Optimised user plane routing in a 5G mobile communications network

Darshan Srinivas

Optimised user plane routing in a 5G mobile communications network

By

Darshan Srinivas

in partial fulfilment of the requirements for the degree of

Master of Science
in Electrical Engineering
Track Wireless Communications and Sensing

at the Delft University of Technology,
to be defended publicly on Friday September 30, 2022 at 10:00 AM.

Thesis Committee:	Ir. Rogier Noldus	TU Delft, Ericsson
	Dr. Remco Litjens, MSc	TU Delft, TNO
	Dr. Ir. Przemyslaw Pawelczak	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/

Preface

This thesis marks the finalisation of my studies at the Delft University of Technology for the Master of Science degree in Electrical Engineering in the track of Wireless Communication and Sensing. This mission has been accomplished due to the support of many individuals.

First, I would like to thank Ir. Rogier Noldus for agreeing to be my daily Supervisor. Working under his guidance and having discussions about the telecom field has been a great experience. I thank him for his valuable guidance, patience, constant supervision and timely suggestions during this study. It is with his feedback and guidance that I have been able to produce this quality work and come up with good outcomes. I am grateful for his support.

Next, I would like to thank Dr. Remco Litjens, MSc and Dr. Ir. Przemyslaw Pawelczak for giving their valuable feedback and agreeing to be a part of my thesis committee.

I want to express my appreciation for all my friends, who have been kind, understanding and helpful during this journey. Without them, doing MSc during a pandemic would have been very difficult.

Finally, I would like to thank my family, who has been a constant support throughout my journey. It is my pleasure to dedicate this accomplishment to them.

Darshan Srinivas

Delft, September 2022

Abstract

With the increase in the number of user devices and applications, the 5G systems (5GS) user plane is bound to be burdened. More work towards independent scaling and optimisation of the 5GS user plane has to be done. The 4G base stations (eNodeB) are mainly deployed as monolithic units, whereas, in 5GS, the 3GPP 5G Next Generation base station (gNodeB) can be divided into Radio Unit (RU), Distributed Unit (DU) and Centralised Unit (CU). The CU can be divided into two logical components, the Centralised Unit-Control Plane (CU-CP) and the Centralised Unit-User Plane (CU-UP), extending Control and User Plane Separation (CUPS) approach into RAN. With the introduction of CUPS into RAN, more independence will be provided to the user plane. Moreover, using Open-RAN (O-RAN), all these components can be deployed at different locations as Virtual Network Functions (VNFs). We propose a 5GS architecture in this study, which optimises the user plane.

The key objectives of this thesis are twofold. Firstly, to compare the impact of co-locating the CU-UP with the DU at a distributed edge cloud location against co-locating the CU-UP with the CU-CP at a centralised regional cloud location. Secondly, with the aid of functional split design, virtualisation and O-RAN, we want to explore whether dynamically deploying the CU-UP and DU on a single physical host machine at a distributed location while centralising the CU-CP can enhance RAN development. To achieve this, we consider the 3GPP architecture as a reference and propose a new architecture and enhanced communication mechanism between CU-UP and DU. An analytical model was designed to evaluate the proposed architecture's latency gains in the IP transport network, and a simulation model was designed to evaluate the proposed architecture's communication latency. Furthermore, flow diagrams involving signalling of PDU session establishment are also presented.

We present an analysis and overall evaluation of the proposed architecture by comparing it with the reference architecture based on practical architectural aspects and PDU session signalling diagrams. The results of the calculation model and output of the simulation model indicated a significant latency improvement when the new architecture is employed. The new architecture found that, on average, 1.5 ms/packet of midhaul delay was reduced. And based on the flow diagram comparisons, it was found that the new architecture introduces overhead in terms of control plane signalling.

Acronyms

5G-NR 5G New Radio.

5GC 5G Core.

5GS 5G systems.

AI/ML artificial intelligence and machine learning.

AMF Access and Mobility Management Function.

BBU Baseband Unit.

C-RAN Centralised/Cloud RAN.

CAPEX capital expenditure.

CN Core Network.

CNF Cloud-native Network Function.

CoMP Coordinated Multi-points.

COTS Commercial off-the-shelf.

CP control plane.

CPRI Common Public Radio Interface.

CRC Cyclic Redundancy Check.

CU Centralised Unit.

CU-CP Centralised Unit-Control Plane.

CU-UP Centralised Unit-User Plane.

CUPS Control and User Plane Separation.

D-RAN Distributed RAN.

DL downlink.

DN Data Network.

DRB Data Radio Bearer.

DU Distributed Unit.

E-UTRAN Evolved-UMTS Terrestrial Radio Access Network.

E1AP E1 Application Protocol.

E2AP E2 Application Protocol.

E2SM E2 service model.

eCPRI enhanced Common Public Radio Interface.

eMBB enhanced Mobile Broad-Band.

eNodeB 4G base stations.

EPC Evolved Packet Core.

ETSI European Telecommunications Standards Institute.

F1AP F1 Application Protocol.

FCAPS Fault, Configuration, Accounting, Performance, Security.

FWA fixed wireless access.

GGSN gateway GPRS support node.

gNodeB 3GPP 5G Next Generation base station.

gNodeB-CU gNodeB-Centralised Unit.

gNodeB-CU-CP gNodeB-Centralised Unit-Control Plane.

gNodeB-CU-UP gNodeB-Centralised Unit-User Plane.

gNodeB-DU gNodeB-Distributed Unit.

GTP-C GPRS Tunnelling Protocol for control plane.

GTP-U GPRS Tunnelling Protocol for user plane.

HLS High Layer Split.

IoT Internet of Things.

IPC Inter-process communication.

KPI Key Performance Indicators.

LLS Low Layer Split.

LTE Long Term Evolution.

MIMO Multiple Input Multiple Output.

MME Mobility Management Entity.

mMTC massive Machine-Type Communication.

MNOs Mobile network operators.

N1 Non Access Stratum (NAS) interface.

NAS-MM NAS-Mobility management.

NAS-SM NAS-Session Management.

NB-IoT Narrowband Internet of things.

near-RT RIC Near-Real Time RAN Intelligent Controller.

NF Network Function.

NFV Network Function Virtualisation.

NG-RAN Next Generation-Radio Access Network.

NGAP NG Application Protocol.

NIC Network Interface Card.

Non-RT RIC Non-Real Time RAN Intelligent Controller.

NR-MAC Medium Access Control layer.

NR-PDCP Packet Data Convergence Protocol layer.

NR-PHY Physical layer.

NR-RLC Radio Link Control layer.

NR-RRC Radio Resource Control layer.

NR-SDAP Service Data Adaption protocol layer.

NRF Network Function Repository.

O-CU-CP O-RAN-Centralised Unit-Control Plane.

O-CU-UP O-RAN-Centralised Unit-User Plane.

O-DU O-RAN-Distributed Unit.

O-RAN Open-RAN.

O-RU O-RAN Radio Unit.

OPEX operational expenditures.

OS Operating systems.

P-GW Packet-Gateway.

PCF Policy Control Function.

PDCP-C PDCP for CP stack.

PDCP-U PDCP for UP stack.

PFCP Packet Forwarding Control Protocol.

PLMN public land mobile network.

QoS Quality of Service.

RAN radio access network.

RIC RAN Intelligent Controller.

RNL radio network layer.

RoHC Robust Header Compression.

RRH Remote Radio Head.

RRM Radio resource management.

RU Radio Unit.

S-GW Serving-Gateway.

SBA Service-Based Architecture.

SBI Service Based Interface.

SDN Software Defined Network.

SGSN serving GPRS support node.

SMF Session Management Function.

SMO Service Management and Orchestration.

SRB Signalling Radio Bearers.

TB Transport Block.

TEID Tunnel Endpoint Identifier.

TNL transport network layer.

TR technical reports.

TS technical specifications.

UDM Unified Data Management.

UDR Unified Data Repository.

UDSF Unstructured Data Storage Function.

UE User Equipment.

UHD Ultra-High Definition.

UL uplink.

UP user plane.

UPF User Plane Function.

URLLC Ultra-Reliable Low Latency Communication.

V2X vehicle-to-everything.

VM virtual machines.

VNFs Virtual Network Functions.

vNIC virtual Network Interface Card.

VR/AR Virtual Reality/Augmented Reality.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides the necessary background information for the presented research study. In this chapter, Section 1.1 introduces the reader to the present day 5G and its requirements. In Section 1.2 and Section 1.3, significant architectural improvements and crucial technologies of the 5G systems (5GS) are elaborated to bring out the essence of this research study. Section 1.4 and Section 1.5 explain parts of the challenges faced by radio access network (RAN) deployment, and the research objective of this research study is mentioned, respectively. Section 1.6 provides the methodology adapted to reach the goals of this research. Finally, Section 1.7 present the outline of this thesis.

1.1 Present-day 5G

With the growing Internet of Things (IoT) usage, the 5G network will have to handle billions of devices and many more links and connections. The 5GS must provide top-notch service in maintaining low latency and a variety of data. The operational performance of 5GS (spectral efficiency, data rates, latency) should considerably improve [1]. Figure 1.1 depicts the past, present and future predictions of Global mobile network data traffic compared to fixed wireless access (FWA).

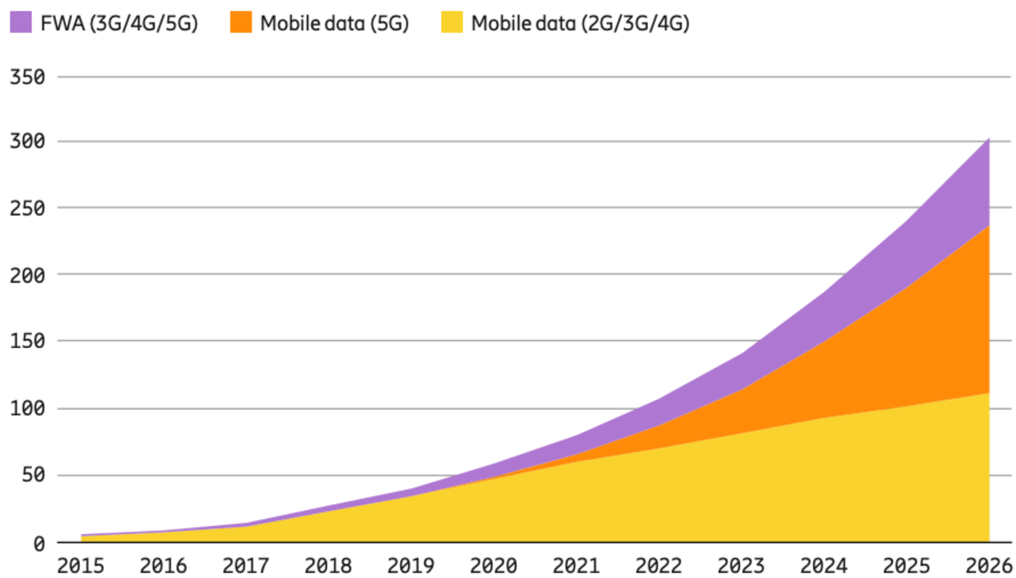


Figure 1.1: Global mobile network data traffic (EB per month) [2]

In 2025, there is a forecast of 10.3 billion mobile connections and 25 billion IoT devices. Furthermore, nearly 30 percent of these connections will depend on 5G communication [3]. On average, a smartphone user consumed about 10GB of data monthly in 2020, expected to reach 35GB/month by the end of 2026. Excluding the fixed wireless access (FWA) traffic, 49 exabyte (EB)/month was consumed by mobile data traffic globally in 2020. The global mobile network data excluding the FWA is forecast to have a five-fold increase and reach 237 EB/month [2]. 5G system is set to define the next significant transformation in the communication industry. The three prominent use cases expected to be supported by 5G are:-

- enhanced Mobile Broad-Band (eMBB) is designed to handle significantly risen data rates, denser deployments and very high data volume, including highly crowded areas, provide flawless coverage, and maintain improvised throughput during high user mobility [1]. Well-known applications such as Virtual Reality/Augmented Reality (VR/AR), Ultra-High Definition (UHD) Video, and other video-related content will be served under the eMBB category.

- Ultra-Reliable Low Latency Communication (URLLC) handles applications with high-priority information, which has critical latency and reliability requirements. Public safety, remote surgery and vehicle-to-everything (V2X) are some applications that fall under this use case.

- massive Machine-Type Communication (mMTC) is the 5G equivalent of Narrowband Internet of things (NB-IoT) of LTE-A. It manages an immensely high number of devices attached, which requires low data rates and power consumption. Industrial automation and smart cities are a few example applications of this use case.

Figure 1.2 depicts a few example applications of the above explained use cases.

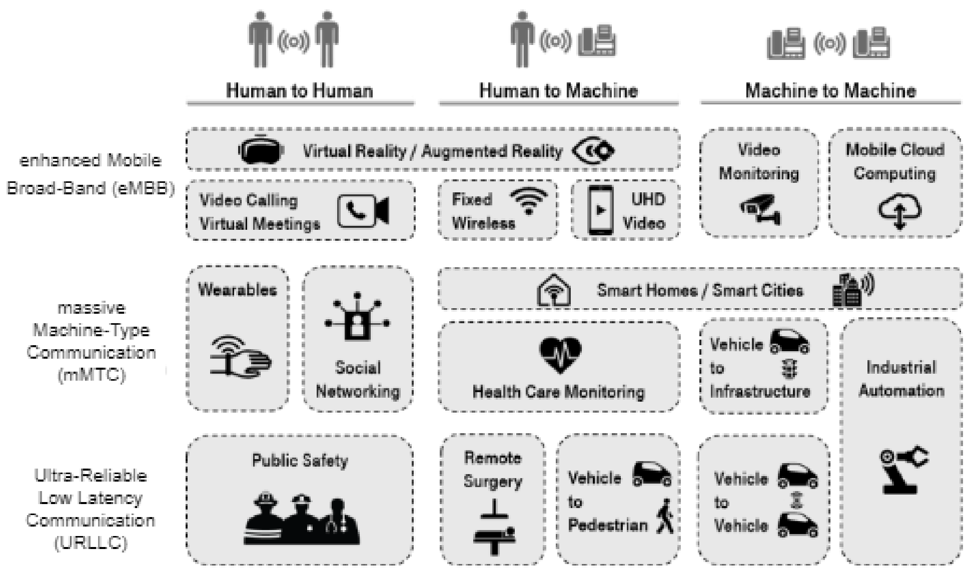


Figure 1.2: 5G use case scenarios [4]

What should the 5G system be capable of handling? International mobile telecommunications-2020 (IMT-2020) defines the minimum technical requirements of the radio interface's Key Performance Indicators (KPI) for 5G. According to IMT-2020 evaluation requirements, the peak data rate expected for an eMBB application is 20 Gb/s for downlink (DL) and 10 Gb/s for uplink

(UL). The minimum requirement of control plane latency for eMBB and URLLC applications is 20 ms. The maximum allowed user plane latency for eMBB and URLLC is 4 ms and 1 ms, respectively. Furthermore, around one million devices/km^2 have to be supported by mMTC [5].

1.2 Advancements in 5GS architecture

1.2.1 CUPS approach

In 2.5G and early 3G network, the control plane signalling and the user plane data are handled by both serving GPRS support node (SGSN) and gateway GPRS support node (GGSN), which is an inefficient approach. In Evolved Packet Core (EPC) of 4G, most control signalling and user plane data were separated, but to a limited extent. The Mobility Management Entity (MME) is responsible for handling most of the control signalling, such as UE registration, setting up/closing data sessions, and selecting the Serving-Gateway (S-GW) and the Packet-Gateway (P-GW). The primary role of S-GW and P-GW is to process the user plane data. However, they are still responsible for some control signalling of user plane management, such as inter-eNB handovers in the user plane, bearer context establishment or modification, policy enforcement, charging support and lawful interception. Figure 1.3 represents the EPC architecture without the CUPS approach.

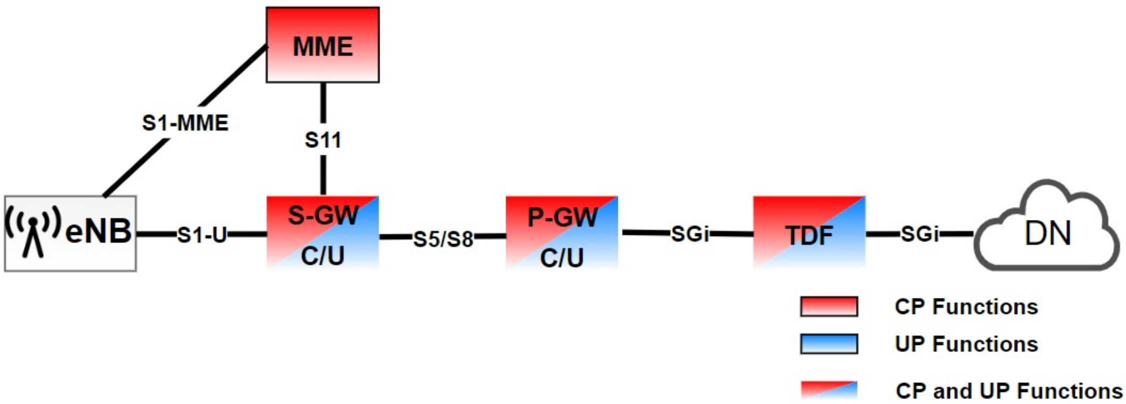


Figure 1.3: EPC architecture without CUPS

Every use case has a different amount of signalling, an aggregate of Core Network (CN) signalling and access network signalling, and the proportion of control signalling and user data differs. In some use cases, the user data is minimal, but the control signalling is too large (e.g., banking applications where authentication signalling is critical) and vice versa (e.g. online video streaming). With the surge of smart devices, video content, and other applications, the user data traffic handled by the network has increased. Also, the demand for low latency and Quality of Experience is increasing. To meet these requirements efficiently, In release 14, 3GPP[1] went a step further and introduced Control and User Plane Separation (CUPS) in EPC [6]. This approach separates the control and user plane functionality of the S-GW, P-GW, and traffic

[1]The 3GPP is a group of standard development organizations (SDOs) that develop technical specifications (TS), technical reports (TR), and standard protocols for cellular telecommunication. The "3GPP releases" log the progress made by the 3GPP SDOs.

detection function (TDF), providing an architectural enhancement in EPC. S-GW-C, and P-GW-C form the control plane. S-GW-U and P-GW-U together form the user plane. Figure 1.4 depicts the EPC architecture with CUPS approach.

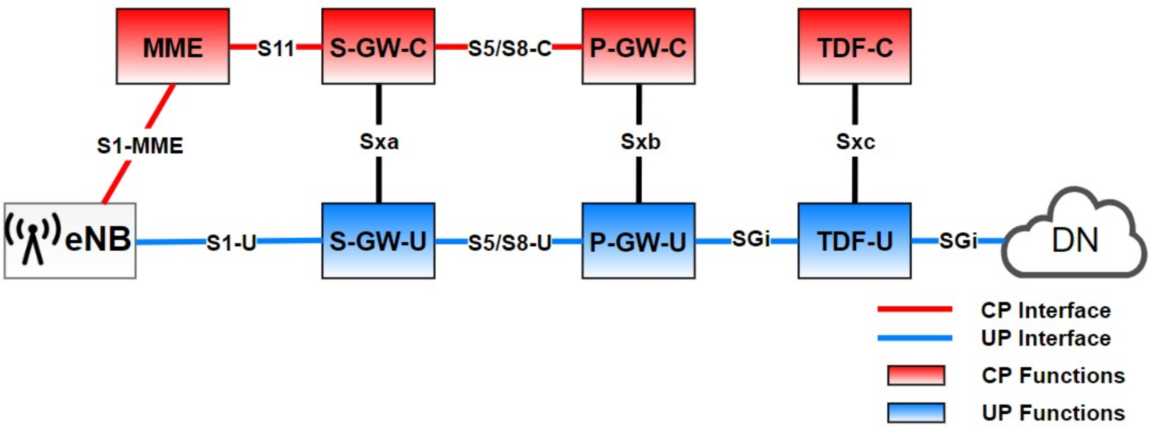


Figure 1.4: EPC architecture with CUPS

The foremost advantage of the CUPS approach is independent scaling of control and user plane functions. Based on an increased User Equipment (UE) data usage requirement, more user plane entities can be added with just the existing number of control plane nodes. It helps achieve critical latency requirements by selecting the user plane nodes closer to the RAN. Also, the changes or upgrades in CP and UP functions are independent.

In 5GS, CUPS is upgraded according to the 5GS architecture. The P-GW-C, along with the session management role of the MME, forms the Session Management Function (SMF). Furthermore, the PGW-U and S-GW-U are merged into one Network Function (NF) called User Plane Function (UPF) (see also Figure 3.3). In EPC, the CUPS approach is limited only to the CN. In 5GS, the CUPS approach can be implemented even in RAN (see also Section 3.2.4). Figure 1.5 depicts the NG-RAN with CUPS approach.

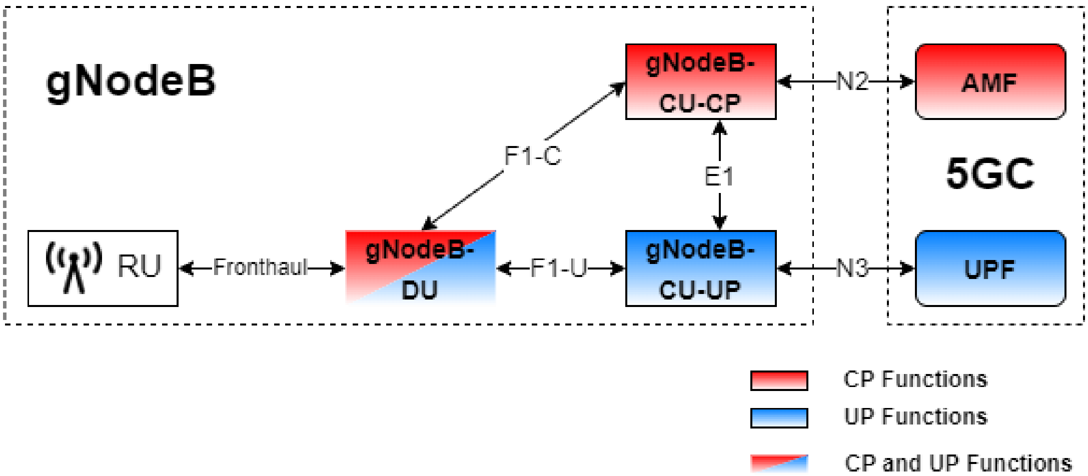


Figure 1.5: NG-RAN gNodeB and 5GC with CUPS approach extended up to gNodeB-DU.

In release 15, 3GPP released a functional split method with eight options for RAN deployment, where the functionalities of the 3GPP 5G Next Generation base station (gNodeB) are split into the so-called Centralised Unit (CU) and the Distributed Unit (DU). Furthermore, using option 2.2 [7] of functional split, the CU can be split into Centralised Unit-Control Plane (CU-CP) and Centralised Unit-User Plane (CU-UP) extending CUPS up to the DU. The Functional split is explained in detail in Section 3.2.2.

1.2.2 O-RAN

Expanding on the European Telecommunications Standards Institute (ETSI) NFV standard architecture and SDN, Open-RAN (O-RAN) alliance offers MNOs numerous multilayer cloud deployment choices by employing COTS hardware and virtualisation software in the form of virtual machines (VM) or containers [8]. In this work, O-RAN reference architecture is chosen to exploit the advantages of NFV. The architecture, Network Functions (NF), impact on operational expenditures (OPEX)[2] and capital expenditure (CAPEX)[3] , and other advantages of O-RAN are further explained in the Section 3.3.

1.3 Crucial technology enablers

This section elaborates on a few technology enablers contributing toward the virtualisation of 5GS. Virtualisation is essential in this study as 5G deployments are relatively more straightforward, and O-RAN depends on virtualisation.

1.3.1 Network function virtualisation

By developing standard virtualisation technology to combine various network equipment types onto industry-standard large volume servers, switches, and storage, Network Function Virtualisation (NFV) seeks to alter how network operators plan and manage networks and network services [9]. NFV alters network designs by encapsulating network operations in software that runs on various industry-standard server hardware. Without the requirement for additional equipment, the software may be dynamically relocated to or instantiated in numerous places within the network as needed [9]. The 5G network architecture is expected to be a programmable network platform with flexibility and adaptability to enable various services and varying application requirements, [10]. As explained earlier in Section 1.1, the requirements of 5G are roaring high. NFVs are envisioned to meet these requirements by enhancing the RAN architecture and its functionality and reducing the cost burden on Mobile network operators (MNOs). In NFV, the network functionalities are implemented as Virtual Network Functions (VNFs). The VNFs are implemented on cloud deployments or servers with high computational power using virtualisation mechanisms (see Section 4.2.1). For example, instead of employing dedicated baseband processing hardware units, cloud computing resources are used for signal processing in NFV. This resource agglomerating will enhance flexibility and minimise computational requirements and cost [11].

1.3.2 Software Defined Networking

5GS may supply cellular phones with various sophisticated communication options and services. There are several issues with this service approach. The administration of a large

[2]OPEX includes electricity, site rentals, maintenance, and salary for site engineers.
[3]CAPEX expenses constitute creating the network, such as designing the network, RF and router hardware costs, software licenses, line installations, and power and cooling required for each cell site.

number of devices running various services is one of its biggest challenges. Software Defined Network (SDN) is considered a crucial solution to address this issue [12]. SDN makes the deployments more flexible, energy and cost-efficient, and scalable. The separation of the control and user planes is one of the critical ideas of SDN. By directing all control responsibilities to a controller, SDN simplifies network configuration and maintenance. The controller handles all the control signalling, whereas the user plane dedicates itself entirely to forward user data. The user plane consists of simple switches which forward the user data based on the routing table provided by the controller. Figure 1.6 depicts the architecture of SDN.

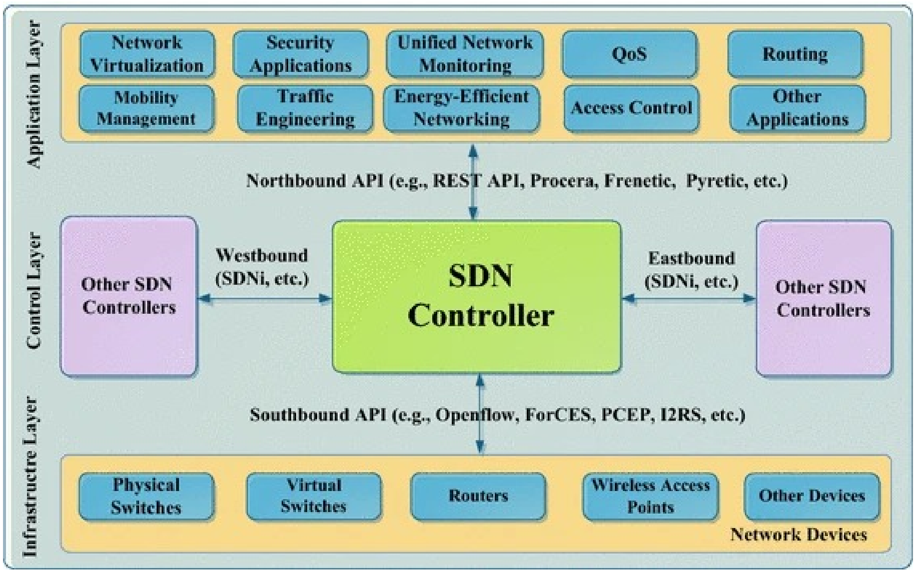


Figure 1.6: SDN Architecture [13]

The application, control, and infrastructure layers are the three core layers of the SDN architecture. Often known as the user plane, the infrastructure layer utilises forwarding devices such as virtual and physical switches to forward user data. These switches perform data forwarding based on the instructions received from the controller via the southbound interfaces. The SDN controllers (such as RYU controller [14]) manage the control layer/control plane and perform control functions through southbound interfaces (such as Openflow [15]). The controllers can also be connected to other controllers using the eastbound/westbound interfaces. The application layer is situated above the control layer and connected via northbound interfaces. The application layer consists of SDN applications such as monitoring, QoS management, and routing. The network needs of SDN applications are sent to the controllers via northbound interfaces such as REST API or Java API [13].

