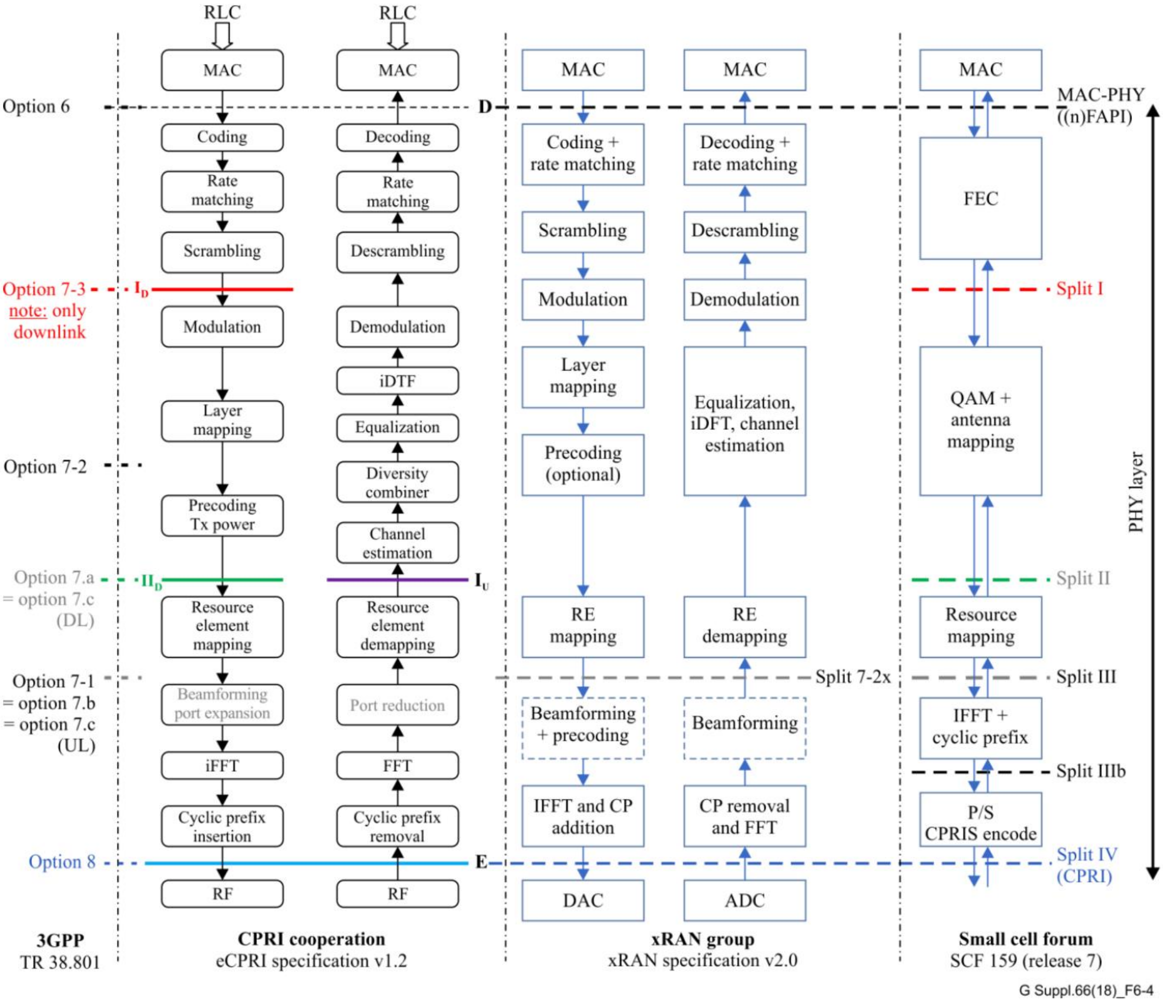


Figure 14. Detailed block diagram of the lower layer functional splits proposed by different organizations.

One of the most notable lower layer functional splits is the 7-2x split, which was originally introduced by xRAN and later adapted by the O-RAN alliance. The alliance is gaining increasing amount of interest, and some of the biggest network equipment vendors have joined its ranks. The split allocates digital beamforming, cyclic prefix (CP) addition/removal and FFT functions to the RRU, while higher layer functions starting from resource element (RE) mapping/demapping are left to the BBU. This means that the data is transferred over the fronthaul as frequency-domain IQ-samples. As the 7-2x split allocates digital beamforming to the RU, the fronthaul capacity requirements for the split can be estimated with Equation 5. For the fronthaul interface, ORAN has adapted two packet-based protocols: eCPRI and Radio over Ethernet (RoE), focus of this thesis being in the former. To prevent eCPRI from becoming a vendor proprietary protocol, O-RAN has stepped in to standardize the interface in a way that true openness can be achieved. [21, 27]

2.3 5G New Radio

Considering the diverse set of use-cases ranging from extreme mobile bandwidth, ultra-low latency and massive amount of connection, the air interface used in LTE is not enough for 5G in terms of flexibility and efficiency. 5G new radio (5G NR) is a latest radio access technology designed specifically for the requirements of 5G. It introduces the use of new frequency bands as well as multiple design enhancements and techniques to overcome the shortcomings of LTE.

2.3.1 Frequency ranges

The most obvious way to increase data rates of a communications system is to widen the bandwidth of the used radio channel. While this sounds straight forward, there is much more complexity to the topic because the usage of the frequency spectrum is heavily licenced to minimize the effect of interference. The licensed frequencies used by LTE range from a few hundred megahertz up to 3.5 GHz. These frequencies are very crowded and have very little space for 5G's bandwidth requirements. Therefore, two distinct frequency ranges have been defined for 5G NR. Frequency range 1 (FR1) contains frequencies from 450 MHz to 7.125 GHz, while frequency range 2 (FR2) spans from 24.25 GHz up to 52.6 GHz. [28]

FR1, commonly known as the "sub-6 GHz" 5G due its upper limit originally being at 6 GHz, consists of some of the frequency bands used in previous mobile technologies along with new, unused bands. While not offering as much new spectrum as FR2, frequencies in FR1 undergo less radio channel attenuation, which leads to better cell coverage. The data rates of the limited FR1 frequency bands have been extended using techniques such as beamforming and carrier aggregation. [28]

FR2 frequencies operate with millimeter wavelength radio signals, which is why the frequency range is also called millimetre wave (mmW) 5G. The main benefit of operating in FR2 frequencies is the amount of available spectrum, which enables the use of wide transmission bandwidths, while not excluding additive techniques used in FR1 frequencies. In 5G, the maximum carrier bandwidth for FR2 is 400 MHz, which would be tricky to fit in the FR1 bands. The flip side of FR2 frequencies is that millimeter waves are associated with higher attenuation, reducing the cell size. This can be overcome by deploying FR2 cells more densely. By concentrating the FR2 deployments into locations where the high data rates are needed, such as city centers, while enhancing the 5G coverage with FR1 cells, the stress on the overused lower frequencies can be kept as minimum and the overall network efficiency is maximized. [28]

2.3.2 CP-OFDM

Orthogonal Frequency Division Multiplexing (OFDM) is a data multiplexing technique in which the available frequency channel is divided into multiple orthogonal carriers called subcarriers, as shown in Figure 15 [29]. The orthogonality between subcarriers mitigates the interference between the signals. OFDM offers multiple benefits compared to wide-band transmissions used in legacy cellular systems, like 2G and 3G. When data is sent parallel on multiple low data rate channels, the effect of frequency selective fading caused by multipath reflections is mitigated. The orthogonality enables the subcarriers to be spaced very closely to each other, which in turn enhances the spectral efficiency. OFDM uses the inverse FFT (iFFT) to multiplex orthogonal subcarriers into concatenated OFDM symbols on the transmitter side, and FFT on the receiver side to de-multiplex the OFDM symbols to original data streams. One closely related concept of OFDM is Orthogonal Frequency Division Multiple Access (OFDMA). By definition, all subcarriers of each symbol can be allocated only to one user in OFDM, meaning that the radio resources are allocated between users only in time-domain. However, in OFDMA, resource allocation can also be done in frequency domain (FD) by allocating different subcarriers within a symbol to different users. [30, 31]

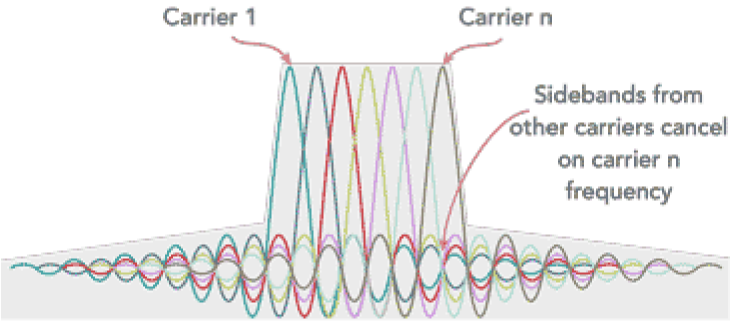


Figure 15. Orthogonal subcarriers of OFDM.

Due to its high spectral efficiency and robustness against various interference sources, OFDM has been used extensively in wireless communication systems. In mobile networks, it has already been utilized in LTE, which uses a subversion of OFDM called cyclic prefix OFDM (CP-OFDM) for downlink transmissions. As for uplink, LTE uses Single-Carrier Frequency Division Multiple Access (SC-FDMA) due to its low power consumption, an important feature for battery-powered user equipment. The way that subcarriers are added together in OFDM makes its peak-to-average power ratio (PAPR) relatively high, putting immense requirements for power amplifiers, thus increasing power consumption. However, this drawback can be mitigated by using techniques such as signal clipping. As for 5G, CP-OFDM is adapted for both DL and UL, making transceiver implementations symmetric for both transmissions. The basic block structure of a CP-OFDM transceiver is shown in Figure 16. [32]

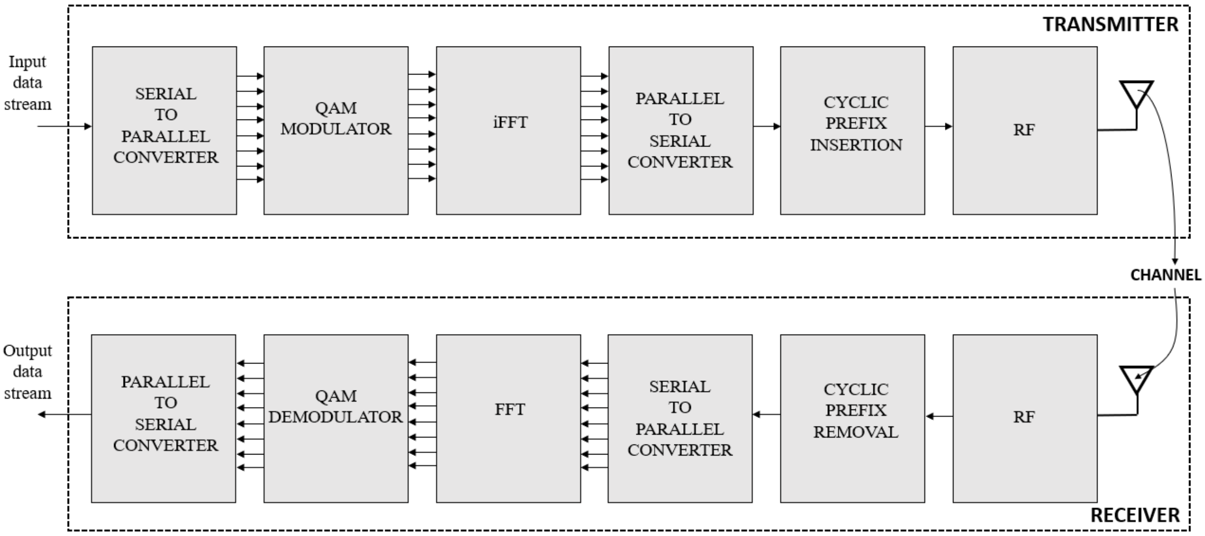


Figure 16. Block diagram of an OFDM transmitter and receiver.

2.3.2.1 QAM

The most widely used modulation method in modern wireless communication systems is the Quadrature Amplitude Modulation (QAM). QAM is based on the modulation of two in-quadrature signals called the in-phase carrier and the quadrature (90° shifted) carrier. In QAM, the baseband data stream is divided into two parallel data streams known as the I and Q streams. The I and Q streams modulate the in-phase carrier and the quadrature carrier respectively, and the sum of the modulated carriers is the output of the modulator. [33]

Figure 17 shows the basic structure of a digital 16QAM modulator. The input bit stream is divided into the parallel I and Q streams. A local oscillator (LO) is used to generate the in-phase and quadrature signals. The I and Q data bits modulate the in-phase and quadrature LO signals respectively. The modulated signal, also called an IQ symbol, is created by adding the two products together. A popular way of representing the IQ data is by mapping the symbol into the complex plane, also known as the constellation diagram, as depicted in Figure 18. The constellation diagram clearly demonstrates how the phase and the amplitude of the modulated IQ signal can be changed just by altering the I and Q components, making QAM a combination of phase and amplitude modulation. [33, 34]

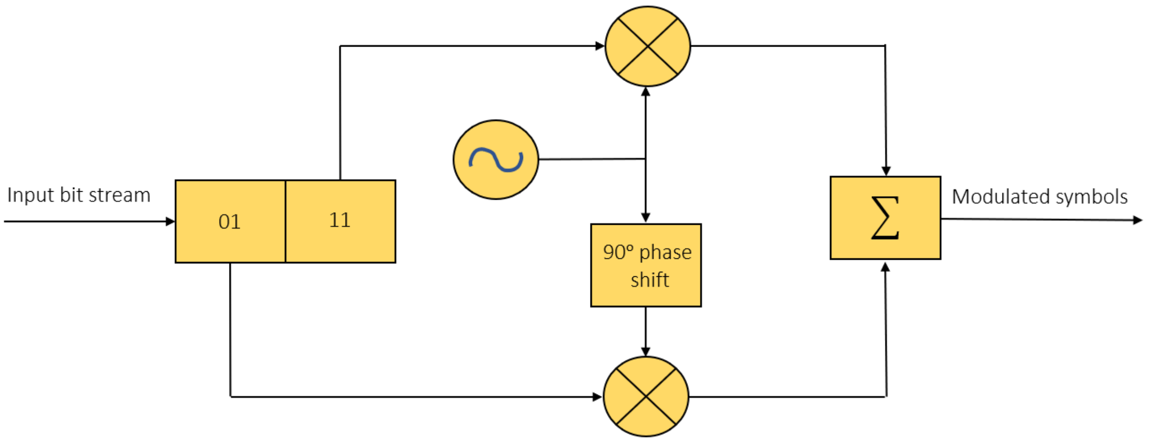


Figure 17. Digital 16QAM modulator.

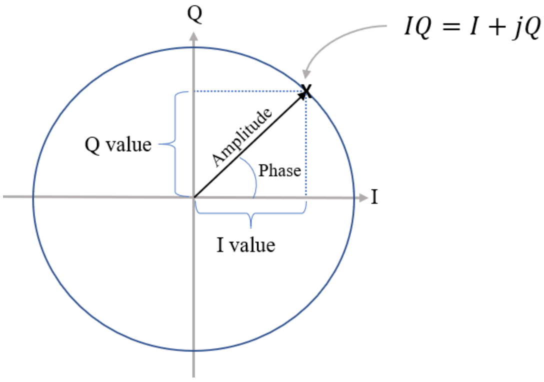


Figure 18. IQ data in complex format.

In digital modulation, the amount of data carried by one modulated IQ signal is dependent on the QAM modulation order. The modulation order determines how many bits are represented by a single IQ symbol. Each modulated symbol has a defined phase and amplitude which can be mapped into the constellation diagram. For example, in 16QAM modulation, there are 16 different constellation points in the constellation diagram, each corresponding to different phase and amplitude combination and each representing a unique 4-bit sequence. The input bit stream is then encoded into these 4-bit symbols. The 16QAM constellation diagram is shown in Figure 19. By increasing the modulation order, more bits can be transmitted per symbol, raising the overall system throughput. [35]

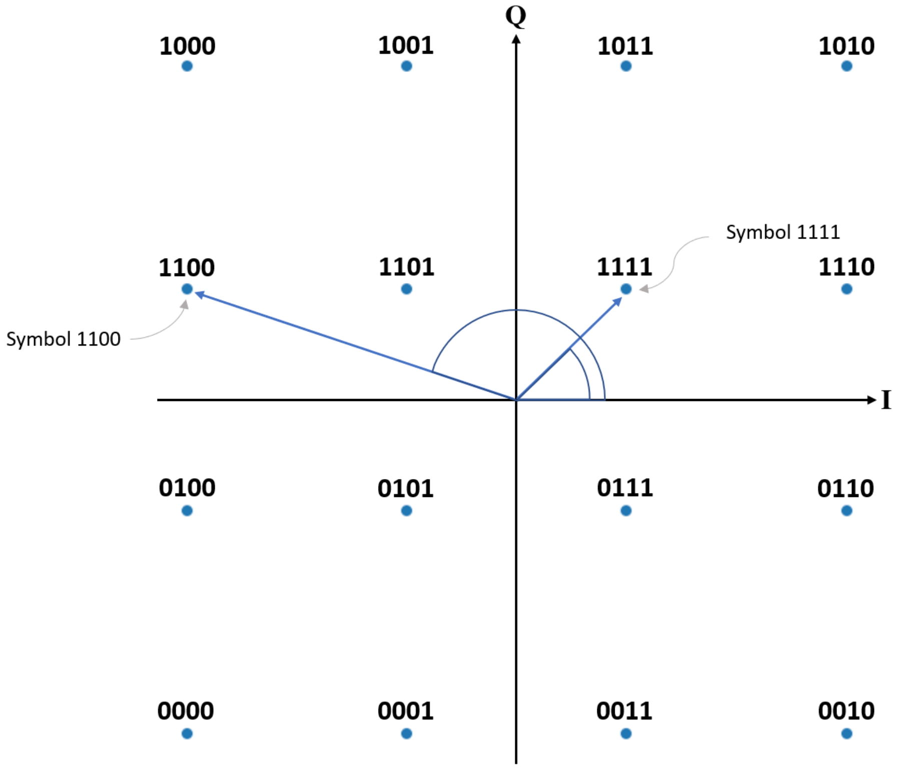


Figure 19. Constellation points of 16QAM modulation

The drawback of QAM lies in its complexity. To accurately detect the received QAM signals, the receiver needs to use linear amplifiers, which are complicated and power-hungry components. Also, when moving to a higher-order modulation, the constellation points are more closely spaced, making the system more prone to errors caused by noise and interference. For this reason, many communications systems allow the order of the QAM modulation to be dynamically changed in respect to the link conditions between the transmitter and receiver. [36]

2.3.2.2 FFT/iFFT

The most essential concept of an OFDM system is the Fast Fourier Transform (FFT), an efficient implementation of the discrete Fourier transform (DFT) algorithm used in wide range of signal processing applications. Essentially, FFT is a mathematical procedure that transfers discrete data from time domain (TD) to frequency domain on the receiver side, and vice versa on the transmitter side with inverse FFT (iFFT). The FFT procedure offers an efficient way to multiplex data onto orthogonal subcarriers: without it, the orthogonality of the subcarriers would be handled by separate modulators, and given that an OFDM waveform can consist of thousands of subcarriers, the total number of modulators would be senselessly high. Instead, the iFFT block of the transmitter takes an array of N modulated subcarriers as input, which together correspond to one OFDM symbol (X_k). A sequence of time domain samples (x_n) are then generated with the iFFT operation. The general mathematical representations of both DFT and inverse DFT are shown in Equations 6 and 7, respectively. The Fast Fourier Transform reduces the processing time of the general form DFT by dividing it into smaller DFT processes. [37, 38]

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi kn}{N}}, \qquad k = 0,1,\dots,N \tag{6}$$

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{\frac{i2\pi kn}{N}}, \qquad n = 0,1,\dots,N \tag{7}$$

One of the most important characteristics of FFT (and DFT in general) is the so-called circular convolution theorem, which states that the circular convolution of two time-domain sequences, $h(n)$ and $s(n)$, is equal to pointwise multiplication of corresponding frequency domain sequences, $H[k]$ and $S[k]$. The circular convolution theorem is shown in Equation 8. The principle of CP-OFDM is based on this feature. [39]

$$h(n) \circledast s(n) \leftrightarrow H[k] \cdot S[k] \tag{8}$$

where $\circledast$ denotes the autocorrelation function and ↔ the FFT process.

