UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Quality of Experience using Machine Learning and Improved Orchestration in 5G Networks and Beyond

Diploma Thesis

Alexandros Stoltidis

Supervisor: Athanasios Korakis

Volos Greece September 2023

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Quality of Experience using Machine Learning and Improved Orchestration in 5G Networks and Beyond

Diploma Thesis

Alexandros Stoltidis

Supervisor: Athanasios Korakis

Volos Greece September 2023

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ποιότητα Εμπειρίας με Μηχανική Μάθηση και Βελτιωμένη Ενορχήστρωση σε Κυψελωτά Δίκτυα μετά της 5ης Γενιάς

Διπλωματική Εργασία

Αλέξανδρος Στολτίδης

Επιβλέπων: Αθανάσιος Κοράκης

Βόλος Ελλάδα Σεπτέμβριος 2023

Approved by the Examination Committee:

Supervisor **Athanasios Korakis**
Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Antonios Argyriou**
Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Parisis Flegkas**
Assistant Professor, Department of Electrical and Computer Engineering, University of Thessaly

Acknowledgements

This thesis is a message of thanks to everyone who helped me complete my studies throughout this tough but interesting five-year academic journey. I would like to express my profound gratitude to Professor Athanasios Korakis, my outstanding thesis advisor, for his tremendous advice and inspiration during the course of my studies. I am grateful for the opportunity to join NITLAB under his leadership, which exposed me to a world-class research environment. Additionally, Kostas Choumas, a Senior Researcher, a Postdoctoral Associate, and my mentor, made a significant contribution to the development of my academic career and earned my sincere appreciation. Beyond his support in helping me navigate this project by sharing his knowledge, expertise, and constant encouragement, he also offered invaluable advice that aligns with my long-term academic goals. I also want to express my gratitude to my parents, sister, and friends for their everlasting support, love, and encouragement throughout this journey. Finally, I'd like to thank Associate Professor Antonios Argyriou and Assistant Professor Parisis Flegkas for their contributions to my thesis review committee.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Alexandros Stoltidis

Diploma Thesis

Quality of Experience using Machine Learning and Improved Orchestration in 5G Networks and Beyond

Alexandros Stoltidis

Abstract

Since the invention of computer networks and as technological innovation advances, applications become more and more demanding. Wireless cellular networks represent a recent breakthrough in networking, facilitating communication between devices over the air and eliminating the need for physical cables. Devices need enhanced performance, including increased network throughput and reduced latency, to meet the requirements of modern high-performance applications.

This thesis explores and contributes to critical research areas within cellular networks by analyzing existing performance factors and proposing groundbreaking optimization strategies. The core of this thesis focuses on two distinct optimization areas within cellular networks.

The first area involves building and enhancing upon previous research on network queue optimization methods known as Active Queue Management. It addresses the optimization of latency-sensitive algorithms within high-latency environments. In this context, Artificial Intelligence and Machine Learning algorithms play a significant role in performance optimization.

The second area falls within network orchestration research and focuses on controller placement in cellular networks. Controller placement is an optimization problem extensively studied in Software-Defined Networks that we extended to cellular networks. In this area, we model cellular networks using linear programming as a set of linear constraints and objectives. We adapt the linear objective based on the specific tasks required by the network, and we assess the optimal controller placement effectiveness through evaluation within a realistic simulation environment.

Keywords:

5G, Disaggregated, QoS, RIC, AQM, Bufferbloat, AI/ML, Controller Placement, Linear Programming, OpenAirInterface5G, NS-3

Διπλωματική Εργασία

Ποιότητα Εμπειρίας με Μηχανική Μάθηση και Βελτιωμένη Ενορχήστρωση σε Κυψελωτά Δίκτυα μετά της 5ης Γενιάς

Αλέξανδρος Στολτίδης

Περίληψη

Από την εφεύρεση των υπολογιστικών δικτύων και καθώς η τεχνολογική καινοτομία προχωρά, οι εφαρμογές γίνονται όλο και πιο απαιτητικές. Οι ασύρματα κυψελωτά δίκτυα αντιπροσωπεύουν μια πρόσφατη καινοτομία στα δίκτυα, διευκολύνοντας την επικοινωνία μεταξύ συσκευών μέσω του αέρα και εξαλείφοντας την ανάγκη για φυσικά καλώδια. Οι συσκευές χρειάζονται βελτιωμένη απόδοση, συμπεριλαμβανομένης της αυξημένης ταχύτητας δικτύου και της μείωσης της καθυστέρησης, προκειμένου να ικανοποιήσουν τις απαιτήσεις των σύγχρονων υψηλής απόδοσης εφαρμογών.

Αυτή η διατριβή εξετάζει και συμβάλλει σε κρίσιμους τομείς έρευνας στα κυψελωτά δίκτυα, αναλύοντας τους υπάρχοντες παράγοντες απόδοσης και προτείνοντας καινοτόμες στρατηγικές βελτιστοποίησης. Το κύριο μέρος αυτής της διατριβής επικεντρώνεται σε δύο διακριτικούς τομείς βελτιστοποίησης στα κυψελωτά δίκτυα.

Ο πρώτος τομέας περιλαμβάνει τη δημιουργία και την ενίσχυση πάνω στην προηγούμενη έρευνα σε μεθόδους βελτιστοποίησης ουράς δικτύου που είναι γνωστές ως Διαχείριση Ενεργής Ουράς. Ασχολείται με την βελτιστοποίηση αλγορίθμων που είναι ευαίσθητοι στην καθυστέρηση σε υψηλές καθυστερήσεις περιβάλλοντα. Σε αυτό το πλαίσιο, οι αλγόριθμοι Τεχνητής Νοημοσύνης και Μηχανικής Μάθησης παίζουν σημαντικό ρόλο στη βελτιστοποίηση της απόδοσης.

Ο δεύτερος τομέας ανήκει στην έρευνα ορχηστρώσεων δικτύου και επικεντρώνεται στην τοποθέτηση ελεγκτών σε κυψελωτά δίκτυα. Η τοποθέτηση του ελεγκτή είναι ένα πρόβλημα βελτιστοποίησης που έχει εκτενώς μελετηθεί στα Δίκτυα Ορισμένου Λογισμικού και το επεκτείνουμε στα κυψελωτά δίκτυα. Σε αυτόν τον τομέα, μοντελοποιούμε τα κυψελωτά δίκτυα χρησιμοποιώντας γραμμικό προγραμματισμό ως ένα σύνολο γραμμικών περιορισμών και στόχων. Προσαρμόζουμε το γραμμικό αντικείμενο βάσει των συγκεκριμένων εργασιών που απαιτούνται από το δίκτυο, και αξιολογούμε την αποτελεσματικότητα της βέλτιστης τοπο-

θέτησης του ελεγκτή μέσω αξιολόγησης σε ένα ρεαλιστικό περιβάλλον προσομοίωσης.

Λέξεις-κλειδιά:

5G, Disaggregated, QoS, RIC, AQM, Bufferbloat, AI/ML, Controller Placement, Linear Programming, OpenAirInterface5G, NS-3

Table of contents

List of figures

Abbreviations

UE	User Equipment
AQM	Active Queue Management
AI	Artificial Intelligence
ML	Machine Learning
VNF	Virtual Network Function
SDN	Software Defined Networking
NFV	Network Functions virtualization
1G	Fist Generation
FDMA	Frequency Division Multiple Access
2G	Second Generation
TDMA	Time Division Multiple Access
CDMA	Code Division Multiple Access
3GPP	Third Generation Partnership Project
3G	Third Generation
4G	Fourth Generation
IP	Internet Protocol
VoIP	Voice Over IP
OFDM	Orthogonal Frequency Division Multiplexing
MIMO	Multiple-Input Multiple-Output
LTE	Long-Term Evolution
WiMAX	Worldwide Interoperability for Microwave Access
eNB	Enhanced Node B
NB	Node B
5G	Fifth Generation
IoT	Internet of Things

NR	New Radio
gNB	Next Generation Node B
RAN	Radio Access Network
CU	Centralized Unit
DU	Distributed Unit
RU	Radio Unit
SDAP	Service Data Adaptation Protocol
PDCP	Packet Data Convergence Protocol
RRC	Radio Resource Control
RLC	Radio Link Control
MAC	Medium Access Control
PHY	Physical Layer
CP	Control Plane
UP	User Plane
OSI	Open Systems Interconnection
SMO	Service Management and Orchestration
O-RAN	Open RAN
FCAPS	Fault, Configuration, Accounting, Performance, and Security
NF	Network Function
CRUD	Create, Read, Update, Delete
RIC	RAN intelligent Controller
QoE	Quality of Experience
Near-RT RIC	Near-Real-Time RIC
Non-RT RIC	Non-Real-Time RIC
E2AP	E2 Application Protocol
E2SM	E2 Service Model
CN	Core Network
5GC	5G CN
MME	Mobility Management Entity
AMF	Access and Mobility Management Function
SMF	Session Management Function
UDM	Unified Data Management

AAF	Authorization Function
AUSF	Authentication Server Function
SGW	Serving Gateway
PDN	Packet Data Network
PGW	PDN Gateway
PCF	Policy Control Function
NEF	Network Exposure Function
API	Application Programming Interface
NSSF	Network Slice Selection Function
NRF	Network Repository Function
AF	Application Function
RNN	Recurrent Neural Network
BiNN	Bidirectional Neural Network
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
NITLAB	Network Implementation Testbed Laboratory
CERTH	Centre for Research and Technology Hellas
ITI	Information Technologies Institute
OAI	Open Air Interface
COTS	Commercial Off-The-Shelf
KPM	Key Performance Metrics
NS-3	Network Simulator 3
CI/CD	Continuous Integration/Continuous Deployment
eMBB	Enhanced Mobile Broadband
URLLC	Ultra-Reliable Low-Latency Communications
mMTC	Massive Machine-Type Communications
RED	Random Early Detection
CoDel	Controlled Delay
FQ-CoDel	Fair Queuing CoDel
PIE	Proportional Integral Controller Enhanced
SFQ	Stochastic Fair Queuing
DRQL	Dynamic RLC Queue Limit

UPF	User Plane Function
QoS	Quality of Service
QFI	QoS Flow Identifier
PDR	Packet Detection Rule
DRB	Data Radio Bearer
UDP	User Datagram Protocol
ICMP	Internet Control Message Protocol
PDU	Packet Data Unit
RTT	Round-Trip Time
SLA	Service-Level Agreement
NSSI	Network Slice Subnet Instance
Grid-of-Beams	GoB
BF	Beamforming
bMRO	Beam-based Mobility Robustness Optimization
mmWave	Millimeter Wave
TTT	Time-to-Trigger
CIO	Cell Individual Offset
mMIMO	Massive MIMO
SS	Synchronization Signal
CSI-RS	Channel Status Information Reference Signal
TRS	Tracking Reference Signal
DMRS	Demodulation Reference Signal
SSB	Single Side Band
MU-MIMO	Multi-User MIMO
SU-MIMO	Single-User MIMO
TS	Traffic Steering
CC	Component Carrier
CA	Carrier Aggregation
DC	Dual Connectivity
V2X	Vehicle-to-Everything
NR-U	New Radio Unlicensed
RF	Radio Frequency

TX	Transmitters
RX	Receivers
CNF	Cloud-Native Network Function
CPU	Central Processing Unit
PCI	Physical Cell Identity
MEC	Multi-Access Edge Computing
CPM	Congestion Prediction and Management
EHC	Ethernet Header Compression
SON	Self Organizing Networks
SINR	Signal-To-Interference-Plus-Noise Ratio

Chapter 1

Introduction

Wireless communications utilize electromagnetic waves transmitted over the air. Electromagnetic waves reduce reliance on a physical connection, contributing to global interconnectivity. Throughout the years, various wireless communication technologies have emerged, like Radio, WiFi, and Bluetooth. Even though these technologies pose technological milestones, their significance pales compared to wireless cellular networks.