1.4 Research motivation and challenges

CUPS in RAN architecture, virtualisation of Network Functions and Openness in RAN, that is, O-RAN, opens up a significant opportunity for research in RAN and its protocol stack. As we have learnt from Section 1.2.1, the main advantage of the CUPS approach is independent scaling and deployment of CP and UP entities. Finding optimal locations for the control plane (CP) and user plane (UP) entities is essential. It can be determined by placing them at different locations and studying them. Furthermore, with its openness and virtualisation, O-RAN claims to be a promising solution for modern-day requirements. It is essential to study flexibility and the impact of the O-RAN on the 5GS architecture. Its advantages while employing CUPS within RAN approach can also be researched. The above-mentioned 5GS developments enable us to introduce different deployment options and explore their relative merits and demerits. In this thesis study, we want to explore the merits and demerits of locating the CU-UP at a location closer to the UE

The radio coverage of many 5G New Radio (5G-NR) frequencies is less compared to Long Term Evolution (LTE), but the bandwidth is much higher. For 3.5 GHz 5G-NR frequency, the estimated radio coverage depending upon propagation environment is 2057 meters and expected bandwidth is 500 MHz. Whereas, for 265 GHz 5G-NR frequency, the estimated radio coverage depending upon propagation environment is 277 meters and expected bandwidth is 3250 MHz [16]. Furthermore, with the increase in the number of equipment, data consumption and demand for high data rates, more CU and DU units are required to process and forward the information in both UL and DL. Hence, the MNOs must set up more cell sites to provide the required coverage and more CUs and DUs to manage many connections, increasing the Total Cost of Ownership (TCO). The TCO is essential for MNOs to consider when deploying the 5GS. CAPEX and OPEX are two significant parts of TCO. As the requirement for the number of gNodeB deployments increases, the TCO also increases [17]. Moreover, using dedicated and proprietary hardware for RAN deployment in 5GS will further increase the expenses, and an alternate approach must be discussed.

Tight coupling of CP/UP functions in RAN is typical, particularly in the lower layers of the radio protocol stack. Separating CP and UP handling entirely might be difficult, and potential performance impact may arise if the processing is not co-located [18].

1.5 Research objective

In this section, the objectives of this study are presented based on requirements derived for 5G networks and obstacles that arise:

1. Assessment of the 3GPP proposed user plane architecture for the 5GS while employing CUPS approach in RAN, and evaluate the so-called GTP-U tunnelling procedure used in the user plane deployment options.

2. Proposal and assessment of a new user plane architecture for 5GS employing O-RAN (with CUPS approach) and an enhanced GTP-U tunnelling procedure, aiming towards optimising the user plane routing of 5GS

1.6 Methodology

The following states the methodology employed in this study to achieve the objectives indicated in Section 1.5.

1. Obtain the 5G user plane requirements according to the 3GPP standard.

2. Conduct a literature study on the mobile network architectures that employ CUPS and related work which have extended the CUPS approach to RAN.

3. Study the 5GS network architecture, its transport network, crucial technology enablers such as NFV and SDN, and architectural advancements such as O-RAN and CUPS.

4. Obtain an in-depth understanding of 5G RAN and O-RAN.
5. Study the 3GPP proposed 5G-RAN architecture that employs CUPS and consider it as the reference model.
6. Describe a new RAN architecture using O-RAN (Virtualizing the RAN approach) which optimises the user-plane routing in 5GS.
7. Derive latency gains and resource optimisation achieved by the proposed system over the existing system using analytical and simulation model.
8. Use the results to conclude, make recommendations, identify limits, and provide suggestions for future research.

1.7 Thesis outline

The thesis report consists of seven chapters. Chapter 2 provides literature study required for this thesis study. Chapter 3 introduces the network architecture fundamentals of the 5GS, NG-RAN architecture and O-RAN. In Chapter 4, we discuss the 3GPP reference architecture, its protocol stacks and signalling involved in PDU session establishment. It is followed by a proposal for new architecture for optimisation of the user plane, and introduction to its new protocol stack and signalling involved in the PDU session establishment. In Chapter 5, a calculation model and a simulation model used for performing experiments related to this thesis is explained. In Chapter 6, the outcomes of the experiments and also the overall comparisons of the proposed architecture against the reference architecture is presented. Finally in Chapter 7, we conclude the thesis study and also suggest future work that can be done on the subject.

Chapter 2

Literature study

The first 3GPP release for the 5G-NR came out in 3GPP Release 15. 3GPP Release-15 addresses the introduction to the 5GS and 5G-NR. Also, the requirements for the three prominent use cases of 5G (eMBB, mMTC, and URLLC) are described. Initially, for a smooth transition from a 4G system to 5GS, the 3GPP introduced a Non-Stand-Alone architecture (NSA) in Release-15, where EPC supports both 5G-NR and LTE radio technologies along with their respective base stations. In later versions of the same release, the Stand-Alone (SA) approach consisting of a 5G core that supports 5G-NR and 5G RAN was released [19].

After introducing the URLLC in the 3GPP Release-15, 3GPP Release-16 introduces functionalities that enhance the URLLC. Some of the main functionalities introduced are QoS monitoring, enhancements of the session continuity mechanism, and redundant transmissions. 3GPP Release-16, along with the new 5G KPIs, adds Performance assurance property to assure performance of 5GS, where NG-RAN and 5GC have the new 5G performance measurements established. Radio resource utilization, packet delay, PDU session management, QoS management, and handovers are a few measurement categories helpful for this research study [20].

Nayak et al. [21] suggested an SDN architecture that centralises certain RAN functionalities into the CN of the 5GS. The approach moves the control functions of NG-RAN to the 5GC, converting the gNodeB base station into a pure data plane node. By relocating the control functions of RAN, namely the RRC protocol layer and the RRM features, from the gNodeB to the AMF, the gNodeB is changed into a node with just data plane capabilities, which is handled via a standard interface from the core network's centralised control function. By centralising the RRC layer and RRM functionalities, the authors aim to reduce signalling costs between the RAN and the CN and attach (registration)-time for UE by altering the NG interface and removing the NG-AP; and improving mobility management which leads to better system throughput.

Arnold et al. [18] propose a CP/UP split RAN architecture employing the SDN switches and controller approach. According to the architecture, CUs responsible for handling the high layers of the RAN protocol stack are called Central Access Controllers (CACs), hosting CP and UP functions at a centralised location. CACs are separated into a UP section (CAC-U) and a CP section (CAC-C). Furthermore, the lower layers are handled by DU and deployed close to the antenna location. The SDN switches or routers are responsible for transferring UL and DL UP data between CN and UE. The SDN controller enforces the CN mobility management function and is responsible for forwarding the user-requested data to the appropriate antenna and the intended UE. The authors remark that this SDN-based approach is supposed to reduce the control overhead compared to the so-called GPRS tunnelling protocol (GTP) solution used in LTE. In order to communicate with UEs, Radio Resource Control (RRC) messages are generated and sent to the Packet Data Convergence Protocol (PDCP) layer in the UP NF. Once the PDCP layer receives the messages from the CP NFs, the UP NF begins processing the messages, and antennas deliver the messages. Therefore the communication between the UE and CAC-C is happening through CAC-U, which is not a complete separation of CP/UP functions. To manage the interaction complexity between CP and UP, the authors advocate a dedicated interface between CP and UP.

Utilising recent developments in SDN and OpenFlow protocol, the authors of [22] have presented an innovative SDN-based design for the 5G network infrastructure to improve flexibility, administration simplicity, and proprietary independent hardware. The authors have employed SDN on 3GPP 5GS architecture and evaluated the performance of the registration and handover procedure of the control plane. The user plane is considered as the data plane. The network's control plane functions, such as AMF and SMF, are consolidated together and moved on top of an SDN controller. These network services are implemented as applications and communicate with the controller by utilising the SDN controller's northbound interface. Contrarily, the 5G network's data plane (the UPF) is built as an SDN switch. The controller and UPF communicate via Openflow. The Results of the proposed SDN based 5G architecture reduce the end-to-end delay by as much as 18 to 62 percent compared to the conventional 5G architecture.

The 5GS requires high throughput, usually delivered by small cells, but it also leads to frequent cell change due to mobility, creating a control plane overhead. Hence the authors of [23] propose a multi/dual connectivity approach to scale control and user plane of gNodeB independently by deploying the control plane using macro cell and user plane using small cell. CP/UP (C/U) split Multi-connectivity is the new design principle proposed by the authors to handle UP/CP transmission independently for a UE in a multi-connectivity session. The cells in the gNodeB are split into CP-gNB and UP-gNB to manage only the CP and UP functions, respectively. CP-gNB cells are deployed as a macro, with one frequency, for example, <6GHz and UP-gNB are deployed as a small cell with one or more frequencies to deliver high throughput. The study proposes a different protocol architecture for the CP-gNB and UP-gNB of the suggested design. It is evaluated against legacy LTE dual connectivity and found to have gained in UE throughput.

Most of the research in the literature mentioned above proposes a 5G network architecture employing either SDN, NFV, or CUPS considering LTE/Non-Stand Alone (NSA) 5GS as a reference architecture. Hence, in this thesis study, a new 5GS architecture optimising the user plane transport is proposed while considering 3GPP's Stand Alone (SA) 5GS as a reference architecture.

Chapter 3

Network Architecture Fundamentals

This chapter provides an in-depth explanation of the building blocks of this research study. Section 3.1 explains the 5G System Network Architecture, Network functions, and their functionalities. Section 3.2 contains a comprehensive study of RAN, its deployment options, the functional split options, and the architecture of the split CU. In Section 3.3 the O-RAN alliance is introduced, and its architecture is presented. Section 3.4 and Section 3.5 elaborate on GTP-U tunnels and PDU sessions, respectively.

3.1 5GS network architecture

As explained in Chapter 2, 3GPP initially introduced the Non-Stand Alone (NSA) architecture followed by Stand Alone (SA) architecture to ease the transition process from the 4G system to 5GS. Figure 3.1 represents the NSA architecture of 5GS and Figure 3.2 represents the SA architecture of 5GS.

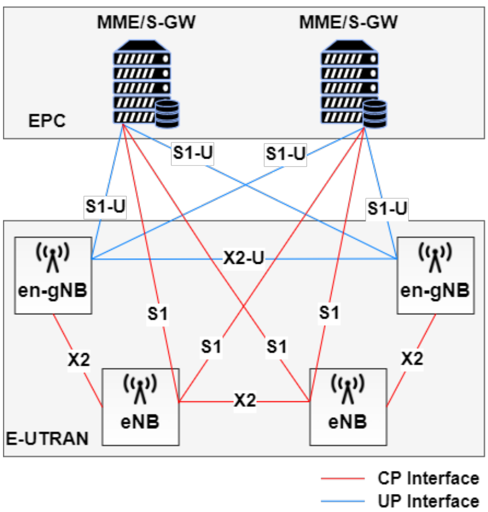


Figure 3.1: The NSA architecture of 5G, where the EPC of 4G system manages the core network. RAN control plane is handled by eNB and RAN user plane is handled by en-gNB[a] [19].

[a]en-gNB represents a gNodeB that can connect with EPC and eNB to support NSA

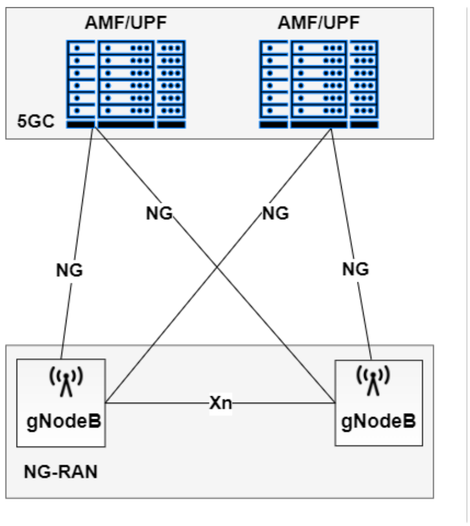


Figure 3.2: The SA architecture for 5GS, where the 5G core of 5GS manages the core network, gNodeB of NG-RAN handles the access network and hence does not require any 4G network entities [19].

In the NSA architecture, the UE is attached to the en-gNB using the 5G-NR interface for user plane connection. And UE is attached to the 4G base stations (eNodeB) using the LTE interface for control plane connection. The EPC of the 4G system acts as the core network. The en-gNB is attached to the EPC via the S1-U interface and to an eNB via the X2 interface. Employing this approach, the user can utilise the 5G-NR capacities without replacing the existing core network, but only the 4G services are made available. The approach is called the NSA as it works alongside the existing 4G Radio-LTE and EPC [19].

In the SA architecture, the UE is attached to the 3GPP 5G Next Generation base station (gNodeB) using the 5G-NR interface, and the CN is supported by 5G Core (5GC). The gNodeB is attached to the Access and Mobility Management Function (AMF) and User Plane Function (UPF) of 5GC via the N2 and N3 interface, respectively. The SA architecture, where the gNodeB is connected to the 5GC using the NG interface, is called the 5GS, the fifth generation of mobile communications [19]. Figure 3.3 represents the reference point architecture of 5GS network.

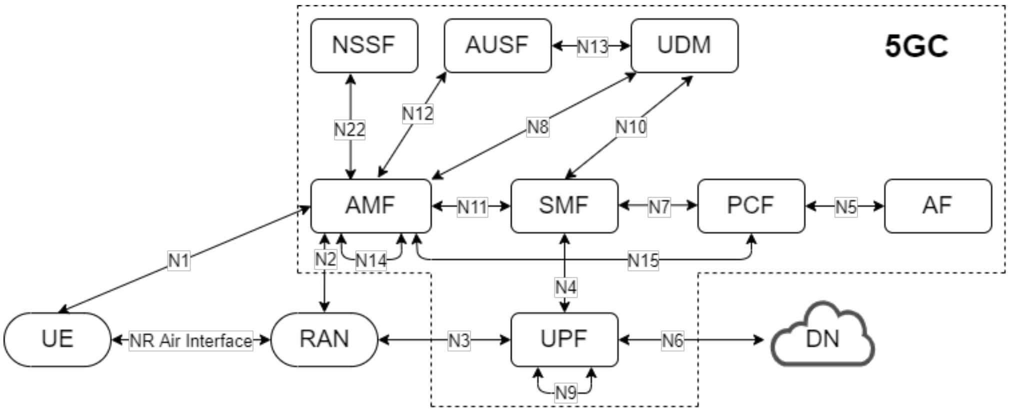


Figure 3.3: Reference point architecture of 5GS network as developed by 3GPP. The 5GC, the UE, and the RAN together form the 5G System (5GS) architecture.

AMF plays a vital role in the 5GC, similar to MME of EPC. It is an essential point of contact to the control plane of 5GC, which can be accessed by UE through RAN. AMF terminates the Non Access Stratum (NAS) interface (N1) and the N2 interface, a control plane interface between RAN and 5GC. Network Function Repository (NRF) assists AMF in selecting an appropriate SMF. All NFs in the 5GC, except UPF, belong to the control plane of the 5GC. UPF is the only NF that belongs to the user plane in 5GC. UPF functionality was handled by the SGW-U and PGW-U (based on the CUPS approach) in EPC of 4G. UPF is connected to SMF over the N4 interface. UPF, under the guidance of SMF, is responsible for managing the UE-requested PDU sessions. Policy Control Function (PCF) coordinates with SMF over N7 interface for policy and charging control of SM-related functionality (For ex: establishing or modifying PDU sessions). PCF is connected to AMF over the N15 interface to enforce access and mobility policies. Unified Data Management (UDM) in 5GC is used for subscription management similar to HSS in EPC. UDM provides the required user subscription data to AMF and SMF over N8 and N10 interfaces. It also stores information about serving AMF for a UE and SMF for a UE PDU session. Figure 3.4 depicts the service based architecture of 5GS network.

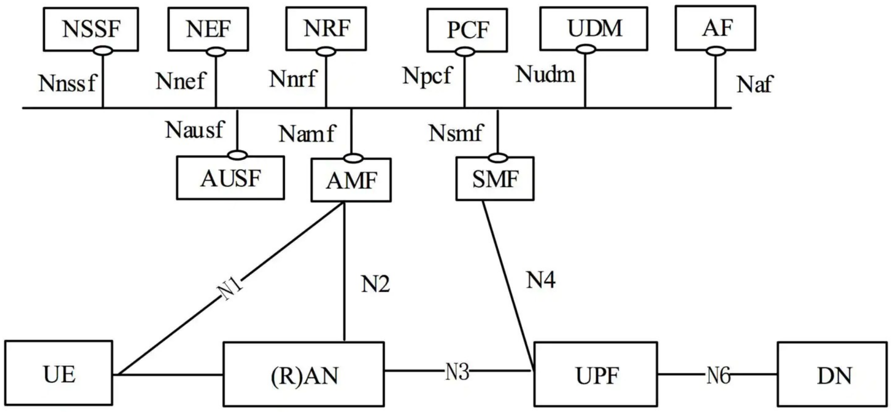


Figure 3.4: Service based architecture of 5GS network [24]

In EPC, the network entities are rigid and require dedicated hardware. A new modular framework called Service-Based Architecture (SBA) was introduced in 3GPP release 15 to virtualise 5GC and increase flexibility. Unlike traditional network entities, the 5GC elements are called Network functions (NFs) and are interconnected via interfaces of a shared structure for data connectivity and other services. NFs are deployed using NFV and SDN. The control NFs are interconnected over Service Based Interface (SBI), which depends on HTTP transport [16]. By SBA, NFs are scalable, and services of any NF are available to any authorised NFs. Multiple NF instances can exist at the same NF set or be accessed from multiple locations. The prominent NFs are AMF, SMF, and AF [19].

3.1.1 NFs and their functionalities

In the control plane of 5GC, none of the NFs, including the UDM, SMF, and PCF, save their data within their VMs or Containers. For stateless NF processes, all of its data are instead saved remotely at the Unified Data Repository (UDR) and the Unstructured Data Storage Function (UDSF) for structured and unstructured 3GPP NF data storage, respectively. In other words, if one of the NFs fails, its previous state and data can be recovered from the UDR and UDSF by initiating the backup NF (stateless), allowing operations to continue

3.1.1.1 AMF

It manages NAS ciphering and integrity algorithms. It can also accommodate policy-related functionalities when required. Registration management, connection management, reachability management, mobility management, and access authentication are a few crucial functionalities of AMF.

3.1.1.2 SMF

SMF in 5GC is similar to session management part of MME from EPC. One of its primary functions is to manage the UPF. It can create, modify, and release the PDU session tunnels between RAN and UPF based on requests sent via NAS message. It instructs UPF in routing data traffic to the right destination. SMF assigns UE IP address, where it is assisted by UPF or DN for this functionality.

3.1.1.3 UPF

UPF implements the QoS flows assigned by the SMF in the user plane. The 5GS Quality of Service (QoS) model is explained in Appendix A. UPF and RAN are connected by the N3 interface. UPF is responsible for providing IP address to the connected UEs and enforcing the policy rules of the user plane part. To support inter and intra-radio access technology (RAT) mobility, UPF acts as an anchor point.

3.1.1.4 PCF

PCF supports the 5GC NFs to implement policy rules. It provides PDU session-related policy control. The UDR provides the subscription information, and the AF provides the service information. The PCF collectively utilises this information to provide QoS authorization for a service data flow. PCF might also consider the requested QoS received from SMF for this calculation.

3.1.1.5 UDM

UDM generates 3GPP Authentication Credentials (AKA). It assists in user identification by storing and managing the 5G Subscription Permanent Identifier (SUPI) for every 5GS user. It authorises access such as roaming, depending on user subscription data. UDM depends on subscription data stored in UDR to deliver these services.

3.1.1.6 AUSF

It facilitates the security aspects of 5GS by Authenticating the 3GPP access and untrusted non-3GPP access.

3.2 Radio access network

The Next Generation-Radio Access Network (NG-RAN) generally provides radio access to 5GS. NG-RAN is a network of 3GPP 5G Next Generation base station (gNodeB), which is a radio access node in 5GS, previously known as eNodeB in 4G systems. Furthermore, each gNodeB consists of three fundamental blocks the Antenna and Remote Radio Head (RRH) at the top of a cell site tower, and the Baseband Unit (BBU) placed at the bottom of the cell site tower or a physical location closer to the 5GC. In gNodeB, the information is processed by a 5G-NR protocol stack, the successor of LTE. It consists of the Physical layer (NR-PHY), Medium Access Control layer (NR-MAC) , Radio Link Control layer (NR-RLC) , Packet Data Convergence Protocol layer (NR-PDCP), Service Data Adaption protocol layer (NR-SDAP) and Radio Resource Control layer (NR-RRC), each having its own set of functions (see also Section 3.2.3).

The gNodeB has many essential functions that resemble Evolved-UMTS Terrestrial Radio Access Network (E-UTRAN), such as

- Performing handover procedure and load balancing.
- Transferring the user data
- RAN sharing functionality allows multiple public land mobile network (PLMN) to access RAN resources.

- Ciphering and deciphering of radio channels
- Establishing and releasing the Connection Management (CM) procedure for the UE
- Forwarding the NAS messages and Header compression

In addition, gNodeB can handle network slicing functionality and co-existence with E-UTRAN [25].

3.2.1 RAN deployment options

The MNOs must decide on the type of RAN deployment according to their requirements. Distributed RAN (D-RAN), Centralised/Cloud RAN (C-RAN) and Next Generation-Radio Access Network (NG-RAN) are examples of RAN deployment options. A detailed explanation about D-RAN and C-RAN are provided in Appendix B. NG-RAN with functional split was introduced by 3GPP to overcome the drawbacks of C-RAN and to support the 5GS requirements.

3.2.1.1 NG-RAN

Because of its centralised baseband processing, the C-RAN (see Appendix B.2) design is ideal for promoting CAPEX and OPEX savings and cell coordination. However, the concept of a resource-efficient and cost-effective RAN implementation is hampered by a contradiction between centralised benefit and transportation resource savings [26]. To counter the fronthaul capacity requirements caused by C-RAN implementation, 3GPP has introduced functional split in release 15 [25] (see also Section 3.2.2). Figure 3.5 represents the NG-RAN architecture.

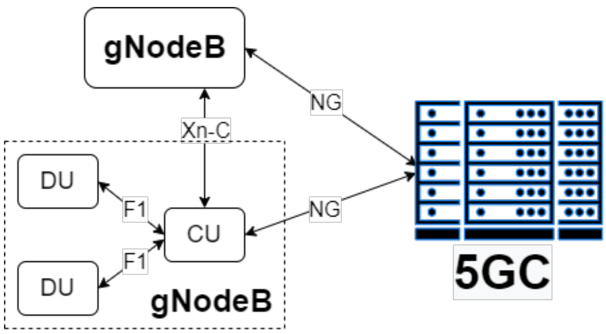


Figure 3.5: The architecture of the NG-RAN wih gNodeB and its interfaces.

In NG-RAN, the NR protocol functions of gNodeB are divided into gNodeB-Centralised Unit (gNodeB-CU) and gNodeB-Distributed Unit (gNodeB-DU) using the functional split options. The high layer NR protocols are processed in gNodeB-CU, and the low layer NR protocols are processed in gNodeB-DU. As the name suggests the gNodeB-CU is located at a more centralised location and the gNodeB-DU is deployed in a distributive manner, closer to the cell site. The NG-RAN architecture comprises a group of gNodeB-CUs and gNodeB-DUs connected to the 5GC through the NG interface. Depending on the termination point, the interface between the NG-RAN and 5GC is referred to as N2 or N3. The F1 interface connects gNodeB-CU and gNodeB-DU, and inter-gNodeB communication happens via the Xn interface.

- **CU:** As the name suggests, the Central Unit is well suited to a centralised location and requires high processing powered data centres [27]. CU is also responsible for handling/-managing the operation of DUs and can manage multiple DU's [28]. The functions of the

CU are dependent on the functional split option selected. As the CU will handle the high layer protocols such as RRC, it generally handles the gNodeB functions such as user data transportation to the core network, Radio resource management (RRM), positioning, and session management.

- **DU:** The Distributed Unit handles the gNodeB functions that generally have large bandwidth requirement and is suitable for distributed deployment. According to the NG-RAN architecture, DU handles low-layer gNodeB functions such as RLC and MAC. DU is controlled by CU and can be connected to only a single CU at a given time [28]. Each DU is capable of handling one or more cell sites.

3.2.2 Functional split of 5G-NR

NR protocol stack responsible for performing the BBU and RRH processing are split into eight partitions, resulting in eight split options for deployment. It gives much more flexibility to the operator in deciding which functions can be deployed in a distributed fashion closer to the UE and which functions to deploy at a centralised location to avail processing benefits. For example, CU and DU can be deployed at the exact edge location when applications have low latency requirements. Whereas for a scenario where latency is not critical, multiple DUs at different edge locations can connect to a CU deployed at a more centralised location. MNOs can build 5G in whichever architecture they believe would best serve their customers. It is anticipated that MNOs will use a variety of functional splits in order to target various markets and geographical regions. Figure 3.6 represents the functional split of NR-protocol stack

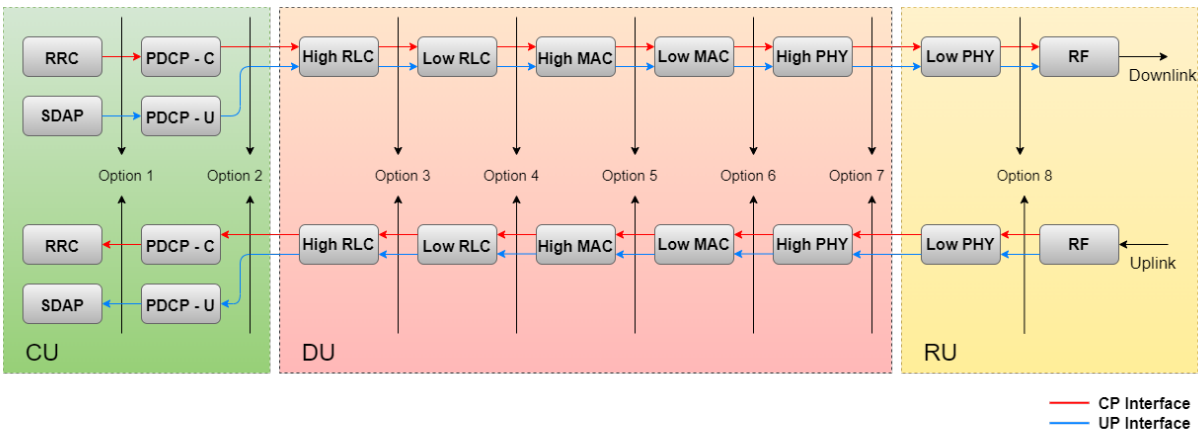


Figure 3.6: Eight possible splits, that divide the functionalities of 5G-NR between CU, DU and RU

By employing the functional split approach, the NR protocol functions can be split according to the MNOs' deployment scenario. Split options 1-5 are called High Layer Split (HLS), and split options 6-8 are called Low Layer Split (LLS). Hence when the two functional splits, HLS and LLS, are employed together, the functions of gNodeB are divided among three different units, namely gNodeB-CU, gNodeB-DU and Radio Unit (RU). 3GPP does not yet standardise RU [29]. We choose Option 2 (see also Section 3.2.4) for HLS split and Option 7 for LLS from the functional split approach to support our research study. The grouping of the functions using the colour coding in Figure 3.6 represents our choice of HLS and LLS.

- **Radio Unit (RU):**

As mentioned above, LLS splits the PHY functions into High PHY and Low PHY. RU in 5GS comprises of Low PHY block and RF unit. A fronthaul interface connects Low PHY, and High PHY blocks

The most common interface between gNodeB-DU and RU is the CPRI (Low PHY/RF split option) employed in C-RAN deployment for LTE. Another interface, enhanced Common Public Radio Interface (eCPRI) used for High PHY/Low PHY split, Option 7.2, is expected to dramatically reduce the fronthaul capacity requirements compared to CPRI and support the 5GS requirements.

3.2.3 Protocol stack of 5G-NR

The 5G-NR protocol stack is developed by keeping the LTE protocol stack as the reference. Due to the CUPS approach, the control and user plane each have a different protocol stack. Apart from a similar protocol stack to LTE, a new protocol layer called SDAP has been introduced in the user plane protocol stack of 5G-NR.

3.2.3.1 User plane protocols

The user-plane protocols support the PDU session service by delivering user data across the radio access layer. Figure 3.7 depicts the 5G-NR user plane protocol stack.

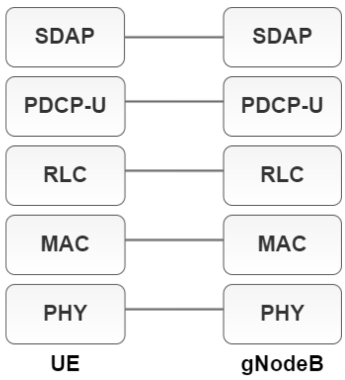


Figure 3.7: User plane protocol stack

- **NR-Physical (NR-PHY) layer :**

  The NR Physical layer manages radio functionalities such as beamforming, multiplexing, and multiple antenna mapping. The NR PHY is responsible for transmitting the information bits to the higher layer via transport channels by mapping the transport channels to physical channels [30]. Channel coding and Random Access procedures also take place at this stage.

- **NR-Medium Access Control (NR-MAC) layer:**

  This layer's fundamental functionalities are managing multiplexing and demultiplexing, hybrid- Automatic Repeat Request (ARQ) transmissions, and allocating the priority among the UEs using dynamic scheduling. In 5G-NR, the MAC layer functionality is extended; for example, the MAC PDU header structure is upgraded to satisfy the low latency requirements. Unlike LTE-MAC, NR-MAC PDU comprises one or more subheaders, each carrying information necessary to decode the respective MAC subPDUs. As the uplink grant defines the Transport Block (TB) size, a UE cannot build an LTE-MAC PDU until it gets an uplink grant. SubPDUs in NR-MAC can be pre-assembled, and when the grant is ready, NR-MAC can add the appropriate padding and concatenate subPDUs. For control signalling, the necessary MAC information is placed at the beginning of the MAC PDU for DL and the end of the MAC PDU for the UL. As explained earlier, the Physical layer transmits the information bits to the MAC layer via transport channels (for example, Radio interfaces such as CPRI/eCPRI). Next, the MAC layer sends the frames to Radio-Link Control (RLC) layer through different logical channels, each distinguished by the type of frames sent [30].