2.3.2.3 Cyclic Prefix

CP-OFDM introduces the concept of a cyclic prefix (CP), a guard interval inserted at the beginning of each OFDM symbol. Even though the addition of CP reduces the total system capacity by introducing overhead to the transmitted signal, it is extremely simple way to battle inter symbol interference (ISI) caused by the delay spread of multipath channel components and inter carrier interference (ICI) caused by Doppler spread. The CP is constructed by taking a replica from the end of the OFDM symbol with a length that is longer than the maximum delay spread of the radio channel. The purpose of cycling samples from the end of the OFDM symbol to the beginning is to make the channel response act as a circular convolution to the transmitted signal. By using the circular convolution theorem shown in Equation 8, it can be stated that the circular convolution of the channel response and the transmitted signal corresponds to multiplication in the frequency domain. This makes channel equalization extremely efficient since the effects of the channel can now be equalized with just a simple multiplication operation. Figure 20 illustrates the concept of CP-OFDM. [40]

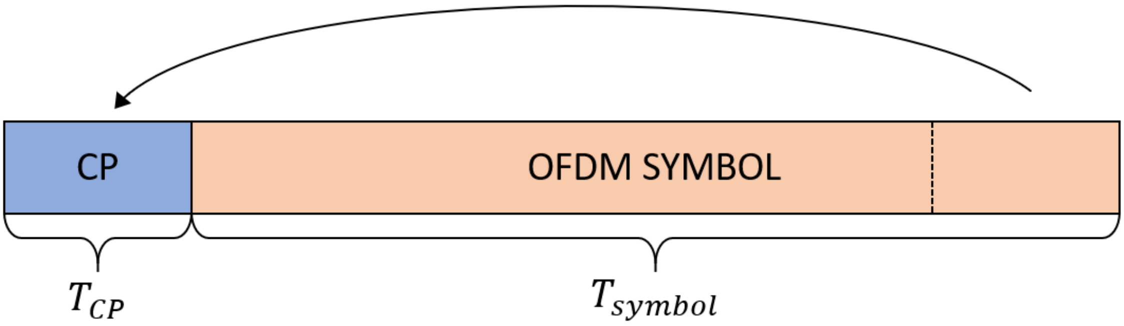


Figure 20. Concept of CP-OFDM

2.3.3 Frame structure

For the most part, the time-frequency resource allocation in 5G NR follows the same basic structure as in LTE. The 5G NR resource grid structure is shown in Figure 21. In frequency domain, 12 consecutive subcarriers make up one physical resource block (PRB), and in time domain the length of one radio frame is 10 ms, further divided into one millisecond long subframes. A subframe in turn consists of slots, and the number of slots in one subframe ($N_{slots\_in\_sf}$) is dependent on the selected 5G NR numerology (μ) as shown in Equation 9:

$$N_{slots\_in\_sf} = \mu + 1 \tag{9}$$

The numerology μ is a concept added in 5G to add flexibility required by the wide range of use-cases and different frequency ranges, and determines the subcarrier spacing (SCS) and symbol length as shown in Table 1. Each slot in 5G NR includes 14 OFDM symbols. Retaining the same basic frame structure of LTE enables 5G to co-exist within the same system as its predecessor technology. [41]

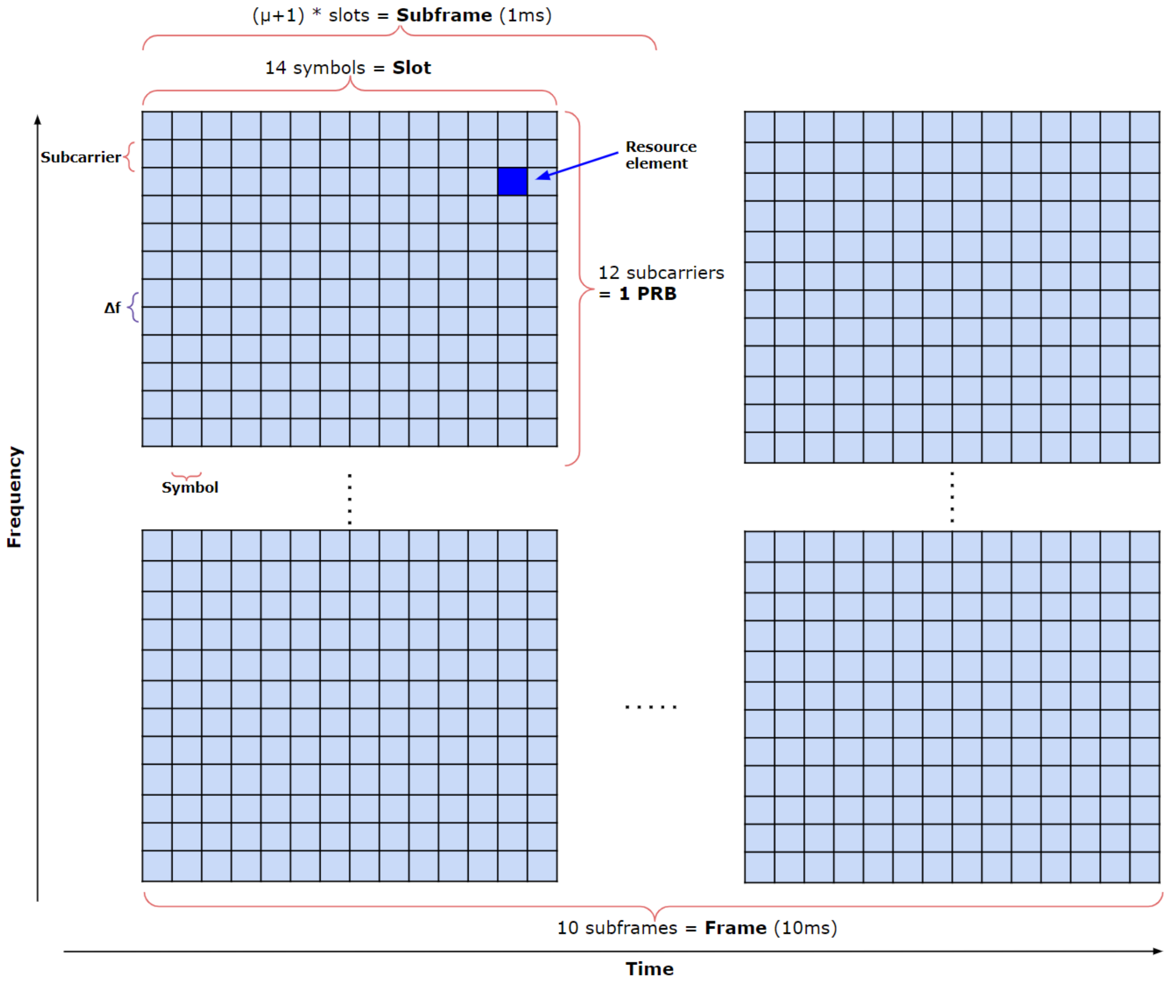


Figure 21. 5G NR frame structure.

Table 1. Flexible numerology configurations in 5G NR.

Numerology (μ)	**0**	**1**	**2**	**3**	**4**
Subcarrier spacing, *kHz* ($15kHz * 2^{\mu}$)	15	30	60	120	240
OFDM symbol duration, μs	66.67	33.33	16.67	8.33	4.17
CP duration, μs (long symbol / other symbols)	5.2 / 4.69	2.86 / 2.34	1.69 / 1.17	1.11 / 0.59	0.81 / 0.29
Slots per subframe	1	2	4	8	16
Symbols per subframe	14	28	56	112	224
Max BW, *MHz*	~50	~99	~198	~396	~397
Frequency range	FR1	FR1	FR1/FR2	FR2	FR2

As seen in table a, the symbol and CP durations get proportionally shorter when moving to higher subcarrier spacing, enabling shorter slot lengths. This property can be used to shorten transmission times, making higher numerologies suitable for low-latency services. However, the shorter CP duration makes higher numerologies insufficient to be used in delay spread intensive environments, such as large macro cells. Therefore, wider subcarrier spacings are most suitable for small radio cells. At mmW frequencies, the wider subcarrier spacing can be used to mitigate increased phase noise caused by high oscillator frequencies. [42, 43, 44, 45]

2.3.4 Massive MIMO and beamforming

In early wireless communications systems, both the transmitter and receiver used only one antenna for transmitting and receiving radio waves. This sort of configuration is commonly known as Single-Input Single-Output (SISO). While being very simple to implement, SISO systems are prone to interference caused by multipath effects and lack efficiency: the whole capacity of the air interface is not utilized. To improve these shortcomings, modern communications systems have moved on to use a configuration called Multiple-Input Multiple-Output (MIMO). In MIMO, multiple antennas and processing chains are used on transmitters and receivers to improve robustness of the channel. Each added antenna provides additional paths (MIMO channels) for the physical radio waves to travel from the transmitter to the receiver. Depending on the chosen transmission technique, the MIMO channels can be used as additional diversity to improve signal-to-noise ratio (same data sent over multiple channels to a single user) or as spatial division multiplexing to increase throughput (different data streams sent over multiple channels to single user). [46]

MIMO is not a particularly new concept in mobile networks: it has been taken to use already in 3G and LTE systems. In these systems, MIMO usually refers to Single-User MIMO (SU-MIMO), in which the MIMO transmissions are sent to a single user at a time. In 5G, the MIMO radios are often called as massive MIMO (mMIMO) radios, referring to the increased amount of antenna elements used per radio. The higher carrier frequencies used in 5G lead to smaller antenna elements, allowing tens, even hundreds of elements to be fit in a single radio system. The larger amount of MIMO channels enables more extensive use of techniques such as beamforming and Multi-User MIMO (MU-MIMO), in which data streams are sent simultaneously to multiple users. MU-MIMO can be implemented by using techniques such as precoding. While SU-MIMO aims to maximize the peak throughput towards a single user, MU-MIMO intends to increase the whole cell capacity. The difference between the two MIMO techniques is illustrated in Figure 22. [47]

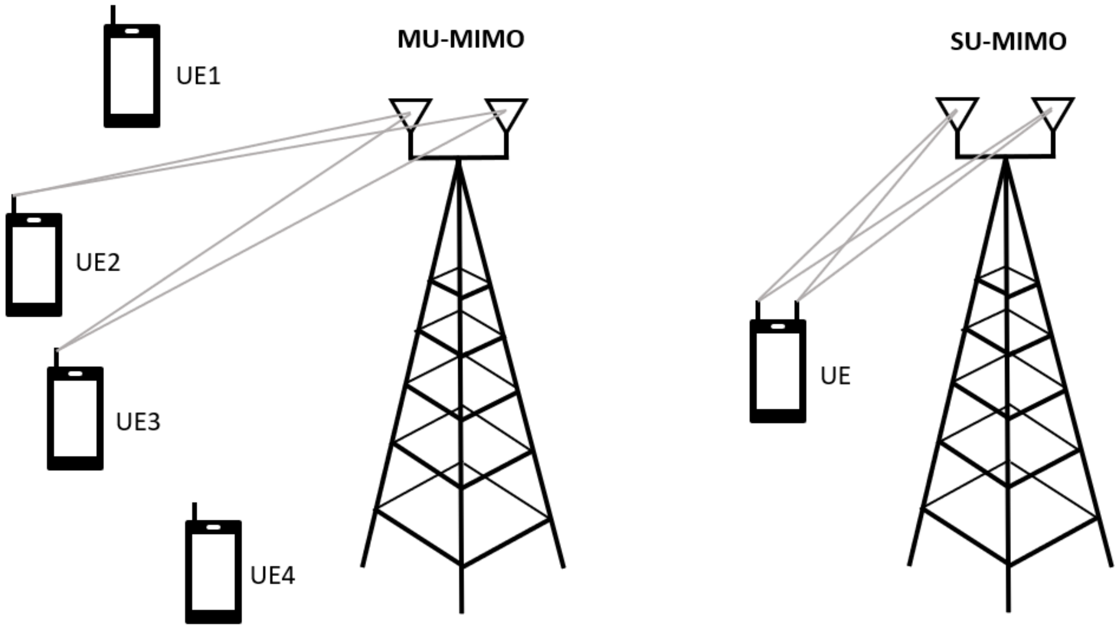


Figure 22. Multi-User MIMO and Single-User MIMO

Beamforming is one of the key techniques in 5G to enhance spectral efficiency of the network. In short, beamforming is a technique used to create a narrow, targeted beam in a specific direction by means of appropriately weighting the magnitude and phase of the signal sent from different antenna elements. In the targeted direction, the phases of the received signals constructively add up, making a strong beam. In other directions, the reverse happens: the opposite phased signals sum up destructively, nullifying the signal in those directions. [48]

There are two main techniques for beamforming: analog and digital. The high-level block diagrams of these beamforming techniques are shown in Figure 23. In analog beamforming, the weighting of the signals happens after digital-to-analog converters (DAC) in the RF domain with analog phase-shifters. Analog beamforming is a simple and cost-effective way to add beamforming gain to the system, but it can only create a single beam at a time. In digital beamforming, the beamforming operation happens in digital domain, which enables multiple beams to be formed with one set of antenna arrays. Digital beamforming is also the basis for MU-MIMO since it enables resource allocation in both time and frequency domain. The drawback of digital beamforming is that it requires separate RF processing chains for each antenna element, making the transceivers more complex and power-hungry. In addition to the main beamforming techniques, hybrid beamforming methods have been developed to combine the simplicity of analog beamforming and the flexibility of digital beamforming. [49]

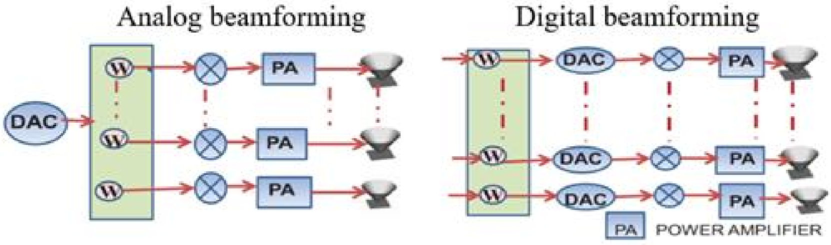


Figure 23. Analog and digital beamforming.

3 5G ECPRI RADIO UNIT

When connected to a cellular network, the UE uses radio waves to establish a direct connection to the RAN. Aside from physical antennas, the first device processing the incoming signals in the RAN is the radio unit. Main functionality of the RU has traditionally been the analog-digital interconversion, utilizing RF functionalities such as signal amplification and filtering. Morever, the radio relays the digital signal to the baseband unit for the complex signal processing tasks. However, with the new BBU-RU functional splits some of the data processing functionalities traditionally done in the BBU are now allocated to the RU. The importance of these more advanced radio units is accelerating since all the latency and capacity requirements for 5G cannot be achieved with the traditional BBU-RU split in which the sole role of the RU is to act as a simple RF processing device. In this chapter, a brief overview is given to an O-RAN based 5G eCPRI radio unit, focus being on the fronthaul interface. Then, the test setup and key performance indicators (KPIs) for UL datapath testing are described.

3.1 Radio unit architecture

A high-level overview of the main functional blocks of an O-RAN 7-2x eCPRI radio unit is given in Figure 24. Closest to the air interface, the analog signal received from the UE is first processed by the RF components and converted into digital domain with analog-to-digital converters (ADC). The cyclic prefixes are then removed from the digitized data, after which the data is transformed into frequency domain with iFFT. Next, the FD data is mapped from antenna streams to data streams with digital beamforming, after which the data streams are encapsulated into eCPRI packets and sent over the fronthaul interface to the BBU. [50]

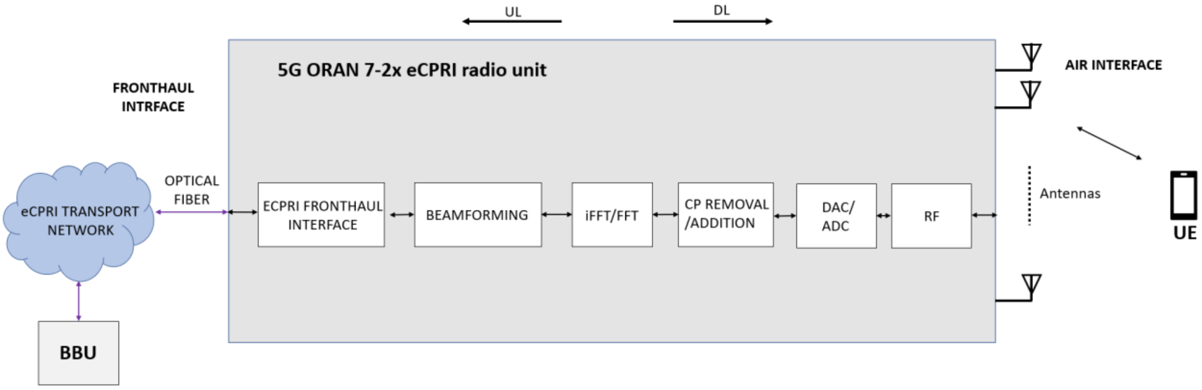


Figure 24. High-level functional architecture of an O-RAN based 5G eCPRI radio unit.

3.2 O-RAN based eCPRI fronthaul

Although eCPRI is meant to be an open interface and a key enabler for RAN vendor interoperability, the specifications provided by the CPRI cooperation are left partly vague, creating a risk of vendor proprietary fronthaul implementations. To prevent eCPRI from becoming a closed interface like CPRI, O-RAN has stepped in to provide its own extensive technical specifications for RAN fronthaul. By unambiguously supplementing the nonspecified parts of eCPRI specifications, the risk of eCPRI becoming a proprietary interface is nullified.

As for the basic data flows, O-RAN has defined four separate planes for the fronthaul interface:

- User plane (U-Plane): Used to transfer user data as frequency domain IQ samples.
- Control plane (C-Plane): Used to transfer real-time control data between the RU and BBU, such as scheduling and beamforming commands.
- Synchronization plane (S-Plane): Used to transform timing related information.
- Management plane (M-Plane): Used for support and management features.

In this thesis, the focus is mostly on the U-Plane data, since it carries the actual modulated IQ samples (user data). [50]

O-RAN specifications provide support for both ethernet and UDP/IP protocol encapsulation for eCPRI, the focus of this thesis being in the former. The modulated IQ samples of user plane are encapsulated under multiple headers, including:

- Ethernet Frame Header
- eCPRI Transport Header
- O-RAN Application Header
- O-RAN Section Header

In addition to the encapsulation specifications provided by the CPRI cooperation (Figure 11), O-RAN specifies two supplemental headers within the eCPRI data payload. The IQ data encapsulation specified by O-RAN is demonstrated in Figure 25. In the following chapters, the content of the headers is described in detail.

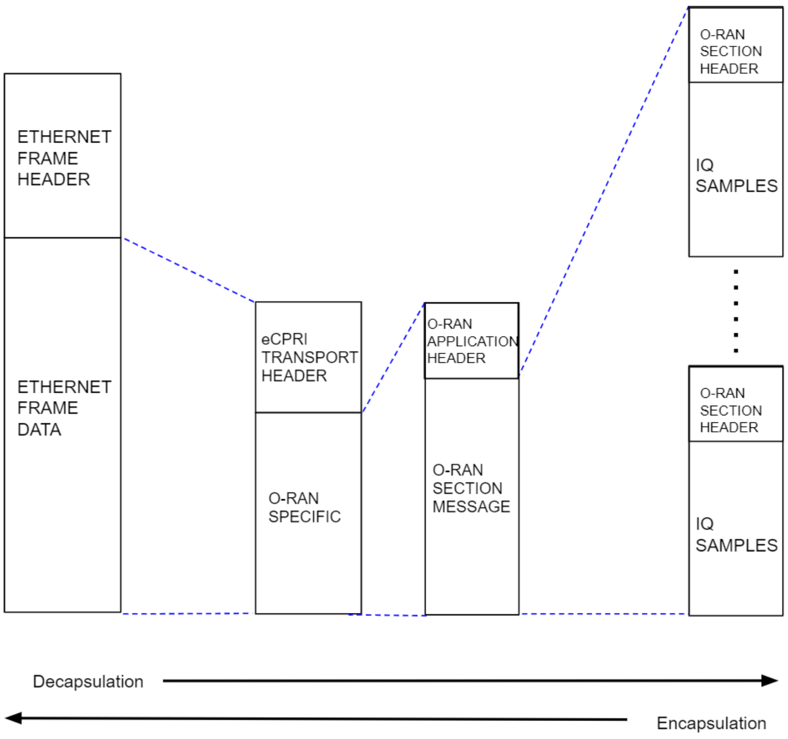


Figure 25. Ethernet eCPRI encapsulation as defined by O-RAN.

3.2.1.1 Ethernet Frame Header

The highest level of O-RAN eCPRI encapsulation happens at the ethernet level. Basic structure of an ethernet frame is shown in figure 26. Along with the ethernet frame header, the figure also depicts other fields that are part of the IEEE ethernet frame format.

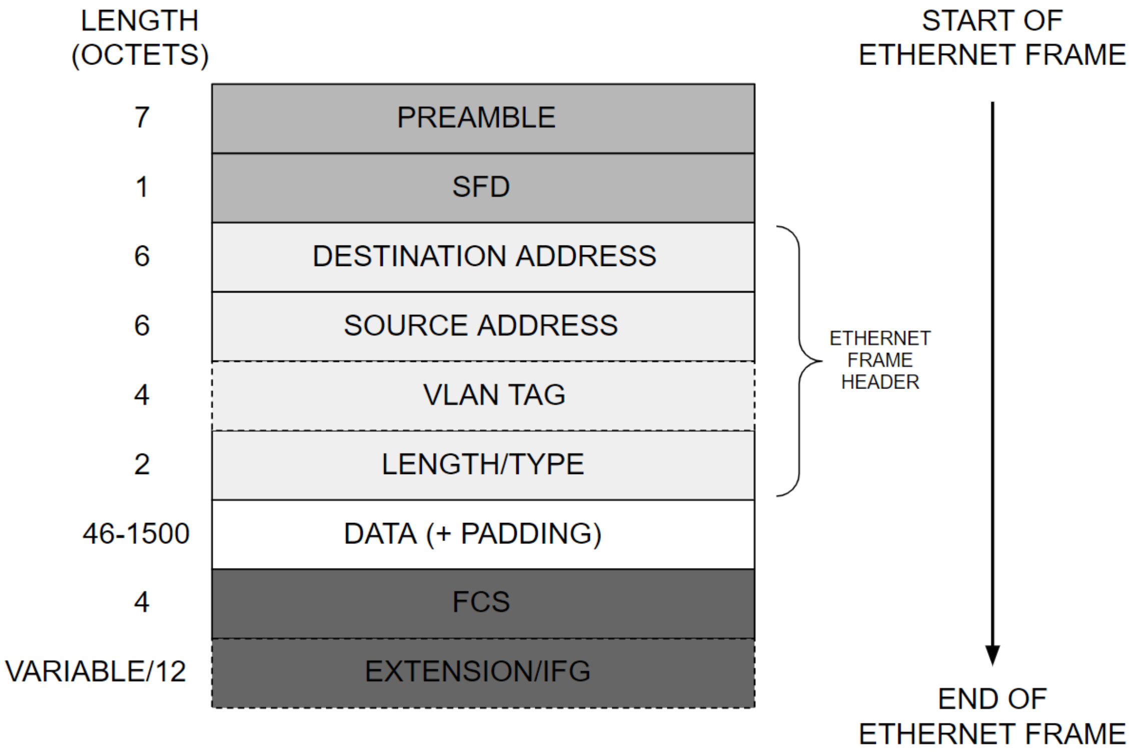


Figure 26. Ethernet frame structure.