The name wireless cellular networks comes from their structure, which resembles the organization of cells in biological organisms at a microscopic level. Cellular metaphorically refers to a geographical area division into multiple cells, like a living organism. A collection of cells forms a cellular network covering a larger geographic region. The cellular architecture allows for efficient frequency allocation amongst cells, signal inference minimization, and overall capacity maximization of the network. Much like living organisms, where the cells interact with each other to form a functioning organism, the cells in cellular networks collaborate in a unified network. A more precise depiction of cellular networks [1] is presented in Figure 1.1.

Cellular networks consist of two fundamental components called base stations, also referred to as cell towers, and user equipment (UE). Telecommunications companies strategically position base stations to efficiently transmit and receive signals to and from user equipment within designated cellular areas or cells. User equipment encompasses a diverse set of mobile devices with radio transceivers ranging from phones and laptops to IoT devices. The transceivers attached to the user equipment enable communications with nearby base stations and seamless transition between neighboring cells without connection quality degradation.

Figure 1.1: Cellular Networks

1.1 Subject Matter and Contribution

This thesis focuses on two distinct areas open for research within cellular networks. The first area tackles bufferbloat in large buffers for optimal performance in modern disaggregated cellular networks, referred to as active queue management (AQM). It enhances previous research by providing and assessing an artificial intelligence (AI) and machine learning (ML) framework, which addresses the optimization of latency-sensitive algorithms within high-latency environments.

The second area falls within network orchestration research and focuses on controller placement in cellular networks. Controller and Virtual Network Function (VNF) placement are optimization problems extensively studied in Software Defined Networking (SDN) and Network Functions Virtualization (NFV), respectively. Given the integration of controllers into cellular networks, extending the scope of the controller placement problem to include these networks becomes a natural progression. This thesis introduces the mathematical model that describes a standard implementation of a cellular network and its controllers. According to the network's requirements, solving the controller placement problem determines the optimal placement. This thesis experiments with different requirements on the same network topology to evaluate the controller placement efficiency.

1.2 Thesis Organization

This thesis structure consists of six distinct chapters. Chapter I provides a comprehensive introduction to the subject matter of this thesis. Chapter II is an overview of each cellular network generation. Chapter III presents a general overview of AI and ML. Chapter IV and Chapter V research AQM and Controller Placement, respectively. Chapter VI marks the end of this thesis.

Chapter 2

Cellular Networks

2.1 Background

Cellular network generations are the evolution of mobile telecommunications standards and technologies. Future cellular network generations are extensions of the previous generations and offer a wider variety of features, broader capabilities, and improved performance, including speed, capacity, and functionality. This section explores the existing cellular network generations, highlighting the few crucial characteristics of each.

2.2 Earlier Generations Overview

2.2.1 First Generation (1G)

The 1G cellular networks marked the inception of mobile telecommunication systems. Introduced in the 1980s, 1G networks utilized analog technology for wireless voice communication. These networks support basic voice-only capabilities, limited capacity, and relatively low call quality.

In 1G networks, distinct frequency channels divide the radio spectrum, with each channel assigned to a specific call. The 1G networks use Frequency Division Multiple Access (FDMA) to access the frequency channel, enabling simultaneous user calls within a given geographical area. FDMA splits the overall available bandwidth into distinct frequency channels, each used by a different user. However, the analog nature of 1G made it susceptible to interference, resulting in inconsistent call quality and a relatively low capacity to support

users.

2.2.2 Second Generation (2G)

Rolled out in the early 1990s, 2G networks improved upon the existing 1G networks by transitioning from analog to digital technology, providing a more efficient and versatile communication platform. The introduction of digital voice transmission in 2G networks results in crisper and more reliable voice calls when compared to their 1G predecessors. Moreover, 2G networks laid the foundation for modern text-based communication by integrating an innovative Short Message Service (SMS), which allowed for the transmission and reception of text messages. The transition from 1G to 2G was a stepping stone in enabling the integration of essential services in cellular networks. Even though modern cellular networks outperform 2G, for its time, the increased speeds and better overall performance enabled users to access web content and email.

The 2G networks employ technologies like Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA) to divide the available spectrum. In TDMA, users share the same frequency channel by allocating different time slots, while in CDMA, users share the whole frequency spectrum with unique codes.

2.2.3 Third Generation Partnership Project (3GPP)

Established in 1998, the 3GPP influences the development of mobile telecommunication network standards and specifications. For 3GPP's establishment and maintenance, a wide range of organizations and institutions from across the globe collaborate to develop and implement global mobile communication standards. The primary and most crucial objective is to create and enforce globally accepted standards, including network architectures, protocols, air interfaces, and services. With 3GPP, devices, and networks from different manufacturers and operators can seamlessly communicate. Finally, future generations of cellular networks after the introduction of 2G are specified in 3GPP.

2.2.4 Third Generation (3G)

The 3G, introduced in the early 2000s, represented a substantial advancement, offering better voice quality, faster speeds with increased bandwidth, and low latency with support for

data capabilities and data-intensive applications. It provided exceptionally high data transfer rates, improving the internet access experience and enabling users to engage more easily with internet content and send or receive emails. It allowed users to access websites, search internet content, and transmit and receive multimedia content like videos, images, or audio. The improved quality and speed also laid the foundation for services like media streaming, video calling, and mobile TV destined for future generations. Additionally, the increased capacity of 3G networks increased the number of simultaneous connections within a cell, reducing call drops and improving call quality. Finally, 3G supports global roaming, where devices can seamlessly travel between countries that support 3G and maintain connectivity.

2.2.5 Fourth Generation (4G)

The 4G cellular networks represent a leap in communication technology compared to 3G. Rolled out in the late 2000s and early 2010s, 4G outperforms 3G cellular networks in speed, capacity, and user experience. The 4G offers high data speeds with a substantial increase in download and upload speeds compared to 3G. High speed allows high-definition video streaming and rapid file downloads and uploads. Low latency is another breakthrough in 4G. Users can enjoy a smoother, more responsive, and more immersive online gaming experience and an increased communication quality with video conferencing. The architecture of 4G, based on Internet Protocol (IP), enables the integration of voice and data services. An example of this all-IP-based architecture is the voice over IP (VoIP) services, which allow users to make voice calls over data networks. Moreover, enhancements in handoff mechanisms improve connection consistency, offering seamless mobility, where users move between cells and do not experience disruptions in connectivity.

One key feature incorporated in 4G that also shaped future generations of mobile networks was the efficient spectrum utilization. The 4G utilizes Orthogonal Frequency Division Multiplexing (OFDM) and Multiple Input Multiple Output (MIMO) to maximize spectrum efficiency. OFDM and MIMO allow base stations to serve more users simultaneously within a given frequency band, improving overall network capacity.

The two most widely adopted 4G technologies were the Long-Term Evolution (LTE) and the Worldwide Interoperability for Microwave Access (WiMAX). LTE, in particular, gained widespread popularity due to its high data rates, scalability, and compatibility with existing 2G and 3G networks. Finally, an LTE base station is named Enhanced Node B (eNB), an

evolution from 3G's Node B (NB).

2.3 Fifth Generation (5G) and Beyond

The 5G network is a pivotal milestone in the networking and telecommunications industries. Like the previous generations, 5G builds upon its predecessors and transcends the boundaries of speed, capacity, and latency. The 5G networks facilitate applications ranging from autonomous vehicles and Internet of Things (IoT) devices to Augmented Reality (AR) and Virtual Reality (VR).

One of the most notable improvements of 5G, compared to 4G, is its remarkable high speeds, which outperform 4G by a significant margin. This significant increase in data rates enables seamless streaming of high-definition content and almost instantaneous downloads. Latency is also minimized in 5G, allowing for more deployment of applications that require low response times, such as remote surgeries. The simultaneous increase of speeds with decreased latency also eliminates lag in real-time gaming and other interactions. The significantly increased capacity of 5G networks can support massive amounts of IoT and smart devices without performance compromisations.

The term used in the telecommunication industry for 5G networks is New Radio (NR). NR is the global standard for a unified 5G wireless air interface. As part of the evolution from 4G's LTE to 5G's NR, the eNB in now rebranded as the Next Generation Node B (gNB).

2.3.1 Radio Access Network (RAN)

Contrary to the 4G's monolithic eNB, 5G's gNB design incorporates virtualization and cloud-native capabilities. To optimize the gNB's functionality and cloudify the network, 3GPP introduced functional splits where the gNB disaggregates into multiple entities. These entities are the Centralized Unit (CU), the Distributed Unit (DU), and the Radio Unit (RU). The most commonly used functional split option is the 3GPP 7.2x split [2] illustrated in Figure 2.1. According to this split, CU supports the higher layers of the protocol stack, such as Service Data Adaptation Protocol (SDAP), Packet Data Convergence Protocol (PDCP), and Radio Resource Control (RRC). On the other side, DU supports the lower layers, including Radio Link Control (RLC), Medium Access Control (MAC), and Physical Layer (PHY). The CU consists of two components, the control plane (CP) and the user plane (UP), called

CU-CP and CU-UP, respectively. The CU-CP and CU-UP contain the RRC and the SDAP layers, respectively. Additionally, Figure 2.1 depicts the protocol stack layers in control and user planes, with the green and the red vertical lines, respectively, as described in [3].