- **NR-Radio-Link Control (NR-RLC) layer:**

  The NR-RLC layer, primarily based on the LTE-RLC, is in charge of data transport. Based on segment offset and sequence numbers, the RLC performs segmentation and re-segmentation. Also, it performs error correction using ARQ. NR-RLC can reduce latency in two ways. Firstly, no Service Data Units (SDUs) reordering in NR-RLC reduces the latency and transfers the frames to the following layer without waiting for previously lost packets to be re-transmitted. Secondly, unlike LTE-RLC, concatenation of RLC SDUs is not supported, and multiple RLC-PDUs can be transferred to the MAC layer per Transmission Time Interval (TTI). Hence, to reduce latency, assembling the RLC PDUs can be done before receiving the scheduling decision from MAC layer [30].

- **NR-Packet Data Convergence Protocol (NR-PDCP) layer:**

  The PDCP layer manages user data transport, re transmission management, and reordering. Header compression, decompression, and integrity protection are all handled by the PDCP layer. In NR, the reordering feature is relocated from RLC to PDCP. The NR introduces a novel PDCP function called duplication, which re sends the user frames to the gNodeB numerous times, ensuring that at least one copy is accurately received. Suppose more than one copy of the same PDU is received, then PDCP eliminates the duplicates. URLLC's high-reliability criteria are satisfied by transmission over several cells using the duplication function [30].

- **NR-Service Data Adaptation Protocol (NR-SDAP) layer:**

  As the QoS requirements of various applications and use cases have started to vary on a large scale, handling those requirements in LTE was difficult. Thus, the SDAP layer has been introduced to handle the QoS requirements while connected to 5GC. QoS flows, and data radio bearers make up each PDU session. According to their needs, IP packets are connected with QoS flows and have a unique identification called QoS Flow Identifier (QFI). Therefore, SDAP maps a QoS flow from the 5GC to a data radio bearer while also identifying the QFI [30].

- **Layer 2 data flow:** In the downlink, IP packets received from the application are processed by the Layer 2 protocols. The packets received by any layer are referred to as SDUs; that particular layer processes the SDUs, adds the header, and outputs them as PDUs of that layer. The IP Packets are first processed by the SDAP layer and add an SDAP header to it, making it SDAP PDU. The PDCP layer receives the SDAP PDU, processes it, and adds a PDCP header, making it PDCP PDUs. Similarly, RLC and MAC PDUs are generated. Uplink works in a similar way. Figure 3.8 represents a Layer 2 Data Flow example, where a PDCP PDU is segmented into two smaller RLC SDUs. Furthermore, two RLC PDUs are concatenated by MAC to generate a TB and are forwarded to the Phy layer with Cyclic Redundancy Check (CRC) for further processing. Out of the two RLC PDUs concatenated by MAC, one corresponds to the IP packet (n) and a segment of an IP packet (m). Figure 3.8 represents the flow of data through the L2 protocols.

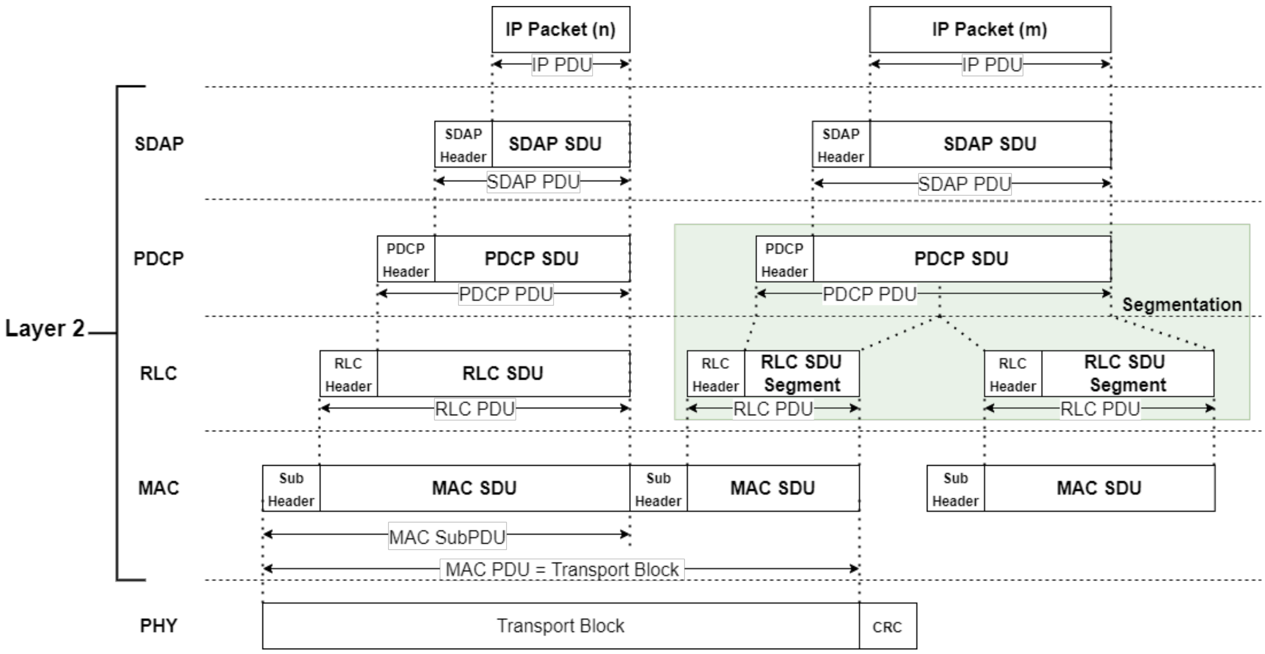


Figure 3.8: Layer 2 data flow

3.2.3.2 Control plane protocols

The central signalling node of the 5G-NR core is the AMF. The only control plane protocol that is not present in the user plane is the RRC protocol. Signalling Radio Bearers (SRB) are used in 5G-NR to transmit messages between the RRC and the NAS layer. Figure 3.9 depicts the 5G-NR control plane protocol stack.

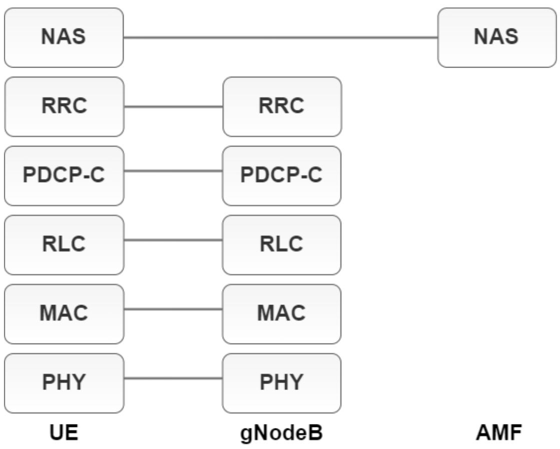


Figure 3.9: Control plane protocol stack

- **NR-Radio Resource Control (NR-RRC Layer):**

  Access Stratum (AS) and Non-Access stratum (NAS) messages are collectively called system information-related messages and are broadcast by RRC. The RRC handles the security, mobility, and QoS management functions between a UE and NG-RAN. The RRC layer alone handles the control plane signalling linking UE and gNodeB. Messages for the establishment, maintenance, and release of the RRC connection are transferred between the UE and gNodeB by RRC protocol. With the assistance of RRC, the radio link failures are detected and recovered. Like LTE, IDLE and CONNECTED are two RRC states that 5G-NR supports. Additionally, to reduce the energy usage and support extended battery life of IoT and mMTC-devices, a transitional state between IDLE and CONNECTED called INACTIVE state is introduced in 5G-NR [30].

3.2.4 The architecture of gNodeB-CU-CP/gNodeB-CU-UP split

3GPP has prominently discussed Split option 2 of NR in Release 15. This split option deploys RRC, SDAP and PDCP protocols at CU and deploys bandwidth-hungry and latency-sensitive RLC, MAC, and PHY layer protocols at DU. The CUPS approach for EPC, introduced in 3GPP Release-14, is enhanced in 3GPP Release-15 by extending it to the gNodeB. Split option 2.2 adapts the CUPS approach and the CU can be split into gNodeB-Centralised Unit-Control Plane (gNodeB-CU-CP) and gNodeB-Centralised Unit-User Plane (gNodeB-CU-UP). It is achieved by placing RRC and PDCP for CP stack (PDCP-C) functions in gNodeB-CU-CP and placing SDAP, and PDCP for UP stack (PDCP-U) functions in gNodeB-CU-UP [16]. This split gives rise to a new interface named E1 for gNodeB-CU-CP and gNodeB-CU-UP to communicate with each other. Furthermore, the F1 interface is split into its control plane part (F1-C interface) and user plane part (F1-U interface). F1-C interface connects gNodeB-CU-CP and gNodeB-DU. F1-U interface connects gNodeB-CU-UP and gNodeB-DU. Figure 3.10 depicts the block diagram of NG-RAN with CUPS approach.

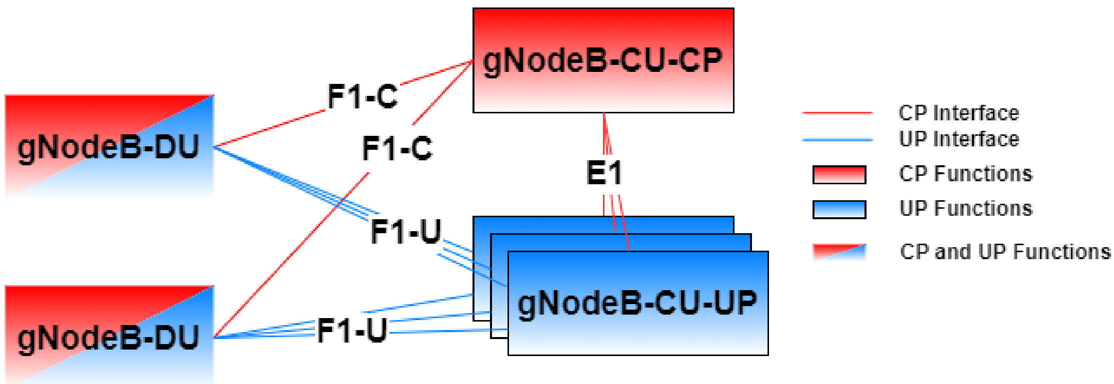


Figure 3.10: Block diagram of gNodeB-CU-CP and gNodeB-CU-UP split, according to this architecture, a single gNodeB-CU-CP, several gNodeB-CU-UP's, and gNodeB-DU's constitute a gNodeB .

According to [31], the gNodeB-CU-CP creates and supervises the connection between the gNodeB-DU and gNodeB-CU-UP.

- A gNodeB-DU can be connected to multiple gNodeB-CU-UP's if they are connected to the same gNodeB-CU-CP [28].
- A gNodeB-CU-UP can support multiple gNodeB-DU's if they are connected to the same gNodeB-CU-CP [28].

3.2.4.1 E1 Interface

gNodeB-CU-CP and gNodeB-CU-UP are logically connected using the E1 point-to-point interface, which helps them communicate by transferring signalling information.

The E1 interface handles the transfer of many interface management procedures, such as the *gNB-CU-UP E1 Setup procedure, gNB-CU-CP E1 Setup procedure, gNB-CU-UP Configuration Update procedure and gNB-CU-UP Status Indication procedure*. Either gNodeB-CU-CP or gNodeB-CU-UP can initiate the E1 Setup procedure[1]. The gNodeB-CU-UP can signal its capacity information to the gNodeB-CU-CP using the *E1 setup procedure and gNB-CU-UP Configuration Update procedure* over the E1 interface. The overloaded or non-overloaded status of gNodeB-CU-UP can be informed to the gNodeB-CU-CP using the E1 gNB-CU-UP Status Indication function over the E1 interface.

The E1 interface is dedicated to exchanging control information between the endpoints and cannot be used for forwarding user data. It can be used for exchanging UE-associated data such as UE IP address and Bearer context management function[2], executing UE trace functions and exchanging Non-UE-associated information such as load management procedures and measurement results transfer procedures of gNodeB-CU-UP. The E1 interface handles the exchange of the so-called NG-U DL GTP TEID and F1-U UL GTP TEID for the NG-U tunnel and F1-U tunnel, respectively (see also Section 3.5).

3.2.4.2 F1 Interface

The F1 interface supports the CUPS in RAN. When Option 2.2 is used, the F1 interface is split into its control plane part (F1-C) and user plane part (F1-U).

- **F1-C:**

  gNodeB-DU and gNodeB-CU-CP are logically connected using the F1-C point-to-point interface. The F1-C interface handles the transfer of many interface management procedures, such as the *F1 Setup procedure, gNB-CU Configuration Update procedure, gNB-DU Configuration Update procedure and gNB-DU Status Indication procedure*. RRC Message Transfer procedures such as Initial UL RRC message transfer procedure and UL/DL RRC message transfer procedure are handled by the F1-C interface. The other control plane signalling transferred by the F1-C interface includes system information, paging, UE tracing, and positioning procedures. The load management procedures handled by the F1-C interface transfer the Resource Status Reporting Initiation procedure and Resource Status Reporting procedure.

- **F1-U:**

  gNodeB-DU and gNodeB-CU-UP are logically connected using the F1-U point-to-point interface. The F1-U interface handles only the user plane and does not handle any control plane procedures.

[1] Using the E1 Setup procedure, the application-level data needed for the gNodeB-CU-UP and the gNodeB-CU-CP are exchanged to inter-operate on the E1 interface correctly.

[2] Bearer Context information of a UE is stored in gNodeB-CU-UP and used for communicating about that particular UE over the E1 interface. It contains the information necessary to provide continuous user plane services to the UE.

3.2.5 5GS transport network

The ambition of linking all devices that benefit from a connection and supporting a wide range of services is handled by the next-generation 5G mobile infrastructure. In order to combine the multiple technological domains of radio, transport, and cloud, the 5G transport networks must provide the requisite capacity, latency, and flexibility [32]. Figure 3.11 represents the 5GS transport architecture.

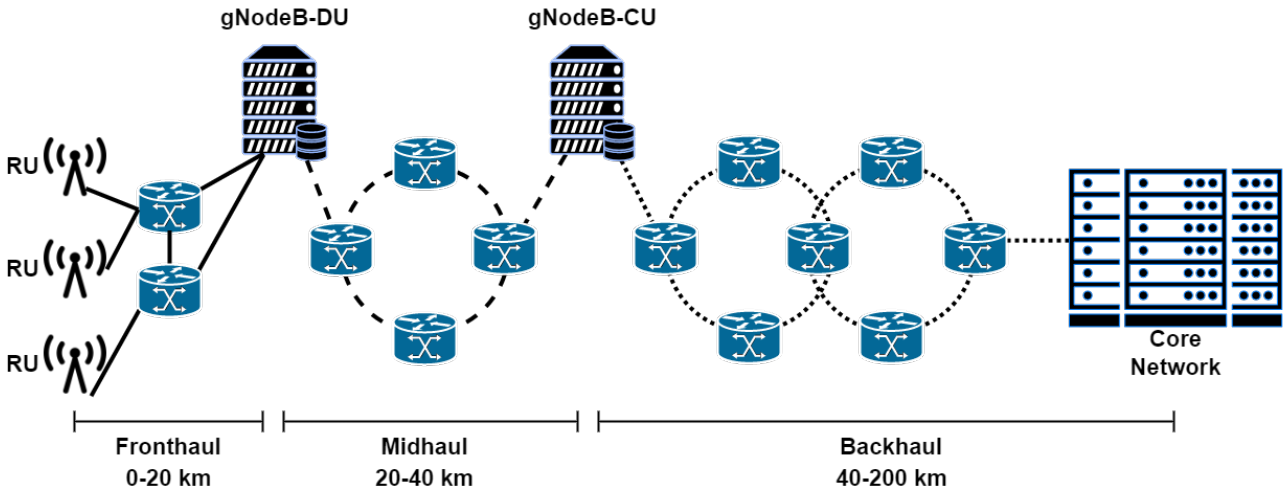


Figure 3.11: 5GS transport architecture

As we employ both HLS and LLS in this research study, the gNodeB is split into three entities: RU, gNodeB-DU and gNodeB-CU. The 5G transport network consists of the Fronthaul, Midhaul, and Backhaul to connect RU, gNodeB-DU, gNodeB-CU and the 5GC. The Fronthaul transport network connects and transports the information between RU and DU and is deployed as a star or ring topology. A ring topology supports the Midhaul transport network between DU and CU. Further, the Backhaul connects the CU and Core Network, supported by ring or mesh topology. The transport network requirements of the 5GS is tabulated in Table 3.1.

	Fronthaul	Midhaul	Backhaul
Distance	0-20 km	20-40 km	40-200 km
Link Capacity	100G+ links	100G+ links	400G+ links
Latency	50-200 µs	1-2 ms	<10 ms

Table 3.1: Transport network requirements [29]

The RAN and Core NFs are to be deployed on computing machines at cloud locations. These cloud locations can be located at different positions and are divided into edge cloud, regional cloud and central cloud based on their distance from the cell site. Table 3.2 mentions the cloud locations and their approximate distance form the cell site.

	Edge Cloud	Regional Cloud	Central Cloud
Distance from UE	15 km	55 km	200 km

Table 3.2: Cloud options along with approximate distance from the cell site

3.3 O-RAN

As the 5GS aims to support a wide range of applications and use cases, the MNOs will need flexible, scalable and cost-effective RAN deployment options to satisfy per-UE requirements. Traditional RAN deployments are rigid as they depend on proprietary hardware and vendor-defined communication interfaces. During any upgrade in the network, scaling from small to large, the physical hardware must be replaced over the entire wireless network, which consumes time, money, and human resources. Because of the proprietary nature of the hardware and interfaces, the physical hardware has to be replaced by the same vendor who originally supplied them so that the hardware remains compatible with the rest of the network [33].

As previously discussed in Section 1.1, due to the low radio coverage area, the 5GS requires many cell sites. Using proprietary hardware for all these large instalments will increase the CAPEX. Also, with increasing requirements for applications and the rise in the number of expected antenna deployments in the 5GS, network complexity is bound to be severe and human intervention is no longer optimal for operating RAN [34]. These drawbacks can be overcome, and the scope for improvement can be increased by virtualising RAN, promoting self-driven RAN and using standardised-open interfaces; O-RAN is a one-stop solution for all these requirements.

O-RAN Alliance is a coalition of institutions and industries in the RAN industry, working towards the development of open standard, keeping open RAN and intelligent RAN as two main goals [35]. O-RAN opens up immense opportunities for third-party entities and increases the scope for Commercial off-the-shelf (COTS) hardware resulting in higher flexibility and cost reduction [36]. To promote self-driven RAN, O-RAN intends to utilise artificial intelligence and machine learning (AI/ML) techniques. Moreover, by making all the O-RAN components run on the standardised APIs, the MNOs will have more extensive deployment options for implementing the same functionalities using alternative components from multiple vendors, making it a multi-vendor ecosystem.

As explained earlier in Section 1.2.2, the O-RAN approach is built upon NFV technology which increases the flexibility of RAN deployments by decoupling the software and hardware elements. In this work, the O-RAN approach is employed to realise the CUPS approach in the gNodeB and the virtualisation of RAN. Furthermore, O-RAN aids this study by realising our requirement for dynamic deployment of the so-called O-CU-UP and O-DU as VNFs.

3.3.1 O-RAN architecture

One of the essential principles of O-RAN architecture is to remain consistent with the 3GPP architecture and interface specifications [37]. The O-RAN architecture can be divided into the management and radio sides. Figure 3.12 illustrates the O-RAN architecture.

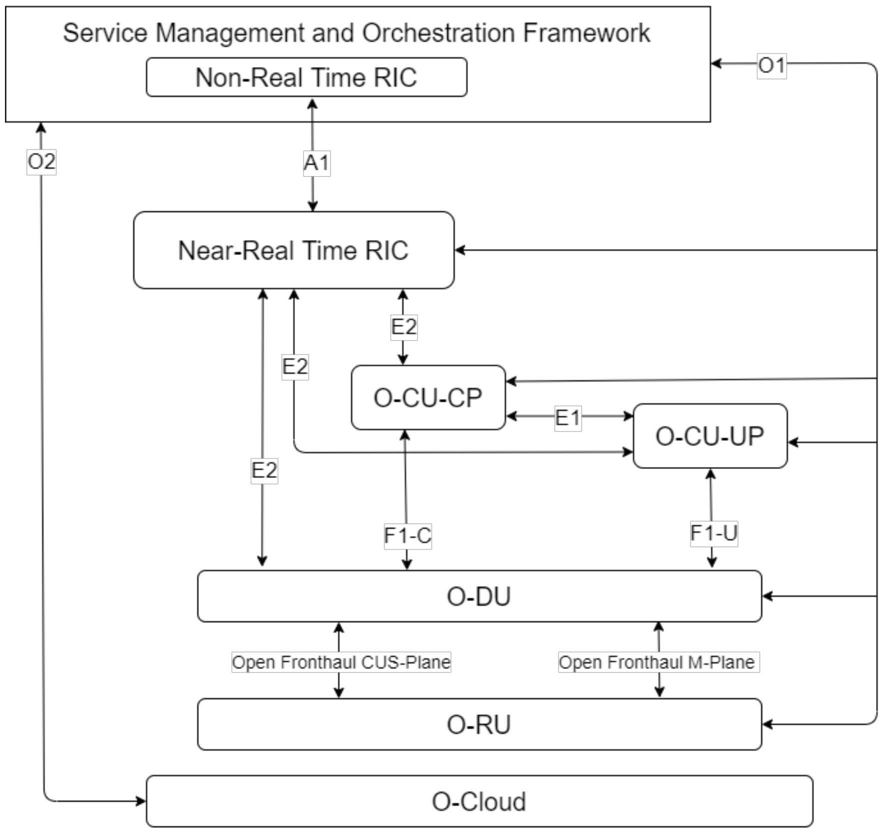


Figure 3.12: O-RAN architecture

Service Management and Orchestration (SMO) and Non-Real Time RAN Intelligent Controller (Non-RT RIC) are part of the management side. Near-Real Time RAN Intelligent Controller (near-RT RIC), O-RAN-Centralised Unit-Control Plane (O-CU-CP), O-RAN-Centralised Unit-User Plane (O-CU-UP), O-RAN-Distributed Unit (O-DU), and O-RAN Radio Unit (O-RU) constitute the radio side. O-CU-CP, O-CU-UP, and O-DU are logical O-RAN nodes that terminate at the E2 interface and are addressed as E2 nodes. The functionalities of O-CU and O-DU remain in line with 3GPP standardisation. O-RU resembles RRH. Instead of the proprietary eCPRI fronthaul interface, the Open Fronthaul interface (also called the LLS interface) is employed to connect the O-DU and O-RU units (the connection between the High Phy and the Low Phy blocks). [37].

Every O-RAN entity is connected to SMO over the O1 interface for Fault, Configuration, Accounting, Performance, Security (FCAPS) support. The presence of the Non-RT RIC in the SMO framework demonstrates that the Non-RT RIC has access to the capabilities of the SMO framework, including the ability to exert influence over the information transmitted through the O1 interface. It also uses the data collected at SMO and the "O-RAN nodes provisioning" function of SMO to decide which artificial intelligence and machine learning (AI/ML) model can be utilised to achieve RAN optimisation. Non-RT RIC connects and forwards information to near-RT RIC over the A1 interface to support RAN optimisation. The E2 interface connects the near-RT RIC to one or more E2 nodes manufactured by different vendors in O-RAN architecture. The RIC optimises RAN through control loops such as Request/Response procedures, feedback loops and control actions. Further details about control loops is provided in Appendix C.1

O-Cloud is connected to SMO over the O2 interface to manage and maintain its workflow orchestration. O-Cloud is a cloud computing platform that provides physical infrastructure nodes that meet O-RAN requirements. It hosts the relevant functions of the O-RAN architecture (such as the near-RT RIC, the O-DU and the control and user plane part of O-CU), the software platform (such as Virtual Machine Monitor, Operating Systems), and the management and orchestration functions [37].

3.3.2 RAN Intelligent Controller (RIC)

As the 5GS evolves, there are developments in RAN specifications such as functional split, CUPS approach, and network slicing, making the 5G RAN architecture increasingly complicated. MNOs are also reconsidering how they control, run, and manage the RAN to reach its full potential. A feasible approach to managing them is to automate network administration and operations by allowing the network to learn and make decisions independently. Moreover, the new developments necessitate sophisticated RAN virtualisation in conjunction with SDN. Hence, RAN Intelligent Controller (RIC) is required to optimise, configure, and control the RAN operation efficiently.

The RIC is deployed as a VNF or a Cloud-native Network Function (CNF), making it a scalable platform for third-party applications. NFV provides the Software operating environment and the cloud infrastructure required for the RIC. Apps are installed on the software platform employing SDN to orchestrate and control the network to realise network automation. On the SDN's northbound interfaces, applications are set up to handle gNodeB functions like mobility management, admission control, and interference management. The network regulations of apps can be enforced on the gNodeB nodes through a southbound interface [38].

- **Working methodology of RIC**

  To increase RRM skills, the control features of RIC use analytical and data-driven methodologies, including sophisticated AI/ML technologies. At the L1/L2/L3 protocols stack of gNodeB, a vast amount of data, statistics, and failure information can be gathered and utilised to build models. Models are trained using AI/ML to enable intelligent administration and control of the RAN. Models such as massive Multiple Input Multiple Output (MIMO) patterns, UE mobility patterns, and expected QoS patterns could be trained or obtained. When combined with additional network-wide context, the trained models may be used to control fine-grained network operations, such as RRM, interference management and mobility management. MNOs can design and own the core AI/ML algorithms, with certain portions being proprietary. Every individual operator can customise the RAN's behaviour by implementing policies and models tailored to their specific needs and objectives.

There are two types of RIC namely, Non-RT RIC and Near-RT RIC. An introduction to Non-RT RIC is provided in Appendix C.2. And near-RT RIC is explained in detail below.

3.3.2.1 Near-Real Time RIC

As the name suggests, the near-RT RIC is a software platform that works near-real-time with timescales of $\geq$10 ms, but $<$1 s. It hosts 3rd party applications called xApps. The xApps control the E2 nodes (O-CU-CP, O-CU-UP, and O-DU) through the E2 interface southbound of near-RT RIC [38]. The E2 interface assists the xApps in gathering near-real-time data from the E2 nodes on a cell basis or per UE basis and provides value-added services.

Open RAN components (E2 nodes) and resources under its control can be controlled and optimised in near real-time by near-RT RIC.

The Non-RT RIC connected to the northbound of near-RT RIC via the A1 interface provides policy-based guidance and necessary information to the near-RT RIC, which helps the near-RT RIC control, override, and supervise the E2 nodes for any E2 service model (E2SM)[3] described function. With the data collected from all the E2 nodes and input from the Non-RT RIC, the RAN is optimised and efficiently manages available resources.

Based on the E2 node's capability, the near-RT RIC controls the RRM functional allocation between the near-RT RIC and the E2 node [39]. If near-RT RIC fails, the E2 node will continue to deliver services, but the services provided by the Near-RT RIC may be unavailable.

This thesis work employs near-RT RIC in our proposed approach as it can supervise E2 nodes (we are interested in controlling O-CU-CP, also see Section 4.2), provides fine-grained services per UE level and operates at a timescale ≥10 ms, but <1 s.

- **xApps:**

  It is any 3rd party app that is developed specifically to work on the near-RT RIC. xAPPS communicate with near-RT RIC over the near-RT RIC APIs. The E2 interface facilitates a direct connection between the RAN function (E2 nodes) and xApp. Applications of this type embody multiple micro services and will identify what data it collects and what data it gives at the on boarding time. Cloud infrastructure to the xApps is provided by the software operating platform of near-RT RIC.

- **Radio-Network Information Base (RNIB):**

  A near-RT RIC database stores the latest RAN information, history of changes in the network state, and information from E2 nodes. It also collects information related to cells, bearers, UE's and Flows. Any xAPP on the near-RT RIC can request this information and receive it as a service.

3.3.2.2 E2 Interface

An open and point-to-point interface called the E2 interface has to be established using the E2 Setup procedure to connect the near-RT RIC to an E2 node. The E2 node is responsible for initiating this operation. The E2 interface handles the control signalling procedures between the near-RT RIC and any E2 nodes. An E2 node can inform the near-RT RIC about all the functions and procedures it can perform via the E2 interface. E2 interface allows the near-RT RIC to take control of specific E2 node procedures and functions. A set of policies to be used when specific events occur are sent to an E2 node using the E2 interface. Using the E2 interface, the near-RT RIC can interrupt the local processing of an E2 node, suspend an ongoing RRM procedure and directs it to transfer the pertinent information to the near-RT RIC for further processing. The near-RT RIC and an E2 node must have a dedicated E2 interface connection. It is possible to differentiate each E2 node and provide each Node with its own set of capabilities using this dedicated connection. During the event of an E2 interface failure, the E2 node will operate independently of the near-RT RIC.