The length of each field is described as octets, which corresponds to 8 bits, or one de facto byte. The fields of an ethernet frame include:

- *Preamble*: 7 octets long, used to notify the receiver that a new frame is coming.
- *Start of Frame Delimiter (SFD)*: 1 octet, always set as 10101011. Used to indicate the start of a new frame.
- *Destination address*: 6 octets, contains the MAC address of the receiver's ethernet interface.
- *Source address*: 6 octets, contains the MAC address of the sender's ethernet interface.
- *VLAN tag:* 4 octets, consists of the 2 octets long tag protocol identifier (TPID) and the 2 octets long tag control information (TCI). Only used in VLAN-aware networks.
- *Length/Type*: 2 octets, has two different meanings depending on the value of the field. If the value is less than or equal to the decimal value 1500, it indicates the *length* of the *Data* field in octets. If the value is greater than or equal to the decimal value 1536, it indicates the *type* of protocol that is encapsulated inside the Data field of the ethernet frame, also known as EtherType.
- *Data*: 46-1500 octets, contains the encapsulated data of higher layers. If data length is less than 46 octets, it is padded with zeros to meet the minimum length.
- *Frame Check Sequence (FCS)*: 4 octets, contains cyclic redundancy check (CRC) checksum value, which is used to detect corrupted frames.

- *Extension/Inter-Frame Gap (IFG)*: *Extension* is used to extend the length of the ethernet frame, if the length of the frame is less than the minimum length.

The EtherType for eCPRI is defined as the hexadecimal number 0xAEFE and must be present in the Length/Type field for eCPRI data. To guarantee frames being unambiguously routed, each eCPRI node needs to have a unique MAC address within the context of the ethernet network they are connected to. [51, 52]

3.2.1.2 eCPRI Transport Header

The eCPRI Transport Header is encapsulated within the data field of the Ethernet frame. O-RAN has adapted the fields used in this header straight from the eCPRI specifications. The overall structure of the header is shown in Figure 27.

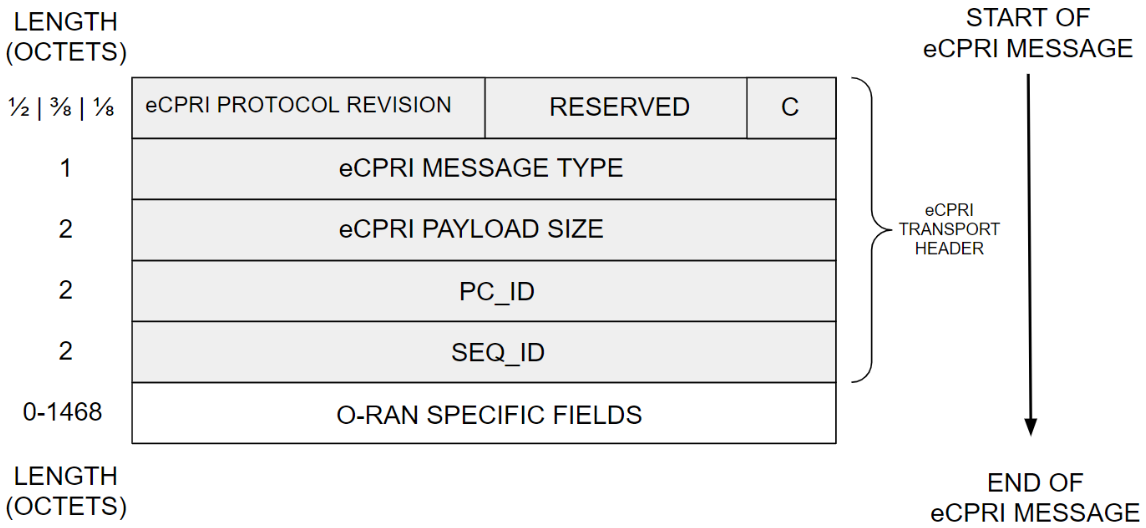


Figure 27. eCPRI Transport Header structure

A brief description for each field of the eCPRI Transport Header is given below:

- *eCPRI protocol revision:* 4 bits, indicates the eCPRI protocol version.
- *Reserved:* 3 bits, reserved for future eCPRI specification use.
- *C:* 1 bit, indicates if concatenation is used. If enabled (set as 1), allows multiple eCPRI messages to be fit within one Ethernet payload.
- *eCPRI message type:* 1 octet, indicates the type of eCPRI service. For IQ data, this field is set as zero.
- *eCPRI payload size:* 2 octets, indicates the length of the eCPRI data payload in octets. Does not include the number of padding octets.
- *PC_ID:* 2 octets, used to identify specific data flows. The field is divided into four smaller sub-fields: RU_port_ID (differentiates logical flows of the RU, such as MIMO layers or PRACH channel), CC_ID (differentiates carrier components),

BandSector_ID (differentiates different bands and sectors supported by the RU) and the DU_Port_ID (differentiates processing units of the BBU).

- *SEQ_ID:* 2 octets, used to identify messages within a series of messages so that the receiver knows when all messages have been received and if re-ordering is needed.
- *O-RAN specific fields:* 1-1468 octets, contains the O-RAN headers and IQ samples.

3.2.1.3 O-RAN Headers and IQ data format

As for the O-RAN specific data of the eCPRI packets, a two-layered header approach is used for U-Plane. This means that the eCPRI data payload consists of two O-RAN headers and IQ-data. The first O-RAN header is the O-RAN Application Header. The main purpose of the application header is to provide crucial information about the location of the data samples so that they can be unambiguously arranged in the time domain. After the application header is the section specific data. The term “section” refers to different types of U-Plane data, for example user data carried in Physical Uplink Shared Channel (PUSCH), or Physical Random-Access Channel (PRACH) data. The section specific message consists of a section header, which identifies the individual sections and indicates the frequency domain allocation of the IQ data, and the actual IQ data samples, which are sent in 12 I and Q sample pairs (one PRB). In a single eCPRI packet, data from only one section type can be included, but multiple section messages from the same type can be combined, each section message having its own section header. [50]

A more detailed look on the different data fields within the O-RAN specific messages are shown in Figure 28. In the figure, different colors are used to identify different headers and sections of the message, yellow being the eCPRI transport header, pink being the application header, darker green (and blue) being the section header, and lighter green (and blue) being the I and Q data samples. It is important to note that the octet numbering of Figure 28 is based on the assumption that the length of each I and Q sample is exactly 1 octet/8 bits. However, the actual length of the samples can be configured between 1 and 16 bits [50].

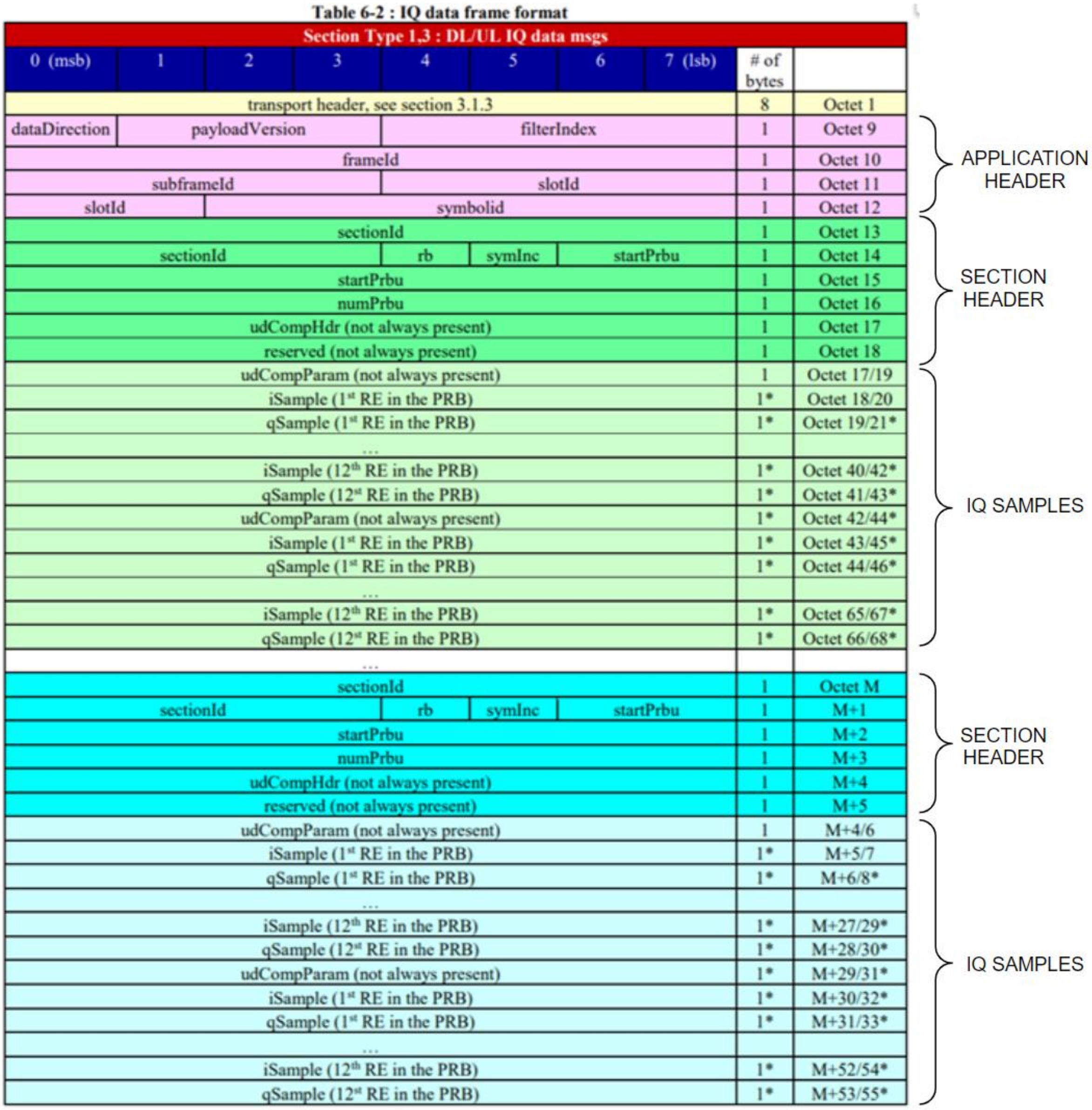


Table 6-2 : IQ data frame format

Section Type 1,3 : DL/UL IQ data msgs									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
transport header, see section 3.1.3								8	Octet 1
dataDirection	payloadVersion			filterIndex				1	Octet 9
frameId								1	Octet 10
subframeId				slotId				1	Octet 11
slotId		symbolid						1	Octet 12
sectionId								1	Octet 13
sectionId				rb	symInc	startPrbu		1	Octet 14
startPrbu								1	Octet 15
numPrbu								1	Octet 16
udCompHdr (not always present)								1	Octet 17
reserved (not always present)								1	Octet 18
udCompParam (not always present)								1	Octet 17/19
iSample (1st RE in the PRB)								1*	Octet 18/20
qSample (1st RE in the PRB)								1*	Octet 19/21*
...									
iSample (12th RE in the PRB)								1*	Octet 40/42*
qSample (12st RE in the PRB)								1*	Octet 41/43*
udCompParam (not always present)								1*	Octet 42/44*
iSample (1st RE in the PRB)								1*	Octet 43/45*
qSample (1st RE in the PRB)								1*	Octet 44/46*
...									
iSample (12th RE in the PRB)								1*	Octet 65/67*
qSample (12st RE in the PRB)								1*	Octet 66/68*
...									
sectionId								1	Octet M
sectionId				rb	symInc	startPrbu		1	M+1
startPrbu								1	M+2
numPrbu								1	M+3
udCompHdr (not always present)								1	M+4
reserved (not always present)								1	M+5
udCompParam (not always present)								1	M+4/6
iSample (1st RE in the PRB)								1*	M+5/7
qSample (1st RE in the PRB)								1*	M+6/8*
...									
iSample (12th RE in the PRB)								1*	M+27/29*
qSample (12st RE in the PRB)								1*	M+28/30*
udCompParam (not always present)								1*	M+29/31*
iSample (1st RE in the PRB)								1*	M+30/32*
qSample (1st RE in the PRB)								1*	M+31/33*
...									
iSample (12th RE in the PRB)								1*	M+52/54*
qSample (12st RE in the PRB)								1*	M+53/55*

Figure 28. IQ data frame format as defined by O-RAN.

Below, a short description of each field defined in the IQ data frame format of O-RAN is given:

Transport header: 1 octet. See Chapter 3.2.1.2.

Application header

- *dataDirection:*1 bit. Direction of data (0: UL, 1: DL)
- *payloadVersion:* 3 bits. Describes the protocol version, set as 001 by default.
- *filterIndex:* 4 bits. Defines the channel filter used between eCPRI and the air interface. For standard PUSCH data, 0000 is used. For PRACH, different values are used to indicate the used PRACH format.
- *frameId:*8 bits. Counter for 10 ms radio frames.
- *subframeId*: 4 bits. Counter for 1 ms sub-frames within a 10ms radio frame.
- *slotId:* 6 bits. Slot number within a 1 ms sub-frame.

- *symbolid:* 6 bits. Symbol number within a slot.

Section header

- *sectionId:* 12 bits. Identifies the individual data sections. Used to map U-Plane data sections to corresponding C-Plane messages.
- *rb:*1 bit. Used to indicate if all resource blocks are used (0), or only every other RB (1).
- *symInc:*1 bit. Used to indicate if the symbol number should be incremented to the next symbol. For U-Plane data, each message is assumed to carry data for only one symbol, so this parameter is set to 0.
- *startPrbu:* 10 bits. Indicates the starting PRB of a U-Plane data section.
- *numPrbu:* 8 bits. Indicates the number of PRBs within the U-Plane data section. The usage of *startPrbu* and *numPrbu* must ensure that no data sections can overlap.
- *udCompHdr:* 8 bits. Defines the compression method (*udCompMeth*) and IQ sample bit width (*udIqWidth*). If compression method is defined in M-Plane, this field is absent.
- *reserved:* 8 bits. Reserved for future use, set as all zeros. If compression method is defined in M-Plane, this field is absent.

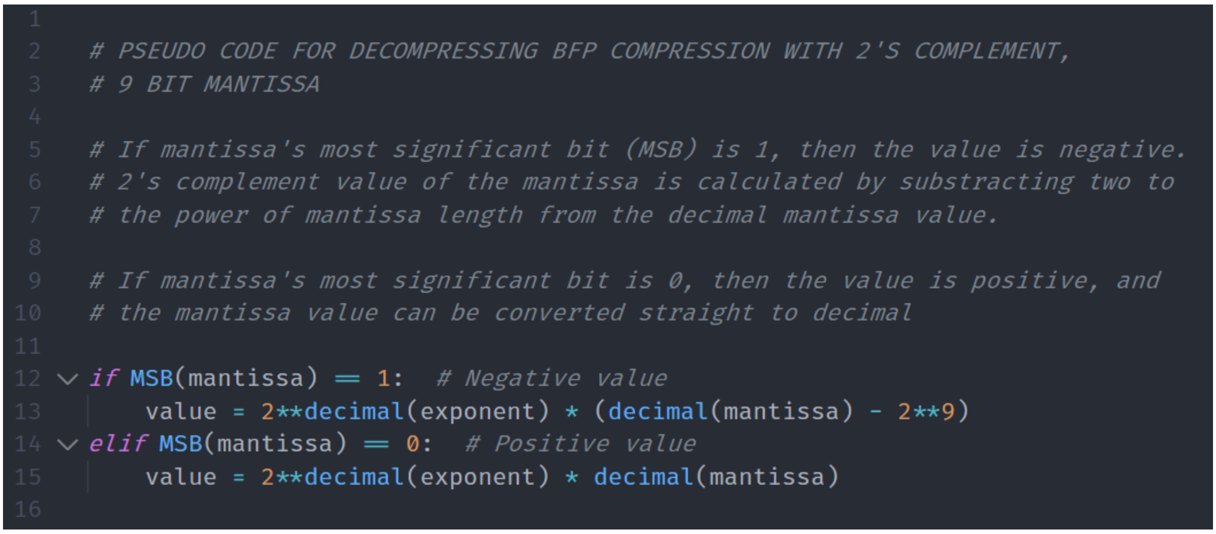
IQ samples

- *udCompParam:* 0-16 bits, depending on the used *udCompMeth.* Provides parameters for the used compression method, i.e. for block floating point (BFP), this field is 1 octet long, first four bits being set as zero and the last four bits indicating the exponent value for the next 12 IQ value pairs (1 PRB).
- *iSample:* 1-16 bits, defined by *udIqWidth*. In-phase sample of one RE. For BFP, this field is the mantissa of the I value.
- *qSample:* 1-16 bits, defined by *udIqWidth*. Quadrature sample of one RE. For BFP, this field is the mantissa of the Q value.

In this thesis, it is assumed that the IQ compression method is defined as BFP compression in M-Plane messages, meaning that both *udCompHdr* and *reserved* fields are absent from the eCPRI packets. BFP compression is a method of implementing floating point arithmetic on a fixed-point processor. The block floating point values consist of two parts: the mantissa, which represents the digits of the value in 2’s complement format, and the exponent, which represents the magnitude of the value. Therefore, the *iSample* and *qSample* fields represent the digit value of I and Q samples in 2’s complement format, the most significant bit (MSB) indicating the sign of the value (MSB 1 implying a negative value and MSB 0 a positive value). The actual I and Q value scale according to the exponent value. Equation 10 shows the general decompression format of a block floating point number. [50]

$$value = 2^{exponent} * mantissa \tag{10}$$

The pseudo-code shown in Figure 29 demonstrates how the sample value is calculated from a BFP compressed, 9-bit long mantissa with 2’s complement representation. If the sign of the mantissa is indicated as negative with MSB 1, the 2’s complement if calculated by subtracting two to the power of 9 (length of the mantissa) from the decimal value of the 9-bit mantissa.



```

# PSEUDO CODE FOR DECOMPRESSING BFP COMPRESSION WITH 2'S COMPLEMENT,
# 9 BIT MANTISSA

# If mantissa's most significant bit (MSB) is 1, then the value is negative.
# 2's complement value of the mantissa is calculated by substracting two to
# the power of mantissa length from the decimal mantissa value.

# If mantissa's most significant bit is 0, then the value is positive, and
# the mantissa value can be converted straight to decimal

if MSB(mantissa) == 1:  # Negative value
    value = 2**decimal(exponent) * (decimal(mantissa) - 2**9)
elif MSB(mantissa) == 0:  # Positive value
    value = 2**decimal(exponent) * decimal(mantissa)

```

Figure 29. 2’s complement calculation for mantissa size of 9 bits.

3.3 Uplink Datapath Testing

Most communication networks are built based on the assumption that the users need more resources for receiving data in downlink direction than for uploading data in uplink direction. However, with the boom of mobile livestreaming and AR/VR applications, the data rate and latency requirements have made UL arguably as important as DL. Although the most applications still receive more data in downlink than they transmit in uplink, the role of uplink as the “initiator of content” has risen: if uplink is slow, the time requesting and receiving content becomes noticable. Therefore, validating UL performance and features has become more crucial.

3.3.1 Test setup

The test setup used in this thesis is shown in Figure 30. The setup is a basic conducted test setup for datapath testing in radio unit integration. Compared to a real network, the UE is replaced with a signal generator/analyser and the BBU is replaced by a BBU emulator. The testing is done in conducted mode, meaning that RF cables are used to replace the real-life over-the-air radio channel. The term “datapath” refers to the route that the data takes when it goes through the radio.

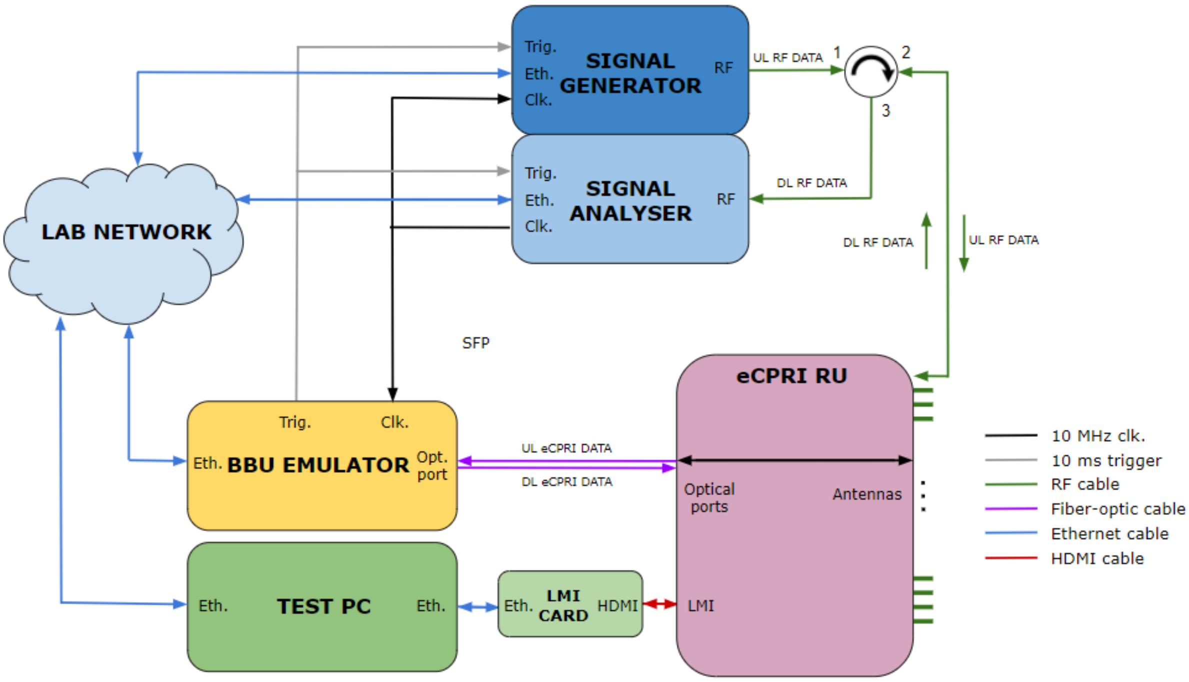


Figure 30. Radio unit test setup for datapath testing.