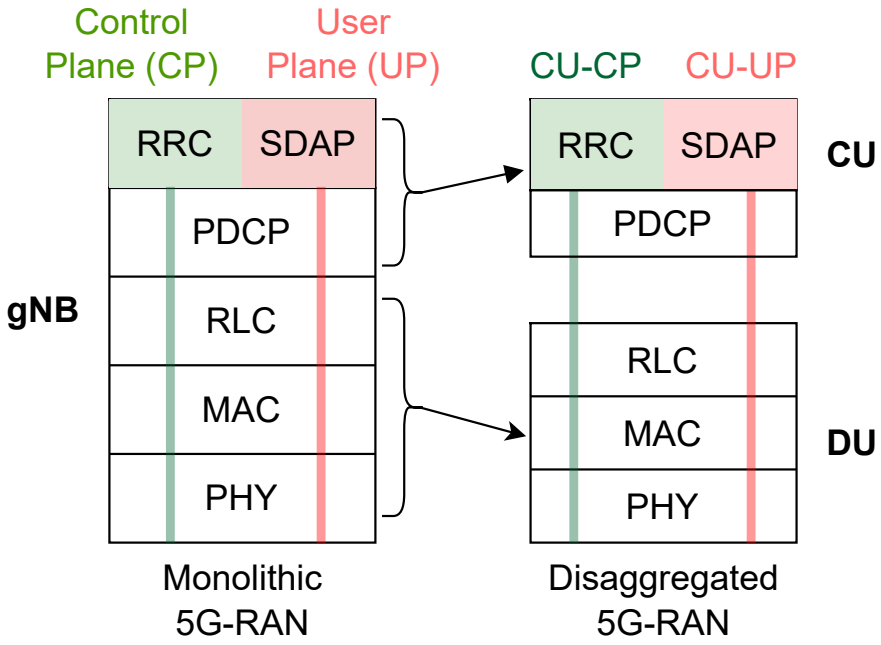


Figure 2.1: Control and user planes in monolithic and disaggregated gNB.

Each protocol layer performs different actions contributing to achieving end-end connectivity. The SDAP protocol is a state-of-the-art user, or data, plane protocol introduced in 5G. It is responsible for packet adaptation for distinct services and ensures that data is treated appropriately based on quality requirements such as latency sensitivity and priority. The RRC is a control plane protocol first introduced in 3G and extended in consequent generations. It is responsible for managing radio resources and controlling radio connections between a UE and its corresponding base station. The PDCP is a data plane protocol introduced in 4G and extended in 5G cellular networks. It assists in IP packet transmission optimization while also performing header compression, encryption, and integrity protection. The RLC is a protocol layer in 3G, 4G, and 5G cellular networks. It contributes to radio link reliability by assuring data packet delivery between the sender and receiver. The RLC supports packet segmentation, reassembly, error detection, and lost or corrupted packet retransmission. The MAC layer resides within the Open Systems Interconnection (OSI) model. It controls access to a shared communication medium and assists in collision avoidance while ensuring its efficient and fair usage. The PHY layer is the lowest in the OSI model, representing the physical properties of the communication media, such as the cable, the wireless signal, the electrical voltage, and

even special hardware, such as the Universal Software Radio Peripheral (USRP). The PHY layer or RU encodes the raw binary data transmitted over the physical media and manages modulation, encoding, and signal transmission.

Figure 2.2 illustrates the CU-DU-RU mapping. Each CU can have multiple DU entities, while a DU only connects to a single CU. Accordingly, each DU can have multiple RU entities, while an RU entity only connects to a single DU.

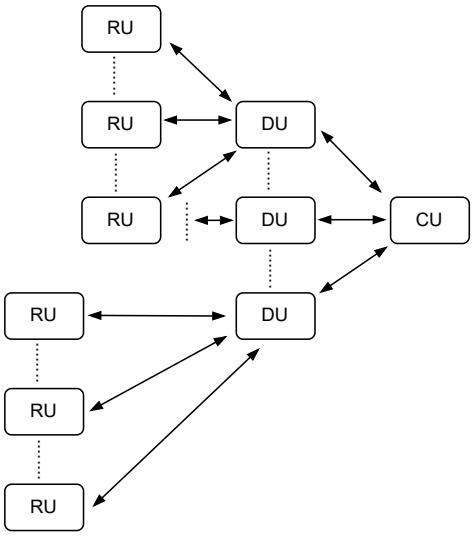


Figure 2.2: CU-DU-RU Mapping.

2.3.2 Service Management and Orchestration (SMO)

The SMO supports various orchestration and management services, extending its reach beyond traditional RAN management [4]. The SMO is a crucial component of the Open RAN (O-RAN) architecture since it is responsible for

1. Providing a comprehensive Fault, Configuration, Accounting, Performance, and Security (FCAPS) interface to network functions (NFs),

2. Handling extensive, long-term RAN optimization efforts and

3. Managing and orchestrating O-Cloud resources, encompassing resource discovery, scaling, FCAPS, software management, and performing Create, Read, Update, Delete (CRUD) operations on resources.

The SMO's role is pivotal, ensuring the seamless functioning of the O-RAN ecosystem, which spans from network performance to dynamic O-Cloud resource management.

2.3.3 RAN intelligent Controller (RIC)

The RIC [5] is a cloud-native software-defined component crucial for managing the open and virtualized RAN network, including its resources and functions, supporting 4G, 5G, and beyond cellular networks. It also supports intelligence and third-party applications for achieving optimal client quality of experience (QoE). The RIC is either a Near-Real-Time RIC (Near-RT RIC) or a Non-Real-Time RIC (Non-RT RIC). The main differences between the Near-RT RIC and the Non-RT RIC are the timescale they manage events and resources and the communication interfaces they incorporate. To begin with, Near-RT RIC operates in timescales between 10ms and 1s while Non-RT RIC in timescales greater than 1s. As presented in Figure 2.3, Near-RT RIC supports E2 and A1 interfaces, while the Non-RT RIC supports O1, O2, and A1 interfaces. The following section is an overview of each of these interfaces.

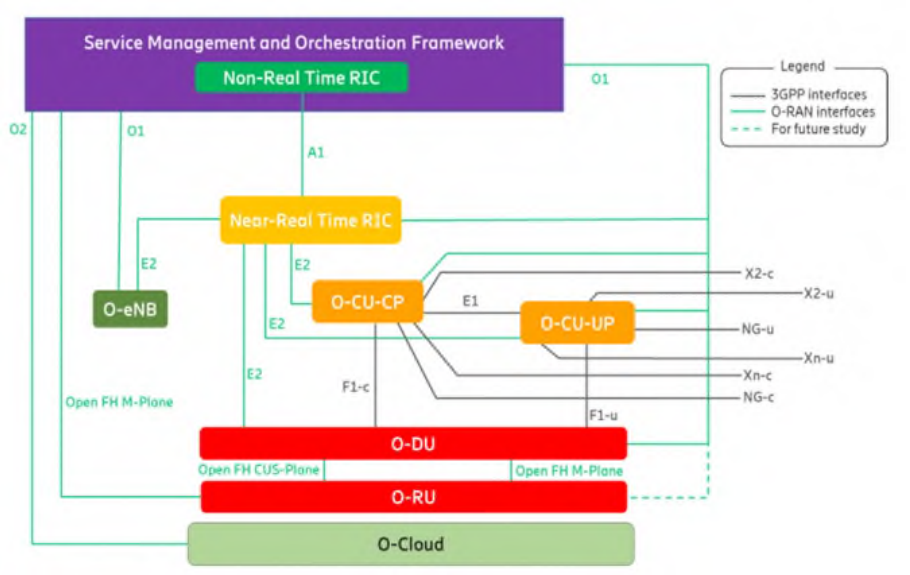


Figure 2.3: Control and user planes in monolithic and disaggregated gNBs.

2.3.4 Network Interfaces and Protocols

O1 and O2

The O1 interface connects the Non-RT RIC with the various RAN entities. Its primary purpose is to enable the exchange of non-real-time information and configurations between the Non-RT RIC and the base stations. The O1 interface supports configuration parameters provisioning, policies, and network optimization directives from the Non-RT RIC to the RAN

elements, enhancing network efficiency, capacity, and quality of experience (QoS) without constraining the real-time processing.

The O2 Interface facilitates the communication and exchange of information, insights, and data regarding network-wide optimization strategies, traffic trends, and resource utilization between Non-RT RIC instances within a network. Non-RT RIC instances can work together to optimize network performance and improve the user experience through the O2 interface. Thus, O2 enables the exchange of non-real-time data and analytics while allowing coordination directives sharing to ensure the network operates consistently and efficiently.

A1

The A1 interface is a standardized and open communication channel that facilitates the exchange of critical data, information, and commands between a Non-RT RIC and a Near-RT RIC, including non-real-time data like performance metrics, network status, and configurations. So, not only is the A1 interface a bidirectional interface that acts as a bridge between the Near-RT RIC and the Non-RT RIC, but it also allows for communication between different Near-RT RIC entities via the Non-RT RIC.

E2

The E2 interface is a standardized communication channel that facilitates the smooth operation of the Near-RT RIC and the various RAN entities. This bidirectional communication channel allows the exchange of real-time information, commands, and insights between the Near-RT RIC and the RAN. Near-RT RIC's E2 enables network elements to collaborate to improve performance, adjust to changing circumstances, and enhance the QoE. The O-RAN specification defines two fundamental E2 concepts, the E2 Application Protocol (E2AP) and the E2 Service Model (E2SM). The E2AP is a general set of protocols that the Near-RT RIC and the RAN nodes use to communicate. Services defined by action types that use these protocols are Report, Insert, Policy, Control, and Query. The E2SM, on the other hand, is the description of services exposed to the Near-RT RIC by the RAN nodes. The Near-RT RIC's xApps subscribe, using the E2AP, to E2SM's registered RAN functions as illustrated in Figure 2.4.