The E2 Application Protocol (E2AP) [40] handles procedures that define how the communication happens between the E2 node and near-RT RIC and supports certain near-RT RIC services, namely Report, Insert, Control, and Policy which communicate the occurrence of an event, measurements and provides control of the E2 nodes to the near-RT RIC via the E2 interface. A detailed explanation of the four services is provided in Appendix C.3 Different combinations of these services form an E2SM.

[3] E2SM is a combination of near-RT RIC services offered by E2 node

The below-mentioned E2AP procedures are used to implement the RIC Services.

- E2AP RIC Subscription procedure: near-RT RIC uses the RIC Subscription procedure to request an E2 subscription on an E2 node. It instructs the E2 node to send an event trigger when a specific event occurs. The subscription is active until the subscription deletion procedure takes place.

- E2AP RIC Indication procedure: E2 nodes use the RIC Indication procedure for sending REPORT/INSERT service messages associated with detecting event triggers of a previously accepted subscription to near-RT RIC.

- E2AP RIC Control procedure: The near-RT RIC uses the Control procedure to resume or begin a particular functionality in an E2 node.

3.4 GPRS tunnelling protocol for the user plane (GTP-U)

The GPRS tunnelling protocol (GTP) was initially developed for packet tunnelling within GPRS. Later changes and enhancements have been made to suit the requirements of the new technologies. The GTP has three versions, GPRS Tunnelling Protocol for control plane (GTP-C), GPRS Tunnelling Protocol for user plane (GTP-U) [41] and GTP prime (GTP') to get tunnelling charging information from PCF. In 5GS, the main focus is given to GTP-U. The GTP-U tunnels are uni-directional tunnels used to tunnel the T-PDUs[4] between the sending and receiving GTP entities. Two uni-directional GTP-U tunnels are used between two entities to form a bi-directional GTP-U tunnel. Each GTP-U tunnel has a Tunnel Endpoint Identifier (TEID) allocated by the receiving GTP entity. The TEID assists the GTP-U entity in identifying the tunnel belonging to a particular UE in which the T-PDUs must be transferred. The TEID is exchanged between the sending entity and the receiving entity using control plane protocols such as NG Application Protocol (NGAP) [42], E1 Application Protocol (E1AP) [43], F1 Application Protocol (F1AP) [44], and Packet Forwarding Control Protocol (PFCP) [45] (see also Section 4.1.1.1)

From Figure 3.13 we see that the GTP-U employs a UDP/IP tunnelling mechanism to transport G-PDUs[5]. The G-PDUs are transported over an IP network as the sending and receiving GTP entities are placed over different geographical locations. UDP/IP/Eth stack constitutes the end-to-end transport protocols between the two entities. The UDP header consists of source and destination port numbers. The destination port number 2152 is pre-defined for GTP-U messages. IP header consists of source and destination IP addresses corresponding to sending and receiving GTP entities.

[4] T-PDU is the user data (commonly IPv4 or IPv6 packets), which is tunnelled inside the GTP-U tunnels
[5] G-PDU are the GTP-U messages, which is T-PDU along with the GTP-U header

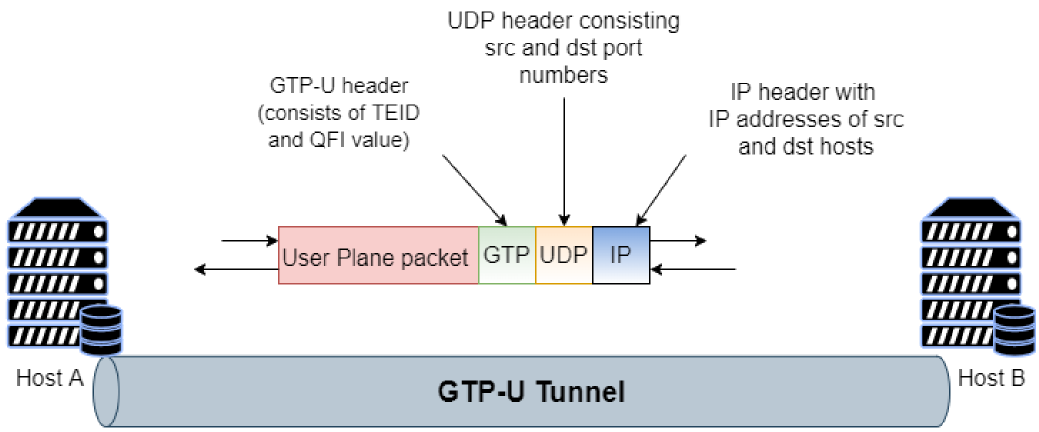


Figure 3.13: GTP-U tunnel between two hosts along with GTP-U message in UL/DL

The GTP-U TEID and IP address of the receiving GTP-U entity together forms tunnel info. The tunnel info is helpful for service continuity during mobility. When a UE is mobile and moves from gNodeB_1 to gNodeB_2, the SMF provides the UPF about tunnel info of gNodeB_2 of that UE, and hence the UPF can continue forwarding DL traffic to the new tunnel info. On the other side, the SMF provides the existing tunnel information of the serving UPF to the gNodeB_2 to facilitate UL data forwarding.

3.5 PDU session

PDU session provides a logical user plane connectivity for a UE towards the Data Network (DN), which is used for exchanging PDUs between the UE and the DN. UE provides a PDU session ID, unique per UE, which can be used to identify one of UE's PDU sessions. Setting up a default QoS Flow between UE and UPF via gNodeB is the primary goal of the PDU session establishment procedure. An introduction to 5GS Quality of Service (QoS) is provided in Appendix A. A PDU session supports one or more QoS Flows. UE uses the default QoS Flow to exchange data or signal other DN services, such as a voice server, to set up additional QoS Flows for handling more demanding traffic. Each PDU session supports a single session type (IPv4, IPv6, IPv4v6, ethernet, unstructured), and in this study, we have chosen IPv4 packets as it is mainstream and used for several applications.

3.5.1 PDU session in CP/UP split RAN architecture

In each PDU session, the PDUs are tunnelled between the UPF and the UE using two types of tunnels. Firstly, a bi-directional tunnel called Data Radio Bearer (DRB) connects the UE and the gNodeB. Secondly, the GTP-U tunnel, where two unidirectional tunnels form a bi-directional tunnel. Figure 3.14 represents the PDU session in CP/UP split RAN architecture. When CUPS architecture is employed in RAN, the GTP-U messages are communicated between UPF and gNodeB-CU-UP over the N3 interface and between gNodeB-DU and gNodeB-CU-UP over the F1-U interface.

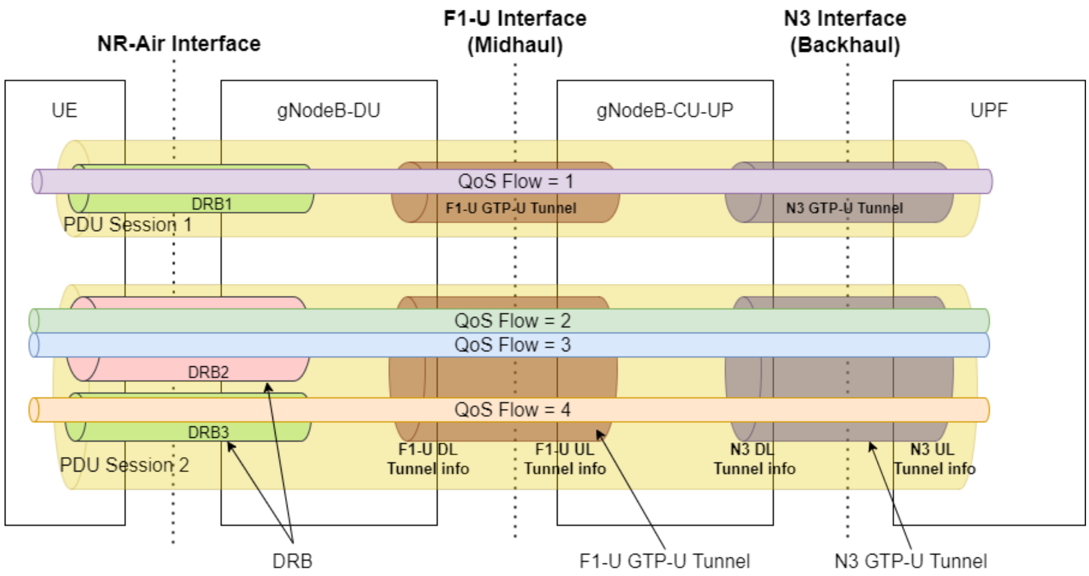


Figure 3.14: PDU session involving N3 GTP-U tunnel connecting UPF and gNodeB-CU-UP, F1-U GTP-U tunnel connecting gNodeB-CU-UP and gNodeB-DU and DRBs connecting gNodeB-DU and UE

A pair of unidirectional GTP-U tunnels used to connect gNodeB-DU to gNodeB-CU-UP for UL/DL connection is called F1-U tunnel and elaborated as follows:

- The UL tunnel which connects towards the gNodeB-CU-UP is called the F1-U UL tunnel and is used for forwarding UE's UL data from gNodeB-DU to gNodeB-CU-UP. It is identified by F1-U UL tunnel info which consists of F1-U UL GTP-U TEID and IP address of gNodeB-CU-UP

- The DL tunnel connecting towards gNodeB-DU is called the F1-U DL tunnel and is used for forwarding UE's DL data from gNodeB-CU-UP to gNodeB-DU. It is identified by F1-U DL tunnel info which includes F1-U DL GTP-U TEID and IP address of gNodeB-DU

Furthermore, another pair of unidirectional GTP-U tunnels that connects gNodeB-CU-UP to UPF for UL/DL connection is called N3 tunnel and elaborated as follows:

- The UL tunnel which connects towards the UPF is called the N3 UL tunnel (also called CN tunnel) and is used for forwarding UE's UL data from gNodeB-CU-UP to UPF. It is identified by N3 UL tunnel info which consists of N3 UL GTP-U TEID and IP address of UPF.

- The DL tunnel connecting towards gNodeB-CU-UP is called the N3 DL tunnel (also called as AN tunnel), and is used for forwarding UE's DL data from UPF to gNodeB-CU-UP. It is identified by N3 DL tunnel info which encloses N3 DL GTP-U TEID and IP address of gNodeB-CU-UP

Chapter 4

Reference architecture and new architecture

In this chapter, the reference architecture and the proposed architecture for 5G user plane are presented. Section 4.1 gives a detailed explanation of the 3GPP architecture, used as reference architecture in this study. In Section 4.2 a proposal for a new user plane architecture is presented. It consists of in-depth explanation of the proposed architecture and also mentions its advantages compared to the reference architecture.

4.1 3GPP reference architecture

In this section, first, the 3GPP reference architecture is introduced. Secondly, the protocol stacks of a PDU session in the reference architecture are explained. It is followed by the potential enhancement opportunity and finally concludes with the signalling call flow diagram of the PDU session establishment procedure.

With the architectural advancements such as CUPS, SBA and gNodeB-DU/gNodeB-CU disaggregated RAN, the 5GS can be deployed in numerous ways, and its NFs can be deployed at different locations with the support of dedicated interfaces introduced by 3GPP. In 5GS, the choice of architecture and location for deploying its NFs is up to the MNOs. MNOs can select the deployment of their choice that they find suitable for their applications. However, Figure 4.1 shows the architecture of a deployment recommended by 3GPP; this architecture is the most widespread choice for option 2.2 splits. gNodeB-CU being deployed more towards the centralised location has the advantage of handling multiple DUs, statistical multiplexing gain and control over numerous cell sites, assisting mobility. Hence we consider it a well-suited reference architecture for this study.

As shown in Figure 4.1, the reference architecture consists of the gNodeB disaggregated into RU, gNodeB-DU and gNodeB-CU. A RU is co-located with the antenna tower at the cell site, and gNodeB-DU is deployed at an edge cloud location. The gNodeB-CU is further split into its CP and UP functions. gNodeB-CU-CP and gNodeB-CU-UP functions are deployed at a regional cloud location, and the UPF is located at a more centralised location in the central cloud. All the NFs are deployed on dedicated hardware. The interfaces connecting the nodes remain in line with Section 3.1 and Section 3.2.4.

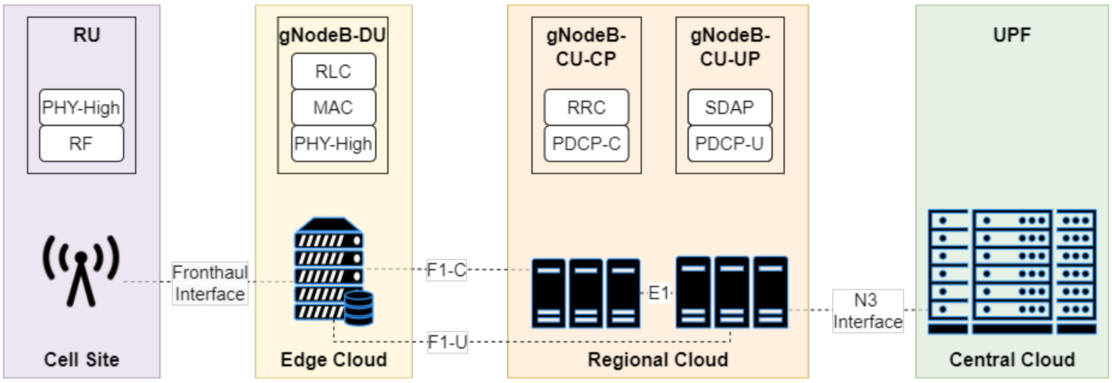


Figure 4.1: 3GPP reference architecture for 5GS, employing CUPS approach in RAN

In the above-shown architecture, even though the RAN functions such as gNodeB-CU-UP are deployed as NFs, they can be dynamically allocated only if those locations are equipped with the dedicated hardware required for hosting those RAN functions.

4.1.1 Protocol stack for the PDU session

For the well-functioning PDU session to exist, the connection between UE and the DN exists over two types of interfaces, the NR-Air interface and New Generation (NG) Interfaces such as F1, N2, N3, N4 and N11. The protocols over these two interfaces are grouped into CP and UP protocols, which helps establish control plane and user plane between UE and the DN. The control plane protocols are responsible for establishing, and controlling the PDU sessions, managing the active connection between the UE and the DN and supervising the transfer of RRC encapsulated NAS messages. Further, the user plane protocols are responsible for carrying out the PDU transfer between the UE and the DN via Access stratum and packet core network [46]. RU is not considered in the protocol stacks below to keep the complexity at a minimum. The functions of RU are considered to be done at gNodeB-DU itself, eliminating fronthaul only for the explanation purpose of the protocol stacks.

4.1.1.1 Control plane protocol stacks of a PDU session

Figure 4.2 and Figure 4.3 illustrate the relations between the control plane protocol stacks responsible for establishing the PDU sessions[1]. The NFs in the both Figure 4.2 and Figure 4.3 play a very important role in establishing, modifying and controlling the PDU sessions. The F1-U tunnel info of both ends is communicated between gNodeB-DU and gNodeB-CU-UP via gNodeB-CU-CP over the F1-C and E1 interface. The N3 tunnel info of both ends is exchanged between gNodeB-CU-UP and UPF via gNodeB-CU-CP, AMF and SMF over E1, N2, N11 and N4 interface. A detailed explanation of the exchange of these messages and the procedure of PDU session establishment is provided in Section 4.1.2.

The NAS-Mobility management (NAS-MM) and NAS-Session Management (NAS-SM) components of the NAS protocol is transferred over the N1 (see Figure 3.3). NAS-MM is exchanged between UE and AMF, responsible for registration and connection management, and controls the user plane connectivity. The NAS-SM is exchanged between the UE and SMF via the AMF (AMF does not alter the content of the NAS-SM message). It is mainly used for PDU session establishment, modification and release.

[1] As these relations are not linear in nature, that is, gNodeB-CU-CP is connected to both gNodeB-CU-UP and gNodeB-DU, they are split into two figures for explanation purpose

The radio network layer (RNL)[2] and transport network layer (TNL)[3] are used in NG-RAN protocol stacks. The F1-C protocol stack comprises F1AP at the RNL, which transfers the control plane messages associated with the UE over F1 and is built upon stream control transmission protocol (SCTP) [47] over IP. While SCTP ensures signalling messages to be delivered, IP provides point-to-point (P2P) communication to exchange the application layer PDUs.

The NG-C interface maps to the N2 interface. RNL in the N2 interface is based on the NGAP to facilitate the exchange of UE-associated control plane messages. Furthermore, the TNL is based on SCTP/IP protocol stack. N11 interface connects AMF and SMF; the signalling messages between these two entities are exchanged as the SBA messages using HTTP/2 protocol.

As explained before, the communication between gNodeB-CU-CP and gNodeB-CU-UP happens over the E1 interface. The RNL of the E1 interface is based upon the E1AP, which transfers signalling information between the two logical nodes connected by E1. Furthermore, the TNL of E1 is also based on SCTP/IP. The N4 interface splits the control plane (SMF) and user plane (UPF) in the 5GC. PFCP is used over this interface, providing control to the SMF over UPF to handle packet management. The PFCP messages are exchanged over UDP/IP transport.

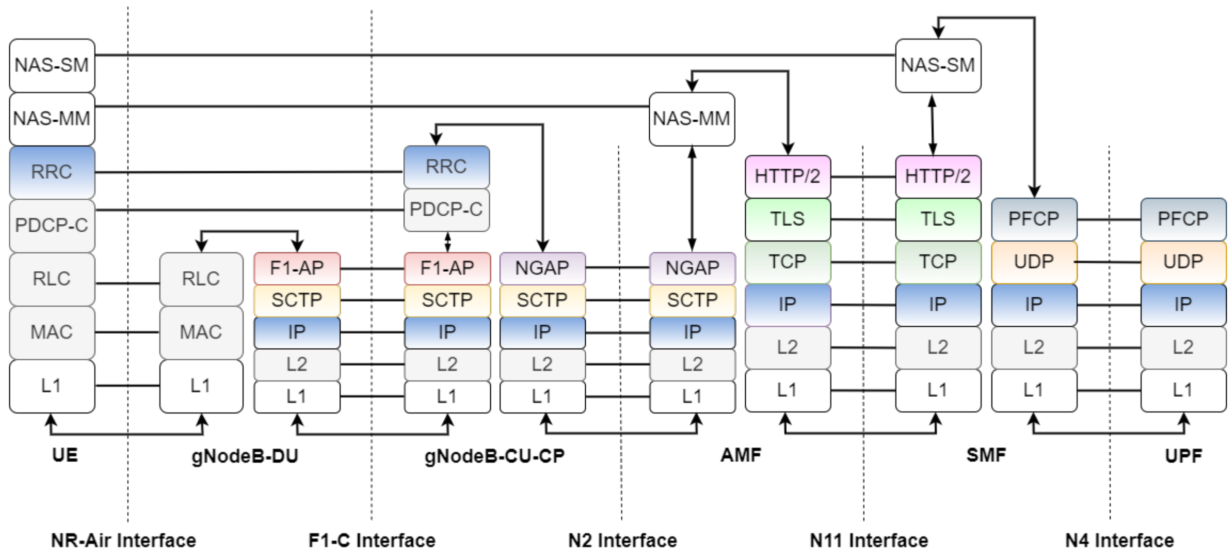


Figure 4.2: End-to-End control plane protocol stack of a PDU session from UE to UPF in the reference architecture. These are the protocols that control the PDU session signalling

[2]The protocol at RNL performs the RAN related functions of the logical node
[3]The TNL protocols are responsible for the transport of user plane and signalling information

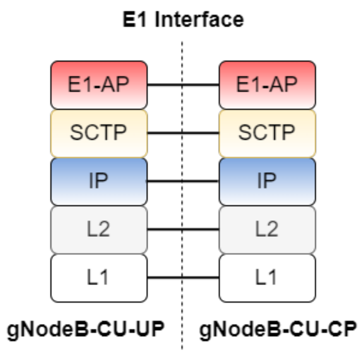


Figure 4.3: E1 interface protocol stack, part of control plane protocol stack of a PDU session connecting gNodeB-CU-CP and gNodeB-CU-UP

4.1.1.2 User plane protocol stacks of a PDU session

Figure 4.4 illustrates the user plane protocol stack responsible for transporting data PDUs between UE and the DN. The PDU layer at the UE and UPF relates to PDUs transmitted between UE and the DN during a PDU session. For explanation purposes, let us consider the downlink direction of the packet. As explained in Section 3.5.1, the PDUs are tunnelled using the GTP-U tunnelling mechanism over N3 and F1-U interfaces. The GTP-U TEID, along with the outer IP address[4] is used to identify the tunnels that match the packet requirements. First, at UPF, the PDUs are encapsulated as GTP-U messages using the GTP-U protocol. Further, the GTP-U messages are transmitted over the N3 interface using UDP/IP transport. At the receiving side of gNodeB-CU-UP, the GTP/UDP/IP/Eth headers are decapsulated and then SDAP and PDCP-U processing are performed.

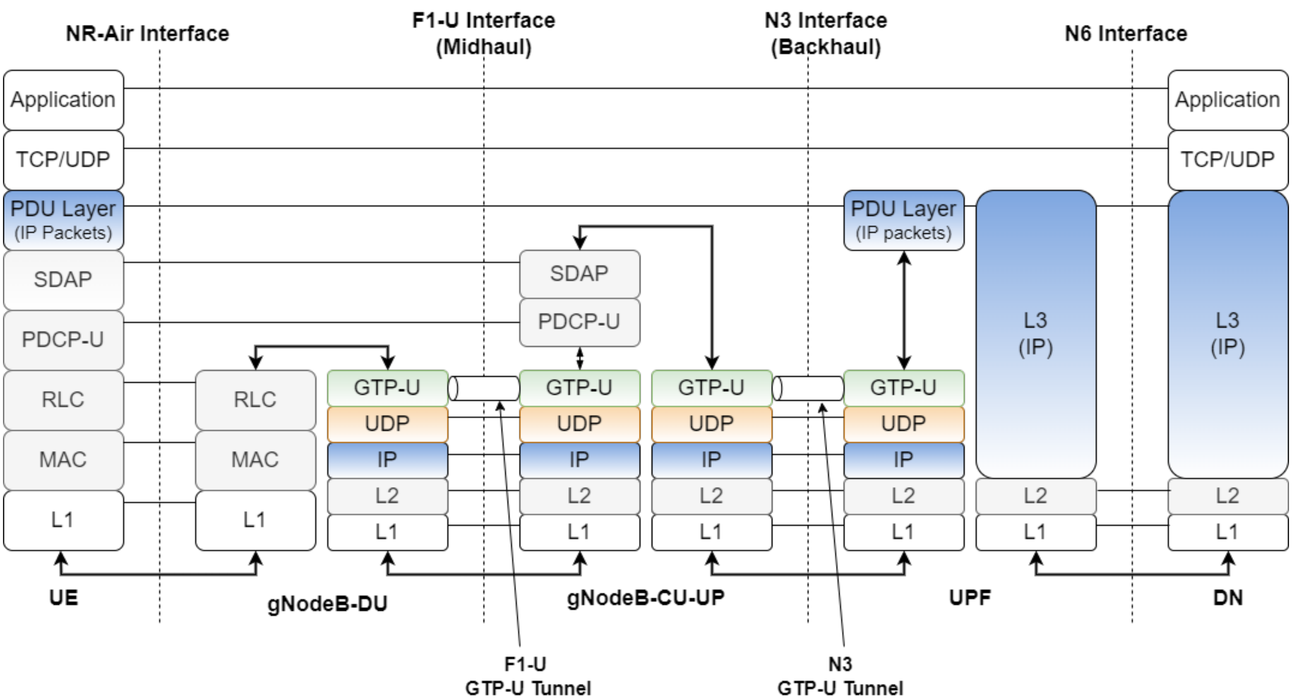


Figure 4.4: End-to-End user plane protocol stack of a PDU session from UE to DN in the reference architecture. These protocols transports user data across the access stratum, implementing the PDU session

[4]In this case, outer IP address is the IP address of the entity hosting GTP-U

After the SDAP and PDCP-U processing is performed on the PDUs, the PDCP PDUs are encapsulated into GTP-U messages and transported over the F1-U interface using UDP/IP transport. The gNodeB-DU decapsulates the GTP/UDP/IP/Eth headers and performs RLC, MAC and PHY layer processing on the PDCP PDUs. The PHY layer transport blocks are transmitted to UEs within DRBs over the NR-Air interface.

Figure 4.5 illustrates an example data flow of a data packet in the downlink direction

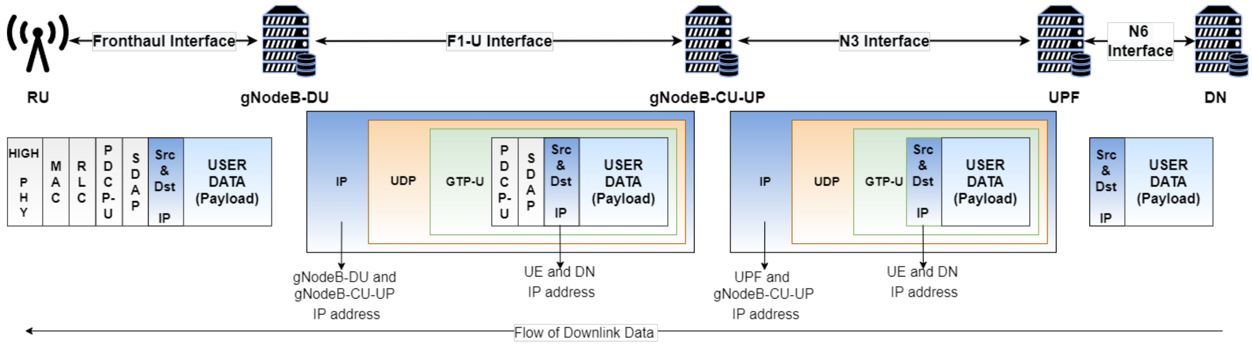


Figure 4.5: DL user plane data flow in the reference architecture

4.1.2 Signalling involved in the 3GPP defined PDU session establishment procedure

Figure 4.6 shows the signal flow that happens during a PDU session establishment procedure when the 3GPP proposed architecture and interfaces are employed.

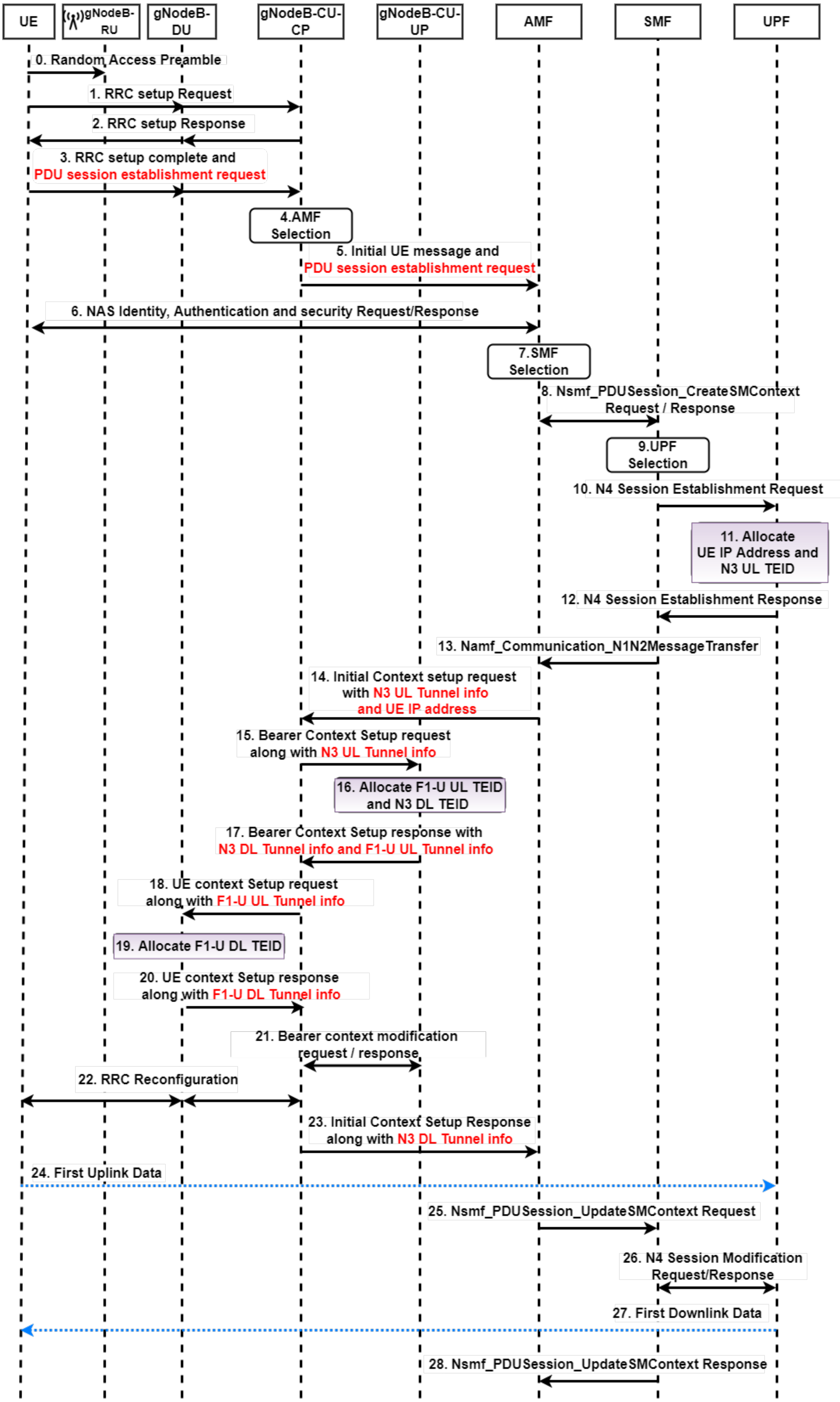


Figure 4.6: Signalling of PDU session establishment procedure for 3GPP defined architecture

1. A RRC Setup Request is sent from UE to gNodeB-CU-CP via gNodeB-DU; The message transfer is done via SRB 0 and involves the establishment of SRB 1.