The datapath test setup consists of different test hardware, each fulfilling a specific task. A signal generator is used to generate the UL RF signal which emulates the wireless signal transmitted by the UE. The generator is connected to one or more antenna ports of the radio unit. As for receiving and analysing DL data, a signal analyser is connected to the same antenna ports as the generator. This poses a danger to damage both the signal generator and analyser by letting in power from unwanted directions, especially if the radio operates in TDD mode where the UL and DL both use the same RF cable to carry data to/from the radio's antenna ports. Therefore, a circulator must be used to block the signal from being routed to wrong ports. On the fronthaul side of the radio, the BBU emulator is connected to the radio's optical ports via optical fiber. The role of the BBU emulator is to generate eCPRI formatted data in DL, and capture eCPRI packets in UL for further analysis. The conventional capture format is the Packet Capture (PCAP) file format, which stores the captured packets in a binary format.

In UL testing, the signal analyser is only used to provide a 10 MHz reference clock signal for the generator and the BBU emulator to keep them synchronized. The radio uses over-the-fiber synchronization methods, e.g. Precision Time Protocol (PTP), to synchronize itself with the BBU emulator. To ensure that the TDD pattern is obeyed and no overlapping between DL and UL transmission happens, the generator must know the precise moment when to transmit data to the radio. The correct timing is provided by the BBU emulator, which maintains a 10 ms frame trigger signal to the generator and the analyser. All configurations to the test setup are done from a test PC, which has a remote connection to the test hardware via the laboratory network, and a direct Local Area Network (LAN) connection to the radio unit via the Local Management Interface (LMI) card.

3.3.2 Key performance indicators

After the UL data capture has been retrieved with the BBU emulator, the quality of the data is analysed with the key performance indicators (KPIs).

3.3.2.1 Signal power and power budget

Determining the signal power is a good first indicator for verifying data quality. Moreover, validating the IQ power level at the eCPRI interface is especially important in O-RAN radios: due to openness of the interface and vendor interoperability requirements, the power level needs to be exactly at the level specified by the O-RAN specs. O-RAN defines the IQ power level of one subcarrier (or resource element) in decibels relative to full-scale (dBFS) as shown in Equation 11. The unit dBFS is a logarithmic measurement unit which assigns 0 dBFS as the maximum power level. For example, a power level of -15 dBFS indicates that the measured power is 15 dB below the maximum subcarrier power. Parameter FS is the full-scale reference value of one I or Q value squared: with a signed 9-bit mantissa and a 4-bit exponent, FS equals $(2^{2^4-1} * 2^{9-1})^2 = 2^{46}$. [50]

$$\begin{aligned} P_{dBFS} &= 10 * \log_{10}(I^2 + Q^2) - 10 * \log_{10}(FS) \qquad (11) \\ &= 10 * \log_{10}\left(\frac{I^2 + Q^2}{FS}\right) \end{aligned}$$

To verify that the power level is at the correct level, the dBFS power calculated at the eCPRI interface (P_{dBFS}) needs to be compared to the power level of the signal transmitted by the signal generator (P_t). To do this, all the gains and losses of the test setup, also known as the power budget, need to be estimated. Figure 31 shows the points in the signal path where the transmitted power (P_t) is amplified and attenuated.

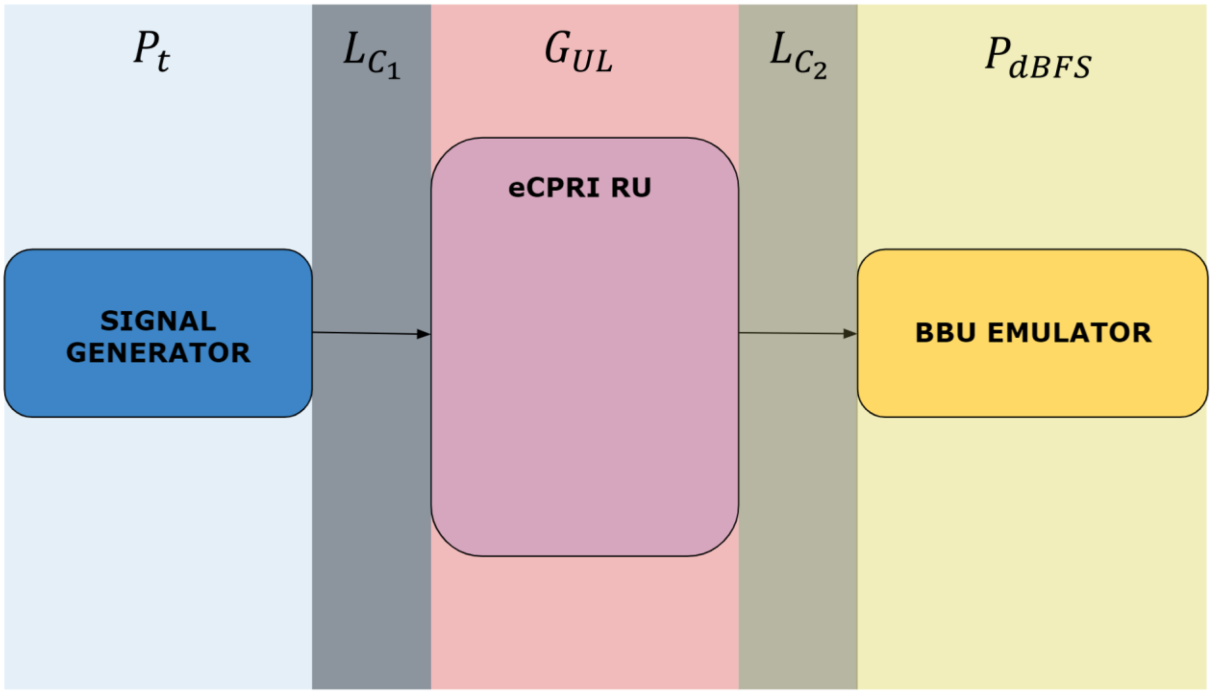


Figure 31. Test setup power budget.

Before the signal reaches the antenna ports of the radio, it undergoes attenuation by different components (L_{C_1}). The absolute attenuation depends on multiple factors, such as the RF cable length, external attenuators and circulator loss. Within the radio, the UL signal power is scaled according to O-RAN specifications, as shown in Equation 12.

$$G_{UL} = R_{interface(dBFS)} + G_{ref(dBm)} + G_{corr(dB)} \tag{12}$$

Uplink gain, G_{UL}, consists of the sum of three parameters. $R_{interface}$ determines the smallest non-zero IQ power level that can be carried by one subcarrier and is defined as $10 * \log_{10}\left(\frac{1}{FS}\right)$ dBFS. G_{ref} is a reference power level, set as 152 dBm by O-RAN. G_{corr} is an additional gain correction parameter that is conventionally set as zero. [50]

The losses between the radio's fronthaul interface and the data capture point in the BBU emulator are marked as L_{C_2} in Figure 31. However, this connection is usually estimated as lossless due only consisting of optical fibre and optical ports. Thus, the overall power budget can be express as shown in Equation 13.

$$P_{dBFS} = P_t - L_{C_1} + G_{UL} - L_{C_2} \tag{13}$$

$$P_{dBFS} = P_t - L_{C_1} + \left(R_{interface(dBFS)} + G_{ref(dBm)} + G_{corr(dB)}\right)$$

$$\Rightarrow P_t = P_{dBFS} + L_{C_1} - \left(R_{interface(dBFS)} + G_{ref(dBm)} + G_{corr(dB)}\right)$$

If the power level P_t calculated from the eCPRI interface power P_{dBFS} matches the actual transmission power of the generator, the gain applied by the radio is set up according to O-RAN specifications.

3.3.2.2 Symbol timing offset

When the RU receives the waveform transmitted by the UE, the optimal signal sampling would happen exactly at the signal peaks. However, this would require both the RU and UE to be synchronized to the same physical master clock, which is not an option in a wireless system. Additionally, phenomena such as multipath propagation and Doppler shift introduce their own share of randomness in the signal sampling due intersymbol interference (ISI) and intercarrier interference (ICI). Even in conducted testing environment, where the effects of the wireless channel are negated, timing offsets happen simply due thermal shift in analogue components of the radio. Therefore, timing offsets are considered as “fact-of-life” in wireless systems. However, with the usage of Cyclic Prefix in OFDM systems (CP-OFDM), the system is made more robust against timing offsets with the slight cost of overall data throughput decrease.

The effect of timing offset can be observed in the constellation of the received signal, as shown with a 16QAM signal in Figure 32. When the symbol timing is ideal, the received IQ data samples fall into the 16QAM reference points. When the signal is sampled with a sample offset, the constellation starts to rotate. However, if the sampling offset gets larger than the CP length, the circularity between the transmitted signal and channel impulse response is lost and signal quality degrades due ISI. [53]

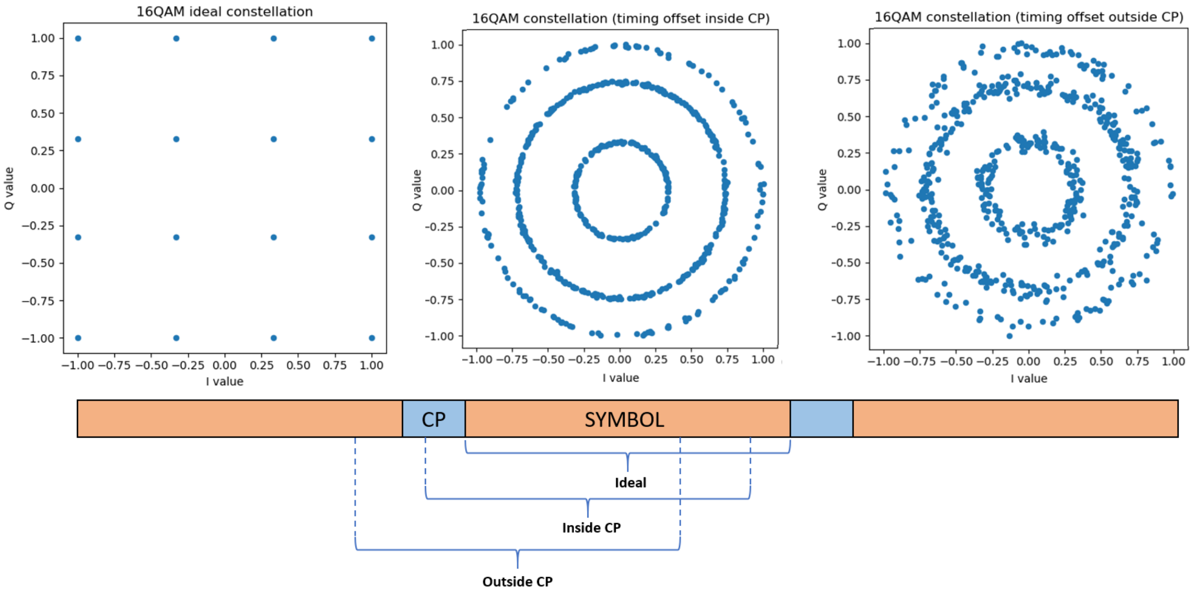


Figure 32. Effect of symbol timing offset on constellation.

The rotation phenomenon can be explained with the properties of FFT transformation. Equation 14 shows how the time domain sampling offset affects the frequency domain symbol X_k.

$$X_{k,N_{off}} = \sum_{n=0}^{N-1} x_{n+N_{off}} e^{-\frac{i2\pi k(n+N_{off})}{N}} = \sum_{n=0}^{N-1} x_{n+N_{off}} e^{-\frac{i2\pi kn}{N}} e^{\frac{i2\pi kN_{off}}{N}} \tag{14}$$

$$= X_k e^{\frac{i2\pi kN_{off}}{N}}, \qquad k = 0,1,\dots,N$$

where $X_{k,N_{off}}$ is one OFDM symbol, affected by an offset of N_{off}, and $x_{n+N_{off}}$ is one time domain sample, sampled with an offset of N_{off}. Each de-modulated subcarrier experiences a unique phase shift, as denoted by the second exponential part of equation 14.

In wireless systems, the timing offset is conventionally detected and estimated by using known patterns in the received signal called reference signals. 3GPP has defined three different reference signals for 5G NR uplink transmissions: Demodulation Reference Signal (DMRS), Phase-tracking Reference signal (PTRS) and Sounding Reference Signal (SRS). DMRS provides channel estimates to the receiver and is used for the demodulation of data from the corresponding channel. PTRS is used to estimate oscillator phase-noise, especially in the mmW frequencies where the effect of phase noise is notable. SRS provides information about the channel over the full channel bandwidth and is used in resource allocation. The focus of this thesis is in DMRS since it provides the needed information for timing and phase offset estimation while being present within all PUSCH transmissions. [54]

The PUSCH DMRS symbols are mapped to the resource grid according to 3GPP specifications ([55]). In short, the mapping depends on four main parameters:

- Mapping type (A or B): Defines the time domain position of the first DMRS symbol within a slot.
- Configuration type (1 or 2): Defines the frequency domain density of the DMRS symbols.
- Additional position (0, 1, 2 or 3): Defines the number of DMRS symbol occurrences in time domain.
- Maximum length (single or double): Defines if a single DMRS occurrence occupies one symbol or two consecutive symbols.

Figure 33 shows an example on how the DMRS symbols are mapped into one uplink data slot/resource block with mapping type A, configuration type 1, additional position 1 and single symbol maximum length.

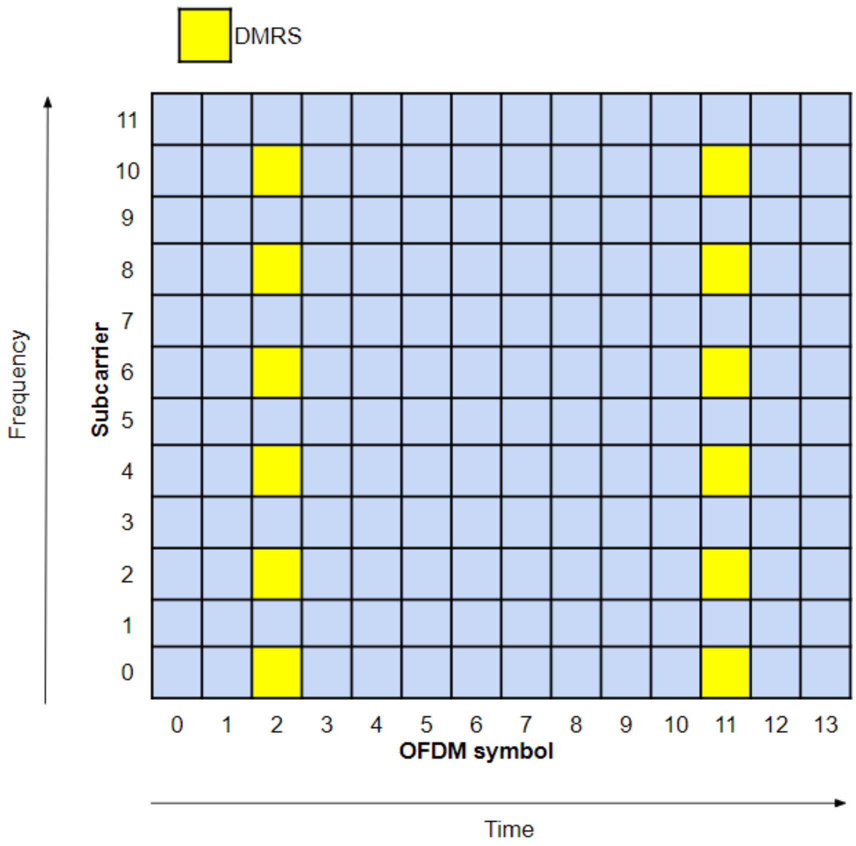


Figure 33. DMRS symbol mapping for mapping type A, configuration type 1, additional position 1 and single symbol maximum length.

The reason why DMRS signals are particularly useful in channel estimation and demodulation is that they are created using Zadoff-Chu sequences. Zadoff-Chu sequences are Constant Amplitude Zero Autocorrelation (CAZAC) type sequences which possess attractive properties for channel estimation. These sequences have a constant amplitude which reduces signal PAPR, thus enabling efficient power amplifier usage. Additionally, Zadoff-Chu sequences possess ideal auto-correlation properties: when autocorrelating the signal with a time-shifted (cyclically shifted) version of itself, the result is zero. In practice this means that the timing offset between Zadoff-Chu sequences can be acquired accurately just by cross-correlating the received signal $s_r(n)$ with the transmitted reference sequence $s_{ref}(n)$ in time domain. Alternatively, cross-correlation between two signals can be expressed as a convolution by time-reversing (complex conjugating) the other signal, which can be further expressed as a pointwise multiplication operation in frequency domain by using the circular convolution theorem (as discussed in Chapter 2.3.2.2). This process is shown in Equation 15. [56, 57]

$$s_r(n) \star s_{ref}(n) = s_r(n) \circledast \overline{s_{ref}(n)} \leftrightarrow S_r[k] \cdot \overline{S_{ref}[k]} \qquad (15)$$

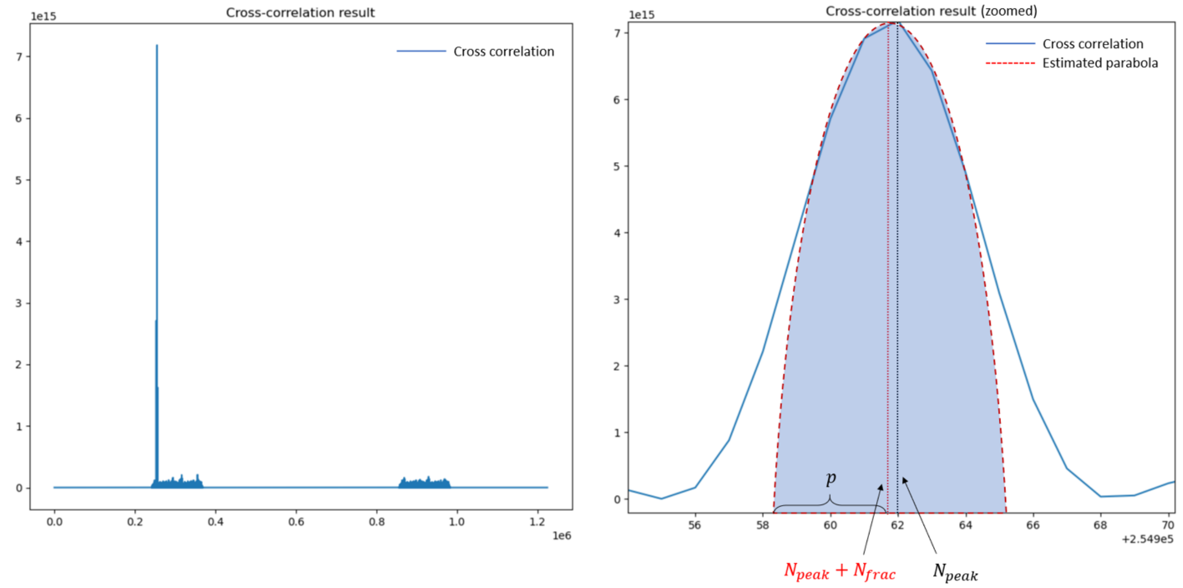
where $\star$ denotes the cross-correlation operation, $\circledast$ the autocorrelation operation, $\leftrightarrow$ the FFT transformation, and $\overline{s_{ref}(n)}$ the complex conjugate of the transmitted sequence $s_{ref}(n)$.

If the reference DMRS symbols are detected within the received signal, the cross-correlation produces a clear peak indicating the DMRS symbol position in the received data. The sample value corresponding to this peak (N_{peak}) can be calculated as shown in Equation 16. Since cross correlation can return both negative and positive values, the absolute value is taken from the result.

$$N_{peak} = argmax(|s_r(n) \star s_{ref}(n)|) \tag{16}$$

where *argmax* returns the sample value which corresponds to the maximum value of $|s_r(n) \star s_{ref}(n)|$ [58].

Equation 16 always returns an integer value for N_{err}, which is not the accurate value for the cross-correlation peak. This issue is illustrated in Figure 34 which plots the absolute value of an example cross correlation (left side figure). Since ross correlation is evaluated with integer sample accuracy, the correlation peak is quite angular, as shown in the zoomed-in plot of Figure 34. In reality, the peak should be a round parabola. In Figure 34, this observation is highlighted with an artificial parabola representing the sub-sample peak of the cross-correlation. The fractional part of the timing offset can be calculated with a center-of-mass algorithm as shown in Equation 17 [59].



(a) Whole cross correlation.

(b) Zoomed-in cross correlation peak.

Figure 34. Absolute value of cross correlation.

$$N_{frac} = \frac{\sum_{l=N_{off}-p}^{N_{off}+p} |s_r(l) \star s_{ref}(l)|^2 * l}{\sum_{l=N_{off}-p}^{N_{off}+p} |s_r(l) \star s_{ref}(l)|^2} \tag{17}$$

where N_{frac} is the fractional timing offset and p is the approximated half-width of the peak.

By adding the fractional offset value to the integer peak value of the sample offset, the index of the “real” peak is acquired. When the expected position of the DMRS symbols is known, the sampling offset of the received data can be acquired as shown in Equation 18.

$$N_{off} = N_{peak} + N_{frac} - N_{DMRS} \tag{18}$$

where N_{DMRS} is the expected sample position of the DMRS symbols.