The Near-RT RIC can retrieve information from the RAN via the Indication Report service. The E2SM, after receiving such a request, periodically reports the information requested

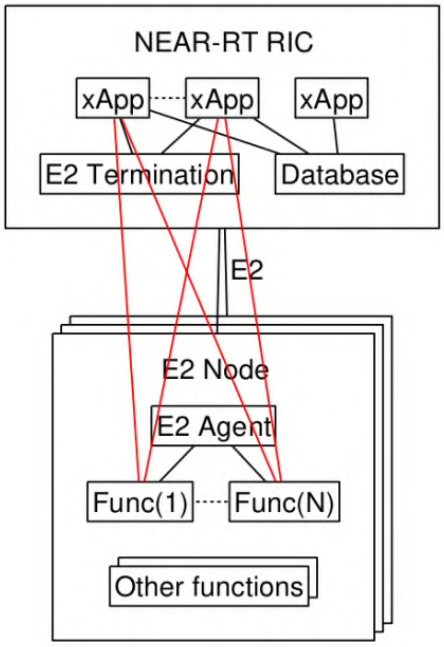
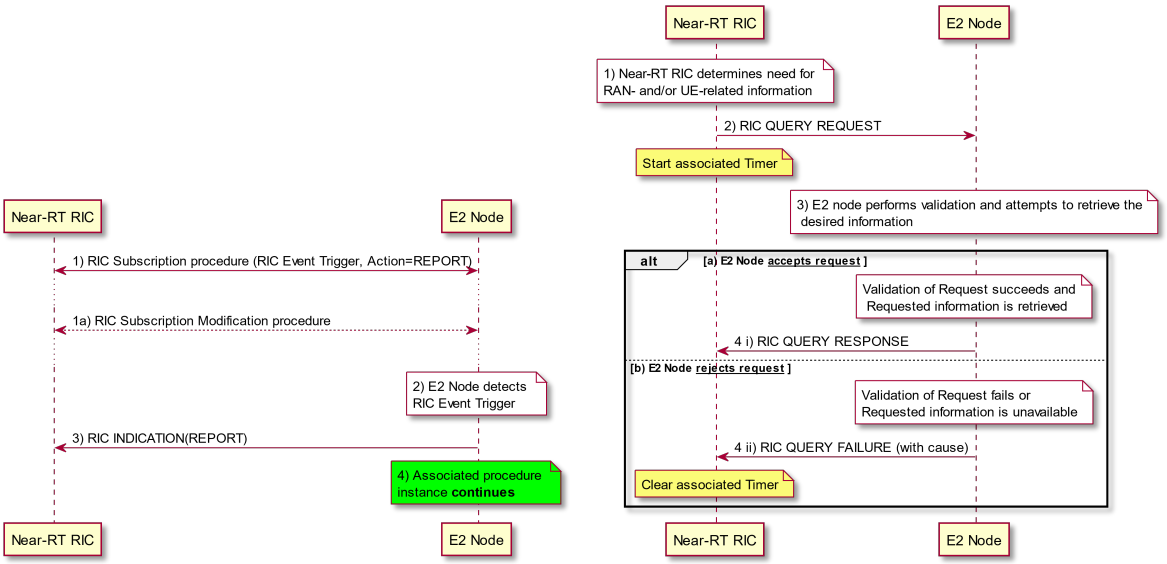


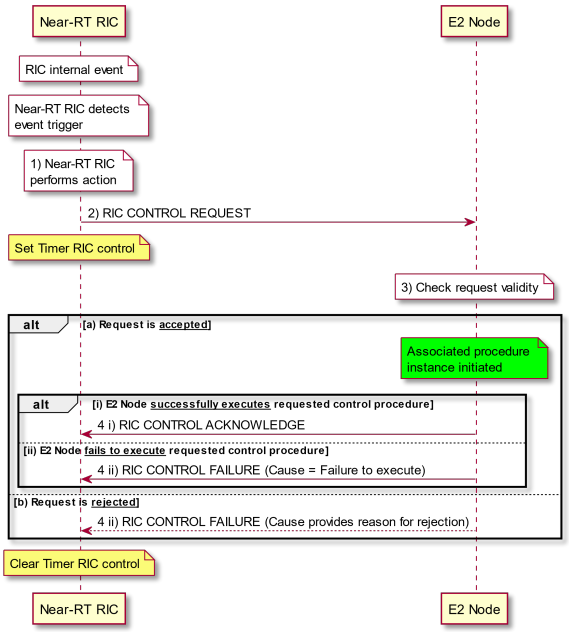
Figure 2.4: Near-RT RIC E2AP and RAN E2SM.

to the Near-RT RIC as seen in Figure 2.5i. The Near-RT RIC can directly control a RAN entity via a Control message as seen in Figure 2.6i. However, a more realistic implementation of a control message is alongside the Indication Insert, used for RAN monitoring. Contrary to Indication Report, where the RAN entity reports periodically back to Near-RT RIC, in Indication Insert, RAN suspends its functionality till it receives a control message, as seen in Figure 2.6ii or in Figure 2.6iii. The Near-RT can also enforce policies in a RAN entity with a Policy message, depicted in 2.6iv. Finally, the Near-RT can retrieve information from RAN via a Query message as in 2.5ii.

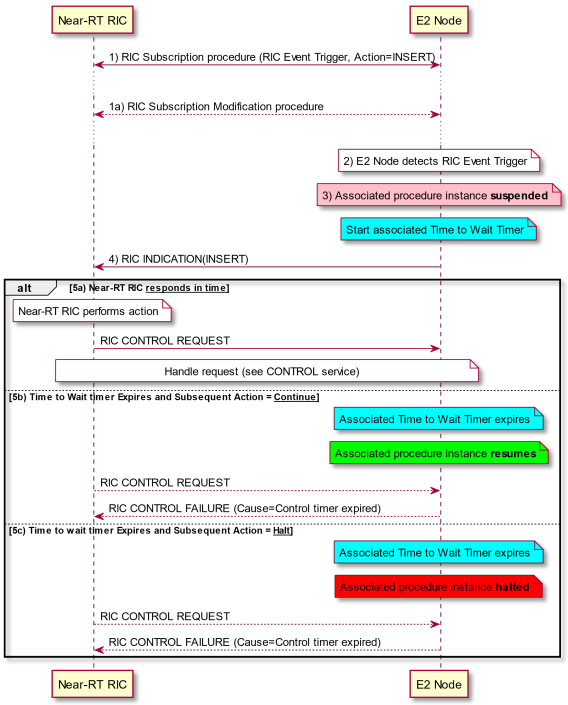


(i) E2 Indication Report. (ii) E2 Query.

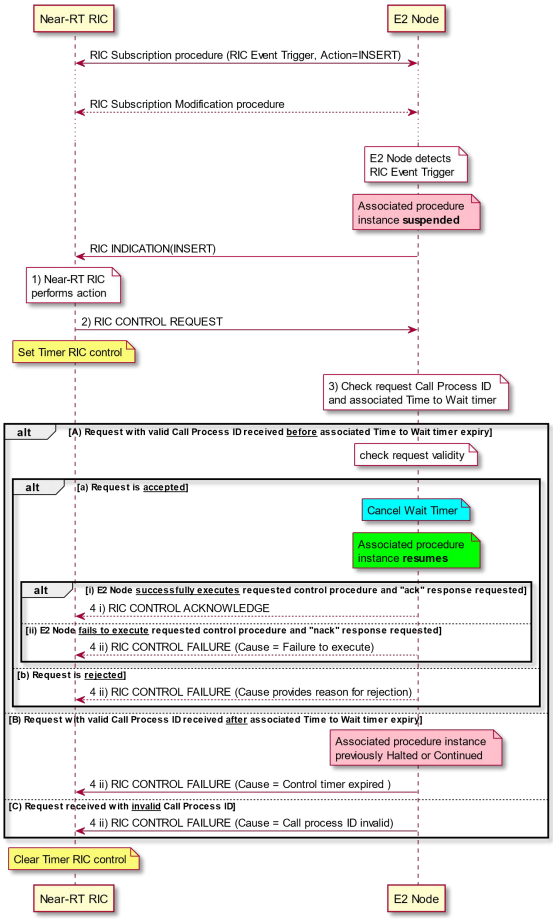
Figure 2.5: E2 Indication Report and Query.



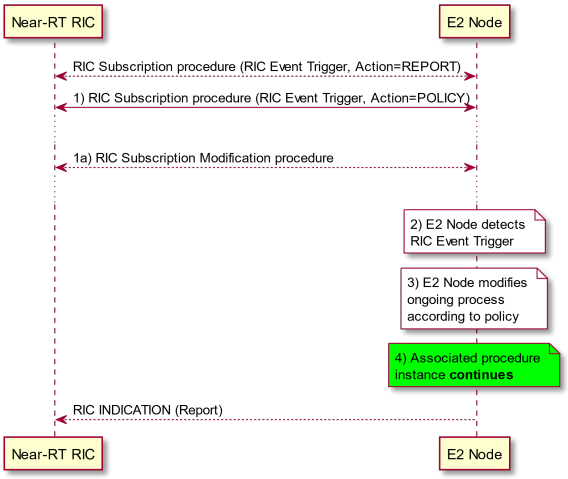
(i) E2 Control.



(ii) E2 Insert and Control.



(iii) E2 Insert and Control.



(iv) E2 Policy.

Figure 2.6: E2 Control and Policy.

2.3.5 Core Network (CN)

The 5G CN, often known as 5G core or 5GC, is a fundamental component of the 5G architecture. It orchestrates the various RAN components while managing a diverse range of services. The main focus of 5GC is to provide secure and private end-to-end connectivity and improve the overall user experience. The 5GC has a cloud-based design, leveraging virtualization and dockerization, with its advantage being scalability and resource utilization efficiency. The 5GC consists of multiple components, as illustrated in 2.7, that interact with each other to meet the requirements of modern connectivity.

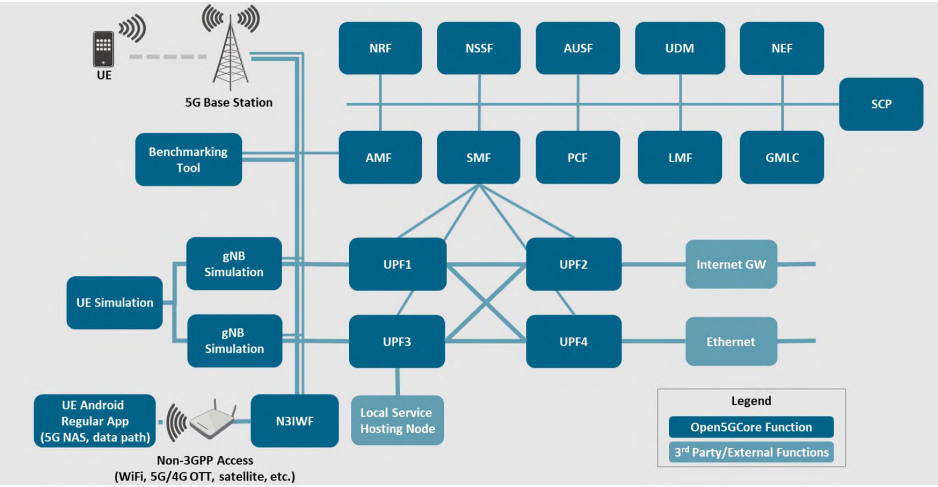


Figure 2.7: Core Network.