2. A RRC Setup Response is sent from gNodeB-CU-CP to UE via gNodeB-DU; The message transfer is done via SRB1.

3. The RRC Setup complete message, used to certify the successful completion of an RRC connection setup, is coupled with a Registration Request[5] and sent from UE to gNodeB-CU-CP via gNodeB-DU in the designated NAS-Message field. SRB1 is involved in this signalling.

4. AMF selection is made by gNodeB-CU-CP.

5. The Initial UE Message, including PDU Session Establishment Request, is sent from gNodeB-CU-CP to AMF. The message also includes the Registration Request received from the user equipment in the RRC Setup Complete message.

6. Identity, Authentication and security request-response messages are exchanged between AMF and UE

7. SMF Selection is made by AMF

8. Nsmf_PDUSession_CreateSMContext Request is sent from AMF to the SMF if the AMF does not already associate with the SMF for the requested PDU session. The SMF sends Nsmf_PDUSession_CreateSMContext Response to the AMF as an acknowledgement of the request. In this step, the PDU session establishment request gets accepted by the SMF, and it processes the request by creating an SM context on the SMF and replies back the SM Context ID to the AMF.

9. UPF selection: SMF selects UPF according to the PDU session requirements made by the UE

10. N4 Session Establishment Request message is sent from SMF to UPF, where SMF requests UPF to allocate UE IP address and N3 UL TEID (CN GTP-U TEID)

11. The UPF allocates UE IP address and N3 UL TEID (CN GTP-U TEID).

12. UPF sends the N4 Session Establishment Response message to the SMF: It is an acknowledgement sent by the UPF which contains the UE IP address and N3 UL tunnel info (CN tunnel info) requested by the SMF.

13. SMF sends a Namf_Communication_N1N2MessageTrasfer message containing PDU session ID, information about PDU sessions such as QoS information and N3 UL tunnel info (CN tunnel info) is transferred to AMF.

14. AMF send an Initial context setup request message to gNodeB-CU-CP. The message consists of Registration Accept message, PDU session established message in PDU session status, N3 UL tunnel info (CN tunnel info), UE IP address allocated by UPF, and PDU session resource setup request.

[5] The Registration Request is an initial NAS message and triggers the N1 NAS signalling connection establishment.

15. The gNodeB-CU-CP sends a BEARER CONTEXT SETUP REQUEST to gNodeB-CU-UP. The message includes the N3 UL tunnel info (CN tunnel info) of the N3 tunnel to set up the bearer context in the gNodeB-CU-UP. The gNodeB-CU-CP determines flow-to-DRB mapping and transmits the resulting SDAP and PDCP-U configurations to the gNodeB-CU-UP.

16. The gNodeB-CU-UP allocates F1-U UL TEID for the F1-U tunnel and N3 DL TEID (AN GTP-U TEID) for the N3 tunnel

17. The gNodeB-CU-UP responds to gNodeB-CU-CP with BEARER CONTEXT SETUP Response. The message contains the F1-U UL TEID for the F1-U tunnel and N3 DL TEID (AN GTP-U TEID) for the N3 tunnel.

18. gNodeB-CU-CP sends F1 UE context setup request to gNodeB-DU. The message includes the F1-U UL tunnel info of the F1-U tunnel and directions to set up one or more bearers in the gNodeB-DU.

19. Allocate F1-U DL TEID for the F1-U tunnel

20. The gNodeB-DU sends the UE CONTEXT SETUP RESPONSE message to the gNodeB-CU-CP. It also contains the F1-U DL tunnel info of the F1-U tunnel.

21. The gNodeB-CU-CP transmits the BEARER CONTEXT MODIFICATION REQUEST message to the gNodeB-CU-UP, which contains the F1-U DL tunnel info. The gNodeB-CU-UP acknowledges back.

22. RRC reconfiguration
    - gNodeB-CU-CP to UE, via gNodeB-DU: RRC reconfiguration
    - UE to gNodeB-CU-CP, via gNodeB-DU: RRC reconfiguration complete

23. gNodeB-CU-CP sends an INITIAL CONTEXT SETUP RESPONSE to AMF, indicating a successful PDU session establishment. The message also includes the N3 DL tunnel info (AN tunnel info) of the N3 tunnel, which should be forwarded to the UPF (specified per PDU session).

24. First UL data to the UPF

25. The AMF sends the Nsmf_PDUSession_UpdateSMContext Request message to the SMF. In this message, the AMF passes the N2 SM information[6] received from the RAN to the SMF; It includes the N3 DL TEID for the NG-U tunnel (AN GTP-U TEID).

26. With the N4 Session Modification Request, the SMF intends to share the AN tunnel info and the related forwarding rules with the UPF. The UPF acknowledges an N4 Session Modification Response to the SMF. Further, the UPF will transfer any downlink packets of the PDU session that have been buffering to the UE.

27. First DL data to the UE

[6] N2 SM information contains information about the PDU sessions such as QoS information and N3 UL TEID (CN tunnel info).

28. SMF sends a Nsmf_PDUSession_UpdateSMContext Response to the AMF using which the SMF signs up for the UE mobility event notification from the AMF by executing the Namf_EventExposure_Subscribe service operation, using which the AMF updates the location and movement of the UE

4.1.3 Motivation for new architecture

As the above mentioned reference architecture is part of 3GPP, it meets all the requirements of 5GS described by 3GPP. But, certain aspects of it can be improved and optimised.

By introduction of CUPS into RAN, the CP and UP functions of the RAN has been separated successfully. But even after employing CUPS within RAN, most of the deployments including the reference architecture allocate CU-CP and CU-UP together at a same location. It is not an optimal use of CUPS approach and not an optimal choice for CU-UP allocation. Hence for CUPS to be implemented efficiently, scaling and deploying of CU-UP has to be discrete according to the application requirements.

Even though the 5GS NFs in the reference architecture are virtualised, it requires proprietary hardware for NF deployments. Hence, CU-UP instance cannot be created at a particular location unless it has the required dedicated hardware. Also, the CU-UP instance for a PDU session is created upon request from CU-CP as seen in Figure 4.6. Dynamic deployment of CU-UP based on changing loads and system requirements is not a feasible at CU-CP level. A smart controller above CU-CP is required to aid the CU-CP for dynamic deployment of CU-UP and to decide the optimal location to deploy CU-UP based on load balancing. Furthermore, when the functional split approach is employed, the gNodeB is split into multiple different entities based on their functions. The co-ordination between these entities is difficult and needs tight coupling between them.

Processing the gNodeB-CU-UP in a more centralised location means that SDAP and PDCP headers are added as part of the packet length and this overhead must travel long distances between the gNodeB-CU-UP and gNodeB-DU, implying they have to go through more routing nodes.

The GTP-U message transfer goes through UDP/IP/Eth encapsulation/decapsulation over N3 and F1-U interface. Taking the F1-U interface into consideration, we can have more efficient communication between the gNodeB-CU-UP and gNodeB-DU if they are co-located. By this, we can optimise the overhead, processing power and time delay caused by UDP/IP/Eth stack.

To summarise, an architecture that exploits the CUPS approach within RAN, including fully virtualised NFs deployable on COTS hardware and co-locates the CU-UP and DU, has to be designed and evaluated.

4.2 Proposal for a new architecture

In this section, first, the new architecture is proposed. Secondly, the virtualisation mechanisms used for deploying NFs on COTS hardware are explained. Third, the explanation of the role of the near-RT RIC. Followed by the advantages of our proposed architecture are explained. Fourth, the Inter-process communication mechanism and how it is adapted in this study is explained. Section 4.2.5 elaborates on the PDU session's CP and UP protocol stack for the proposed architecture. Finally, the signalling call flow diagram of the PDU session establishment in the proposed architecture is explained.

Figure 4.7 illustrates our proposed architecture in which the O-RAN NFs are deployed in various cloud locations; the CU is split into O-CU-CP and O-CU-UP and is connected using the E1 interface.

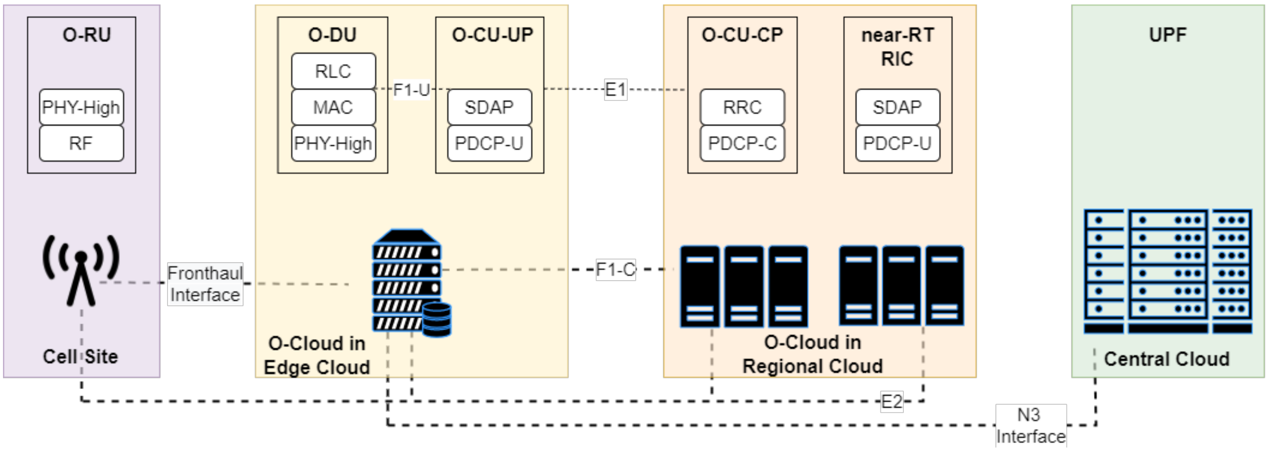


Figure 4.7: Proposed architecture for 5GS, employing O-RAN and CUPS approach in RAN

The O-DU is connected to the O-CU-CP and O-CU-UP using the F1-C and F1-U interfaces. The near-RT RIC is deployed in the regional O-Cloud location and communicates with the O-CU-CP, O-CU-UP and O-DU using E2 interfaces. Using the concept of NFV, all the E2 nodes (O-CU-CP, O-CU-UP and O-DU) are deployed as VNFs on different containers. O-DU and O-CU-UP are deployed as VNFs on different containers running on the same physical host machine at O-Cloud in an edge cloud location. Furthermore, O-CU-CP is deployed as a VNF in a container whose execution environment is more of a centralised location on an O-Cloud at a regional cloud. The near-RT RIC is also deployed on an O-Cloud in a regional cloud location. The 5GC components such as UPF are deployed in the central cloud.

The RAN NFs in the proposed architecture are fully O-RAN. The logic behind employing O-RAN in this thesis is as follows. Firstly, the proposed architecture needs to support the dynamic deployment of NFs. Second, because it is virtualised and open, it can be installed on COTS hardware, which is not proprietary and can be commonly supported by more edge locations. Thirdly, O-RAN includes an intelligent controller that can be used to make deployment decisions, overlook all the NF deployments, and drive them to better co-ordination. Finally, it is in line with the specifications and interfaces of 3GPP, hence easier for comparison study.

4.2.1 Virtualisation mechanisms

As explained in Section 1.3.1, VNFs are deployed on cloud computing technologies using virtualisation mechanisms/techniques. Virtual machines and Containers are the two most common mechanisms [46]. Bare metal hypervisors and Unikernels are the other two options. The choice of the virtualisation technique is based on the objective. Figure 4.8 illustrates all the mechanisms mentioned above. They are elaborated below:

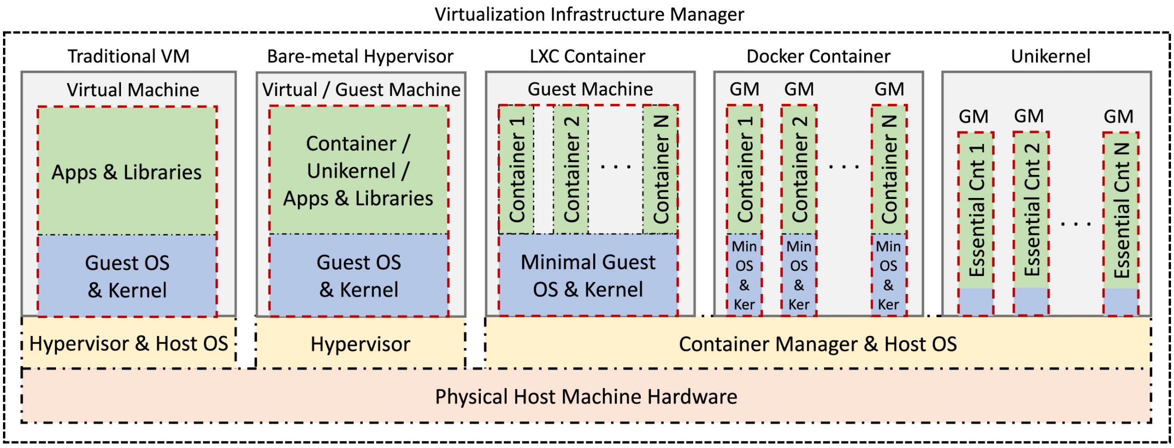


Figure 4.8: Types of virtualisation mechanisms with high-level NFV architecture [36]

- **VM:** Through virtualisation, applications can be operated in environments that are functionally isolated from the physical hardware (physical host machine). These environments divide the servers into virtual machines consisting of a guest Operating System (OS) and kernel. The Physical host machine has an OS of its own called Host OS. In simple words, VMs are OS or application environments installed on the host OS, imitating the physical hardware by providing resources to run the VNFs [36], [46].

- **Bare-metal Hypervisors:** The working of Bare-metal hypervisors is almost similar to the standard VMs. It stands apart from standard VMs, as it does not require a Host operating system.

- **Containers:** The term container refers to a software package that includes all the code and resources needed to run in a virtual setting. The Host OS and kernel are shared between the containers, but namespace isolation isolates them from each other. Compared to VMs, Containers require a minimal guest OS. Linux containers [48] and Dockers [49] are the two widely used containers [36].

We choose the Container mechanism with Docker for our approach as it aids in simpler Interprocess communication, and has lower overhead compared to VMs. It also has the option of sharing process-IDs between two containers [46].

4.2.2 Role of near-RT RIC in the proposed architecture

As explained earlier, in the 3GPP approach, the gNodeB-CU-UP resources required for the requested PDU session are created at an appropriate location selected by gNodeB-CU-CP. In the proposed architecture, the O-CU-UP instance for the requested PDU session is deployed dynamically as a VNF on the same physical host machine where the O-DU VNF handling that particular PDU session is deployed.

When allotting O-CU-UP resources requested by the PDU session, the near-RT RIC intervenes and overrides the O-CU-CP using the Control service (see Appendix C.3). It instructs the O-CU-CP to deploy the required O-CU-UP resources for the requested PDU session in the same execution environment where the O-DU handling that particular PDU session is deployed. The steps elaborating the procedure of near-RT RIC intervening and controlling are explained below:

1. Using the RIC Subscription procedure (see Section 3.3.2.2), the xApp on the near-RT RIC subscribes to a Bearer Context Setup function of O-CU-CP, asking for an event trigger when the O-CU-UP resources have to be set up (Bearer Context setup) for the PDU session of a particular type requested by any UE. Furthermore, the O-CU-CP acknowledges the request.

2. When the Bearer Context setup occurs during the PDU session establishment, the O-CU-CP detects the subscribed event being triggered. Furthermore, using the RIC Indication procedure (see Section 3.3.2.2), the O-CU-CP will send an INSERT message to the near-RT RIC informing that an event trigger has been detected to set up the O-CU-UP resource for a requested PDU session (Bearer Context setup). Along with sending the insert message, the O-CU-CP suspends the procedure, sets a timer, and waits for near-RT RIC to respond (also see Section 4.2.6 step 15).

3. Using the RIC Control procedure (see Section 3.3.2.2), the near-RT RIC directs the O-CU-CP to set up the O-CU-UP resources for the requested PDU session (Bearer Context setup) in the same physical host machine as the O-DU handling that particular PDU session at O-Cloud in edge cloud location (also see Section 4.2.6 step 16).

4.2.3 Advantages

The advantages of the proposed architecture over the reference architecture can be explained as:

1. The overhead caused by SDAP and PDCP headers contributes to routing delay. In a deployment where O-CU-UP and O-DU are executed as VNFs on different containers at a same edge cloud environment, there are fewer or no nodes between them. Hence the nodal delay caused by SDAP and PDCP header processing is less or non-existent.

2. As we have learned from Figure 4.4 the user plane protocol stack F1-U interface employs GTP-U tunnelling upon UDP/IP/Eth to transfer the user plane PDUs. In our approach, as the O-CU-UP and the O-DU are executed as VNFs on different containers on a single O-Cloud physical host machine at O-Cloud in the edge cloud location, the IP-based routing is unnecessary. Instead of carrying the GTP-U messages over the UDP/IP transport, they can be communicated between the O-CU-UP and O-DU using Inter-process communication (IPC). As we are eliminating the UDP/IP routing, the DL GTP-U messages need not go through UDP/IP/Eth header encapsulation and decapsulation at O-CU-UP and O-DU, respectively; and vice-versa for UL GTP-U messages.

3. The IPC between CU-UP and DU mentioned above is possible exclusively when the two entities are located on a single O-Cloud physical host machine. Assume that the O-Cloud physical host machine handling the O-DU is heavily loaded; hence, the CU-UP cannot be deployed on the same physical host machine. Instead, it has to be deployed on a different physical host machine, and the communication mechanism has to be changed to the traditional IP routing. The required dynamic deployment of CU-UP and communication mechanisms suitable for PDU session types can be deployed as xApps and handled by near-RT RIC.

4.2.4 Inter-Process Communication

The process running on Operating systems (OS) can be classified as an Independent process [7] and the co-operative process [8]. When two or more co-operative processes run on the same physical host machine, they can exchange information via host OS. The OS facilitates a communication mechanism called Inter-Process communication, using which information can be passed from one process to another. IPC aids in communication between two co-operative processes. In our thesis, O-CU-UP and O-DU are considered co-operative processes as they exchange data. The two processes which use IPC follow the Client-Server model. The server process sends data, and the client process receives the data.

Shared memory IPC is one of the most efficient communication mechanisms. Under this mechanism, communication between two co-operative processes happens via shared memory in the user memory space. The memory segment used for shared memory IPC is created by one of the co-operative processes. The other process attaches itself and gets access to the shared memory by mapping to the shared segment. The attributes of the shared memory, such as access permissions and size of the memory block, are defined when the shared memory is created.

It is not possible for processes running on different hosts to communicate with one another via shared Memory. Therefore O-CU-UP and O-DU must be deployed as containers on the same physical host machine for the proposed approach to work as expected.

To initiate the communication between any two processes, shared memory must be created, and the address of the created shared memory must be exchanged between the two processes. For the transfer of UL GTP-U messages from O-DU to O-CU-UP, the O-CU-UP acts as the receiving entity and creates the shared memory dedicated to UL communication. Furthermore, O-CU-UP sends the shared memory address to O-DU via O-CU-CP. Now O-DU can send UL packets to O-CU-UP using the UL shared memory address.

For the transfer of DL GTP-U messages from O-CU-UP to O-DU, the O-DU will play the role of receiving entity and creates the shared memory dedicated to DL communication. Furthermore, O-DU sends the shared memory address to O-CU-UP via O-CU-CP. Now O-CU-UP can send DL packets to O-CU-UP using the DL shared memory address.

For the explanatory purpose, consider DL UP data flow. The GTP-U process in O-CU-UP generates the DL GTP-U messages by encapsulating the PDCP PDUs with a GTP header; and further writes it to the shared memory location. The GTP-U process in O-DU is already waiting for packet arrival at the shared memory location, hence reads the DL GTP-U messages on arrival from O-CU-UP via shared memory. For the UL UP data flow, consider vice versa of the above explanation.

4.2.5 Protocol stacks of a PDU session in the proposed architecture

4.2.5.1 Control plane protocol stacks of a PDU session

The control plane protocol stack of PDU session in the proposed architecture is shown in figure 4.9 and figure 4.10. The E2 control plane protocol stack illustrated in Figure 4.10 is a new addition to support the role of near-RT RICs function in establishing PDU sessions.

[7]An independent process is a process that does not influence or is not influenced by any processes being executed on the system. There is no data exchange between any two independent processes

[8]A co-operative process is a process influenced by other processes being executed on the system. There is an exchange of data between the two processes

As O-RAN E2 nodes map to gNodeB nodes of 3GPP, O-CU-CP, O-CU-UP, O-DU, and O-RU will employ the same protocol stack employed in gNodeB-CU-CP, gNodeB-CU-UP, gNodeB-DU and RU.

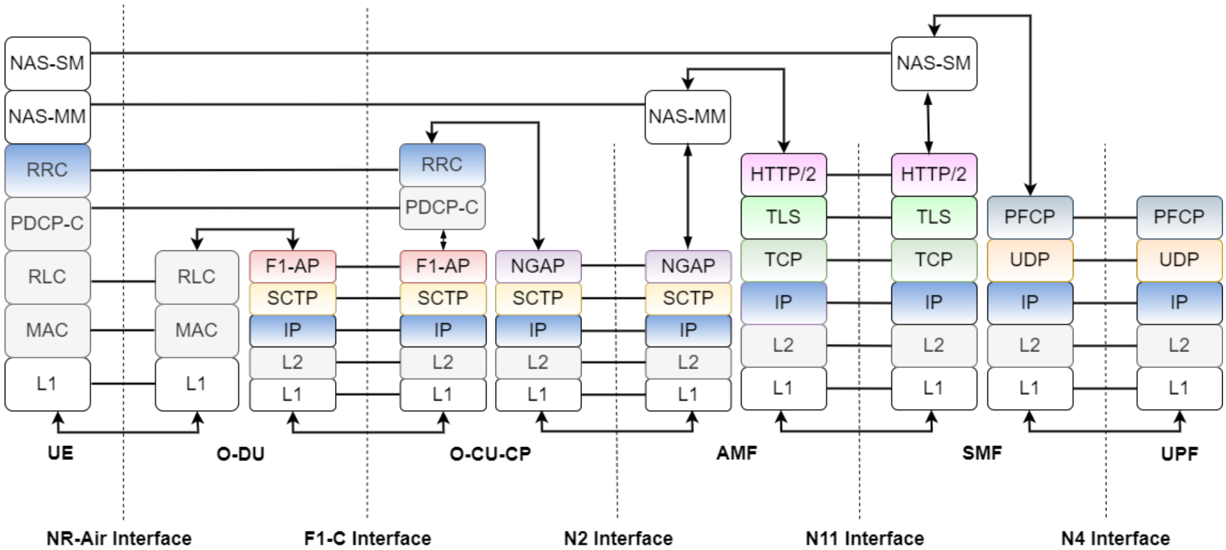


Figure 4.9: End-to-End control plane protocol stack of a PDU session from UE to UPF in the proposed architecture. These are the protocols that control the PDU session signalling in the proposed architecture.

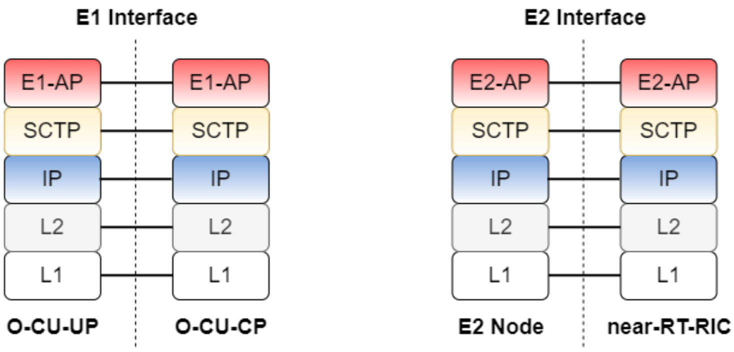


Figure 4.10: E1 and E2 interface protocol stack, part of a PDU session's control plane protocol stack connecting O-CU-CP to O-CU-UP and E2 node to near-RT-RIC, respectively.

Compared to the explanation provided in Section 4.1.1.1, there are changes in the contents of F1-U tunnel info and we have the addition of E2 protocol stack as near-RT RIC is a part of the proposed control plane of the PDU session. The changes proposed are explained below:

1. **Proposing enhanced F1-U tunnel (eF1-U tunnel):** In the reference architecture, the F1-U tunnel uses the outer IP address[4] for IP routing. In the proposed architecture, we use IPC to communicate the GTP-U messages over the F1-U tunnel. As we propose changes in the F1-U tunnel, it is referred to as the enhanced F1-U tunnel (eF1-U tunnel).

   For implementing the eF1-U tunnel, we propose that instead of using the outer IP address[4], the shared memory address of the F1-U GTP-U process should be employed for setting up a tunnel and transferring GTP-U packets. As explained in Section 4.2.4, for the eF1-U UL tunnel, the O-CU-UP creates the shared memory dedicated to UL communication.

Furthermore, the UL shared memory address, along with the F1-U UL TEID, is called enhanced F1-U UL tunnel info (eF1-U UL tunnel info). eF1-U UL tunnel info is sent to O-DU via O-CU-CP (See step 18 of Section 4.2.6).

Also, from the explanation in Section 4.2.4, we know that for the eF1-U DL tunnel, the O-DU creates the shared memory dedicated to DL communication. Furthermore, the DL shared memory address, along with the F1-U DL TEID, is called enhanced F1-U DL tunnel info (eF1-U DL tunnel info). This information is sent to O-CU-UP via O-CU-CP (See step 21 Section 4.2.6). The N3 UL tunnel info and N3 DL tunnel info remain unchanged.

2. As explained above, the near-RT RIC is employed for setting up the O-CU-UP resources, which is a part of the PDU session establishment procedure. Hence, it exchanges application level signalling messages related to the procedure with O-CU-CP. E2AP is used to facilitate this communication over the E2 interface between O-CU-CP and near-RT RIC. The transport of E2AP messages over the E2 interface is facilitated by TNL based on the SCTP/IP stack.

Apart from the changes mentioned above, other functions, application protocols, interfaces, and the messages exchanged to control the PDU session remain in line with Section 4.1.1.1.

4.2.5.2 User plane protocol stacks of a PDU session

Figure 4.11 illustrates the user Plane protocol stack of the PDU session in the proposed architecture.