After acquiring the sampling offset, the samples of each subcarrier can be "de-rotated" back to their original places. This procedure is shown in Equation 19, in which the rotation caused by sampling offset N_{off} is corrected by multiplying the samples with an exponential term, opposite to the rotation.

$$X_k e^{\frac{i2\pi k N_{off}}{N}} e^{\frac{-i2\pi k N_{off}}{N}} = X_k \tag{19}$$

3.3.2.3 Phase offset

Typically, in wireless systems, the phase of the received signal is unknown at the receiver. This causes the down-converted baseband signal to experience a common phase rotation: all REs are rotated with the same amount of phase offset in the constellation plane. This effect is shown in Figure 35. Although common phase offset does not necessarily tell much about the signal quality, detecting and correcting it is very important before further analysis on the data can be done. Mathematically speaking, the phase offset φ_{off} can be expressed as an exponential multiplication in both time and frequency domain, as shown in Equation 20. [59]

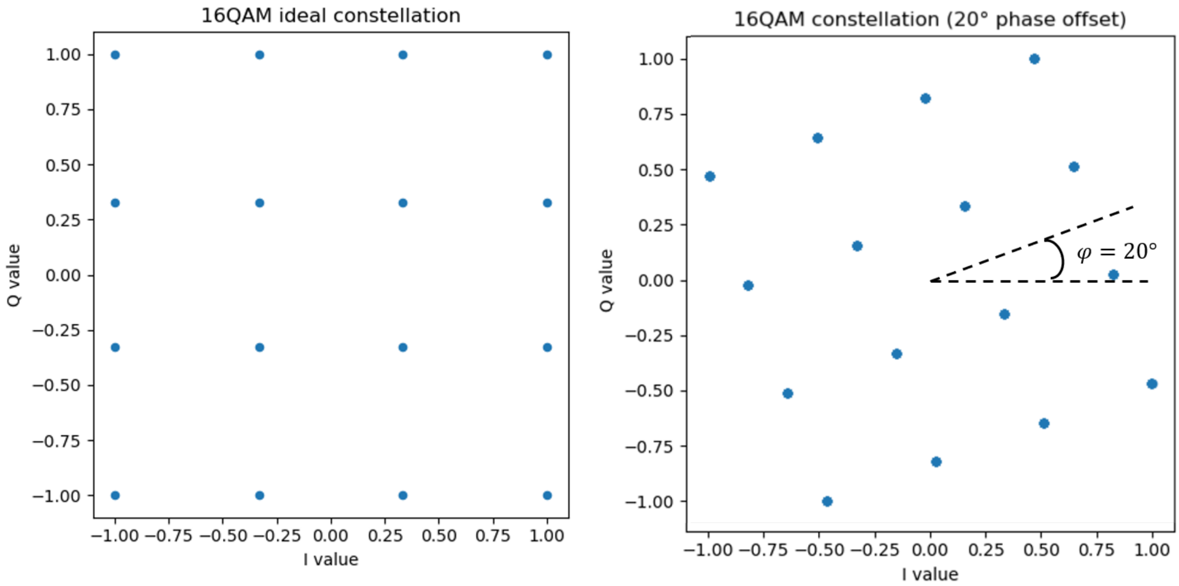


Figure 35. Effect of carrier phase offset on constellation.

$$X_{k,\varphi_{off}} = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi kn}{N}} e^{i\varphi_{off}} = X_k e^{i\varphi_{off}}, \qquad k = 0,1,\dots,N \tag{20}$$

where φ_{off} is the phase offset in radians.

Common phase offset can be recovered with similar methods as used in timing offset estimation. Instead of estimating the peak position of the cross-correlation between the received and reference signals, the point of interest is the angle of the peak, also known as complex argument arg. This calculation is shown in Equation 21. [60]

$$\varphi_{off} = arg\left(\left|s_r(n) \star s_{ref}(n)\right|\right) \tag{21}$$

As the phase offset affects all the subcarriers equally, it is corrected by rotating all samples by the opposite amount of phase shift, like shown in Equation 22.

$$X_k e^{i\varphi_{off}} e^{-i\varphi_{off}} = X_k \tag{22}$$

3.3.2.4 Error Vector Magnitude

After the timing and phase offsets have been corrected on the captured data, the quality of the received signal can be analysed. A common quality metric used in modern OFDM systems is the Error Vector Magnitude (EVM). EVM is a compact metric to measure the effects of different impairments of the wireless system. Even in conducted testing environment, where the effects of the over-the-air channel are not present, the signal experiences some distortion due inevitable RF impairments in the analogue components of the receiver chain. [61]

Figure 36 illustrates EVM in the constellation plane. When a received sample undergoes amplitude and/or phase distortion, it deviates from the corresponding reference point in the constellation plane. This deviation is represented as an error vector between the received and reference constellation point, EVM being the magnitude of the error vector. The fact that EVM describes both amplitude and phase errors makes it especially figurative metric for determining signal quality.

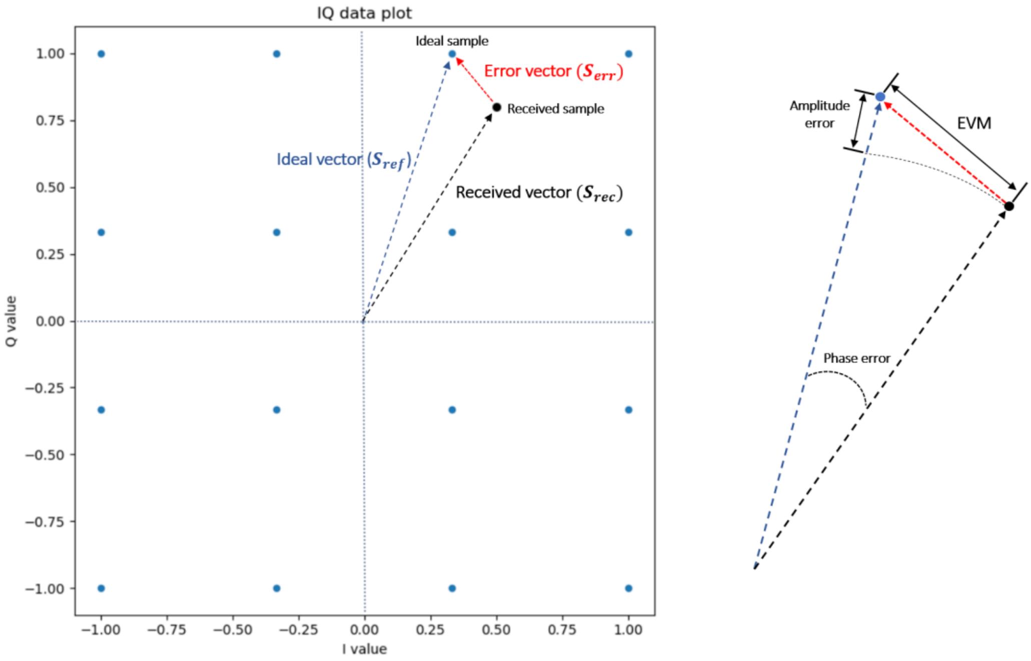


Figure 36. EVM illustrated in the constellation plane.

In high order modulations, i.e. 256QAM, the constellation points are spaced very close to each other, making less room for errors. Therefore, EVM requirements for high modulation orders are tougher than for low modulation orders. Since O-RAN doesn't explicitly define target EVM levels for the eCPRI interface, the signal quality requirements are usually defined in radio specifications. EVM is conventionally expressed in percentages as shown in Equation 23. [60]

$$EVM(\%) = \frac{\sqrt{\frac{1}{N}\sum_{n=0}^{N-1}\left|S_{ref}(n) - S_r(n)\right|^2}}{\sqrt{\frac{1}{N}\sum_{n=0}^{N-1}\left|S_{ref}(n)\right|^2}} \times 100\% \tag{23}$$

$$= \sqrt{\frac{\frac{1}{N}\sum_{n=0}^{N-1}|S_{err}(n)|^2}{\frac{1}{N}\sum_{n=0}^{N-1}\left|S_{ref}(n)\right|^2}} \times 100\%$$

In the above equation, S_{ref} is the reference vector, S_r is the received vector, S_{err} is the error vector, and *N* is the number of samples used in the calculation.

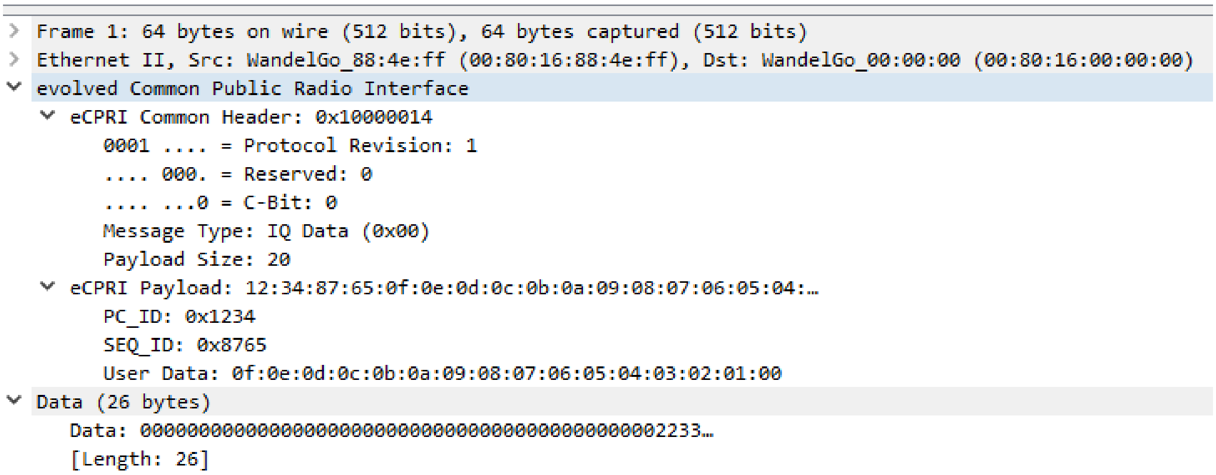
4 CURRENT ECPRI DATA ANALYSIS TOOLS

When radio's UL datapath is tested in unit integration, the end goal is to validate that the quality of data is at required level when it enters the fronthaul interface. However, there are not many analysis tools available for eCPRI data analysis at unit integration level, and the existing tools are missing some essential analysis features. Moreover, the most tools that are used currently in manual testing are not automation friendly. Thus, a new tool is developed in this thesis, specifically for analysing UL data that is captured from the O-RAN based eCPRI fronthaul. In this chapter, the currently used eCPRI analysis tools are evaluated by listing their main advantages and disadvantages. The names of inhouse tools have been abstracted, and figures presented in this chapter have not been directly generated with them: instead, plots have been re-created manually just to give an impression of how the tools presents the results.

4.1 Wireshark

Wireshark is a free, open-source network packet analyser tool. It is commonly used for live network monitoring and traffic capturing, supporting a wide variety of different network interfaces such as ethernet and Wireless Local Area Network (WLAN). Moreover, Wireshark is also able to analyse PCAP formatted captures taken from external network interfaces.

For eCPRI analysis, Wireshark has an official plugin that has capabilities to decode eCPRI packets which have been formatted according to eCPRI specifications [62]. This means that the plugin can parse the fields of the eCPRI Transport header. While this is useful in the early phases of eCPRI protocol analysis, the plugin does not provide any support of O-RAN specific headers, making the plugin insufficient for any further analysis. Luckily, Wireshark offers a way to create custom-plugins for data-parsing, and plugins have been created for parsing O-RAN formatted eCPRI data. These plugins are very handy for validating the ethernet header fields, eCPRI transport header fields and O-RAN specific header fields. The biggest limitation with Wireshark plugins is that they are not very dynamic in terms changing the hard-coded configurations: for example, if the mantissa size of the IQ samples is changed, the user needs to use a separate Wireshark plugin that supports the new mantissa size. In Figure 37, an example eCPRI PCAP capture file (.pcap) is parsed using the official Wireshark eCPRI plugin.



```
> Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
> Ethernet II, Src: WandelGo_88:4e:ff (00:80:16:88:4e:ff), Dst: WandelGo_00:00:00 (00:80:16:00:00:00)
v evolved Common Public Radio Interface
  v eCPRI Common Header: 0x10000014
      0001 .... = Protocol Revision: 1
      .... 000. = Reserved: 0
      .... ...0 = C-Bit: 0
      Message Type: IQ Data (0x00)
      Payload Size: 20
  v eCPRI Payload: 12:34:87:65:0f:0e:0d:0c:0b:0a:09:08:07:06:05:04:…
      PC_ID: 0x1234
      SEQ_ID: 0x8765
      User Data: 0f:0e:0d:0c:0b:0a:09:08:07:06:05:04:03:02:01:00
v Data (26 bytes)
    Data: 000000000000000000000000000000000000000000002233…
    [Length: 26]
```

Figure 37. eCPRI packet parsed by Wireshark

In automatization perspective, the packet parsing functionalities of Wireshark can be integrated to many different programming languages via available wrapper modules, such as python-libcap for python and Jpcap for java. These wrappers enable the usage of Wireshark in a way that the data can be parsed to automated environments. However, the wrapper libraries itself require operating system specific PCAP libraries to be installed on the host machine, making the automation setup platform specific. [63]

Wireshark is quite useful for checking the header fields of the captured eCPRI traffic. However, it is not an IQ data analysis tool, and should not be expected to become one. Capabilities to analyse IQ data are limited to checking if the IQ samples have any data in them. Further analysis regarding the quality of data is not possible to do within the tool, making it inadequate to use after the very initial stage of eCPRI data validation where metrics such as power level and EVM become the point-of-interest.

4.2 Analysis tool A

Analysis tool A is a Nokia inhouse tool, originally developed for LTE data analysis and visualisation. The tool is known for its extensive features, supporting multiple different data formats, LTE/NR carrier configurations and signal types. Due to ease-of-access via almost any internet browser, the tool has become extremely popular analysis tool at every stage of integration. Here, the most important features of the tool are introduced from radio unit integration perspective.

For 5G eCPRI analysis, Analysis tool A offers data parsing functionalities, as well as some signal analysis features. The so-called Trace View enables the capture to be parsed and viewed similarly to Wireshark, while introducing carrier- and protocol-specific parametrization, such as RAT technology and mantissa size. Compared to Wireshark, the Trace View of Analysis tool A is much more effortless to use in general protocol analysing due having extensive header filtering capabilities for O-RAN based eCPRI packets. In radio unit integration perspective, the Trace View is mostly used to check the various header fields of the eCPRI packet.

Another useful view for radio unit integration is the IQ constellation view, which plots the I and Q samples of the capture in the constellation plane. In most cases, the constellation diagram of the captured data ends up looking like circles, like shown in Figure 38 for a 16QAM modulated data capture. This is because the tool does not have the capability to correct timing nor phase offsets, resulting in a raw constellation plot with rotated IQ samples. This view can be used as a quick sanity check for UL data quality: If the “doughnuts” are sharp and clear, the data has only rotated in phase, and very likely could be un-rotated back by correcting timing and phase offsets. On the other hand, this sort of analysis is not 100% reliable due relying on visual analysation of the constellation circles and does not provide any concrete metric for automatic analysis.

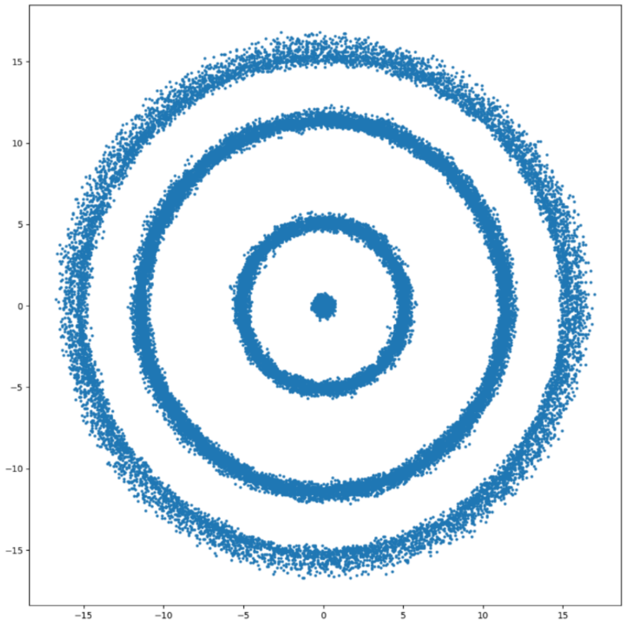


Figure 38. Raw IQ constellation plot.

A mention-worthy feature of Analysis tool A is the resource grid plotting, which arranges the captured IQ samples into a 5G NR time-frequency grid. This enables users to visualize which symbols and subcarriers are carrying data and can be used to validate radio features such as TDD switching. Different symbols of the grid can be colored based on their amplitude, making the resource grid plot a handy way of visualizing the location of reference symbols such as DMRS.

Overall, Analysis tool A is the most versatile tool that is available for eCPRI data analysis. Unfortunately, the necessary timing and phase offset estimation and correction features are missing, making proper data analysis such as EVM measurement impossible. Also, the tool does not offer any application programming interface (API) for remote or script usage, which makes integration to automated environments difficult.

4.3 Analysis tool B

The last available tool covered in this thesis is a Nokia's inhouse tool called Analysis tool B. It is a MATLAB based tool, designed specifically for IQ data analysis from captures, including most of the necessary features from decoding eCPRI formatted data to EVM measurement. Even though the tool is made in MATLAB, its usage does not necessitate owning a MATLAB licence due to being a compiled, standalone application. Instead, the users only need the MATLAB Compiler Runtime (MCR) which is available for free. However, since compiled versions of MATLAB scripts are operating system specific, the correct version of the tool needs to be installed for the operating system in use.

The main feature of Analysis tool B is the PCAP capture parser and analyser. The tool allows users to configure most of the important parameters for IQ analysis, including carrier specific parameters and mantissa size. Although straightforward to use, the user interface can sometimes be quite clunky due to MATLAB not offering adequate enough GUI creation tools.

As for the UL data analysis at the unit integration level, Analysis tool B has almost all desirable features covered. The tool is capable to calculate timing offset, power level and EVM for the capture, along with other useful features such as plotting the frequency spectrum and time domain view of the IQ data. In addition to overall frame analysis, the tool can also do symbol-by-symbol analysis. Visually, the results are presented clearly, and most of the plots are relevant for data analysis on radio unit integration level. However, the tool cannot plot the 5G NR resource grid view of the captured data, which is a notable downside.

In addition to the GUI, Analysis tool B offers an interface for automated usage. The interface is run as a separate background process and the communication with it is done by executing commands on the command line. The results of the analysis are outputted into multiple text files, and the tool also saves all the generated MATLAB figures as screenshots.

While being the most suited data-analysis tool available for eCPRI captures, Analysis tool B has achieved limited popularity due to requiring relatively complex, platform specific setup and having a clunky MATLAB user interface. Additionally, the source code of the tool is not distributed, making any local modifications impossible.

4.4 Summary

Table 2 summarises the advantages and shortcomings of the three covered eCPRI analysis tools used in radio unit integration. Overall, none of the tools satisfy all the needs of eCPRI protocol and IQ data analysis in radio unit integration level. Thus, a new tool is required to combine all the necessary features of the existing tools, while also covering their main shortcomings.

Table 2. Summary of existing eCPRI analysis tools.

Tool	**Advantages**	**Shortcomings**
Wireshark	+Supports PCAP captures +Custom plugin development support +Good for quick header field checks +Wrapper modules exist for automatization	-No IQ data analysis features -Not dynamic in terms of changing field bit-widths -Automatization requires platform-specific PCAP libraries
Analysis tool A	+Supports PCAP captures +Quick and easy to use via any web browser +Good for quick header field checks +IQ power level estimation +Visual IQ data analysis with raw IQ constellation plots +Resource grid view	-No correction possibility for timing or phase offsets -No support for EVM measurement -No possibility to integrate to external scripts for automation
Analysis tool B	+Supports PCAP captures +Necessary IQ data analysis features: timing offset, power level, EVM +IQ data plots with corrected timing and phase offsets +Symbol specific analysis +Interface for automatic usage	-Clunky MATLAB interface -Platform specific MCR setup -Platform specific executables -No resource grid plot

5 IMPLEMENTATION OF ECPRI PCAP ANALYSER

Every tool needs a name that is simple but descriptive. For the data analysis tool developed in this thesis, the name "eCPRI PCAP Analyser" is chosen because it covers the main purpose of the tool while being easy to remember. As for the actual implementation, creating a solid design plan is the most crucial part of developing any tool: especially when a tool is created from the ground-up, like in this thesis. Not only should the design choices be made in a way that the tool can achieve what it is intended to do, but also to support effortless implementation of future features. In this chapter, the implementation of the new tool is then introduced by listing the feature requirements and presenting the selected design methods, programming language, high-level code structure and user interfaces. The tool is then evaluated using two main metrics: significance for testing and performance.