The 5GC decomposes and extends the functionality of the 4G core network's Mobility Management Entity (MME) into Access and Mobility Management Function (AMF) and Session Management Function (SMF) The AMF manages user access and mobility in the network. The SMF ensures efficient utilization of network resources by handling the establishment, modification, and termination of data sessions. The Unified Data Management (UDM) stores and manages user-related data. This data includes subscription profiles, authentication credentials, and service entitlements, enabling seamless user experiences across different devices and services. The Authentication and Authorization Function (AAF) or Authentication Server Function (AUSF) performs authentication and authorization of users and devices to the network. It ensures the security and integrity of network connections. The UPF is an evolution of the 4G Serving Gateway (SGW) and Packet Data Network (PDN) Gateway (PGW), collectively known as SPGW. Its role is to manage the data plane within the network, handling the routing and forwarding of user data packets. The Policy Control

Function (PCF) manages policy and service quality enforcement. It ensures the allocation of network resources according to predefined policies and that different services receive the desired quality and priority. The Network Exposure Function (NEF) provides an Application Programming Interface (API) that allows third-party users and corporations to deploy applications and services and interact with the core network. The Network Slice Selection Function (NSSF) determines the appropriate network slice for a user or device in network slicing. The appropriate slice selection is based on their requirements and the available resources and ensures proper resource allocation to services. The Network Repository Function (NRF) assists in service discovery and routing within the core network by maintaining a repository of information about network functions and services. Finally, the Application Function (AF) is responsible for application logic in the core network. It provides an interface for service developers that enables hosting and managing the creation and deployment of new services.

Chapter 3

Intelligence and Machine Learning

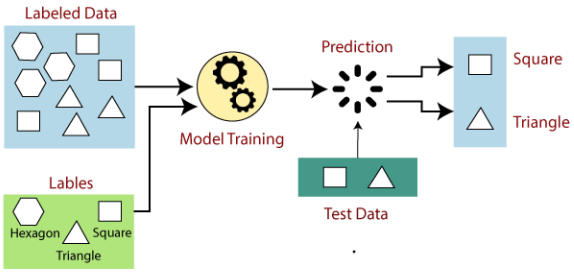
3.1 Introduction to Machine Learning

ML is AI that allows software applications to predict the outcomes of different problems by inventing unique algorithms. There are three types of ML, called supervised [6], unsupervised [7], and reinforcement learning [8], depicted in 3.1i, 3.1ii, and 3.1iii, respectively. Each paradigm encompasses distinct approaches to training algorithms and addressing various challenges within AI. In supervised learning, labeled training data is provided to the ML model, enabling it to examine existing patterns and make predictions or classifications when presented with unseen data. On the other hand, in unsupervised learning, unlabeled data is fed to the model to uncover the relationship and the underlying structure between the given data points and solve for clustering, dimensionality, or reduction. Finally, reinforcement learning models, or agents, adjust their behavior through a sequence of decisions, learning optimal strategies through rewards or punishments within a given environment.

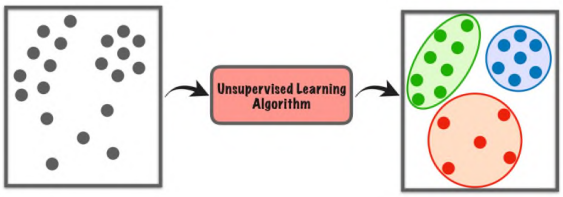
3.2 Neural Networks

Neural networks are a subset of ML, constituting the heart of deep learning algorithms. They are a mathematical implementation of the human brain mimicking the behavior of biological neurons that signal one another to form intelligence.

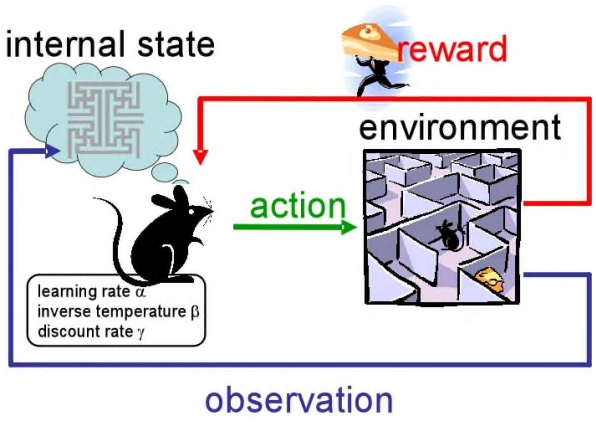
Neural networks, illustrated in 3.2, contain node layers, the first of which is the input layer and the last one the output layer, while the layers in between are called hidden layers. Two nodes can connect through a weighted connection. Each node has its threshold value,



(i) Supervised Learning.



(ii) Unsupervised Learning.



(iii) Reinforcement Learning.

Figure 3.1: Types of ML.

which becomes active when its value exceeds this threshold. Only active nodes forward their values to the next layer through their connections.

Neural networks retrieve data from the input layer, undergo training, learn, and enhance accuracy before exporting and evaluating results at the output layer. The advantage of neural networks is their improved performance in prediction, classification, and clustering problems compared to human performance on similar tasks. Different tasks require different fine-tuned neural network architectures.

3.3 Recurrent Neural Networks (RNN)

RNN is a type of neural network where the input of the current step is the output of the previous step. Contrary to a traditional neural network, where the inputs and the outputs are independent, an RNN can rely on previous information to deduce future data. Having memory states, they can remember information about a sequence based on previous inputs. The hidden layer maintains this information, called a hidden state. Compared to a traditional

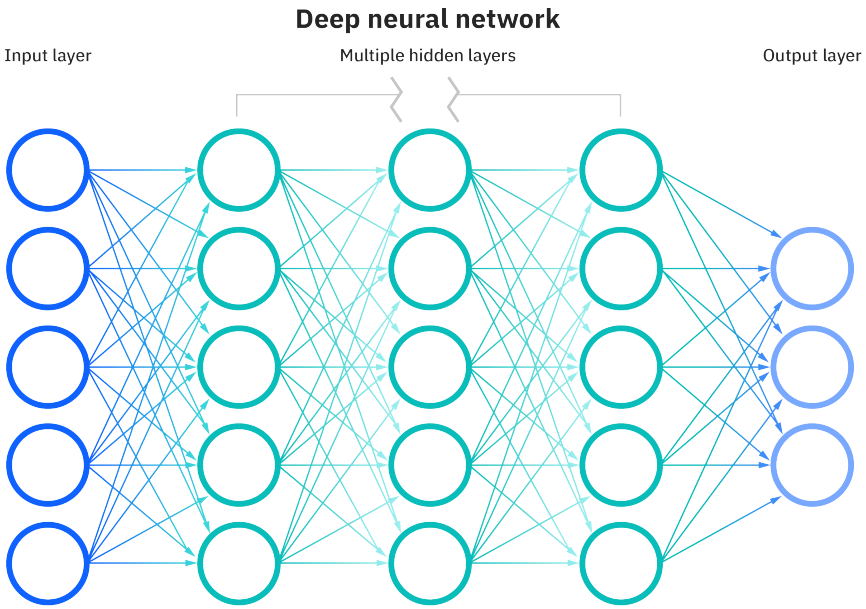


Figure 3.2: Input, Hidden and Output Layers in Neural Networks.

neural network, an RNN uses the same parameters for each input, performing the same task on all inputs or hidden layers to produce the output.

RNNs suffer from two different types of problems, the vanishing gradient and the exploding gradient descent problems. In vanishing gradient the primary shortcoming arises when the amount of layers increases and certain activation functions are added to the neural network. These functions force a large input space to be squished into a smaller one resulting in a training complexity increase as the gradients of the loss function approach zero. The exploding gradient descent, on the other hand, arises when the derivatives of backpropagation are large. When this happens the gradient will increase exponentially as we propagate down the model until they eventually explode. This makes the model unstable and its results might deviate from the correct solutions. Since the aforementioned two problems are observed in RNNs, two different RNN variants were introduced, the Long Short-Term Memory (LSTM) neural networks and Bidirectional Neural Networks (BiNN).

3.3.1 BiNN

In BiNN the input information flows in both direction and then the output of both directions are combined to produce the input. The input data is very important in BiNN, and thus they are used in Natural Language Processing (NLP) and in Time-Series analysis problems.

3.3.2 LSTM

LSTM works by reading and writing the most useful information that produces valid output. Redundant information if discarded since it is not important to the RNN when making predictions. The functionality previously described is achieved with three new RNN gates, which are illustrated in Figure 3.3 and act as filters that classify which information to forward to the next layers and which one to forget based on the input.

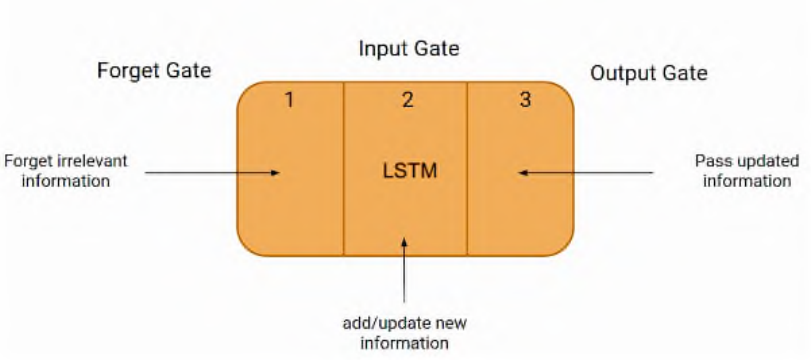


Figure 3.3: LSTM Node

Chapter 4

Experimental Tools

4.1 Introduction

In this chapter, we explore the experimental tools employed in this thesis. Subsequent sections provide an analysis of the hardware and software resources, encompassing testbed, platforms, frameworks, libraries, repositories, programming languages, and databases.

4.2 NITLAB and NITOS Testbed

Since 2007, the Network Implementation Testbed Laboratory (NITLAB) [9] at the University of Thessaly (UTH) team has developed and maintained a Network Implementation Testbed utilizing Open Source platforms facility called NITOS [10]. NITLAB is affiliated with the Centre for Research and Technology Hellas (CERTH) [11], while NITOS is one of OneLab Federation's [12] facilities, accessed through the OneLab portal.