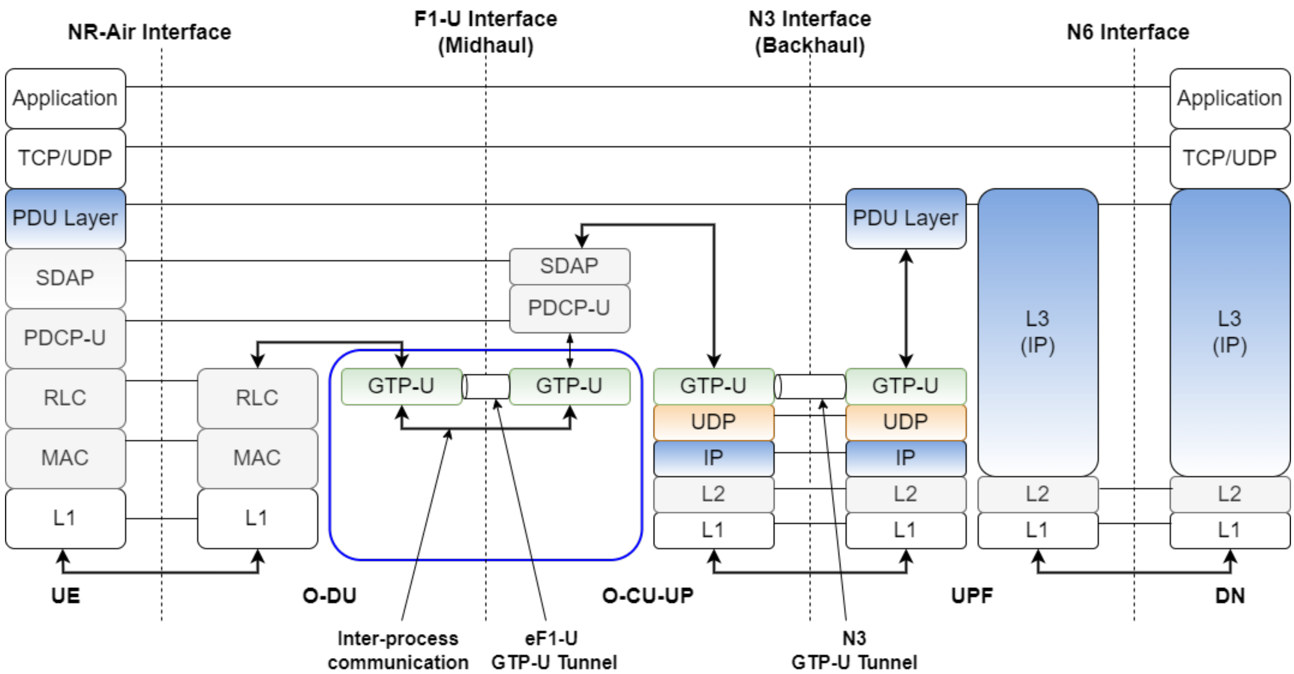


Figure 4.11: End-to-End user plane Protocol Stack of a PDU session from UE to DN in the proposed architecture. These protocols transports user data across the access stratum, implementing the PDU session

For explanation purposes, let us consider the downlink direction of the packet. The application data packets are sent from DN to UPF as IP packets. The PDU layer at the UE and UPF relates to PDUs transmitted between UE and the DN during a PDU session. The GTP-U messages are exchanged between UPF and O-CU-UP over the N3 interface as previously explained in Section 4.1.1.2. Once the O-CU-UP decapsulates the GTP/UDP/IP/Eth headers from the GTP-U messages received from UPF, it processes the packets with SDAP and PDCP headers. After PDCP-U processing, the PDCP PDUs are encapsulated with the GTP-U protocol. These GTP-U messages are transmitted to O-DU in the eF1-U tunnel over the F1-U interface using Inter-process communication. The O-CU-UP uses the shared memory address of the GTP-U process at the O-DU and the F1-U DL TEID to match the packets to their respective tunnels. The process is vice versa in the UL direction of packet flows.

Figure 4.12 illustrates an example data flow of a data packet in the proposed user Plane in downlink direction

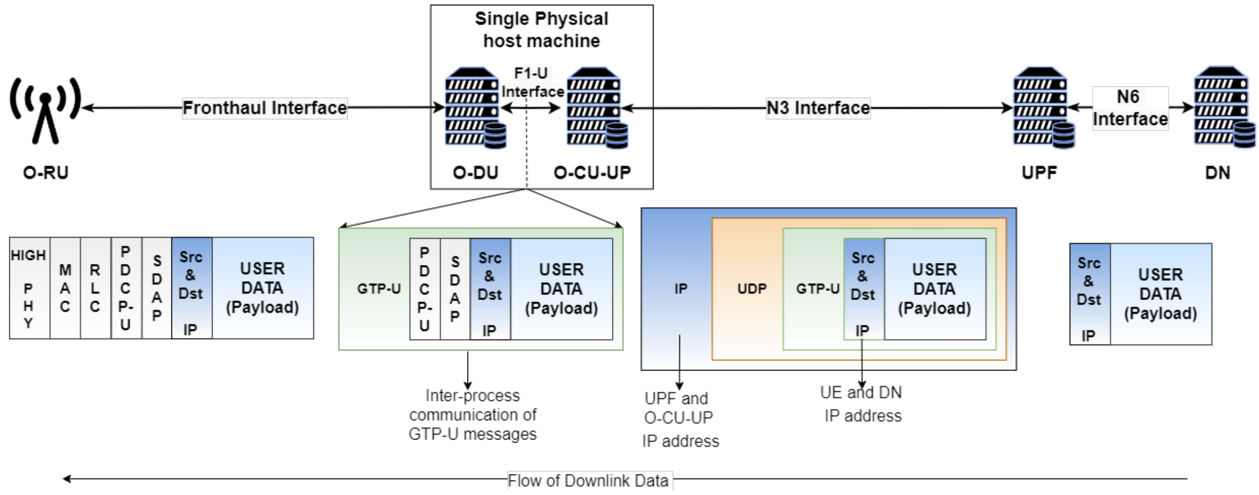


Figure 4.12: DL user plane data flow in the proposed architecture

4.2.6 Signalling involved in the PDU session establishment procedure for the proposed architecture

Figure 4.13 illustrates the call flow during the PDU session establishment procedure for the proposed architecture. The enclosure indicates the changes that are made from Figure 4.6, the call flow diagram of the reference architecture.

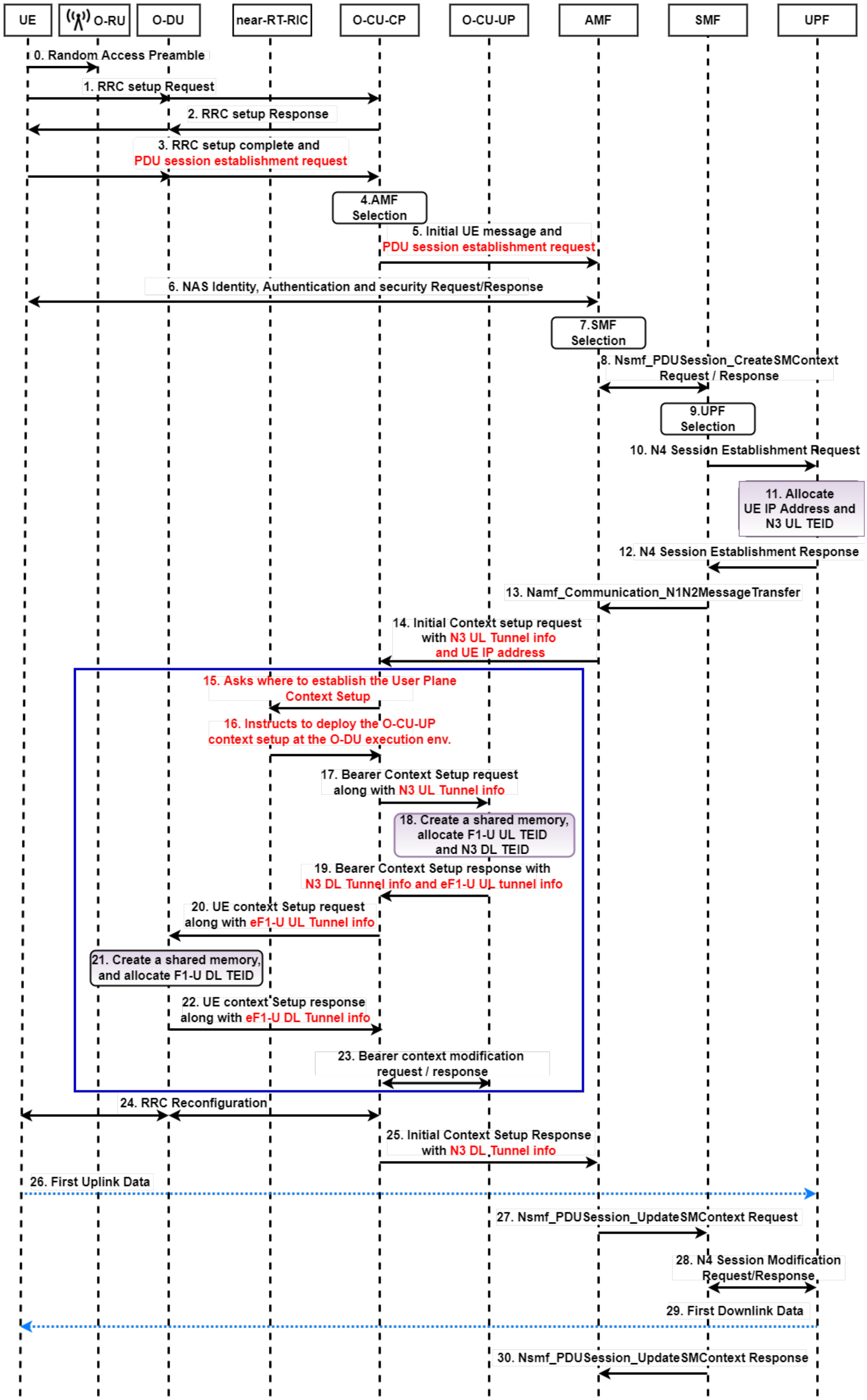


Figure 4.13: Signalling of PDU session establishment procedure for the proposed architecture

For the above Figure 4.13, Steps 1-14 are defined in Section 4.1.2[9]

15. The O-CU-CP informs near-RT RIC that the O-CU-UP bearer context setup event has occurred and, in turn, asks where to deploy the O-CU-UP instance.

16. The near-RT RIC controls/instructs O-CU-CP to allocate the necessary O-CU-UP resources at the same O-Cloud in the edge cloud location as the O-DU for the particular PDU session.

17. The O-CU-CP sends a BEARER CONTEXT SETUP REQUEST to O-CU-UP. The message includes the N3 UL tunnel info (CN tunnel info) of the N3 tunnel to set up the bearer context in the O-CU-UP. The O-CU-CP determines flow-to-DRB mapping and transmits the resulting SDAP and PDCP-U configurations to the O-CU-UP.

18. The O-CU-UP creates a shared memory for receiving UL GTP-U messages and allocates F1-U UL TEID. The address of the shared memory for UL and the F1-U UL TEID together forms the eF1-U UL tunnel info of the eF1-U tunnel. The O-CU-UP also allocates N3 DL TEID (AN GTP-U TEID) for the N3 tunnel

19. The O-CU-UP responds to the BEARER CONTEXT SETUP REQUEST by forwarding the N3 DL tunnel info of the N3 tunnel and eF1-U UL tunnel info of the eF1-U tunnel.

20. O-CU-CP sends F1 UE context setup request to O-DU. The message includes the eF1-U UL tunnel info of the eF1-U tunnel and directions to set up one or more bearers in the O-DU.

21. The O-DU creates a shared memory for receiving DL GTP-U messages and allocates F1-U DL TEID. The address of the DL shared memory and the F1-U DL TEID together forms the eF1-U DL tunnel info of the eF1-U tunnel.

22. The O-DU sends the UE CONTEXT SETUP RESPONSE message to the O-CU-CP. It also contains the eF1-U DL tunnel info of the eF1-U tunnel.

23. The O-CU-CP transmits the BEARER CONTEXT MODIFICATION REQUEST message to the O-CU-UP, which contains the eF1-U DL tunnel info. The O-CU-UP acknowledges back.

Steps 24-30 are defined in Section 4.1.2[9]

[9]When referring back to Section 4.1.2 for the use of steps in proposed architecture, the naming conventions of the nodes have to be changed from gNodeB terminologies (gNodeB-CU-CP,gNodeB-CU-UP, and gNodeB-DU) to O-RAN terminologies (O-CU-CP, O-CU-UP, and O-DU)

Chapter 5

Modelling, Calculations and Simulations

In this chapter, we aim to calculate the gains of our system in two steps. Firstly, in Section 5.1, we evaluate how the transport latency of the IP transport network changes based on the location of CU-UP deployment. Secondly, in Section 5.2, we evaluate the communication latency gains obtained in the user plane by the proposed IPC between O-CU-UP and O-DU compared to the traditional UDP/IP/Eth communication protocol stack.

5.1 Calculation of transport latency gains in IP transport network due to processing the CU-UP closer to the DU

In this section, we present the model to calculate the overhead delay caused by carrying the SDAP and PDCP headers over the routers in the IP user plane transport network. Using option 2.2 (PDCP/RLC) split of the functional split, CU is split into the CU-CP and CU-UP. In this thesis study, we have proposed an architecture in which O-CU-UP and O-DU are executed as VNFs on a single physical host machine at edge cloud environment reducing a considerable amount of distance between the two entities and the number of routers the PDCP PDUs have to pass through. The source IP packets are transported between UPF and CU-UP over the backhaul, and then the source IP packets with SDAP/PDCP overhead are transported between gNodeB-DU and gNodeB-CU-UP over the midhaul.

Packet length contributes to nodal delays. In this section, we want to calculate the delay caused by the SDAP/PDCP header; hence the enhanced communication protocol stack is not yet considered here but will be discussed in the next part. The communication protocol stack GTP/UDP/IP/Eth remains the same between UPF/CU-UP and CU-UP/DU for both the scenarios shown in Figure 5.1.

5.1.1 Processing CU-UP at a centralised regional cloud

Scenario A of Figure 5.1 is the reference architecture, where the PDCP-U and SDAP processing is performed at a centralised regional cloud location where SDAP/PDCP headers are added (for DL) / removed (for UL). In this scenario, two cases, with and without the Robust Header Compression (RoHC) [50], are considered and discussed below:

1. **Case 1: Without RoHC**

   The 1432 bytes of payload from the source has 28 bytes of UDP/IPv4 header length. At the regional cloud location, gNodeB-CU-UP processing is done and adds four bytes of

SDAP and PDCP headers. Further, with the GTP/UDP/IP/Eth communication protocol stack, 1514 bytes of information are transmitted for every 1432 bytes of payload from the reference point to the edge cloud location.

2. **Case 2: With RoHC**

   The ROHC compresses the UDP/IP header of the IP data packet sent from the data network. The 1432 bytes of payload from the source has 28 bytes of UDP/IPv4 header length, which is reduced to 1-3 bytes after ROHC (we assume the maximum length of 3 bytes for calculation). The previously 1460 bytes of IP packet is reduced to 1435 bytes (1432 bytes of payload + 3 bytes of ROHC header) after header compression. Further, the regional cloud location adds 4 bytes of SDAP and PDCP headers. Along with the GTP/UDP/IP/Eth communication protocol stack, 1489 bytes of information are transmitted for every 1432 bytes of payload from the reference point to the edge cloud location.

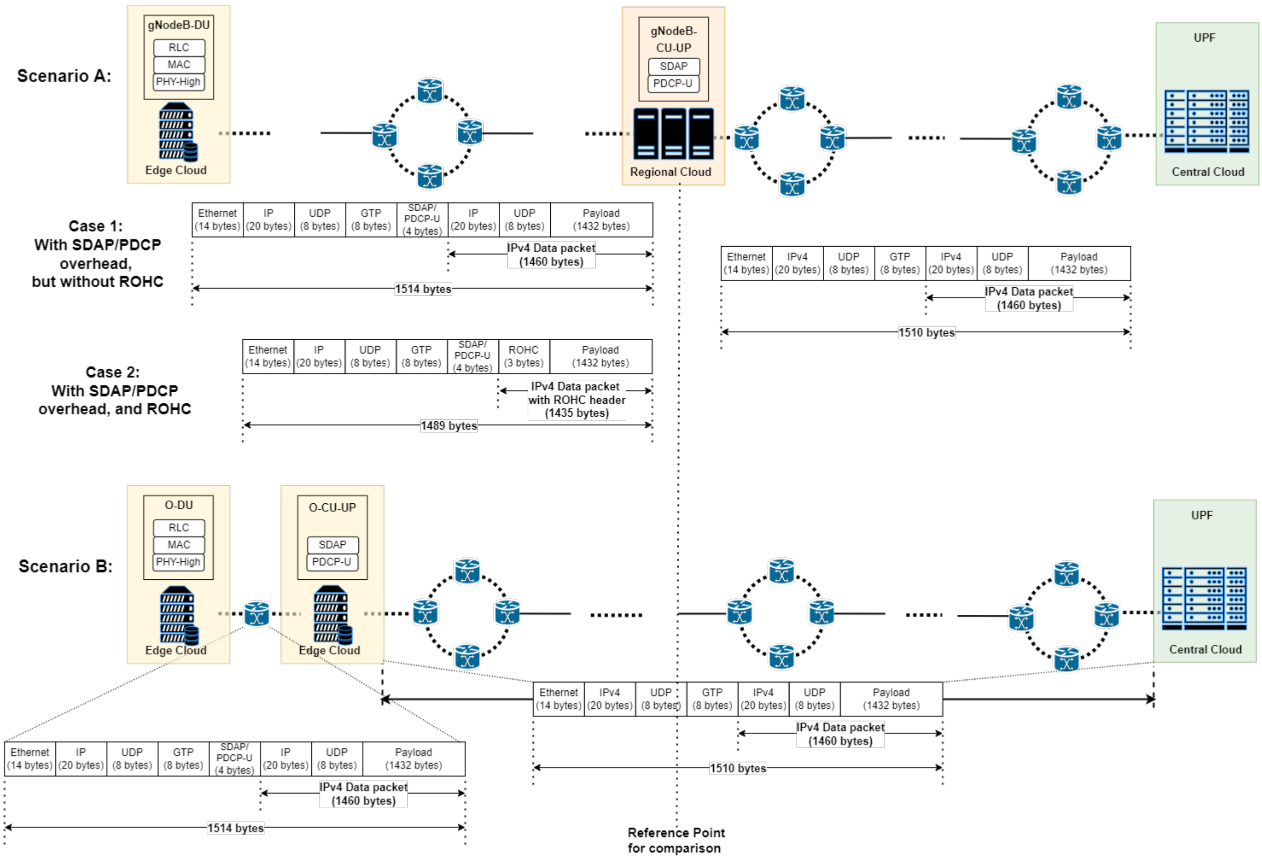


Figure 5.1: User plane transport model employed for nodal delay calculations

5.1.2 Processing CU-UP at a edge cloud

The scenario B of Figure 5.1 represents the proposed architecture where the O-CU-UP processing is happening at the edge location due to which SDAP and PDCP headers are added (for DL) / removed (for UL) at the edge location saving the data overhead in the midhaul. One single router connecting the O-CU-UP and O-DU represents traditional network routing, as the enhanced communication between O-CU-UP and O-DU is not considered. Instead, GTP/UDP/IP/Eth communication protocol stack is used because we want to measure the delay caused transportation by SDAP/PDCP headers solely.

5.1.3 Calculation of nodal delay

Scenario A and scenario B can be compared, considering regional cloud as a reference point. The 1432 bytes of payload from the source has 28 bytes of UDP/IPv4 header, along with the GTP/UDP/IP/Eth communication protocol stack 1510 bytes of information that originated from UPF continues to travel through the regional cloud to edge cloud location. The transmission of 1510 bytes of information for every 1432 bytes payload is continued from the reference point to the edge cloud location. And then, 1514 bytes of data are passed through one single router between O-CU-UP and O-DU.

Hence to find the transport delay caused by the SDAP/PDCP header in the user plane, we calculate and compare the delay caused by three different packet sizes (1514 bytes, 1489 bytes and 1510 bytes) at one single router.

IP transport is one of the major transport networks used in wired connections (Optic fibre/Ethernet cable) in recent technologies, including the user plane of 5GS. From Figure 5.1, we can see that the user plane transport network is built upon several routers between user plane NFs. Each router contributes to a delay called as nodal delay (d_{nodal}) [51]. Figure 5.2 illustrates the components of a nodal delay. Nodal delay is composed of four types of delay namely, processing delay (d_{proc}) [51], propagation delay (d_{prop}) [51], queuing delay (d_{queue}) [51], and transmission delay (d_{trans}) [51]. Hence the total nodal delay is given by Eq.5.1:

$$d_{nodal} = d_{proc} + d_{prop} + d_{queue} + d_{trans} \tag{5.1}$$

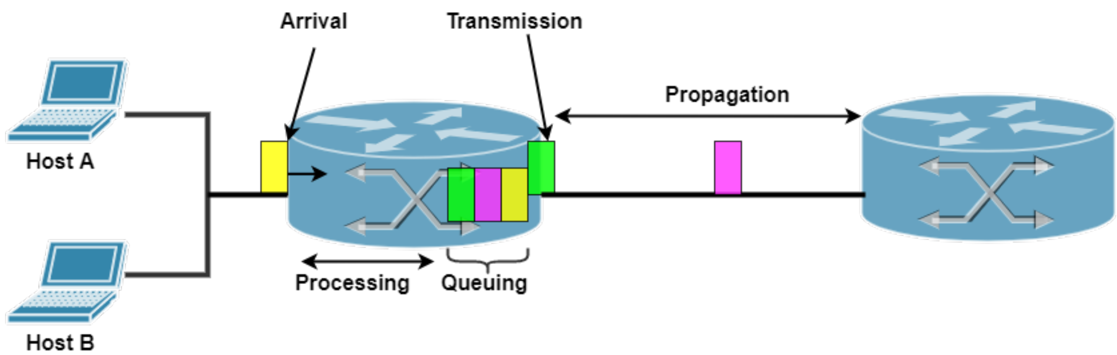


Figure 5.2: Components of a nodal delay

The processing and propagation delays are not affected by the different packet sizes and hence are not considered. As queuing delay and transmission delay are affected by the packet size; hence it is considered in our study and considered as the nodal delay from here after. And the Eq.5.1 can be re-written as:

$$d_{nodal} = d_{trans} + d_{queue} \tag{5.2}$$

Eq.5.2 is considered as the nodal delay in this study. The effect of packet length on transmission delay and queuing delay are explained below.

5.1.3.1 Transmission delay

In IP-based routing, a packet is transmitted to a physical link after being processed by the router (L3 processing of the IP packets). According to Figure 5.2, transmission delay is the time consumed to transfer all the packet bits onto the physical link. It is not affected by the distance between two nodes. Instead, it depends on the link speed and the packet length in bits. The transmission delay is modelled in line with [51]. The packet size in bits is denoted by *(L)* bits, and the speed/link speed/transmission rate of the link between two routers is denoted by *(R)*. Eq.5.3 represents the transmission delay.

$$d_{trans} = \frac{L}{R} \tag{5.3}$$

5.1.3.2 Queuing delay

Since packets are served on First In, First Out (FIFO) priority, a typical approach in packet-switched networks, a packet can only be transmitted after all the early arrived packets are served. So, the packets have to wait in the queue to be served. According to Figure 5.2, the queuing delay considered in this study is the queue formed by the packets waiting to be transmitted onto the physical link. The queuing delay is modeled according to [51], [52] and [53].

In this study, we use the M/M/1 queue, a basic model in the study of queueing theory. The M/M/1 queue title indicates that the system has an i.i.d. exponentially distributed inter-arrival rate (λ) for incoming packets, an i.i.d. exponentially distributed service rate (μ) for serving the packets, one server and queue that is infinitely long [52]. Traditionally, the inter-arrival rate of IP packets (λ) is characterised by the Poisson process, a well-known random process. The average queuing delay (d_{queue}) for M/M/1 system is given by Eq.5.4,

$$Queuing\ delay\ (d_{queue}) = \frac{1}{\mu - \lambda} - \frac{1}{\mu} \tag{5.4}$$

The incoming traffic in bps denoted by (H) can be considered as the product of average packet size (L) and average arrival rate (λ), and Eq.5.5 represents the incoming traffic in bps.

$$Incoming\ traffic\ in\ bps\ (H) = Packet\ size\ (L) \times Packet\ arrival\ rate\ (\lambda) \tag{5.5}$$

Furthermore, the average arrival rate (λ) in packets per second (PPS) is given by,

$$Packet\ arrival\ rate\ (\lambda) = \frac{Incoming\ traffic\ in\ bps\ (H)}{Packet\ size\ (L)}\ pps \tag{5.6}$$

With link speed R and average packet size L, the service rate (μ) can be written as;

$$Packet\ service\ rate\ (\mu) = \frac{Link\ speed\ (R)}{Packet\ size\ (L)}\ pps \tag{5.7}$$

By considering Eq.5.6 and Eq.5.7, we can re-write Eq.5.4 as:

$$Queuing\ delay\ (d_{queue}) = \frac{L}{R - H} - \frac{L}{R} \tag{5.8}$$

From Eq.5.8, we see that queuing delay depends on the packet size, incoming traffic and link speed. The link utilisation is defined as the ratio of average arrival time over average service time, which helps us understand the traffic intensity in the system.

$$Link\ utilisation\,(\rho) = \frac{\lambda}{\mu} \tag{5.9}$$

As packet size and service rate are constant, link utilisation is directly proportional to arrival rate. The arrival rate is varied from 0 PPS to the maximum packets the link can serve per second; this leads to link utilisation to vary from 0 to 100 per cent, that is, ($\frac{\lambda}{\mu} \leq 1$). When a queue becomes filled up more than it can handle, the delay a packet endures while waiting to be served will grow.

5.2 Simulation model for comparing IPC with UDP/IP/Eth network stack

In this section, we explain the experimental setup used for evaluating the performance of the transfer of GTP-U messages between O-CU-UP and O-DU via the IPC mechanism and compare it with the UDP/IP/Eth communication protocol stack. The test environment is an Intel core i5 Linux machine with the following features,

- RAM: 8 GB
- Host OS: Ubuntu 20.04
- Architecture: x64-based processor
- CPU type: 64-bit operating system
- CPU frequency: 2.71 GHz
- No. of Cores: 2

We consider two GTP-U processes for sending and receiving DL GTP-U messages. Our experiments use a packet size of 1472 bytes (See Section5.1.1). The sending process and receiving processes are executed on individual Docker containers. Ubuntu is used as the minimal guest OS to run each docker container. Both Docker containers that execute the sending and receiving process are run on the same Linux machine. There is no other process running in the background while performing our experiments.

Executing the sending and receiving processes on separate Linux machines for testing the UDP/IP/Eth communication protocol stack would have created a better replica of the reference architecture. To justify, executing the docker of sending and receiving processes on different host machines would crate the time synchronisation problem. Recording time stamps during the sending and receiving processes is crucial for our experiments.

This simulation aims to scientifically determine the latency, necessitating communication latency to be measured precisely. According to suggestions made in [54], our experiments were performed using one of the famous Linux timer APIs, clock_gettime(). For both the communication mechanisms IPC and UDP/IP, we aim to calculate a one-way delay between sending GTP-U process and receiving GTP-U process. On the sending side, the timestamp is recorded after the message is created and before the sending process begins. On the receiving side, the timestamp is recorded after the receiving process is completed and before reading the message. The total delay is the difference between the two recorded timestamps.

According to the PDU session establishment procedure (See Sections 4.1.2 and 4.2.6), First, the receiving GTP-U process shares its F1-U/eF1-U tunnel info, followed by sending process. Hence, the sender process will already have the address of the receiver process before sending the packets.

5.2.1 IPC between O-CU-UP and O-DU using shared memory

For shared memory, we use POSIX IPC with C programming language, as the IPC namespace in the docker systems supports both POSIX IPC and SysV IPC. As we are using POSIX IPC shared memory, we employ the mmap() [55] system call in our execution to attach the sender and receiving process to the shared memory. mmap() uses a temporary file as shared memory. While executing the shared memory IPC, we ensure the receiving process is executed first. The receiving process creates a shared memory object, shares the shared memory address with the sending process, and starts to read from the shared memory location. Now, as the sender process has the shared memory address, it starts sending the packets to the shared memory location.

5.2.2 Communication between O-CU-UP and O-DU using UDP/IP/Eth network stack

We use UDP/IP sockets with C programming language to transfer messages between two systems. In UDP/IP sockets, the sockets are made to bind first, and then the receiver process is executed. The receiving process's UDP port number and IP address are shared with the sending process. The receiving process starts to listen for incoming packets. Next, the sender process is executed and starts sending messages to the receiving process using the UDP port number and the IP address of the receiving process. The sockets implementation takes care of the Eth and layer one processing in the background.

Chapter 6

Results

In this chapter, the results of this thesis study are presented in three sections. Section 6.1 presents the results of latency gains in the IP transport network. Section 6.2 presents the proposed architecture's communication latency gains against the reference architecture. Finally, Section 6.3 presents the overall results obtained by comparing the reference and proposed architecture.

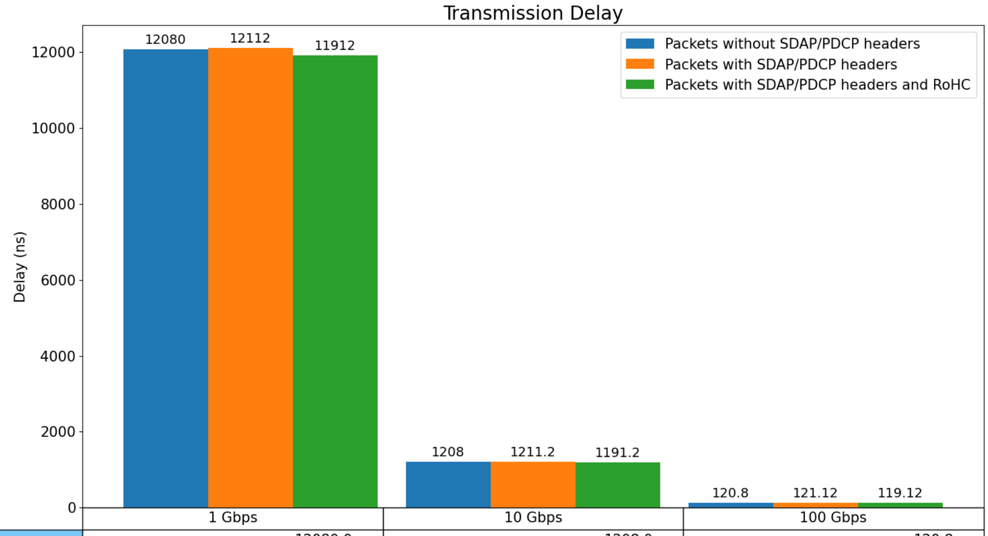
6.1 Latency gains in IP transport network due to processing the CU-UP closer to the DU

In this section, we want to analyse the results of performing the PDCP-U and SDAP processes at a regional cloud location and compare them with the results of processing them at the edge cloud location along with the DU. Three different link capacities of 1 Gbps, 10 Gbps, and 100 Gbps are considered for this calculation.