5.1 Requirements

Starting from the main requirements, the eCPRI PCAP Analyser is specified to include the following features and design choices:

- O-RAN based eCPRI data parsing
    - Ethernet Frame Header fields
    - eCPRI Transport Header fields
    - O-RAN Application and Section Header fields
    - I and Q sample extraction, decompression and IQ sample formation
- PUSCH (user data) analysis
    - Ideal radio channel can be assumed → No need to equalize effects such as carrier frequency offset or phase noise.
    - Signal power level calculation
    - Symbol timing offset estimation and correction
    - Carrier phase offset estimation and correction
    - EVM calculation
    - Constellation plot
    - Time-frequency resource grid plot
- Support for all 5G NR carrier configurations
- Support for any QAM modulation order
- Analysis is based on DMRS reference symbols
    - DMRS symbols loaded from the reference data that the user provides
- Command line interface (CLI) for manual usage
    - Users specify required parameters with command line arguments
- Integration to other scripts possible for automated usage
- Implementation done so that the addition of new features and extension of existing features is effortless
    - Good coding practices must be used with code structure, function naming, commenting etc.
- Results are presented in a way that a human and a machine can understand them
    - Clear plots, reasonably formatted results and simple logging
- Error handling should be implemented so that user-errors and actual implementation errors can be distinguished

- Performance of the tool should be at decent level
- Easily portable between operating systems

5.2 Programming language

For a data analysis tool that requires features such as signal processing and visual data plotting, there exists multiple options for the base programming language. General-purpose language Python has grown in popularity as the backbone for many data-analysis tools due its growing collection of scientific computing packages, being open-source, and portability between operating systems. Also, considering that many of the libraries used in automation are built with Python, choosing Python as the main programming language makes integration to existing libraries very effortless. Thus, Python is selected for implementing eCPRI PCAP Analyser.

The biggest drawback of Python compared to other low-level programming languages is its lack of memory-efficiency. In Python, variables do not need to be declared as a certain type, like an integer or a string: Python takes care of the conversion automatically. Thus, bad performance can become a problem in big Python projects. On the other hand, with good optimization, code structure and coding style Python can be used quite efficiently, while retaining the readability of the code.

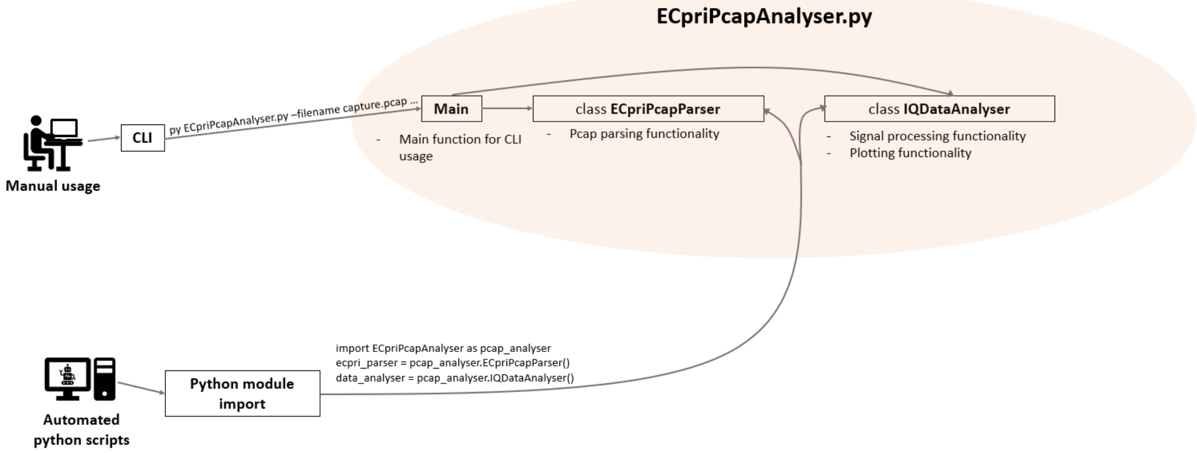
For a data analysis tool that is expected to parse big data captures, process the data with mathematical functions such as FFT, and visually present the results, Python offers very sophisticated and efficient third-party libraries. For this work, four modules were utilized as the main source for additional functionality:

- **Scapy**: Used for reading PCAP files into a byte string. Additionally, Scapy can parse the Ethernet header fields of the packet but has no support for parsing eCPRI/O-RAN headers. Relevant functions include:
  - scapy.utils.rdpcap
- **Numpy**: Used for memory-efficient array structures and advanced mathematical processing. This includes functions such as FFT, cross-correlation, exponential operations and complex-number calculations. Relevant functions include:
  - numpy.array
  - numpy.fft.fft
  - numpy.correlate
  - numpy.exp
- **Matplotlib**: Used for data visualization. As the name suggests, it can create MATLAB-like figures such as scatter plots. Relevant functions include:
  - mp.pyplot.scatter
  - mp.pyplot.plot
  - mp.pyplot.imshow
- **Argparse**: Used for defining and parsing advanced command line arguments. Enables users to specify additional parameters for the tool via the CLI. Relevant functions include:
  - argparse.ArgumentParser.add_argument
  - argparse.ArgumentParser.parse_args

5.3 Code structure

The features of eCPRI PCAP Analyser can be grouped into roughly two groups: PCAP capture parsing features and IQ data analysis features. It would be sensible to logically separate these functionalities in the code. This can be done using object-oriented programming, a programming paradigm based on the concept of objects. An object can be described as a collection of data and functionalities, which are defined and created with a blueprint. In programming, these blueprints are called classes. Classes are a handy way to separate different parts of the code into smaller logical pieces and using them makes the code easier to read and modify.

The simplified code structure and user interfaces of the eCPRI PCAP Analyser are shown in Figure 39. The logic is built within the main script, ECpriPcapAnalyser.py, which is divided into two main classes: ECpriPcapParser and IQDataAnalyser. These classes can be used either directly by importing them to automated python scripts, or via the main function which serves manual usage via the CLI.



```
py ECpriPcapAnalyser.py –filename capture.pcap ...
```

```
import ECpriPcapAnalyser as pcap_analyser
ecpri_parser = pcap_analyser.ECpriPcapParser()
data_analyser = pcap_analyser.IQDataAnalyser()
```

Figure 39. High-level code structure of eCPRI PCAP Analyser.

5.3.1 class ECpriPcapParser

The PCAP parsing functionality is implemented within the ECpriPcapParser class. The high-level block diagram of the class is presented in Figure 40. When an ECpriPcapParser object is created, the parsing process of the capture begins based on the inputted capture file name and mantissa size. First, the PCAP formatted capture is read into a list of Scapy packet objects. Each packet in the list is then iterated, and the ethernet header fields (Ch. 3.2.1.1) are retrieved. This does not require additional parsing of the data due Scapy being able to decode the header fields. An additional check is then done for the Ethernet type field to make sure that the packet is an eCPRI packet. If the type doesn't match the eCPRI EthType (0xAEFE), the packet is discarded from further parsing and the processing moves on to the next ethernet packet. If the packet is confirmed as an eCPRI packet, the processing continues to further parse the Ethernet payload. Since Scapy does not have support for eCPRI or O-RAN field parsing, the payload is returned as sequence of byte literals. In python, byte literals present data as ASCII characters, which are used to represent 8-bits of data (one octet) as a single character. To make the further parsing of data simpler, the ASCII characters are converted into the raw bit format. This way the data can be easily handled bit-by-bit and the parsing can be done directly using the header lengths in bits.

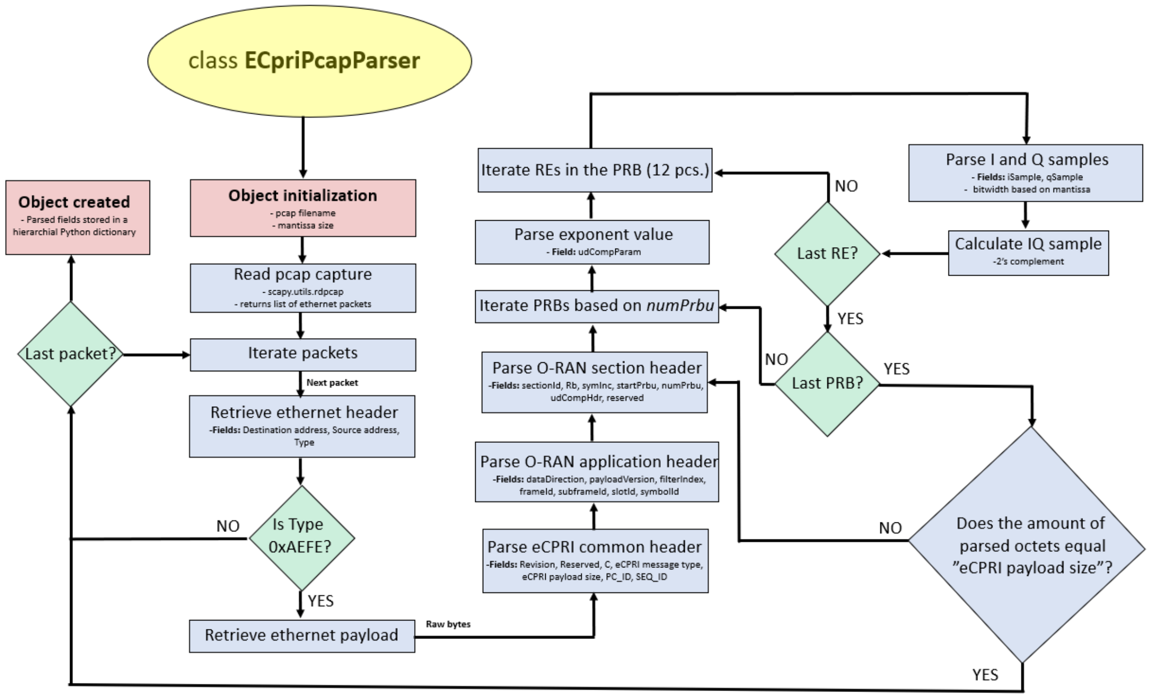
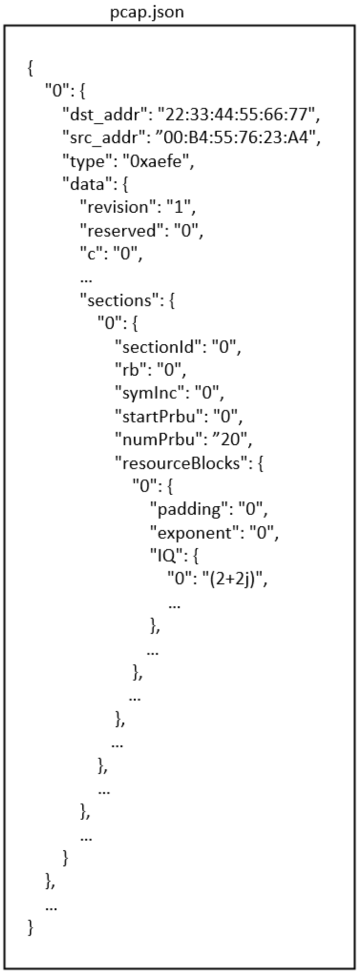

class ECpriPcapParser
Object created
- Parsed fields stored in a hierarchial Python dictionary
Object initialization
- pcap filename
- mantissa size
Read pcap capture
- scapy.utils.rdpcap
- returns list of ethernet packets
Last packet?
Iterate packets
Next packet
Retrieve ethernet header
-Fields: Destination address, Source address, Type
NO
Is Type 0xAEFE?
YES
Retrieve ethernet payload
Raw bytes
Iterate REs in the PRB (12 pcs.)
Parse exponent value
- Field: udCompParam
Iterate PRBs based on numPrbu
Parse O-RAN section header
-Fields: sectionId, Rb, symInc, startPrbu, numPrbu, udCompHdr, reserved
Parse O-RAN application header
-Fields: dataDirection, payloadVersion, filterIndex, frameId, subframeId, slotId, symbolId
Parse eCPRI common header
-Fields: Revision, Reserved, C, eCPRI message type, eCPRI payload size, PC_ID, SEQ_ID
NO
Last RE?
YES
NO
Last PRB?
YES
Parse I and Q samples
- Fields: iSample, qSample
- bitwidth based on mantissa
Calculate IQ sample
-2's complement
NO
Does the amount of parsed octets equal "eCPRI payload size"?
YES

Figure 40. Initialization of an ECpriPcapParser object.

Using the header field lengths presented in Chapters 3.2.1.2 and 3.2.1.3, the eCPRI transport header as well as the O-RAN application and section headers are parsed from the raw bit representation of the ethernet payload. From the section header, the class retrieves the *numPrbu* field, which tells how many PRBs are expected within this data section. This knownledge is then used to parse the IQ sample fields of the PRB (Ch. 3.2.1.3), starting with the common BFP exponent value that is retrieved from the *udCompParam* field, after which I and Q mantissa values are parsed one RE at a time. After parsing one IQ sample pair, the actual complex IQ sample is calculated using the 2's complement method described in Chapter 3.2.1.3. This processing continues until all 12 REs within a PRB have been parsed, and all the PRBs have been iterated within the section. During the parsing, the class keeps track of the number of octets that have been iterated and uses that knowledge to determine if there are any data sections left within the packet. This is done by comparing the count value to the eCPRI Payload Size header field of the eCPRI transport header: If they match, the packet has been fully parsed, and the processing can move on to the next packet.

Python dictionary is a convenient data structure based on key-value pairs. While the different fields of the packets are being parsed, the information retrieved from them are constantly being placed in a hierarchical dictionary format. In Figure 41, the hierarchical structure of the dictionary is demonstrated. After the whole capture has been parsed, the parsed dictionary is saved within the object as an attribute as well as saved into a separate JSON (JavaScript Object Notation) formatted file. JSON file format (.json) is convenient for storing Python dictionaries due also using key-value pair structure.



pcap.json

```
{
  "0": {
    "dst_addr": "22:33:44:55:66:77",
    "src_addr": ”00:B4:55:76:23:A4",
    "type": "0xaefe",
    "data": {
      "revision": "1",
      "reserved": "0",
      "c": "0",
      ...
      "sections": {
        "0": {
          "sectionId": "0",
          "rb": "0",
          "symInc": "0",
          "startPrbu": "0",
          "numPrbu": ”20",
          "resourceBlocks": {
            "0": {
              "padding": "0",
              "exponent": "0",
              "IQ": {
                "0": "(2+2j)",
                ...
              },
              ...
            },
            ...
          },
          ...
        },
        ...
      },
      ...
    }
  },
  ...
}
```

Figure 41. Parsed header fields and IQ data of a PCAP capture in JSON format

5.3.2 class IQDataAnalyser

The second class implemented in the tool is called IQDataAnalyser. The IQDataAnalyser is designed to handle all signal processing and analysing functions of the eCPRI PCAP Analyser. When an object of this class is created, the relevant IQ data is retrieved from the dictionary parsed with the EcpriPcapParser and organized frame-by-frame into NR resource grids. Basically, this means that the IQ samples are mapped into a two-dimensional numpy array that represents the time-frequency grid. The frames from different data flows are sorted based on the PC_ID field of the eCPRI transport header (Ch. 3.2.1.1).

The initialization flow of an IQDataAnalyser object is illustrated in Figure 42. The object creation requires five parameters, first one being the EcpriPcapParser dictionary that carries all the header fields and IQ data of the parsed capture. The other mandatory parameters are SCS (5G NR numerology), number of active subcarriers, resource block offset and the FFT length, all which are used to create the full NR resource grid. The object initialization starts by gathering all the different PC_IDs in the parsed capture. In radio unit integration level testing, the PC_ID field is usually configured to represent a specific spatial stream. The PC_IDs are then iterated so that the radio frames can be recreated for each separate data flow. First, the class finds all the frames for the specific PC_ID by gathering the unique frameID fields. It then proceeds to iterate each of the frameIDs and continues to find the unique subframeIDs for that frameID. Going further, all the slotIDs are retrieved for each of the subframeIDs, and all symbolIDs are retrieved for all the slotIDs. Lastly, the IQ samples for each symbolID are retrieved and mapped on the resource grid.

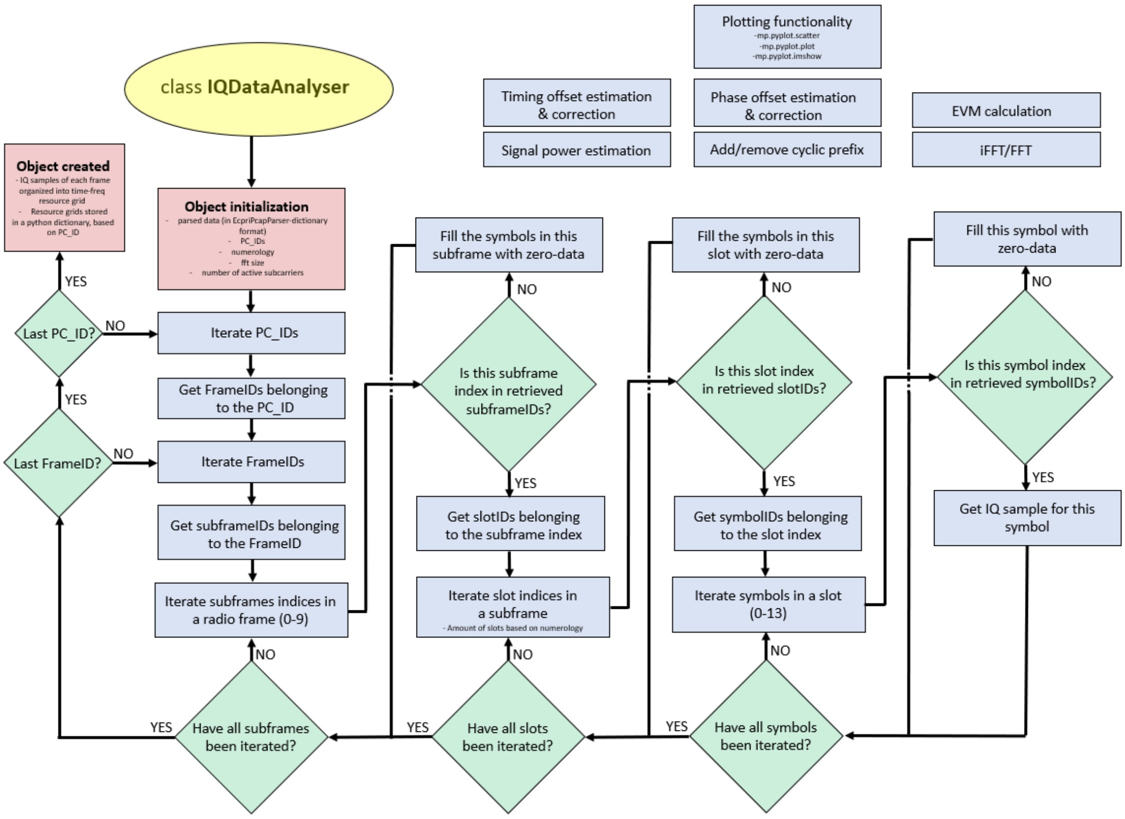

class IQDataAnalyser
Object initialization
- parsed data (in EcpriPcapParser-dictionary format)
- PC_IDs
- numerology
- fft size
- number of active subcarriers
Object created
- IQ samples of each frame organized into time-freq resource grid
- Resource grids stored in a python dictionary, based on PC_ID
Last PC_ID?
YES
NO
Iterate PC_IDs
Get FrameIDs belonging to the PC_ID
Last FrameID?
YES
NO
Iterate FrameIDs
Get subframeIDs belonging to the FrameID
Iterate subframes indices in a radio frame (0-9)
Have all subframes been iterated?
YES
NO
Is this subframe index in retrieved subframeIDs?
NO
Fill the symbols in this subframe with zero-data
YES
Get slotIDs belonging to the subframe index
Iterate slot indices in a subframe
- Amount of slots based on numerology
Have all slots been iterated?
YES
NO
Is this slot index in retrieved slotIDs?
NO
Fill the symbols in this slot with zero-data
YES
Get symbolIDs belonging to the slot index
Iterate symbols in a slot (0-13)
Have all symbols been iterated?
YES
NO
Is this symbol index in retrieved symbolIDs?
NO
Fill this symbol with zero-data
YES
Get IQ sample for this symbol
Plotting functionality
-mp.pyplot.scatter
-mp.pyplot.plot
-mp.pyplot.imshow
Timing offset estimation & correction
Signal power estimation
Phase offset estimation & correction
Add/remove cyclic prefix
EVM calculation
iFFT/FFT

Figure 42. Initialization of an IQDataAnalyser object.

As discussed in Chapter 2.3.3, a radio frame in 5G NR consists of 10 subframes. Moreover, the number of slots within a subframe depends on numerology, and a slot consists of 14 OFDM symbols. In frequency domain, the available bandwidth is divided into subcarriers and the number of subcarriers is determined by the FFT size. Each unique resource element (symbol-subcarrier pair) is represented with an IQ sample. However, in TDD mode, not all symbols are assigned for UL transmission, resulting in missing symbols/slots/subframes in the capture. Additionally, the UL test signals, which are defined by 3GPP, usually do not utilize all the available subcarriers for data transmission, meaning that some of the subcarriers are left empty. Therefore, it is important to know how many active subcarriers the signal is using, as well as their position in frequency domain (determined with the resource block offset). The missing IQ samples in the resource grid are filled with zero-data by the tool.

The resource grids of each stream are ordered into a hierarchical python dictionary, based on which PC_ID stream they belong to. After the object is created, this dictionary can be retrieved as an attribute of the object for further analysis with the other methods of the class, including:

- Data plotting using matplotlib
    - IQ constellation, resource grid and cross correlation plots
- IQ data processing
    - iFFT/FFT
    - Remove/Add Cyclic Prefixes
- Symbol timing offset estimation and correction
    - Equations of Ch. 3.3.2.2.
- Carrier phase offset estimation and correction
    - Equations of Ch. 3.3.2.3.
- EVM calculation
    - Equations of Ch. 3.3.2.4.
- Signal power calculation
    - Equations of Ch. 3.3.2.1.

5.3.3 Main function

The main function offers an extensive sequence of functionalities that utilize the EcpriPcapParser and IQDataAnalyser classes. Its primary purpose is to act as a middleman between the command line interface (CLI) and the two main classes of the script. When a user launches the tool via the CLI, the Main function executes and goes through the parsing and analysis sequence as shown in Figure 43. The results of the analysis are presented visually in log prints and plots, as well as saved in separate PNG (Portable Network Graphics) and JSON files for examination.