NITLAB and, therefore, NITOS have grown over time into a powerful solution for experimenting and assessing new concepts and technologies in networking research. As a result, it has become one of the most well-known experimental hubs in Europe, known for its vast scope in wired and wireless network research capabilities. It serves as a bridge between highly programmable, remote accessible technologies, allowing users from all over the world to reach its full potential, as illustrated by the architectural configuration shown in Figure 4.1. The foundation of this functionality is the open-source control and management framework (OMF), allowing users to perform custom experiments by reserving slices. These slices include computational nodes, which can act as access points, stations, base stations, or simple

devices. Slices also reserve and utilize the frequency spectrum of the testbed through the NITOS scheduler. Its design allows for a smooth interaction with a wide variety of resources, thus expanding its scope and providing a complete platform for experimentation. Finally, NITOS consists of three separate testbeds positioned in different environments.

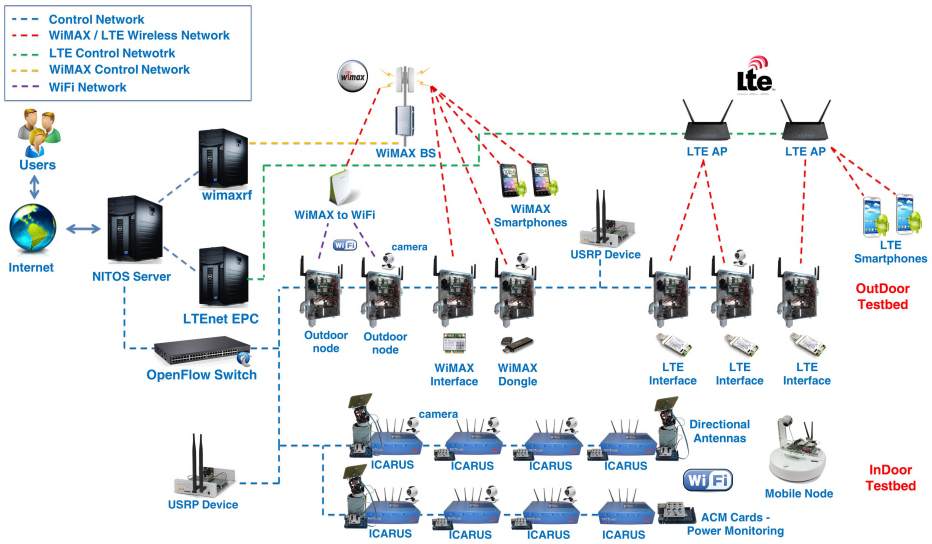


Figure 4.1: NITOS Testbed Architecture.

4.2.1 Outdoor Testbed

The outdoor testbed consists of fifty computational nodes positioned at the exterior of the University of Thessaly's campus building, as illustrated in 4.2. Each node supports multiple wireless interfaces, allowing experimentation with heterogeneous wireless technologies. These technologies can range from Wi-Fi to WiMAX, LTE, and NR. An outdoor testbed's importance lies in its ability to enable testing in real-world environments, where noise and interference are present that reflect the complexity of the real world.

4.2.2 Indoor RF Isolated Testbed

To provide researchers with the opportunity to evaluate power-intensive algorithms and protocols in a controlled and secluded environment, NITOS has developed an indoor testbed consisting of fifty ICARUS nodes. Each node supports multiple wireless interfaces, ranging from Wi-Fi to LTE, WiMAX, and NR. This versatile deployment features cutting-edge components, including directional antennas and advanced prototypes. The strategically arranged

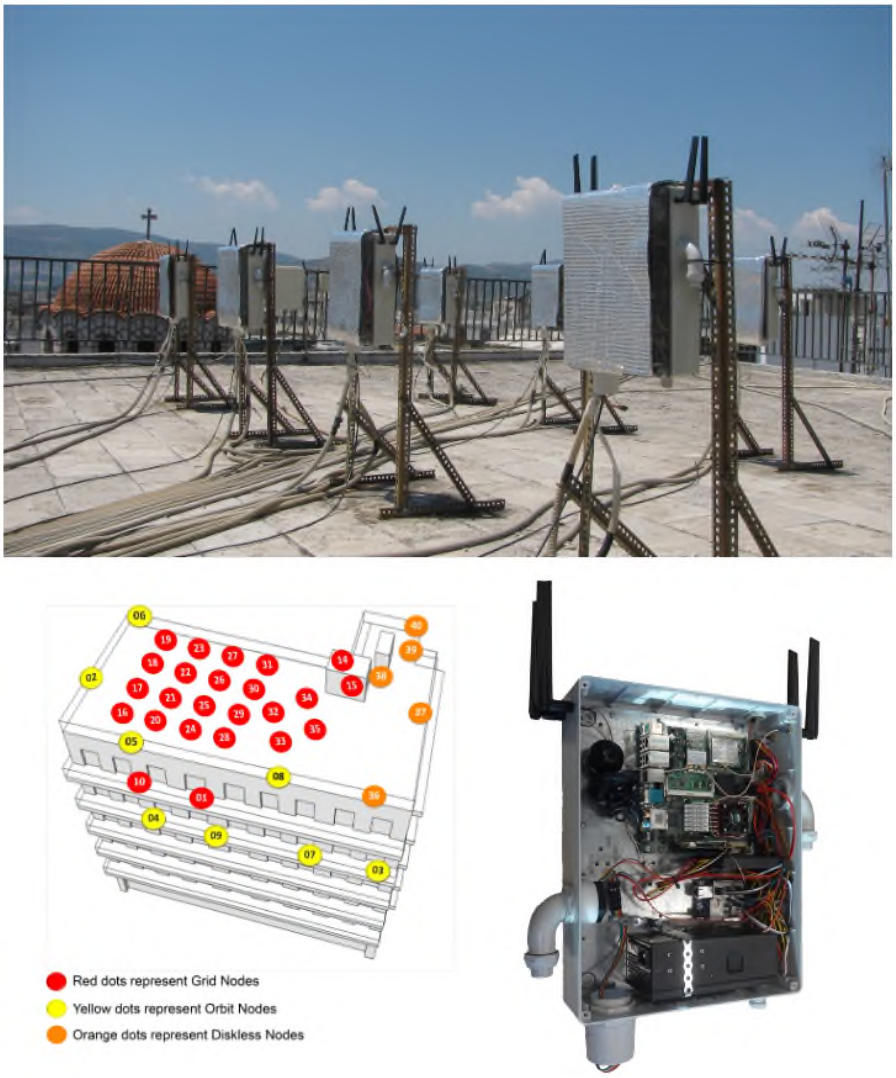


Figure 4.2: Outdoor Testbed.

ICARUS nodes are at a consistent 1.2-meter spacing and share the same height level, creating a symmetrical grid topology characterized by isobaric nodes of equivalent capability. This well-planned layout ensures a uniform and consistent testing environment, demonstrated in Figure 4.3.

4.2.3 Office Indoor Testbed

The office testbed comprises ten high-performance second-generation ICARUS node nodes. These nodes also contain a combination of technologies, including Wi-Fi, WiMAX, LTE, NR, and others. This range of technologies allows experimenters to carefully plan and execute experiments based on real-world parameters within a controlled, predictable office setting. The physically located testbed at the CERTH Information Technologies Institute (ITI)

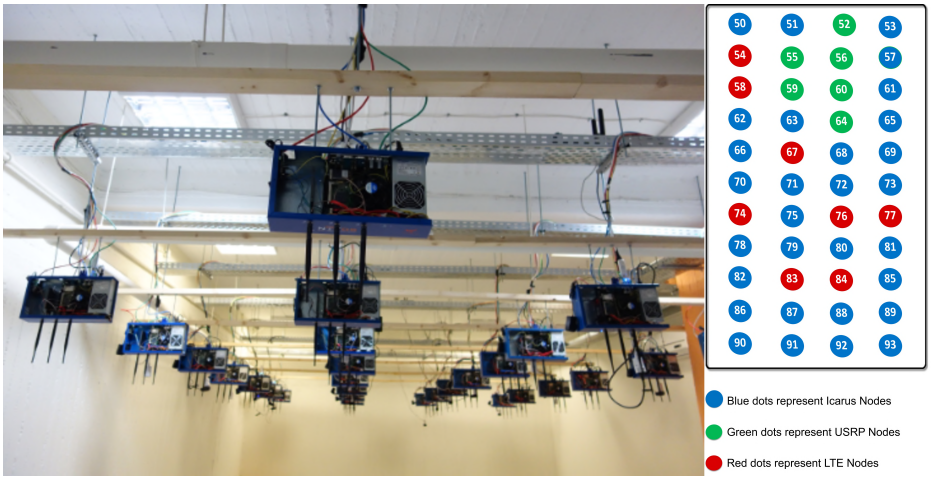


Figure 4.3: Indoor Testbed.

laboratory in Volos city strategically spans its entire third-floor layout. This integration adds to the testbed's experiential depth by creating a naturally heterogeneous environment. This enhancement improves the authenticity of the experimentation, reflecting real-world challenges. Figure 4.4 illustrates the strategic interaction between the technology and the environment.

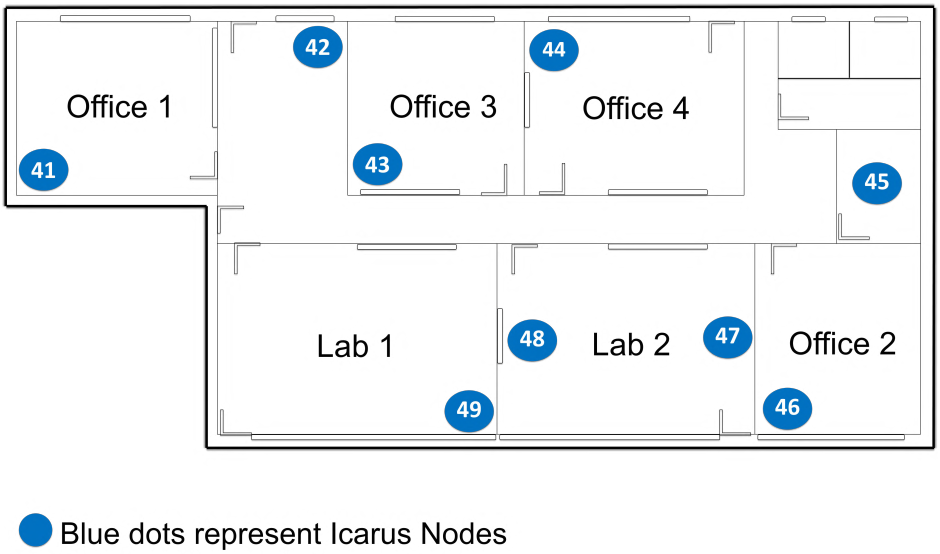


Figure 4.4: Office Indoor Testbed.