6.1.1 Transmission delay

The values of transmission delay according to the calculation from Section 5.1.3.1 is plotted into a bar graph as shown in Figure 6.1. The graph consists of the transmission delay of the three types of packets considered in this study at different link capacities.

From the results we can see that the packets which undergo RoHC and CU-UP processing at the centralised regional cloud location have the least transmission delay as transmission delay is directly proportional the packet size. The packets without the SDAP/PDCP overhead is proven to have less transmission delay than the packets with SDAP/PDCP overhead, without RoHC, due to smaller packet length. The delay caused by the SDAP/PDCP overhead without RoHC, for a single packet is 32ns, 3.2ns, and 0.32ns for 1 Gbps, 10 Gbps, and 100 Gbps respectively.



	1 Gbps	10 Gbps	100 Gbps
Packets without SDAP/PDCP headers	12080.0	1208.0	120.8
Packets with SDAP/PDCP headers	12112.0	1211.2	121.11999999999999
Packets with SDAP/PDCP headers and RoHC	11912.0	1191.1999999999998	119.11999999999999

Figure 6.1: Transmission delay

6.1.2 Queuing delay

In most cases, link utilisation alone cannot adequately characterise the queuing delay statistics. Still, it is an excellent way to get a general idea of the magnitude of queuing delay. To analyse the queuing delays experienced by the three types of packets, we plot the graph of queuing delay vs the link utilisation percentage. The aim is to find the contribution of the SDAP/PDCP headers to the queuing delay in the user plane transport network. In the X-axis, to obtain a queuing delay for different link utilisation values, the link capacity is kept constant, and the incoming traffic is increased at intervals of 10 percent of the link capacity. The Queuing delay for 100 percent link utilisation was impossible to plot as the queuing delay values increase without bound and reach infinity.

The queuing delay for the three types of packets at 1 Gbps link capacity is plotted as a line graph and depicted in Figure 6.2. The queuing delay when link utilisation is 70 per cent is tabulated in Table 6.1. Figure 6.2 and Table 6.1 depict that, similar to transmission delay, the queuing delay of packets with SDAP/PDCP headers and RoHC is the least of all the three types of the packet under consideration due to smaller packet size. In terms of queuing delay, for packets without RoHC, performing the CU-UP process closer to the DU location is advantageous. The 4 bytes of the SDAP/PDCP headers are added to the packets at the centralised regional cloud, increasing the packet size and, in turn, the service time. The queues for packets with SDAP/PDCP headers fill up faster than those without SDAP/PDCP headers.

Beyond a specific limit, the queuing delay per router becomes unacceptable; otherwise, the system will have a high packet loss rate. Hence, we have to limit the link utilisation to a certain point. We consider the maximum link utilisation allowed to be 70 percent; after the 70 percent mark, the queuing delay rises exponentially. We must remember that it is a single hop delay value, which will be multiplied by the number of routers along the path. Also, the authors of [53], [56], [57] also suggest keeping the link utilisation below the 70 percent mark.

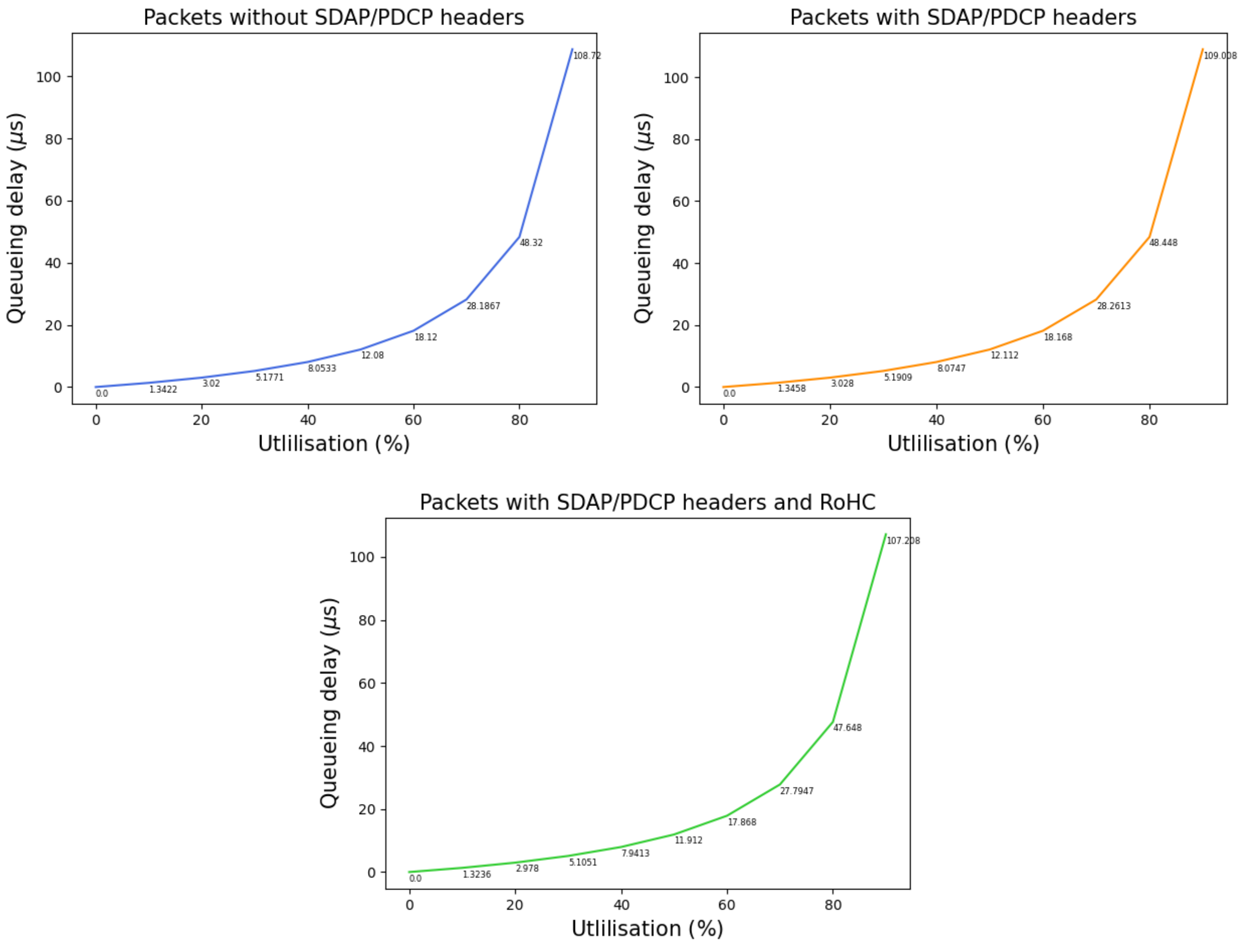


Figure 6.2: Queuing delay at 1 Gbps link capacity

Type of packet	Queuing delay in μs
Packets without SDAP/PDCP headers	28.1867
Packets with SDAP/PDCP headers	28.2613
Packets with SDAP/PDCP headers and RoHC	27.7947

Table 6.1: Queuing delay at 70 % link utilisation of 1 Gbps link capacity

The queuing delaying of the three types of the packet under consideration at 10 Gbps and 100 Gbps are plotted as a graph and are depicted in Figure 6.3 and Figure 6.4. The queuing delay at 70 percent link utilisation of 10 Gbps and 100 Gbps links are tabulated in Table 6.2 and Table 6.3 respectively. The queuing delay pattern in the 10 Gbps and 100 Gbps link capacity for the three types of packets remains in line with the explanation provided for the 1 Gbps link capacity. Even in the case of 10 Gbps and 100 Gbps link capacity, the packets with SDAP/PDCP headers and RoHC have less queuing delay than the other two types of packets. Without RoHC, the packets without SDAP/PDCP headers have less queuing delay.

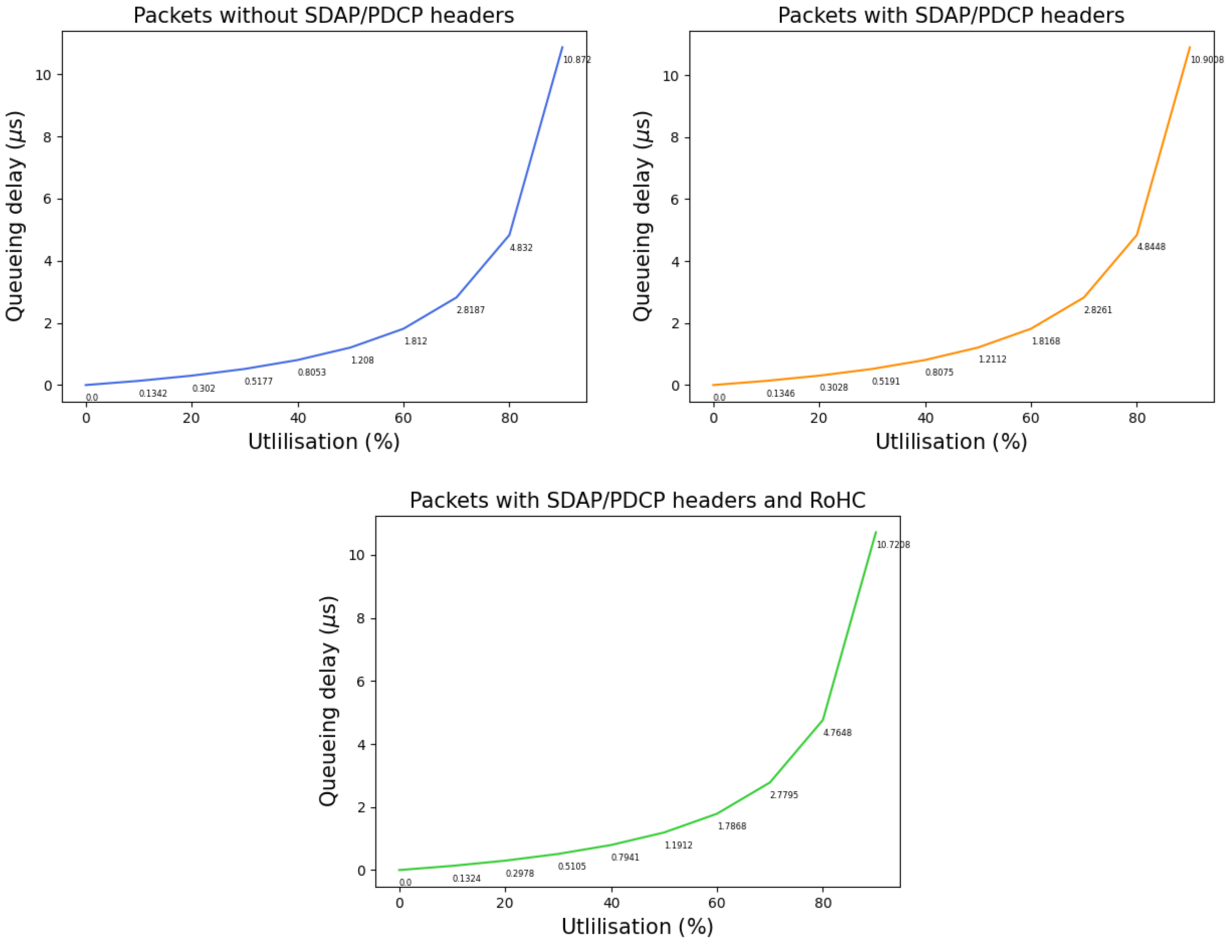


Figure 6.3: Queuing delay at 10 Gbps link capacity

Type of packet	Queuing delay in μs
Packets without SDAP/PDCP headers	2.8187
Packets with SDAP/PDCP headers	2.8261
Packets with SDAP/PDCP headers and RoHC	2.7795

Table 6.2: Queuing delay at 70 % link utilisation of 10 Gbps link capacity

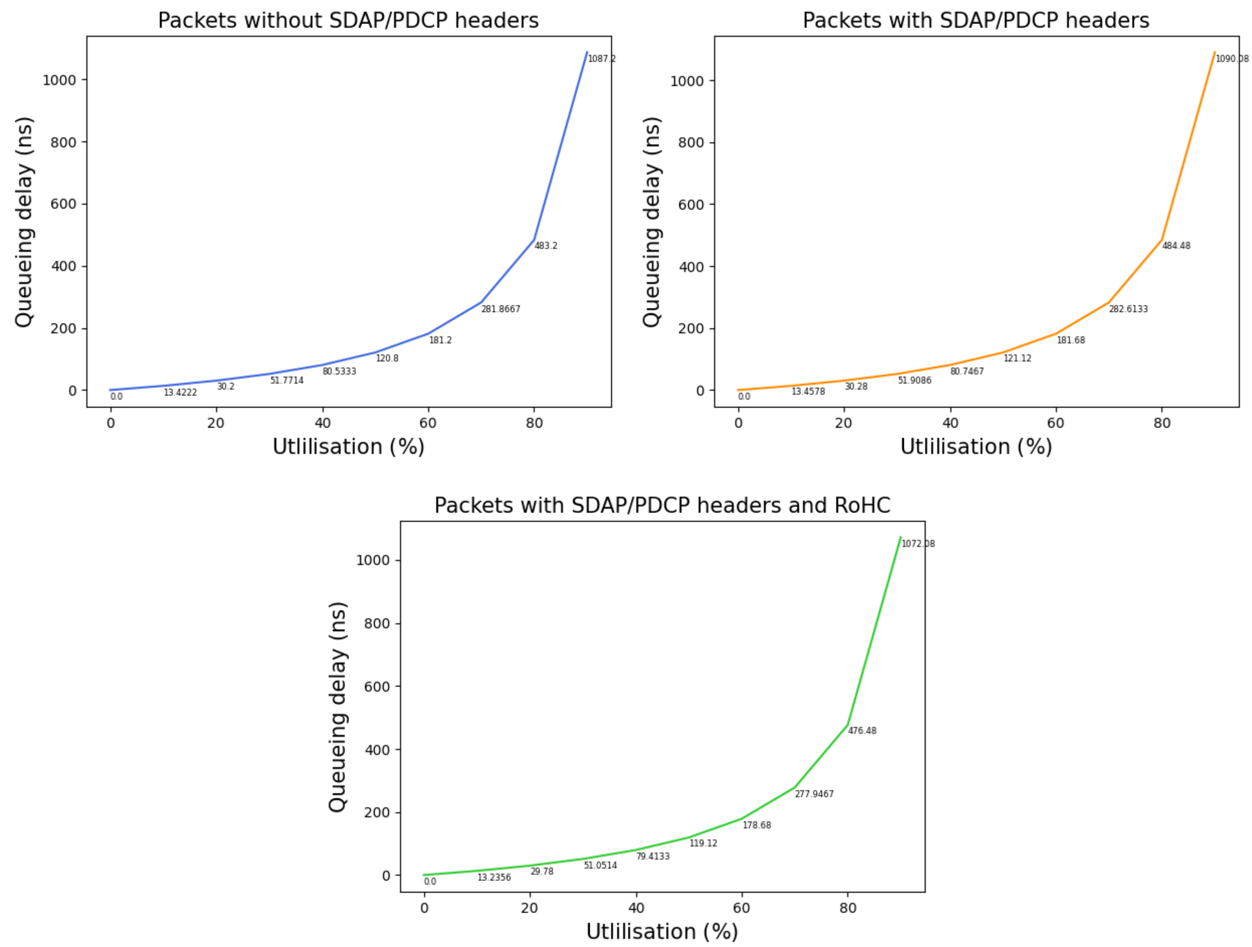


Figure 6.4: Queuing delay at 100 Gbps link capacity

Type of packet	Queuing delay in ns
Packets without SDAP/PDCP headers	281.8867
Packets with SDAP/PDCP headers	282.6133
Packets with SDAP/PDCP headers and RoHC	277.9467

Table 6.3: Queuing delay at 70 % link utilisation of 100 Gbps link capacity

Table 6.4 represents the nodal delay considered in this study, the sum of transmission delay, and queuing delay at 70 percent link capacity. Table 6.4 consolidates the nodal delay of all three types of packets under consideration at three different link capacities. When the link capacity increases by a factor 10, the packet arrival rate and the packet service rate also increase by a factor 10, meaning it can handle ten times the packet. From the table above, we recognise that, instead of using ten links with 1 Gbps link capacity, it is better to use one link with 10 Gbps link capacity. Both handle the same number of packets, but the link with 10 Gbps capacity reduces the delay by 90 per cent. Before considering a link with ten times the capacity, bottleneck and link failures should also be considered.

	Nodal delay at 1Gbps link capacity (ns)	Nodal delay at 10Gbps link capacity (ns)	Nodal delay at 100 Gbps link capacity (ns)
Packets without SDAP/PDCP headers	40266.7	4026.7	402.6876
Packets with SDAP/PDCP headers	40373.3	4037.3	403.7333
Packets with SDAP/ PDCP headers and RoHC	39706.7	3970.6	397.0667

Table 6.4: Nodal delay, sum of transmission delay, and queuing delay at 70 % link utilisation

When CU-UP is placed away from DU, there is more than one router in the midhaul routing path. According to the assumptions in [58], [59] and [60], the number of hops in the midhaul should be limited to 15-20 hops. In our thesis, we assume that 20 routers are present between the edge cloud location and the regional cloud location, which we consider as a reference point in Section 5.1.1. We multiply the delay caused by one router as per our calculation by 19, considering that one router always exists between O-CU-UP and O-DU even in scenario B of Section 5.1.1. Assuming all the existing transport links are not yet upgraded to 100 Gbps link capacity according to 5GS requirements, we consider nodal delay values at 10 Gbps link. The results are tabulated in table 6.5 below.

	Nodal delay due to single hop at 10 Gbps link capacity (μs)	Nodal delay due to 19 hops at 10 Gbps link capacity (μs)
Packets without SDAP/PDCP headers	4.0267	76.5073
Packets with SDAP/PDCP headers	4.0373	76.7087
Packets with SDAP/ PDCP headers and RoHC	3.9706	75.4414

Table 6.5: Nodal delay due to single hop and nineteen hops

From Table 6.5, we can see that, in terms of transport latency due to nodal delays, processing the CU-UP with RoHC at a centralised location is better by 1.065 μs when compared to processing it at a location closer to DU (proposed architecture). Even though the RoHC reduces the transport latency due to nodal delay, it has a trade-off with high power consumption. According to [61], the PDCP layer consumes 80 percent of the L2 layer's total power consumption, most of which is consumed by the RoHC protocol. Since RoHC contributes the majority of L2 power consumption, we can reduce power consumption in RAN environments and also improve the battery performance of UEs by omitting RoHC. Furthermore, according to [62], the PDCP layer consumes 71 per cent of the L2 layer's total execution time, out of which the RoHC protocol consumes 71 percent of the PDCP layer execution time. Since the results provided in the literature were not per packet, it is impossible to compare the gain in transport latency due to RoHC in our system against the delay caused by the execution of RoHC.

It is essential to consider the packet size when deciding to implement RoHC. RoHC plays a prominent role in applications like Voice over IP (VoIP). When packet sizes are as small as 32 bytes, the information has to be divided into small packets. Hence packets are more frequent, and so are the headers. But for a packet size of approx. 1500 bytes, more information can be sent in a packet. So, the occurrence of packets is less and so are the headers.

From the results in Table 6.5, it is clear that when RoHC is not implemented, the delay caused just by adding 4 bytes of SDAP/PDCP headers over 19 hops is 201.4 ns/packet. In other words, when the proposed architecture (CU-UP processing closer to DU) is employed, we could save around 201.4 ns/packet in delay. Hence, taking just the transport latency due to nodal delays into consideration, when RoHC is omitted, the proposed architecture is advantageous over the reference architecture.

6.2 Communication latency of GTP-U messages using IPC v/s network stack

In this section, we compare the results of simulating the communication of the GTP-U messages via UDP/IP/Eth communication protocol stack and shared memory IPC. We start the simulations by first sending 100 packets using the two mentioned communication mechanisms and, further, sending 1000 packets to record the change in behaviour of the communication mechanisms as the load increases. Note that the GTP-U header processing is not done during the simulations.

According to the state-of-the-art, the GTP-U tunnelling happens only through UDP/IP/Eth protocol stack. The transfer of GTP-U messages using the IPC mechanism will provide a communication mechanism with less delay between O-DU and O-CU-UP located on a single O-Cloud physical host machine.

The one-way communication latency between the sending and receiving process for the two communication mechanisms in discussion is plotted as a line graph in Figure 6.5. At the UDP/IP/Eth stack, the first packet experiences a high delay due to socket binding[1]. The UDP/IP/Eth mechanisms plots show a slight "peak-and-trough" trend, elaborated further. Even though some extent of packet loss can be expected in UDP communication, for transmission of 100 packets, there was 0 percent packet loss.

There is no high initial delay in the shared memory mechanism, as the receiving entity has created the memory location and is actively waiting for the packets. The communication delay in shared memory IPC is consistent throughout, with less deviation from the average delay mentioned in Table 6.6. There are no packets lost using the shared memory IPC.

As explained in Section 4.2.3, higher delay in UDP/IP/Eth compared to shared memory IPC can be justified by the time consumed for processing UDP, IP, and Eth headers in their respective layers at both sending and receiving end. Whereas in shared memory, there is no header processing required.

Furthermore, for sending the packets from sending process to receiving process using the UDP/IP network stack, the packets have to go through the physical layer consisting of the Network Interface Card (NIC)[2]/virtual Network Interface Card (vNIC)[3]. For the sending and receiving to happen on a physical machine, the packets are copied from the user memory space of the sending process to the kernel space. From the kernel space, it is mapped to the memory of the NIC of sending machine and then sent to the receiving machine via UDP/IP routing. At the receiving end, the messages are mapped from the NIC of receiving side to the kernel space and, at last, copied to the receiving process user memory space.

[1] used to know which program and particular port to send the datagrams
[2] When the communication is between two physical host machines
[3] When the communication is within the same physical host machine

As Shared memory IPC uses shared memory in user memory space, messages do not need to traverse the physical link, which contributes to the better performance of the shared memory.

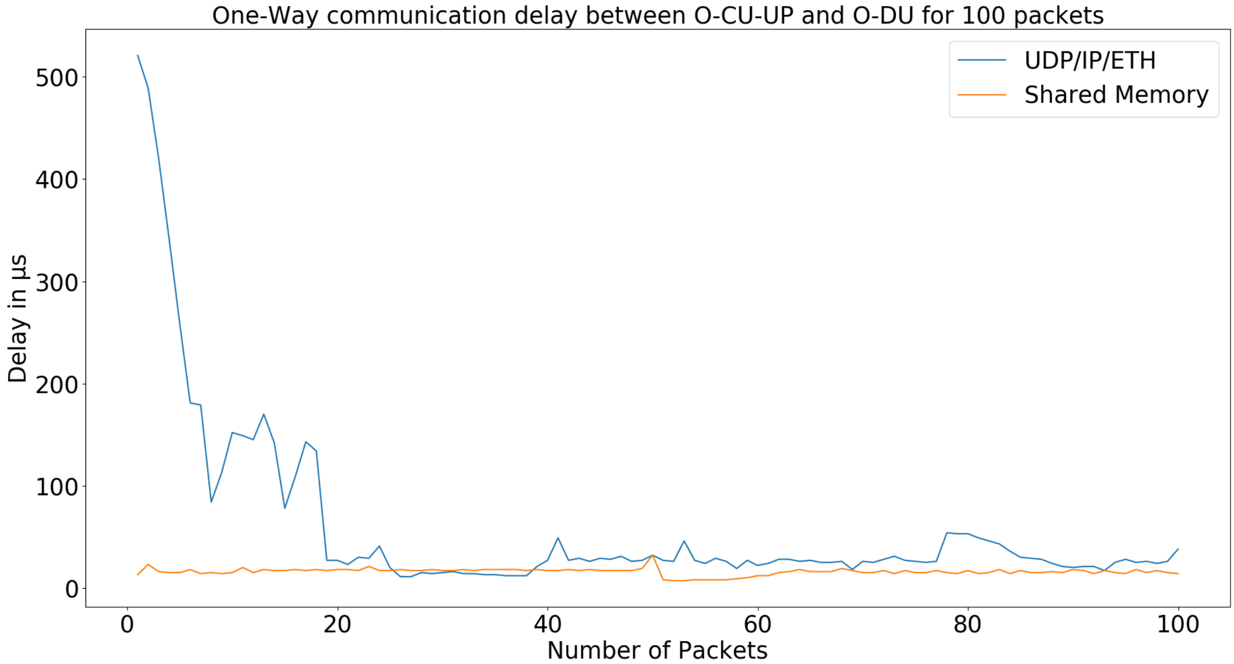


Figure 6.5: One way communication delay for 100 packets

Type of Communication	Communication delay in μs
UDP/IP/Eth communication	Avg = 59.81, Min = 11, Max = 521
Shared Memory IPC	Avg = 15.89, Min = 7, Max = 32

Table 6.6: One way communication delay description

Figure 6.5 indicates that the shared memory IPC performs better than the UDP/IP/Eth communication. Moreover, the delay description can be analysed using Table 6.6. The average delay of the UDP/IP/Eth mechanism is more significant than the Shared memory mechanism. Even though the minimum delay of both the systems are similar, the maximum delay at the UDP/IP/Eth mechanism is comparatively high.

We increase the number of packets to 1000, and the resulting one-way communication delay between the sending and receiving process for the two communication mechanisms is depicted in Figure 6.6. In UDP/IP/Eth, the "peak-and-trough" trend is much larger and visible compared to the delay results for 100 packets. The delay in UDP/IP/Eth communication mechanism started to increase to very high levels, and the peaks were caused due to lining up of packets at receiving end. Once the delay reached the peak, the fall was also accompanied by regular packet loss, as expected from the UDP process. The packet loss helped bringing down and stabilise the delay, but as seen in Figure 6.6, it did not remain at lower values for most packets. The delay continued to increase in the system until it no longer satisfied the 5GS requirements for midhaul. However, the average delay, tabulated in Table 6.7, satisfies the 5GS requirements.

On the other end, the shared memory IPC continues its remarkable performance. The delay continued to follow the same trend as the results of the delay for 100 packets. The proposed system's delay is as low as the nanoseconds range for few packets. The lower delay can also be attributed to high speed and continuous reading of messages at shared memory locations residing in user memory space.

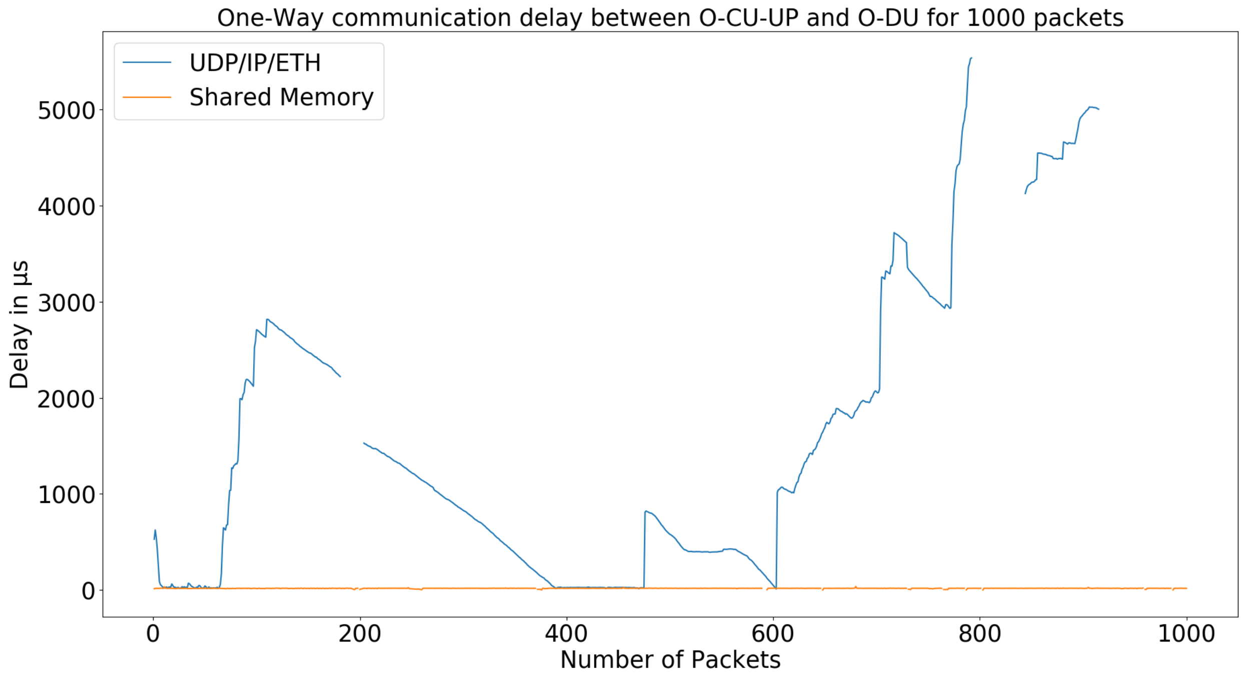


Figure 6.6: One way communication delay for 1000 packets with packet loss

Type of Communication	Communication delay in μs	Avg. Packet loss over seven simulations
UDP/IP/Eth communication	Avg = 1527.92, Min = 10, Max = 5537	21.45%
Shared Memory IPC	Avg = 16.915, Min $\leq$ 0, Max = 36	11.22%

Table 6.7: One way communication gain description

Furthermore, the delay description is tabulated in Table 6.7. In the results of the simulation with 1000 packets, communication delays were also accompanied by packet loss. The percentage of average packet loss for UDP/IP/Eth communication was higher than the shared memory IPC. Shared memory also has another problem than packet loss; Shared memory is persistent in reading messages and will continue to re-read the packets stored in the memory until the new packet shows up. Hence a packet is read multiple times until the next packet is written to the memory by the sending process. It can be avoided by using synchronisation mechanisms like semaphores. When semaphores are used, the receiving entity reads the packet and waits for the arrival of new packets before reading again. From the results discussed in this section and Table 6.7, we can deduce that the proposed user plane architecture has significant gains of 1.5ms in communication delay compared to the reference user plane architecture.