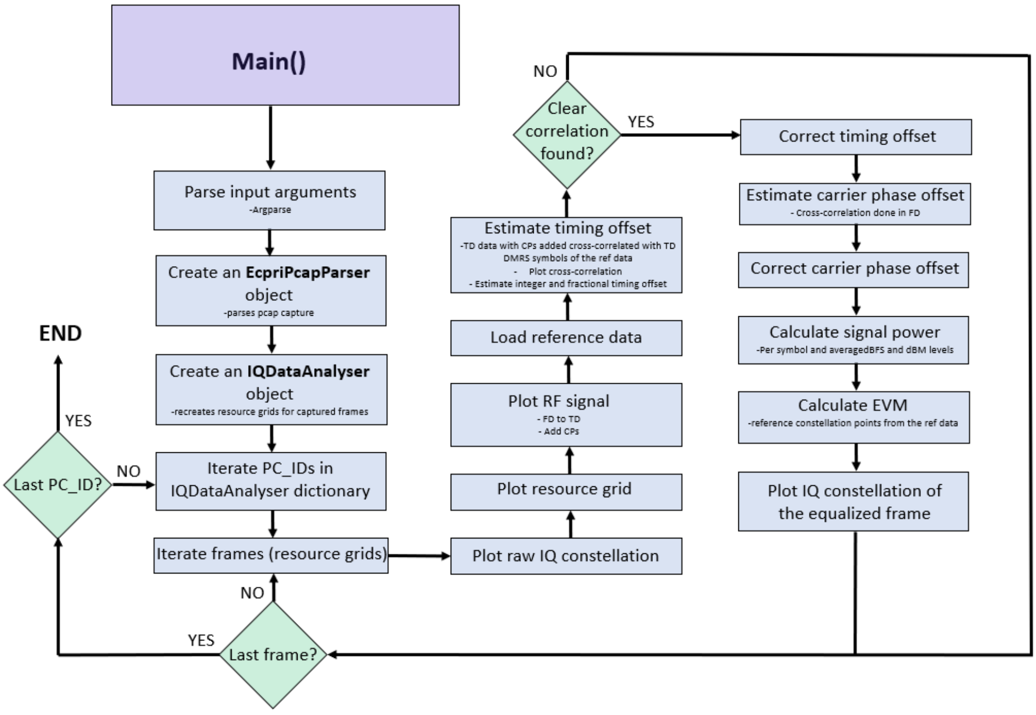
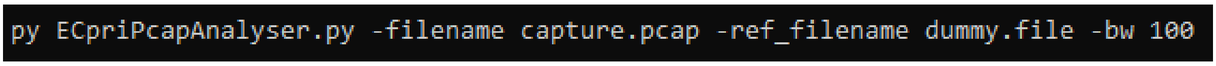

Main()
Parse input arguments
-Argparse
Create an **EcpriPcapParser** object
-parses pcap capture
Create an **IQDataAnalyser** object
-recreates resource grids for captured frames
Iterate PC_IDs in IQDataAnalyser dictionary
Iterate frames (resource grids)
Plot raw IQ constellation
Plot resource grid
Plot RF signal
- FD to TD
- Add CPs
Load reference data
Estimate timing offset
-TD data with CPs added cross-correlated with TD DMRS symbols of the ref data
- Plot cross-correlation
- Estimate integer and fractional timing offset
Clear correlation found?
NO
YES
Correct timing offset
Estimate carrier phase offset
- Cross-correlation done in FD
Correct carrier phase offset
Calculate signal power
-Per symbol and averagedBFS and dBM levels
Calculate EVM
-reference constellation points from the ref data
Plot IQ constellation of the equalized frame
Last frame?
NO
YES
Last PC_ID?
NO
YES
END

Figure 43. Structure of the main function.

When the main function is called, the command line arguments provided by the users are first parsed using the argparse module. Based on the inputted arguments and default arguments set by the tool, the function then proceeds to parse the PCAP capture by creating an EcpriPcapParser object, after which it maps the IQ data of the capture to resource grids with IQDataAnalyser. All PC_IDs within the data are then iterated, and all the found radio frames are taken into analysis, one by one. The first figure the tool plots is the raw IQ constellation plot ("doughnut" plot) and the resource grid of the frame. The frame is then "reversed-engineered" into the digital RF signal by transforming the data from frequency domain to time domain using iFFT and by adding the CPs to the time domain data. After that, the reference data provided by the user is retrieved and the function moves on to timing offset estimation, which is based on Chapter 3.3.2.2. The estimation is done in time domain by cross correlating the RF frame (TD frame with CPs added) with the DMRS symbols of the reference data. This way the timing offset (integer and fractional) can be estimated from the whole length of the frame. The cross-correlation result is then saved into a separate plot. However, if no clear constellation peak is found between the captured data and DMRS symbols, no reliable timing offset estimation can be done, and the tool moves on to the next frame. If the cross-correlation result produces a noticeable peak, the timing offset is then corrected. After the timing is correction, the carrier phase offset is determined and corrected utilizing equations from Chapter 3.3.2.3. Signal power is then calculated from the IQ samples utilizing the equations from O-RAN specs (Chapter 3.3.2.1). Finally, the EVM of the frame can be calculated as described in Chapter 3.3.2.4. The reference constellation points for the calculation are taken from the reference data. Finally, the constellation plot of the timing and phase corrected frame is plotted.

5.4 User interfaces

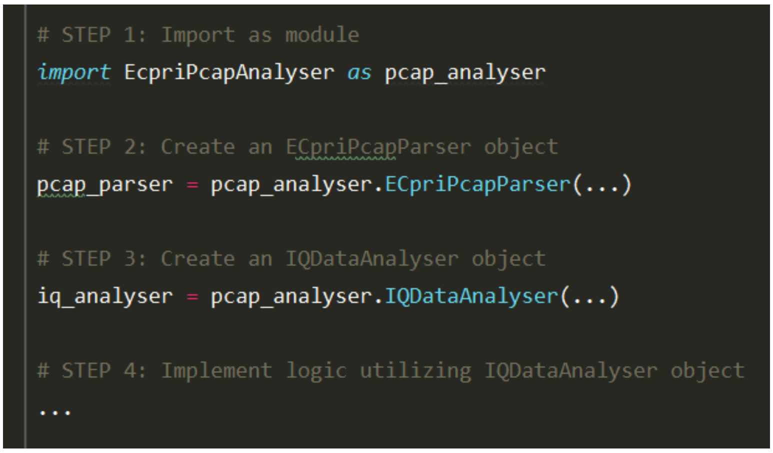
Two user interfaces are implemented in the eCPRI PCAP Analyser, one for manual usage and one for external script integration. To use the tool manually, the user calls the main script EcpriPcapAnalyser.py directly from the CLI and provides as many or as little input arguments as they like, taking the default parameters set by the tool into account. In many use-cases, the user is only required to specify the name of the PCAP capture and the reference file, as well as carrier specific parameters such as bandwidth. An example showing the conventional CLI call for running eCPRI PCAP Analyser is shown in Figure 44.



```
py ECpriPcapAnalyser.py -filename capture.pcap -ref_filename dummy.file -bw 100
```

Figure 44. Example CLI call for running the eCPRI PCAP Analyser.

The second way to use the tool is to integrate it to other python scripts as an imported module. This way the functions and classes defined in EcpriPcapAnalyser.py can be used from other python script, enabling usage in automated environments. Compared to the CLI usage, the module importing offers way more flexibility for using the tool since the features are not limited by the Main function. Instead, the logic for using the tool can be implemented as is seen necessary for the specific use-case. For example, if a test-case is only interested in calculating the captured signal power, it can skip all the unnecessary equalization and plotting functionalities by constructing a function that does only the needed steps to calculate signal power. In Figure 45, the process of importing the eCPRI PCAP Analyser as a python module is shown, as well as the steps required for starting IQ data analysis.



```
# STEP 1: Import as module
import EcpriPcapAnalyser as pcap_analyser

# STEP 2: Create an ECpriPcapParser object
pcap_parser = pcap_analyser.ECpriPcapParser(...)

# STEP 3: Create an IQDataAnalyser object
iq_analyser = pcap_analyser.IQDataAnalyser(...)

# STEP 4: Implement logic utilizing IQDataAnalyser object
...
```

Figure 45. Importing eCPRI PCAP Analyser to external python scripts.

5.5 Outputs

Similarly to the user interfaces, the results produced by the tool must be usable in both manual and automated testing environments. Basically, this means that the results generated by the tool should be in a format that is easily understandable by humans, while also producing outputs that can be used by computers. For example, a human can give an estimate on the data quality just by looking at the IQ constellation plot, but similar visual analysis is quite more difficult for a computer. Therefore, the eCPRI PCAP Analyser is designed and implemented to present results in both visual and text format.

The tool is capable to produce four different kinds of plots: IQ constellation diagrams, RF signal time domain plots, cross-correlation plots and resource grid plots. Example plots produced by the main function are shown in Figure 46. First plot is the raw IQ constellation diagram, in which the IQ samples from the frame are mapped into the IQ data plane. Since the timing and phase offsets are still visible in this figure, the constellation is rotated. Next, the resource grid of the frame is plotted, giving a visual indication where the data is mapped in the time-frequency grid. RF signal plot is used to visualize the captured FD frame as an RF signal and can be used to verify the TDD pattern. When the main function proceeds to estimate timing offset, it plots the cross-correlation result. If the cross-correlation is bad (no clear peak visible in the plot), it likely means that the captured data is either corrupted or noise, or the reference signal is not matching the captured data. Finally, after all the processing and calculations have been done to the captured frame, the equalized IQ constellation is plotted with the main calculation results displayed on top of the figure, including EVM, timing offset and power level. This figure can be considered as the most important output for manual users since it combines all the relevant result into a single plot.

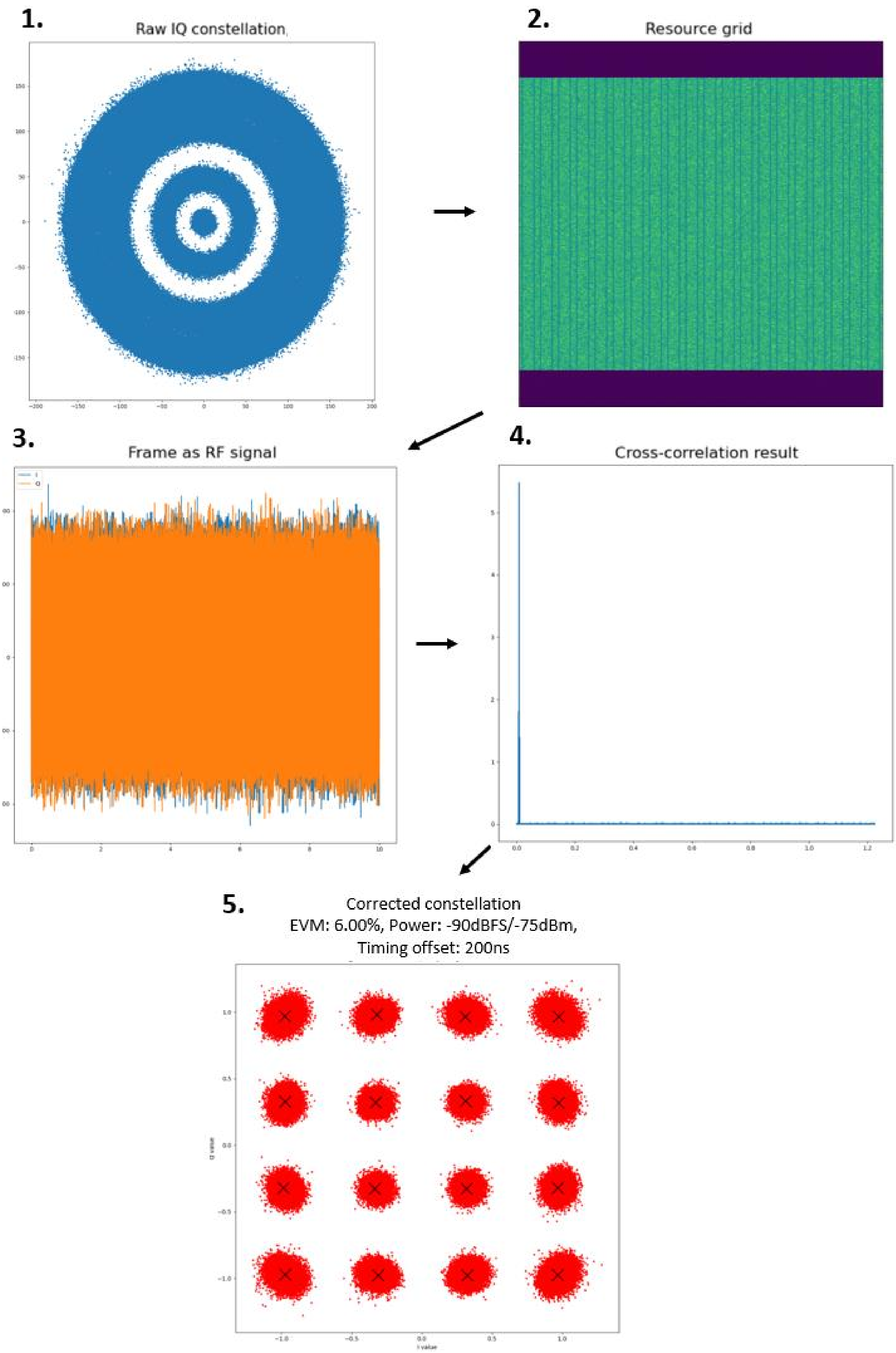
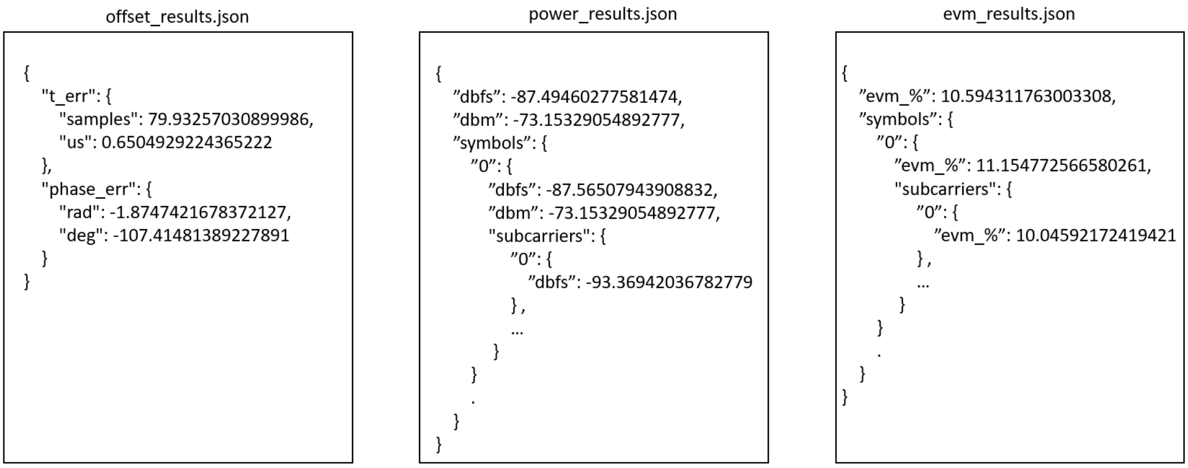


Figure 46. Result figures of eCPRI PCAP Analyser.

Having a clear text-based result format is important for computer-based analysis. Since most of the calculation results done within the tool are organized into python dictionaries, the logical format for saving them is the JSON format since it goes hand-in-hand with python dictionaries. Python also offers convenient built-in functions to read data from a JSON file to a dictionary and vice versa.

Within the main function, all the relevant results are saved into three separate JSON files, one for offset estimates, one for signal power levels, and one for EVM measurements. The structure of these JSON files is presented in Figure 47. The offset_results.json file presents the timing offset in samples and microseconds, as well as the phase offset in both radians and degrees. The calculated power and EVM levels are stored into power_results.json and evm_result.json respectively. Both files possess identical basic structure, presenting the mean value for the whole frame, as well as symbol- and subcarrier-specific values. Power levels are presented in both dBFS and dBm, exception being the subcarrier-specific power levels, which are only expressed in dBFS. All presented EVM levels are expressed in percentages.



offset_results.json

```
{
   "t_err": {
      "samples": 79.93257030899986,
      "us": 0.6504929224365222
   },
   "phase_err": {
      "rad": -1.8747421678372127,
      "deg": -107.41481389227891
   }
}
```

power_results.json

```
{
   ”dbfs”: -87.49460277581474,
   ”dbm”: -73.15329054892777,
   ”symbols”: {
      ”0”: {
         ”dbfs”: -87.56507943908832,
         ”dbm”: -73.15329054892777,
         "subcarriers": {
            ”0”: {
               ”dbfs”: -93.36942036782779
            },
            …
         }
      }
      .
   }
}
```

evm_results.json

```
{
   ”evm_%”: 10.594311763003308,
   ”symbols”: {
      ”0”: {
         ”evm_%”: 11.154772566580261,
         "subcarriers": {
            ”0”: {
               ”evm_%”: 10.04592172419421
            },
            …
         }
      }
      .
   }
}
```

Figure 47. Result JSON files of eCPRI PCAP Analyser.

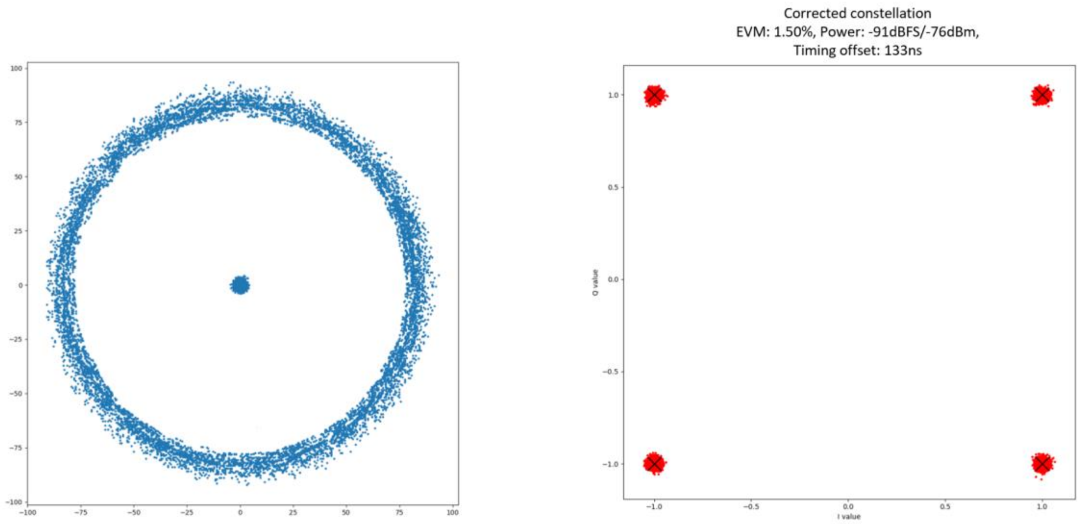
5.6 Evaluation

The first version of eCPRI PCAP Analyser was released in May 2020 to a limited userbase within Oulu’s radio unit integration team. Although the tool was not being used very much at this point, the feedback acquired was crucial for the later stages of development. Many of the bigger bugs, performance problems and issues with user experience were discovered during the first release. When the second release was done more publicly in June 2020, the tool was already quite mature and ready for a bigger userbase.

5.6.1 Usage

Currently, the tool is used in multiple Nokia sites for UL data validation of various O-RAN eCPRI radio units and is gaining increasing amount of interest due to its analysis features, simple user interface and ease of installation. Analysis that was previously done visually by looking at the constellation figure of the raw IQ samples can now be done with concrete metrics such as EVM and power level. This level of analysis at unit integration was not possible with the tools used before the introduction of the eCPRI PCAP Analyser, and it can be estimated that the tool has saved days, even weeks of development time by detecting bugs in the radio's data processing that would have otherwise been ran into in later stages of integration.

The heaviest usage for the tool has been in radio unit regression testing, where datapath is validated to both DL and UL directions. With the previous tools, the UL data quality was reported as the "visual quality of the constellation doughnut". Relying on this sort of analysis sometimes resulted in subpar results being submitted as acceptable. With the new tool, UL regression testing could finally be done using real KPIs. Figure 48 illustrates the difference between the old analysis method (visual analysis) and the new analysis method provided by the eCPRI PCAP Analyser (KPI analysis).



Constellation circle is clear, data *looks* good → UL OK"

(a) Visual analysis (old way)

"EVM, timing offset and power level at *acceptable level* → UL OK"

(b) Analysis using KPIs (new way)

Figure 48. Old analysis method vs. new analysis method.

The tool has also been implemented to an automated testline that runs regression testing for UL datapath validation for an O-RAN 7-2x functional split based eCPRI radio unit. Without automation, validating data quality for all antenna paths of the radio would take a significant amount of dedicated worktime from a tester due the manual steps required for signal generator routing, data capturing and running the analysis. Since this testing is now automated, the hours that were previously spent in manual testing can now be used on other tasks. Additionally, the automated testline can be run 24/7, enabling testing to be done outside regular working hours.

5.6.2 Performance

Even though top-tier performance was not listed as an initial requirement for the eCPRI PCAP Analyser, it will become a more important aspect of the tool when the number of active users increases. Thus, it is good to create efficient baseline benchmarks that can be used in the future to evaluate how much the performance has improved. The performance measurements were run on a standard test PC with the following high-level specifications:

- Windows 10 operating system
- Core i7 @ 3.20 GHz
- 16 GB of RAM

The first benchmark selected for the performance analysis was the eCPRI packet parsing time, as shown in Figure 49. The tool achieved an average parsing speed of 47.6 packets per second. However, since the tool also saves the captured data in a JSON file and loads it on further analysis runs, the parsing time is cut down to a couple of second after the initial run.