4.3 The Open Air Interface (OAI) Platform

Traditionally, telecommunications equipment manufacturers and vendors provided proprietary hardware and software solutions for experimentation and implementation of cellular

networks. The closed nature of these systems resulted in high costs and less flexibility and innovation. The OAI [13] is open-source and provides a 3GPP compliant emulation of 4G and 5G RAN's and CN's key components, aiming to address the previously mentioned limitations. The emulation supports conventional computing platforms and Software Defined Radio (SDR) cards. Users can deploy a 4G or 5G network and inter-operate with either commercial-off-the-self (COTS) UE or a SIM card dongle. The open-source nature of OAI enables experimentation, configuration testing, and modifications on emulated networks. The OAI project includes the RAN, the CN, and the RIC.

4.3.1 RAN

The integration of 4G LTE's eNB and 5G NR's gNB within the RAN aligns with the dynamic standards established by the most recent iterations of the 3GPP. The OAI provides a novel approach to testing that eliminates the need for radio frequency boards, instead relying on software simulations to accurately replicate layers within either the 4G protocol stack or the 5G stack. Additionally, it supports commercial off-the-shelf (COTS) UE emulation, demonstrating its dedication to providing comprehensive and versatile wireless communication solutions. Finally, OAI RAN offers a variety of deployment options, including monolithic and disaggregated deployments of the RAN.

4.3.2 CN

OAI provides two different implementations of the CN, designed for the 4G and 5G environments, respectively. Figure 4.5 illustrates the CN that supports LTE 4G, while Figure 4.6 depicts a specialized basic implementation of the CN that supports NR 5G networks.

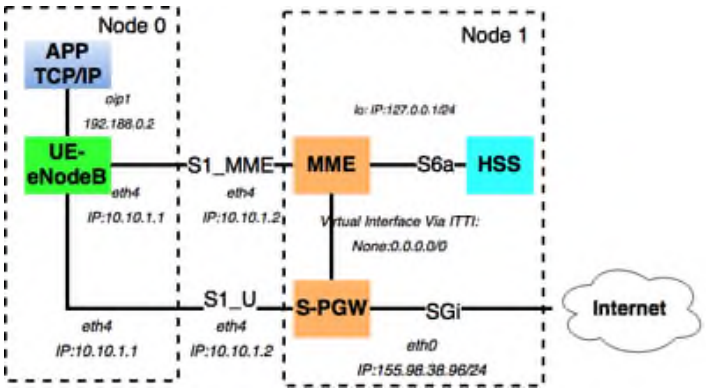


Figure 4.5: OpenAirInterface LTE Core Network.

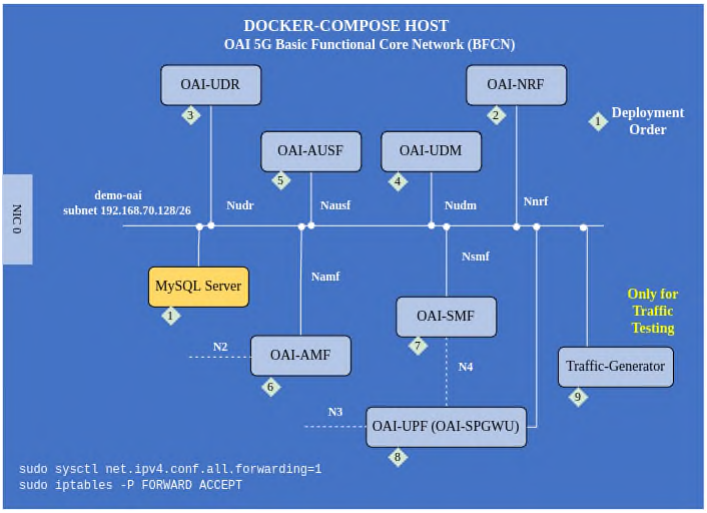


Figure 4.6: OpenAirInterface NR Core Network.

Both CN implementations allow for end-to-end connections between the connected UEs and the Internet.

4.3.3 RIC

As of the current moment, OAI exclusively accommodates Near-RT RIC for its operations. Deploying Near-RT RIC necessitates the utilization of two distinct implementations, contingent upon whether the connection interfaces with an eNB or a gNB.

In the context of eNB connections, the designated Near-RT RIC is called FlexRAN. Conversely, for both gNB and eNB connections, the corresponding Near-RT RIC takes on the appellation of FlexRIC. Whether the connection is with LTE or NR, it is E2 compatible, corresponding to the intended design, shown in Figure 4.7. The Near-RT FlexRIC connects to the RAN in a way that matches the specifications. FlexRIC supports custom xApps and includes a Key Performance Metrics (KPM) xApp that follows the requirements specified in O-RAN.

4.4 The Network Simulator 3 (NS-3)

NS-3 [14] is a state-of-the-art, free-to-use network simulation framework facilitating research, engineering, and development in complex network scenarios under controlled virtual environments. With a comprehensive suite of features and an in-depth understanding of modern programming principles, NS-3 has become an essential resource for network research

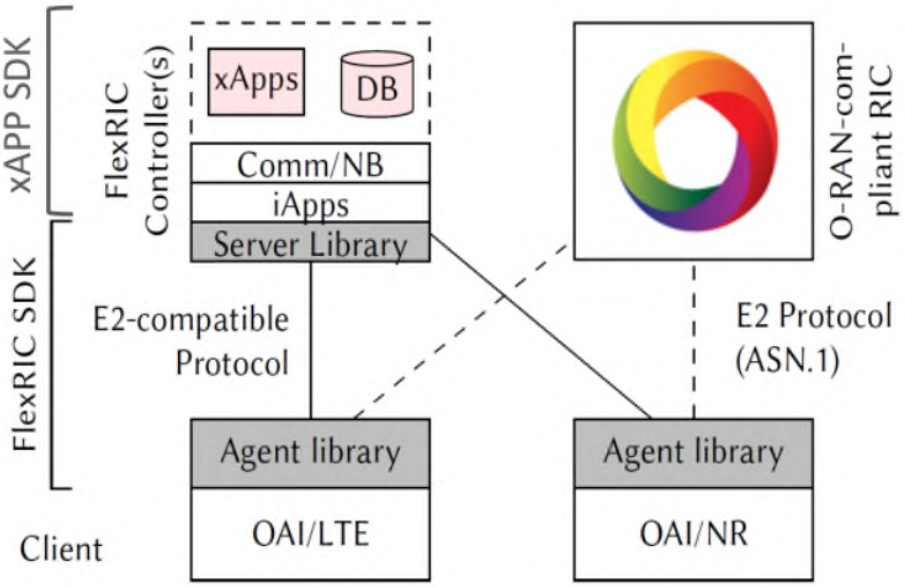


Figure 4.7: FlexRIC and RAN in Open Air Interface

and educational purposes. NS-3 contributes to a comprehensive understanding of network dynamics, functionality, and optimization capabilities through realistic network behaviors, protocols, and topologies. From the simulation of wired and wireless networks to the evaluation of new communication protocols and the testing of novel networking concepts, NS-3 provides an essential platform for innovative, analytical, and insightful work in the dynamic domain of computer networking.

The NS-3 ecosystem is witnessing a significant development with the introduction of the module ns-O-RAN, supporting both 4G and 5G networks [15] [16] [17]. This milestone marks a significant milestone, as it is the first implementation within the framework of NS-3 that fully complies with the stringent specifications mandated by O-RAN. This module has dual functionality, as it implements the 4G protocol stack within NS-3 and integrates the key E2 interface to enable seamless communication with Near-RT RIC [18]. The support of the E2 interface in the ns-O-RAN module is accompanied by two distinctive E2 service models, one for KPM monitoring and the other for RAN control. These service models work together to create a closed-loop control mechanism, enabling dynamic monitoring and orchestration of functions, including traffic control and mobility management. As illustrated in 4.8, the LTE or NR device connects with SCTP to the Near-RT RIC via the e2sim module [19]. All messages passed between the RAN and the Near-RT RIC are encoded and decoded in ASN.1.

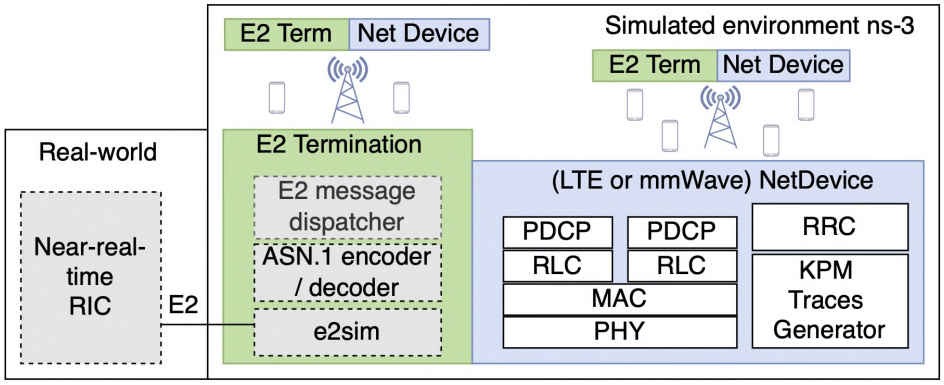


Figure 4.8: NS-3 4G and 5G with Near-RT RIC

4.5 ML Tools and Platforms

4.5.1 NumPy

NumPy [20] is an open-source Python library built for numerical and scientific computing. It introduces and optimizes operations on multi-dimensional arrays, which are highly efficient data structures that store numerical data efficiently for mathematical and statistical manipulation. Additionally, Numpy introduces optimized linear algebra operations for performance and memory efficiency. Its versatility and performance have led to widespread in data science and ML.

The seamless integration of NumPy with other libraries such as SciPy, Matplotlib, Pandas, PyTorch, and TensorFlow poses one of its core strengths. NumPy forms an ecosystem for data analysis, visualization, and computing, enabling researchers and developers to work with numerical data and complex computations in Python.