6.3 Overall architectural comparisons

- **Changes required in the GTP header**
  There are no changes made to the GTP-U tunnelling procedure in the reference architecture. Hence the existing GTP-U header is fully functional and supports the GTP-U tunnelling for the 3GPP specified 5G user plane. Instead of using the UDP port number and IP address in the proposed architecture, we use the shared memory address to communicate with the destination process. Changes must be made to the existing GTP-U header to accommodate the shared memory address. So that while processing the GTP-U header, the system will know where to send the GTP-U packets.

- **Hardware requirements**
  The reference architecture requires two physical host machines dedicated to hosting O-CU-UP and O-DU. In contrast, in the proposed system, both these NFs can be deployed on a single physical host machine, limiting the number of hardware requirements in every edge location.

- **Centralisation gains**
  As explained in Appendix B.2, one of the main advantages of centralising the RAN NFs is CoMP and statistical multiplexing gain. In the proposed architecture, the system can achieve only partial CoMP and multiplexing gain as the O-CU-UP is distributed. In the proposed architecture, even though the O-CU-UP is distributed, the O-CU-CP remains centralised. By employing the proposed architecture, we can still take advantage of the centralised control plane, such as synchronisation of cell cites and enhanced Inter-Cell Interference Control (eICIC). Regarding CU-CP, our CoMP and multiplexing gains are higher than the complete distributed architecture but lower than the reference architecture.

- **Impact of the proposed architecture on F1, E1 interfaces and F1AP, E1AP protocols**
  As explained in Section 4.1.2 and Section 4.2.6, in the 3GPP reference architecture and the proposed architecture, the communication information among much other information of DU and CU-UP is exchanged between each other via CU-CP. Hence, in the proposed user plane architecture, the shared memory address has to be exchanged between O-DU and O-CU-UP via O-CU. The shared memory address is a piece of internal OS information belonging to the physical host machine and is considered sensitive information. No process running outside the physical host machine can access the shared memory location; however, it can be considered a risk if any undesired process running on the same physical host gets access to this information. As this information is exchanged between O-DU - O-CU-UP - O-CU-CP, it is communicated over E1 and F1 interfaces using E1AP and F1AP protocols. We consider it to depend on the existing encapsulation provided by the interfaces and protocols mentioned above. Nevertheless, extra attention has to be paid to the security of the shared memory address. Also, docker's IPC permissions add to the security risk concern. When permissions are given as shareable by "Docker A" for IPC purposes, any other docker on the same physical host machine can access any shared memory if they have the appropriate name of "Docker A" and the shared memory address.

- **Comparisons of handover during mobility**
  In the reference architecture, Suppose a UE moves from one gNodeB-DU to another gNodeB-DU, and both gNodeB-DU are under the same gNodeB-CU-UP; the gNodeB-CU-UP need not be changed. However, in the proposed architecture, O-DU and O-CU-

UP should be at the same location, and both the NFs must be deployed on the same physical host machine. Hence both O-DU and O-CU-UP have to be handed over to the new location during mobility, which results in signalling overhead.

Chapter 7

Conclusions and future work

In this chapter, Section 7.1 describes the conclusions drawn from this thesis study and in Section 7.2 the further research that can be conducted based on the outcome of this thesis study is discussed.

7.1 Conclusions

In this research work, we present an enhanced architecture for optimising the routing of the user plane of the 5GS. The proposed architecture deploys DU and CU-UP on a single physical host machine at a distributed location, intending to optimise the user plane. At the same time, CU-CP remains at a centralised location.

Firstly, the proposed user plane architecture for 5GS optimises the user plane latency and reduces packet loss. The midhaul communication latency contributed by the proposed architecture falls well below the maximum latency allowed at midhaul by 3GPP. Furthermore, on average, 1.5 ms/packet delay can be saved by employing the proposed architecture instead of the reference architecture.

When RoHC is omitted, performing the SDAP/PDCP-U processing at a distributed location according to the proposed architecture provides a gain at the nanoseconds scale compared to the reference architecture. However, the throughput between the regional and edge cloud locations will increase by not having to carry the SDAP/PDCP headers.

Furthermore, the proposed architecture, along with the proposed eF1-U GTP tunnelling, shows promising results and can be concluded as a new alternative architecture to deploy an optimised user plane for 5GS.

Secondly, by observations, when compared to the signalling involved in the 3GPP defined PDU session establishment procedure, we conclude that the signalling involved in the PDU session establishment procedure for the proposed architecture becomes slightly complicated and consumes extra time because of the involvement of near-RT RIC in the procedure. It indicates a signalling overhead and latency in the control plane. However, with the smart and near real-time performance of $\geq$10 ms, but <1 s (see Appendix C.1) by the near-RT RIC, the delay is assumed to be negligible.

Furthermore, the handover in the proposed architecture is complicated compared to the reference architecture. Hence it is recommended that a connection with the target O-DU and O-CU-UP is established before the connection with the source O-DU and O-CU-UP is broken.

Thirdly, regarding RoHC and CoMP, we conclude that when the application's packet size is large, the power and time saved by the proposed architecture are more advantageous than the statistical multiplexing gain obtained by the centralised CPU-intensive RoHC of the reference architecture. Hence RoHC can be omitted.

Considering both transport latency results from Section 6.1 and communication latency results from Section 6.2, the proposed system can be used even for deployments in which RoHC is essential as well. When comparing proposed architecture with reference architecture (RoHC enabled), the proposed architecture falls behind and underperforms[1] by 1.065 μs of transport latency. Furthermore, the proposed architecture has a 1.5 ms communication latency gain compared to the reference architecture. In the proposed architecture, the gain obtained in communication latency is significant compared to the underperformance in transport latency. Hence, if RoHC is a crucial requirement for MNOs, the proposed architecture can be employed by considering performing RoHC at the edge location where O-CU-UP is deployed along with O-DU. By doing this, the MNOs can take advantage of both communication latency gains in midhaul by proposed architecture and reduced bandwidth requirement for fronthaul and over NR-air interface by RoHC.

7.2 Future work

Based on the results and conclusions of this study, we suggest the following as topics for future work.

- **Power consumption analysis due to distributed nature of the proposed architecture**
  We could produce good results in optimising the user plane regarding latency. The distributed nature of the user plane in the presented architecture will impact power consumption. Performance analysis of power consumed by the proposed architecture will be an exciting study. Also, the research can include power consumption analysis on RoHC for the proposed packet sizes and the impact of performing RoHC at the distributed location for applications in which RoHC is critical.

- **Employing the proposed architecture for Mobile Edge Computing (MEC) applications**
  As the proposed architecture proposes to deploy O-CU-UP at edge location, research can be conducted on deploying MEC applications using the proposed architecture. Future work can also deploy the UPF at the edge location and employ the proposed enhanced GTP-U tunnel between UPF and gNodeB-CU-UP.

- **Analysis of handover due to mobility**
  As mentioned in the results and conclusions, handover due to mobility in the proposed architecture is complicated, but the complication scale is not discussed. An in-depth study of handover due to mobility in the proposed architecture, discussing signal flow diagrams and the impact of the handover on the control plane, will constitute good research. The possibility of handover involving fallback to generic PDU session, where CU-UP and DU are located at different host machines, can be inspected.

- **Simulation of O-CU-UP and O-DU with proposed architecture**
  In this study, we simulate part of the communication protocol stack between O-CU-UP and O-DU and find the communication latency. Simulating the proposed architecture

[1] In terms of transport latency, the proposed architecture underperforms when compared to reference architecture with RoHC processing. See table 6.4

with L2 protocols of the NR protocol stack will provide more insight into the processing time consumed by O-CU-UP and O-DU. Considering the time frame before taking this approach is crucial, as it requires implementing five NR protocols (SDAP, PDCP, RLC, MAC, and High-PHY).

Bibliography

[1] "ETSI - Mobile Technologies - 5g, 5g Specs | Future Technology." [Online]. Available: https://www.etsi.org/technologies/5g

[2] "Past and present reports - Mobility Report - Ericsson." [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/reports

[3] 5G Americas White Paper, "3GPP releases 16 & 17 & Beyond," 2021. [Online]. Available: https://www.5gamericas.org/wp-content/uploads/2021/01/InDesign-3GPP-Rel-16-17-2021.pdf

[4] 5G Americas White Paper., "© Copyright 2017 5G Americas- 5G Services and Use Cases Nov 2017 1," pp. 1–52, 2017.

[5] ITU-R, "M2410 - Minimum requirements related to technical performance for IMT-2020 radio interface(s)," *Working Party 5D*, vol. November, no. Report ITU-R M.2410-0, pp. 1–11, 2017.

[6] "Control and User Plane Separation of EPC nodes (CUPS)." [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/1882-cups

[7] "5G NR gNB Logical Architecture and It's Functional Split Options - Techplayon." [Online]. Available: http://www.techplayon.com/5g-nr-gnb-logical-architecture-functional-split-options/

[8] O-RAN White Paper, "O-RAN Use Cases and Deployment Scenarios Towards Open and Smart RAN," no. February, 2020.

[9] M. Chiosi, S. B. Wright, D. Clarke, P. Willis CableLabs, C. Donley, L. Johnson CenturyLink, M. Bugenhagen, J. Feger, W. Khan, C. Cui, H. Deng, and C. Chen, "Network Functions Virtualisation (NFV) Network Operator Perspectives on Industry Progress."

[10] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, "NFV and SDN-Key technology enablers for 5G networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2468–2478, 11 2017.

[11] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Network function virtualization in 5G," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 84–91, 2016.

[12] C. N. Tadros, M. R. Rizk, and B. M. Mokhtar, "Software Defined Network-Based Management for Enhanced 5G Network Services," *IEEE Access*, vol. 8, pp. 53 997–54 008, 2020.

[13] V. G. Nguyen, T. X. Do, and Y. H. Kim, "SDN and Virtualization-Based LTE Mobile Network Architectures: A Comprehensive Survey," *Wireless Personal Communications*, vol. 86, no. 3, pp. 1401–1438, 2 2016. [Online]. Available: https://link.springer.com/article/10.1007/s11277-015-2997-7

[14] "Ryu SDN Framework." [Online]. Available: https://ryu-sdn.org/

[15] "OpenFlow - Open Networking Foundation." [Online]. Available: https://opennetworking.org/sdn-resources/customer-case-studies/openflow/

[16] 5G PPP Architecture Working Group, "View on 5G Architecture," *Version 3.0, June 2019*, no. June, pp. 21–470, 2019.

[17] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for Mobile Networks — A Technology Overview," vol. 17, no. 1, pp. 405–426, 2015.

[18] P. Arnold, N. Bayer, J. Belschner, and G. Zimmermann, "5G Radio Access Network Architecture Based on Flexible Functional Control / User Plane Splits," pp. 0–4, 2017.

[19] 3GPP TR 21.915, "Release 15 Description; Summary of Rel-15 Work Items," *Release 15, 2019*, no. September, 2019.

[20] 3GPP TR 21.916, "Release 16 Description;Summary of Rel-16 Work Items," *Release 16, 2021*, no. September, 2021.

[21] N. M. Akshatha, P. Jha, and A. Karandikar, "A Centralized SDN Architecture for the 5G Cellular Network," *IEEE 5G World Forum, 5GWF 2018 - Conference Proceedings*, pp. 147–152, 2018.

[22] A. Abdulghaffar, A. Mahmoud, M. Abu-Amara, and T. Sheltami, "Modeling and Evaluation of Software Defined Networking Based 5G Core Network Architecture," *IEEE Access*, vol. 9, pp. 10 179–10 198, 2021.

[23] L. Du, N. Zheng, H. Zhou, J. Chen, T. Yu, X. Liu, Y. Liu, Z. Zhao, X. Qian, J. Chi, Z. Chen, and G. Liu, "C/U Split Multi-Connectivity in the Next Generation New Radio System," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, 2017, pp. 1–5.

[24] 3GPP TS 23.501, "System architecture for the 5G System (5GS); Stage 2," *Release 16, 2020*, no. December, 2020.

[25] 3GPP TR 38.801, "Study on new radio access technology: Radio access architecture and interfaces," *Release 14, 2017*, no. June, 2017.

[26] Y. Xiao, J. Zhang, and Y. Ji, "Can Fine-Grained Functional Split Benefit to the Converged Optical-Wireless Access Networks in 5G and Beyond?" *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1774–1787, 2020.

[27] L. M. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5G mobile crosshaul networks," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 1, pp. 146–172, 2019.

[28] 3GPP TS 38.401, "NG-RAN; Architecture description," *Release 16, 2021*, no. January, 2021.

[29] G. Brown, "New Transport Network Architectures for 5G RAN," *Fujitsu*, 2018. [Online]. Available: www.fujitsu.com/us/Images/New-Transport-Network-Architectures-for-5G-RAN.pdf

[30] F. Rinaldi, A. Raschellà, and S. Pizzi, "5G NR system design: a concise survey of key features and capabilities," *Wireless Networks*, pp. 1–16, 10 2021. [Online]. Available: https://link-springer-com.tudelft.idm.oclc.org/article/10.1007/s11276-021-02811-y

[31] 3GPP TR 38.806, "Study of separation of NR Control Plane (CP) and User Plane (UP) for split option 2," *Release 15, 2017*, no. December, 2017.

[32] P. P. Peter¨ohlén, B. Skubic, A. Rostami, M. Fiorani, P. Monti, Z. Ghebretensaé, J. Mårtensson, K. Wang, and L. Wosinska, "Data Plane and Control Architectures for 5G Transport Networks," *Journal of Lightwave Technology, Vol. 34, Issue 6, pp. 1501-1508*, vol. 34, no. 6, pp. 1501–1508, 3 2016. [Online]. Available: https://www-osapublishing-org.tudelft.idm.oclc.org/abstract.cfm?uri=jlt-34-6-1501https://www-osapublishing-org.tudelft.idm.oclc.org/jlt/abstract.cfm?uri=jlt-34-6-1501

[33] N. Hoque Essing, K. Westcott, S. Nesargi, J. Loucks, and C. Wigginton, "Next-gen radio access networks | Deloitte Insights." [Online]. Available: https://www2.deloitte.com/xe/en/insights/industry/technology/technology-media-and-telecom-predictions/2021/radio-access-networks.html

[34] O-RAN White Paper, "O-RAN: Towards an Open and Smart RAN," no. October, 2018.

[35] "About O-RAN ALLIANCE — O-RAN ALLIANCE." [Online]. Available: https://www.o-ran.org/about

[36] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead," *Computer Networks*, vol. 182, no. May, p. 107516, 2020. [Online]. Available: https://doi.org/10.1016/j.comnet.2020.107516

[37] O-RAN Technical Specification, "O-RAN.WG1 O-RAN Architecture Description," *v03.00*, pp. 1–32, 2021.

[38] "Open RAN 101–Role of RAN Intelligent Controller: Why, what, when, how?" [Online]. Available: https://www.rcrwireless.com/20200730/opinion/readerforum/open-ran-101-role-of-ran-intelligent-controller-why-what-when-how-reader-forum

[39] O-RAN Technical Specification, "O-RAN WG 3 Near-Real-time RAN Intelligent Controller Architecture & E2 General Aspects and Principles," *v01.01*, pp. 1–38, 2020.

[40] O-RAN Technical Specification., "O-RAN WG 3, Near-Real-time RAN Intelligent Controller, E2 Application Protocol (E2AP)," *v01.01*, pp. 1–83, 2020.

[41] 3GPP TS 29.281, "General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)," *Release 17, 2022*, no. March, 2022.

[42] 3GPP TS 38.413, "NG-RAN; NG Application Protocol (NGAP)," *Release 16, 2021*, no. July, 2021.

[43] 3GPP TS 38.463, "NG-RAN; E1 Application Protocol (E1AP)," *Release 16, 2021*, no. April, 2021.

[44] 3GPP TS 38.473, "NG-RAN; F1 application protocol (F1AP)," *Release 16, 2021*, no. April, 2021.

[45] 3GPP TS 29.244, "Interface between the Control Plane and the User Plane Nodes: Stage 3," *Release 16, 2022*, no. June, 2022.

[46] S. Ahmadi, *5G NR: Architecture, technology, implementation, and operation of 3GPP new radio standards*. Academic Press, 2019.

[47] R. R. Stewart, "Stream Control Transmission Protocol," RFC 4960, Sep. 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc4960

[48] "Linux Containers." [Online]. Available: https://linuxcontainers.org/

[49] "Home - Docker." [Online]. Available: https://www.docker.com/

[50] L.-E. Jonsson, K. Sandlund, and G. Pelletier, "The Robust Header Compression (RoHC) Framework," RFC 5795, Mar. 2010. [Online]. Available: https://www.rfc-editor.org/info/rfc5795

[51] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach (6th Edition)*, 6th ed. Pearson, 2012.

[52] P. Van Mieghem, *Performance analysis of complex networks and systems*. Cambridge University Press, 2014.

[53] D. Medhi and K. Ramasamy, "Chapter 7 - ip traffic engineering," in *Network Routing (Second Edition)*, second edition ed., ser. The Morgan Kaufmann Series in Networking, D. Medhi and K. Ramasamy, Eds. Boston: Morgan Kaufmann, 2018, pp. 214–258. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128007372000090

[54] A. Venkataraman and K. Kumar Jagadeesha, "Evaluation of Inter-Process Communication Mechanisms."

[55] "mmap(2) - Linux manual page." [Online]. Available: https://man7.org/linux/man-pages/man2/munmap.2.html

[56] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot, "Measurement and analysis of single-hop delay on an IP backbone network," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 908–921, 8 2003.

[57] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, F. Tobagi, and C. Diot, "Analysis of measured single-hop delay from an operational backbone network," Tech. Rep., 2002.

[58] Xilinx, "Fronthaul, midhaul and backhaul."

[59] B. M. Khorsandi and C. Raffaelli, "BBU location algorithms for survivable 5G C-RAN over WDM," *Computer Networks*, vol. 144, pp. 53–63, 10 2018.

[60] B. M. Khorsandi, F. Tonini, and C. Raffaelli, "Centralized vs. distributed algorithms for resilient 5G access networks," *Photonic Network Communications*, vol. 37, no. 3, pp. 376–387, 6 2019. [Online]. Available: https://link.springer.com/article/10.1007/s11107-018-00819-7

[61] P. K. Gupta, R. V. Rajakumar, and C. S. Kumar, "Energy cost analysis of data plane and control plane protocols for 3GPP-LTE," *2012 National Conference on Communications, NCC 2012*, 2012.

[62] D. Szczesny, A. Showk, S. Hessel, A. Bilgic, U. Hildebrand, and V. Frascolla, "Performance analysis of LTE protocol processing on an ARM based mobile platform," *2009 International Symposium on System-on-Chip - Proceedings, SoC 2009*, pp. 56–63, 2009.

[63] Y. Lin, L. Shao, Z. Zhu, Q. Wang, and R. K. Sabhikhi, "Wireless network cloud: Architecture and system requirements," *IBM Journal of Research and Development*, vol. 54, no. 1, 2010.

[64] O-RAN Technical Report, "O-RAN WG2 Non-RT RIC: Functional Architecture," *v01.00*, pp. 1–42, 2021.

Appendix A

5GS QoS model

The 5GS QoS model consists of QoS Flows, the smallest granularity of a traffic flow in 5G at which QoS and pricing can be implemented. The QoS flow can be pre-configured or created during the Protocol Data Unit (PDU) session establishment/modification procedure. The QoS Flow Identifier (QFI) is a unique identifier for each QoS flow, and all packets belonging to a specific QoS Flow in a PDU session are treated equally. SMF is responsible for establishing the QoS Flow and managing them by assigning QFI and QoS profiles to related QoS Flows.

The QoS model offers two different profiles for QoS flows guaranteed flow bit rate (GBR QoS Flows) and non-guaranteed flow bit rate (Non-GBR QoS Flows). The QoS profile comprises parameters common to both GBR and Non-GBR, such as 5G QoS Identifier (5QI)[1] and Allocation and Retention Priority (ARP) information which defines priority level. Additionally, the GBR QoS Flow profile includes Guaranteed Flow Bit Rate (GFBR) for UL and DL and Maximum Flow Bit Rate (MFBR) for UL and DL.

- **Default QoS Flow:**

  In the 5G QoS model, a Non-GBR QoS flow called default QoS Flow with default QoS rules and without any Packet Filter will be established and maintained for the entire duration of the PDU session. When a UL or DL traffic is not matching any packet filters of a UE's PDU session, then default QoS Flow will be utilised to forward this traffic.

[1] 5QI is a scalar that serves as a reference point to 5G QoS characteristics such as Packet Delay Budget, Packet Error Rate and Priority Level

Appendix B

RAN deployment options

B.1 Distributed RAN (D-RAN)

In D-RAN architecture, the BBUs are de-centralised and placed at the base of the cell site tower. The RRH hardware is placed on the tower along with the antenna. The dedicated BBU is placed at the base of the cell site tower or a few meters away from it. Figure B.1 is an illustration of D-RAN.

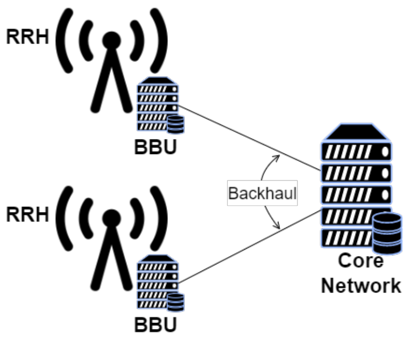


Figure B.1: Distributed RAN, in which BBU processing happens distributively at the cite location

Common Public Radio Interface (CPRI) interface connects the RRH to the BBU, forming the fronthaul. The D-RAN architecture is designed to meet the peak requirements in that region. As the peak utilisation periods are not constant, most of the processing power is underutilised. Hence the resources not being scalable according to the requirement is a main drawback of the D-RAN architecture. Resources such as power, cooling, and maintenance must be explicitly provided to the BBUs at each cell site. The D-RAN cannot support collaborating techniques such as the Coordinated Multi-points (CoMP). Also, signalling in the backhaul causes a delay of 4-15ms [16].

B.2 C-RAN

With the vision of optimising resources consumed by BBUs, a design was introduced in the year 2010 by [63], which has evolved over the years and is now called C-RAN. "C" denotes Cloud and Centralised, two different but interrelated aspects of C-RAN. Figure B.2 represents the approach of centralised RAN and figure B.3 illustrates the cloud RAN.

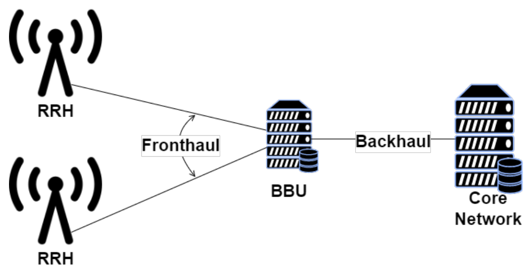


Figure B.2: Centralised RAN, in which the BBU processing happens at a centralised location

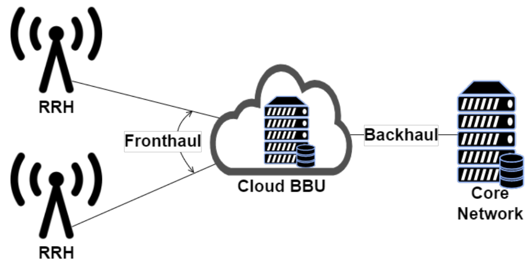


Figure B.3: Cloud RAN, in which the BBU processing happens at a cloud location

C-RAN technology, keeping the concept of pooling as its essence, aims to create a Centralised-BBU by moving multiple BBUs of various base stations into a cloud infrastructure to achieve statistical multiplexing gain [17]. Also, The RRH is easy to implement and maintain. Along with optimising computing resources and energy consumption, C-RAN has added the advantages of being scalable and performing fast handovers. In other words, C-RAN reduces CAPEX/OPEX for the MNOs. Nevertheless, C-RAN implementation has a significant disadvantage of creating high fronthaul capacity requirements [27]. For instance, under the constant development of the wireless spectrum and antenna size, Fronthaul suffers from the high bandwidth requirement to transmit raw in-phase and quadrature (I/Q) samples. By intense centralisation of BBUs into faraway processing units, the transport bandwidth requirement will be 100-fold or perhaps 1000-fold greater than that necessary for 4G. Also, as explained in Chapter 1, the 5G capacity requirements are rising. Hence, C-RAN must be improved to manage the 5GS capacity requirement and the high fronthaul capacity demand created by the C-RAN implementation.

Appendix C

O-RAN

C.1 Control loops

The RIC optimises RAN through control loops such as Request/Response procedures, feedback loops and control actions. The three main control loops supported by the O-RAN architecture are the O-DU Scheduler control loop (<10 ms), near-RT RIC control loop (≥ 10 ms, but <1 s), and Non-RT RIC control loop (≥ 1 s). The role of the control loops depends on the O-RAN entity controlling them. Control loops also depend on other O-RAN entities and functions while updating data to the controlling entity. These loops run simultaneously and communicate with each other only if required by the use cases [37].

C.2 Non-Real Time RIC

Non-RT RIC is a logical function that can control and optimise RAN resources in non-real-time, manages AI/ML workflow, and provides policy-based guidance for applications in near-RT RIC [64]. Also, the Non-RT RIC performs model training and RAN analytics for the near-RT RIC applications. Non-RT RIC platform, as the name suggests, works non-real-time with >1s latency, and the applications hosted on Non-RT RIC are called rApps. Real-time control functions and AI/ML models that have been trained in the non-RT RIC are then transferred to the near RT-RIC via the A1 interface to be executed at the runtime. Multiple near-RT RICs can be connected to a single Non-RT RIC. The Non-RT RIC self-configures all the new radio units, decreasing manual intervention, which is critical for 5G deployments.

C.3 near-RT RIC services

The four near-RT RIC services are elaborated below:

- **REPORT:** Using the Subscription procedure, an xAPP in the near-RT RIC requests the E2 Node to send a REPORT message with requested information whenever a defined event is triggered or at specified intervals. The E2 Node sends a Report message including the requested information using the Indication procedure whenever a specific event triggers or at specified intervals (using a timer function).
- **INSERT:** Using the RIC subscription procedure, the xAPP subscribes to a function of the E2 Node (e.g., UE indicating handover). According to this subscription, whenever the

specified RIC Subscription Event Trigger occurs, the E2 Node has to suspend the ongoing procedure and send an Indication message along with information to identify the paused procedure to the near-RT RIC. The near-RT RIC also mentions the wait timer in the RIC subscription message.

When the E2 Node detects an event trigger, it sends an INSERT message using the Indication procedure and sets a timer waiting for further directions from the near-RT RIC. If there is no response from the near-RT RIC within the wait timer, the paused E2 procedure either resumes or halts.

- **CONTROL:** A CONTROL REQUEST is sent from the near-RT RIC to E2 Node to resume or want to begin a particular functionality in an E2 node. The Control service is initiated when the near-RT RIC senses an event trigger. The cause of the event trigger can either be a previous RIC Indication message from an E2 Node or caused autonomously due to RIC internal event. Once the event trigger occurs, the near-RT RIC decides what action has to be performed and sends it over to the E2 node using a RIC control message; it may include information related to the previously suspended procedure. Once the Control message is received, the E2 node resumes the previously paused procedure or begins a new procedure as instructed by near-RT RIC. Also, if necessary, near-RT RIC expects an acknowledgement from the E2 node.
- **POLICY:** Using the subscription procedure, the near-RT RIC instructs the E2 Node to carry out a particular POLICY following each occurrence of a specified RIC Subscription event Trigger during the operation of the E2 Node. Whenever a trigger event is detected, the ongoing procedure is altered by the E2 Node according to the instructions provided in the Policy description