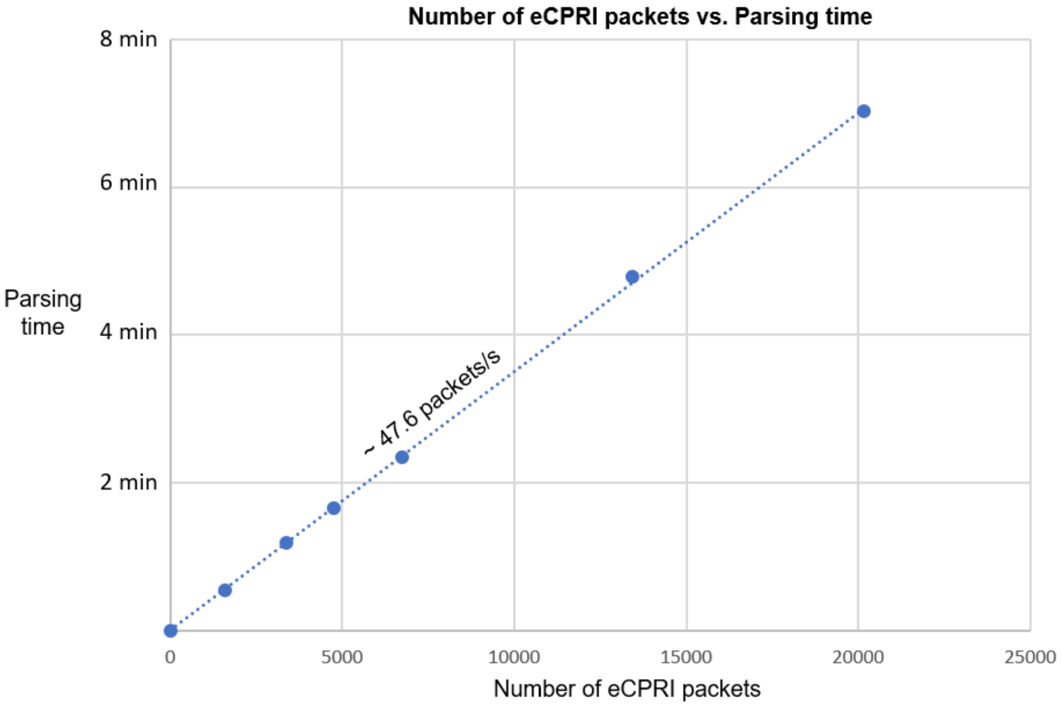


Figure 49. Number of eCPRI packets vs. Parsing time.

IQ data analysis time for different carrier configurations with 16QAM modulation was selected as the second performance benchmark. More specifically, analysis time for carriers with 122.88, 61.44 and 30.72 MHz sampling rates were compared. The results on Figure 50 show that the carrier configuration with the lowers sampling rate, 30.72 MHz, achieved the fastest analysis time of 7.5 seconds. Captures with 61.44 MHz and 122.88 MHz sampling rate were analysed in about 16.5 and 30 seconds, respectively. It can be roughly estimated that doubling the sampling rate doubles the analysis time of the tool.

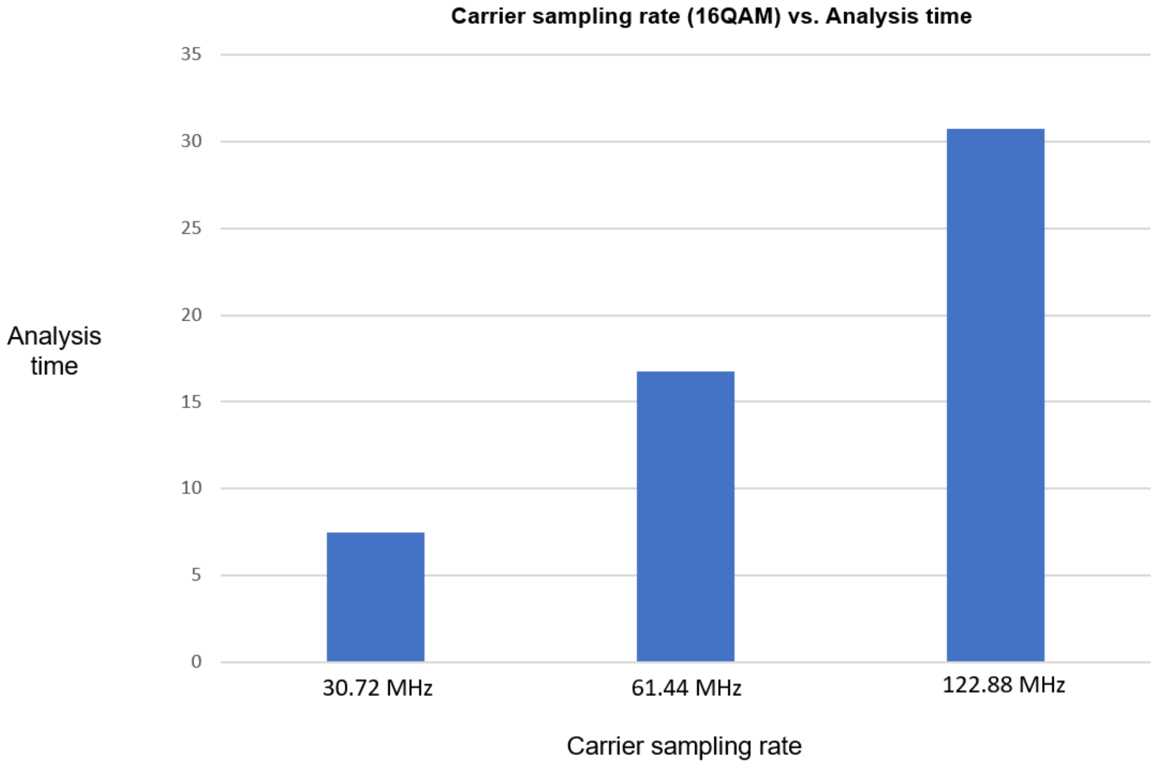


Figure 50. Carrier sampling rate vs. Analysis time

6 DISCUSSION

The demand for the outcome of this thesis was quite high from the get-go, since the eCPRI interface did not have sufficient tools for UL data analysis in radio unit integration teams. While this put a moderate level of pressure for getting the tool released, it also acted as the main motivator to really dig deep into the background of eCPRI and 5G in general.

The most challenging part of this thesis ended up being the overall learning curve related to 5G and digital signal processing. When starting this thesis, the author possessed some high-level theoretical knowledge of what radio access networks are and how they operate, and it was satisfying to see some of this knowledge link up with the practical work of radio unit testing. However, most of the fronthaul related concepts as well as signal processing had to be learned almost from ground up. This made the actual development of the tool a relatively minor part of this work, especially since the author was already quite familiar with Python programming.

The biggest point of improvement for the tool's future development is in its performance. The way that the tool is coded is not particularly efficient, which leads to slower parsing and analysis times compared to the other available eCPRI analysis tools. While performance can become a limitation in Python-based programs, removing inefficient looping and data structures can improve the performance noticeably. A total rework of the code structure is not an excluded option either if the performance becomes a huge hindrance for using the tool.

Another future point of study is how accurately the methods and algorithms of this tool perform. Currently, the tool offers accuracy levels that are suitable for radio unit integration, but there is still a lot of potential for improvement in this area. Concepts like oversampling and machine learning could be used to refine the accuracy and reliability of the used algorithms.

Although the tool has already gained a decent userbase within integration teams, the goal is to attract even more users by introducing new features and improving the user experience. Some of the planned features and improvements for the tool include:

- Support for LTE PUSCH data analysis
- Support for PRACH data analysis
- Channel estimation and equalization
- Frequency offset estimation
- Phase compensation reversal
- Support for UDP/IP and VLAN based eCPRI transport networks
- Graphical user interface

As the popularity of O-RAN grows, so does the importance of eCPRI interface testing since the open interface needs to fulfil strict requirements to ensure vendor interoperability. It is evident that more tools for eCPRI interface testing will come up, evidently leading to high-standard commercial solutions. Therefore, the future of eCPRI PCAP Analyser is challenging but bright: by the addition of new and desired features as well as maintaining support for both manual and automated usage, the tool will remain relevant within Nokia for a long time.

7 SUMMARY

The new 5G mobile network generation aims to enhance the performance of the cellular network in almost every possible aspect, offering higher data rates, lower latencies, and massive number of network connections. To achieve these improvements, 5G introduces multiple new and innovative technological changes to the RAN, including new frequency ranges, cloud RAN architecture, and beamforming. Arguably the most important change from LTE are the new RU-BBU split options for 5G promoted by 3GPP. From the initial eight split option proposals, 3GPP has moved on to further standardize one option as the high layer split by introducing the F1 fronthaul interface, while other organizations have taken a bigger role with the low layer split standardization.

Particularly big conceptual shift introduced with 5G is the open RAN concept, pushed forward by organizations such as the O-RAN alliance. O-RAN aims to standardize the interfaces between different RAN elements, such as the radio unit and baseband unit, in a way that promotes vendor interoperability and lowers the entry barrier for new equipment suppliers. Moreover, the 7-2x split option standardized by O-RAN has risen as the most important option within the different low layer split options. For the 7-2x split, O-RAN has selected the packet-based eCPRI protocol as the fronthaul interface. Compared to its predecessor CPRI, eCPRI has been designed to be more flexible and dynamic in terms of transport network and data-rates.

However, due being a new standard for the fronthaul, proper analysis tools for testing eCPRI formatted data were still lacking. The objective of this thesis was to create a new data analysis tool for UL eCPRI data so that UL data quality could be validated for any O-RAN 7-2x functional split based radio unit. The goal for the tool was to provide concrete data quality metrics for testers, including features such as timing offset estimation based on DMRS symbols, signal power level estimation, and error vector magnitude calculation. The tool produces visual and text-based outputs that can be used in both manual and automated testing.

The tool was implemented with python due to its easy portability between operating systems, vast library support, and object-oriented programming. A simple command line interface was designed for manual usage, while integration to other python scripts was made possible with clear-cut classes for object creation. The code implementation was done so that the addition of new features would be simple and effortless.

The new tool has been taken into use in multiple radio unit integration teams within Nokia and is growing in popularity as a data analysis tool for UL testing. As the O-RAN alliance gains popularity, the importance for eCPRI interface testing also increases, making the new tool more and more significant for Nokia. Although commercial solutions for eCPRI testing will eventually pop-up, the tool will remain relevant with active development of new features.

8 REFERENCES

[1] ITU-R (2015) IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond.

[2] Sami Tabbane (2019) 4G to 5G networks and standard releases: ITU PITA Workshop on Mobile network planning and security, p. 11.

[3] A. Gupta and R. K. Jha (2015). A Survey of 5G Network: Architecture and Emerging Technologies, IEEE Access, vol. 3, p. 1206-1232.

[4] Ericsson (2020) Ericsson Mobility Report, June 2020, p. 20.

[5] ITU-R (2015) IMT traffic estimates for the years 2020 to 2030.

[6] 3rd Generation Partnership Project (2019) 3GPP TR 21.915 V15.0.0.

[7] M. G. Kibria, K. Nguyen, G. P. Villardi, K. Ishizu and F. Kojima (2018). Next Generation New Radio Small Cell Enhancement: Architectural Options, Functionality and Performance Aspects, IEEE Wireless Communications, vol. 25, no. 4, pp. 120-128.

[8] T. V. Kiran Buyakar, H. Agarwal, B. R. Tamma and A. A. Franklin (2019) Prototyping and Load Balancing the Service Based Architecture of 5G Core Using NFV, 2019 IEEE Conference on Network Softwarization (NetSoft).

[9] V. Jungnickel et al. (2014) Software-defined open architecture for front- and backhaul in 5G mobile networks, 2014 16th International Conference on Transparent Optical Networks (ICTON).

[10] C-RAN (2011) The Road Towards Green RAN, Version 2.5.

[11] Bin Guo, Wei Cao, An Tao and D. Samardzija (2012) CPRI compression transport for LTE and LTE-A signal in C-RAN, 7th International Conference on Communications and Networking in China, 8-10 Aug. 2012.

[12] Hector Menendez (Accessed 15.02.2020) Evolution towards Cloud RAN and eCPRI. URL: https://www.nokia.com/blog/evolution-towards-cloud-ran-and-ecpri/

[13] Signalboosters.com (Accessed 15.02.2020). C-RAN - Centralized Vs. Cloud-Based Radio Access Network URL: https://www.signalbooster.com/blogs/news/c-ran-centralized-vs-cloud-based-radio-access-network

[14] 3rd Generation Partnership Project (2020) 3GPP TS 38.300 V15.10.0

[15] L. M. P. Larsen, A. Checko and H. L. Christiansen (2019) A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks, IEEE Communications Surveys & Tutorials (Volume: 21, Issue: 1, Firstquarter 2019), p. 146 – 172.

[16] techplayon.com (Accessed 10.4.2020) 5G NR User Plane Protocol, What's new Over LTE in 5G NR. URL: http://www.techplayon.com/5g-nr-radio-protocol-user-plane-whats-new-lte-5g-nr/

[17] P. Govindarajan (Accessed 10.4.2020) Quality of Service Model in 5G. URL: https://www.awardsolutions.com/portal/shareables/what-is-5G/5G-Training-Online/quality-service-model-5g-prasad-govindarajan

[18] B. Bertenyi, R. Burbidge, G. Masini, S. Sirotkin and Y. Gao (2018) NG Radio Access Network (NG-RAN), River Publishers.

[19] Kevin Murphy (2015) Centralized RAN and Fronthaul, ISE magazine, May 2015.

[20] Ericsson AB, Huawei Technologies Co. Ltd, NEC Corporation, Alcatel Lucent, and Nokia Networks (2015) CPRI Specification V7.0.

[21] ITU-T (2018) 5G wireless fronthaul requirements in a passive optical network context, Series G, supplement 66.

[22] Ericsson AB, Huawei Technologies Co. Ltd, NEC Corporation and Nokia (2019) eCPRI Specification V2.0.

[23] J. G. Andrews et al. (2014) What Will 5G Be?, IEEE Journal on Selected Areas in Communications (Volume: 32, Issue: 6, June 2014), p. 1065 – 1082.

[24] NTT DOCOMO, INC (2016) CU-DU split: Refinement for Annex A (Transport network and RAN internal functional split).

[25] 3rd Generation Partnership Project (2020) 3GPP TS 38.470 V16.1.0.

[26] 3rd Generation Partnership Project (2016) Transport requirement for CU&DU functional splits options, 3GPP TSG RAN WG3 Meeting #93.

[27] A. Umesh, Y. Yajima, T. Uchino and S. Okuyama (2019) Overview of O-RAN Fronthaul Specifications.

[28] 3rd Generation Partnership Project (2019) 3GPP TS 38.101-1 V16.2.0.

[29] electronics-notes.com (Accessed 16.4.2020) 5G Waveforms & Modulation: CP-OFDM & DFT-s-OFDM. URL: https://www.electronics-notes.com/articles/connectivity/5g-mobile-wireless-cellular/waveforms-ofdm-modulation.php

[30] B. Witte (Accessed: 16.4.2020) The basics of 5G's modulation, OFDM. URL: https://www.5gtechnologyworld.com/the-basics-of-5gs-modulation-ofdm/

[31] gta.ufrj.br (Accessed 16.4.2020) OFDMA. URL: https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2014_2/rafaelreis/ofdma_scfdma.html

[32] electronics-notes.com (Accessed 16.4.2020) 5G Waveforms & Modulation: CP-OFDM & DFT-s-OFDM. URL: https://www.electronics-notes.com/articles/connectivity/5g-mobile-wireless-cellular/waveforms-ofdm-modulation.php

[33] allaboutcircuits.com (Accessed 17.4.2020) Understanding I/Q Signals and Quadrature Modulation. URL: https://www.allaboutcircuits.com/textbook/radio-frequency-analysis-design/radio-frequency-demodulation/understanding-i-q-signals-and-quadrature-modulation/

[34] A. Wolke (Accessed 17.4.2020) What's Your IQ – About Quadrature Signals. URL: https://www.tek.com/blog/quadrature-iq-signals-explained

[35] electronics-notes.com (Accessed 17.4.2020) QAM Formats: 8-QAM, 16-QAM, 32-QAM, 64-QAM, 128-QAM, 256-QAM. URL: https://www.electronics-notes.com/articles/radio/modulation/quadrature-amplitude-modulation-types-8qam-16qam-32qam-64qam-128qam-256qam.php

[36] rfwireless-world.com (Accessed 17.4.2020) Advantages of QAM | Disadvantages of QAM | 16QAM, 64QAM, 256QAM. URL: https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-QAM-types.html

[37] dsprelated.com (Accessed 20.4.2020) Mathematics of the DFT. URL: https://www.dsprelated.com/freebooks/mdft/Mathematics_DFT.html

[38] rfmw.em.keysight.com (Accessed 20.4.2020) Concepts of Orthogonal Frequency Division Multiplexing (OFDM) and 802.11 WLAN. URL: http://rfmw.em.keysight.com/wireless/helpfiles/89600B/WebHelp/Subsystems/wlan-ofdm/content/ofdm_basicprinciplesoverview.htm

[39] dsprelated.com (Accessed 20.4.2020) Convolution Theorem. URL: https://www.dsprelated.com/freebooks/mdft/Convolution_Theorem.html

[40] gaussianwaves.com (21.4.2020) Cyclic Prefix in OFDM: hands-on demo in Matlab. URL: https://www.gaussianwaves.com/2016/05/ofdm_and_cyclic_prefix_a_handson_demonstration/

[41] A. Yazar, B. Peköz and H. Arslan (2019) Fundamentals of Multi-Numerology 5G New Radio. Department of Electrical Engineering, University of South Florida.

[42] techplayon.com (Accessed 25.4.2020) Motivation Behind Having Multiple Numerology in 5G NR. URL: http://www.techplayon.com/motivation-behind-having-multiple-numerology-in-5g-nr/

[43] S. Parkvall, E. Dahlman, A. Furuskar and M. Frenne (2017) NR: The New 5G Radio Access Technology. IEEE Communications Standards Magazine (Volume: 1, Issue: 4, Dec. 2017).

[44] techplayon.com (Accessed 27.4.2020) 5G NR Cyclic Prefix (CP) Design. URL: http://www.techplayon.com/5g-nr-cyclic-prefix-cp-design/

[45] H. Holma, A. Toskala and T. Nakamura (2019) 5G Technology: 3GPP New Radio. John Wiley & Sons, p. 34-35.

[46] Rohde&Schwarz (2007) From SISO to MIMO – taking advantage of everything the air interface offers (2). News from Rohde&Schwarz Number 194.

[47] M. Khan, S. Bashir & A. Habib (2014) Semi Round Robin Pairing and Scheduling in Uplink Virtual Multiple Input Multiple Output (VMIMO) communications. Journal of Space Technology. 4. p. 61-66.

[48] M. Passoja (Accessed 29.4.2020) 5G NR: Massive MIMO and Beamforming – What does it mean and how can I measure it in the field?. URL: https://www.rcrwireless.com/20180912/5g/5g-nr-massive-mimo-and-beamforming-what-does-it-mean-and-how-can-i-measure-it-in-the-field

[49] M. Agiwal, A. Roy and N. Saxena (2016) Next Generation 5G Wireless Networks: A Comprehensive Survey. IEEE Communications Surveys & Tutorials (Volume: 18, Issue: 3, thirdquarter 2016), p. 1617 – 1655.

[50] O-RAN Alliance (2019) ORAN-WG4.CUS.0-v01.00.

[51] IEEE (2016) 802.3-2015 - IEEE Standard for Ethernet. p. 108-113.

[52] firewall.cx (Accessed 13.4.2020) VLAN Tagging – Understanding VLAN Ethernet Frames. URL: http://www.firewall.cx/networking-topics/vlan-networks/219-vlan-tagging.html

[53] D. Chang (2008) Effect and Compensation of Symbol Timing Offset in OFDM Systems With Channel Interpolation. IEEE Transactions on Broadcasting (Volume: 54, Issue: 4, Dec. 2008), p. 761 – 770.

[54] techplayon.com (Accessed 20.6.2020) 5G NR Reference Signals (DMRS, PTRS,SRS and CSI-RS). URL: http://www.techplayon.com/5g-nr-reference-signals-dmrs-ptrssrs-and-csi-rs/

[55] 3rd Generation Partnership Project (2020) 3GPP TS 38.211 V16.0.0.

[56] sharetechnote.com (Accessed 20.6.2020) Zadoff - Chu Sequence. URL: https://www.sharetechnote.com/html/Handbook_LTE_Zadoff_Chu_Sequence.html

[57] physicspages.com (Accessed 21.6.2020) Time Reversal, Antiunitary Operators and Wigner’s Theorem. URL: https://www.physicspages.com/pdf/Shankar/Shankar%20Exercises%2011.05%20Time%20reversal.pdf

[58] A. B. Awoseyila, C. Kasparis and B. G. Evans (2009) Robust time-domain timing and frequency synchronization for OFDM systems. IEEE Transactions on Consumer Electronics (Volume: 55, Issue: 2, May 2009), p. 391-399.

[59] K. Hung and D. W. Lin (2008) Optimal Delay Estimation for Phase-Rotated Linearly Interpolative Channel Estimation in OFDM and OFDMA Systems. IEEE Signal Processing Letters (Volume: 15), p. 349 – 352.

[60] H. Li and W. Ye (2015) Study of Digital Modulation Signal Error Vector Magnitude Based on Vector Signal Analyzer. 2015 8th International Symposium on Computational Intelligence and Design (ISCID).

[61] H. Arslan and D. Singh (2006) The role of channel frequency response estimation in the measurement of RF impairments in OFDM systems. 2006 67th ARFTG Conference.

[62] wiki.wireshark.org (Accessed 3.7.2020) evolved Common Public Radio Interface (eCPRI). URL: https://wiki.wireshark.org/eCPRI

[63] wiki.wireshark.org (Accessed 3.7.2020) Libpcap File Format. URL: https://wiki.wireshark.org/Development/LibpcapFileFormat