4.5.2 Pandas

Pandas [21] is an open-source Python library widely used by data scientists and developers due to its robust data processing and analysis capabilities. Pandas is based on NumPy and introduces two distinct data structures, the Series and the DataFrame. The Series is a single-dimensional labeled array that supports a wide range of data types, and the DataFrame is a two-dimensional table-like data structure that divides data into columns and rows, much like a spreadsheet or an SQL table.

Pandas is crucial in machine learning since it enables efficient data cleaning, transformation, and exploration. Pandas allows and simplifies data prepossessing, handling missing

data, dataset aggregation, filtering, grouping, and data visualization using libraries like Matplotlib. One remarkable feature is its versatility when loading data from various file formats and its ability to perform complex mathematical and statistical operations on large datasets. Finally, Pandas simplifies the data analysis workflow and enables efficient data prepossessing, creating ideal conditions for ML.

Pandas can efficiently handle big datasets with various lazy evaluation and memory optimization techniques. It also provides flexible indexing and selection methods, allowing users to access, manipulate, and analyze data. Whether loading data from various file formats, performing complex data operations, or generating summary statistics, Pandas simplifies the entire data analysis workflow and empowers users to derive valuable insights from their data.

4.5.3 TensorFlow and Keras

TensorFlow [22] is an open-source end-to-end ML platform that enables users to create ML models and deploy them on different devices ranging from servers and edge devices to the cloud. TensorFlow helps with data automation, model tracking, performance monitoring, and model retraining and inference. It is used for training and inference of deep neural networks and is available in Python, JavaScript, C++, and Java [23].

Keras is the high-level API integrated into TensorFlow that provides an interface for approaching and solving ML problems with a focus on deep learning. Keras delivers an approachable and highly productive interface, covering every step of the ML workflow. Keras offers vast features ranging from data processing to hyperparameter tuning to deployment. Keras also focuses on enabling fast experimentation [24].

KerasTuner [25] is an easy-to-use, scalable hyperparameter optimization framework that efficiently searches the tuning hyperparameters within a specified search space. KerasTuner can either leverage its already implemented search algorithms or allow the integration of experimental algorithms as extensions to uncover the optimal hyperparameters for a given model and dataset. Finally, KerasTuner works with Python and TensorFlow.

4.6 Docker

Docker [26] is a platform that simplifies software application development, deployment, and management. Docker utilizes containerization technology to encapsulate applications

alongside their dependencies into isolated and lightweight environments called containers. Docker containers come from docker images, constituting the blueprints of an application's environment required for deployment. With docker images, developers can package their applications with necessary libraries, configurations, and runtime environments, ensuring consistent behavior across different host machines. Dockerfiles describe the process of creating a docker image. Dockerfiles are text-based configuration files that specify a set of instructions describing the process followed to assemble an image layer by layer. Therefore, as observed in 4.9 a Dockerfile builds a docker image that, when deployed, becomes a docker container.

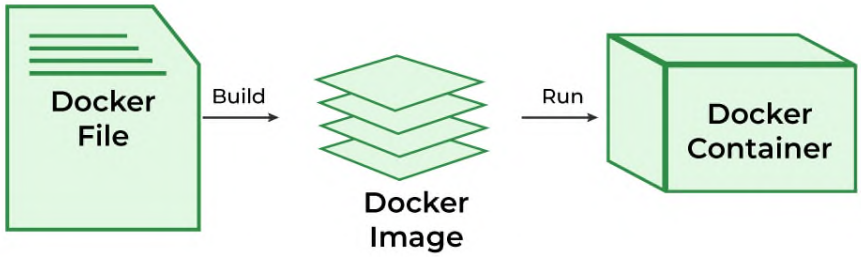


Figure 4.9: Dockerfile to Image to Container

Docker abstracts away the operating system and the underlying hardware, making software development more versatile. Developers can develop the software only once since it can run anywhere. Additionally, docker is lightweight, bringing efficiency to operations. Containers run as separate processes in the user space that share the host's operating system's kernel, Figure 4.10, outperforming virtual machines. Docker is ideal for microservices architectures and continuous integration/continuous deployment (CI/CD) pipelines since it offers performance advantages in container startup times and resource utilization. Docker also supports layering, where a base image can be reused in different containers, reducing storage requirements while minimizing network bandwidth usage. Finally, a little less-known feature of docker is its support for orchestration. Like Kubernetes, docker supports deployment on clusters where software is managed and scaled across container clusters.

4.7 Redis

Redis cache or Redis [27] is an open-source key-value data store optimized for data retrieval. To achieve optimal performance when accessing frequently accessed data, Redis

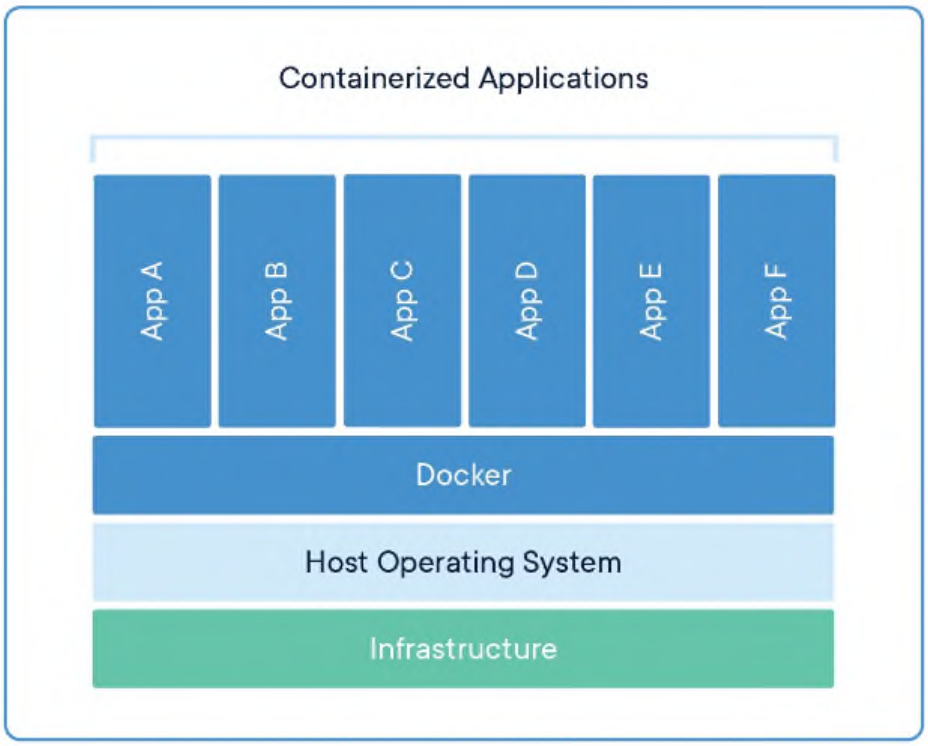


Figure 4.10: Single Machine Hosting Multiple Docker Containers

stores data in memory. Its advantage is that it excels in speeding up applications since it replaces the resource-intensive storage systems with the main memory. The in-memory nature of Redis allows for fast read and write operations, making it ideal for session management, dynamic caching, and real-time analytics. Redis also supports data persistence, replication, and clustering, making it suitable for caching and as a primary data store in specific scenarios.

Redis also supports message passing and queuing with the so-called message brokers. The message broker provides streams, lists, sorted sets, and publish and subscribe utilities. Redis streams are communication channels utilized when building streaming architectures. Additionally, Redis streams are the perfect solution for event sourcing as log-like data structures for persisting data. Redis publish and subscribe, as seen in Figure 4.11, is a lightweight messaging protocol designed to broadcast notification messages in environments where minimal latency under high throughput is required. Finally, Redis lists and sorted sets are the basis required when implementing message queues.

The versatile nature of Redis and its low latency and extensive ecosystem make it the ideal candidate solution for in-memory data caching. Its usage is vital since it enables application performance optimization and scalability.

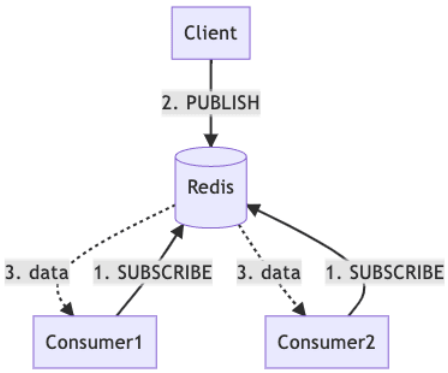


Figure 4.11: Single Machine Hosting Multiple Docker Containers

Chapter 5

Active Queue Management

5.1 Introduction

The fifth-generation (5G) technology promises to revolutionize how we communicate and interact with the world around us. Unlike its predecessors, 5G is not just about faster speeds but also about providing a platform for a range of innovative services. To achieve this, 5G has introduced three service categories [28], called Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and Massive Machine-Type Communications (mMTC). eMBB supports high-speed data applications and services, URLLC focuses on applications that demand ultra-reliable and low-latency communication, and mMTC enables communication between loads of devices while consuming little energy and transmitting at low data rates.

Modern networks employ large buffers for efficient resource utilization, affecting these services with the bufferbloat phenomenon [29]. As network buffers become excessively large, low-latency-sensitive flows face prolonged delays and unavoidable sojourn times within these buffers, leading to high latency and consecutively degraded network performance. High latency reduces the available bandwidth for eMBB, undermines the low-latency requirements of URLLC, and causes network congestion and increased network resource consumption in mMTC. Thus, bufferbloat presents a noteworthy issue in modern networks, resulting in delays, congestion, and suboptimal overall performance. AQM is a technique used to prevent bufferbloat by actively managing the size of the buffers in network devices and entities. It aims to replace traditional passive queue management techniques, such as drop-tail and drop-head, to manage increasing network congestion by maintaining buffers at a reasonable size,

preventing them from becoming extensively large and causing congestion or delays. The literature has introduced several AQM algorithms to manage queues across different protocol stack layers within the 5G architecture. However, many of these algorithms primarily target 5G's monolithic deployment, often neglecting its disaggregated characteristics. Regarding disaggregation, the network protocol stack layers, including their functions and corresponding queues, are split between separate network entities. As a result, these algorithms may prove inadequate in disaggregated network deployments.

5.2 Contribution

As the 5G specification does not address AQM in disaggregated deployments, it is crucial to integrate it within the 5G framework or consider it for future generations of cellular networks. Consequently, our research focuses on formulating a comprehensive approach to manage AQM algorithms within 5G and beyond networks. We leverage AI/ML within each RIC to tackle challenges in disaggregated network deployments. To showcase the effectiveness of using AI/ML in such environments, we evaluate an AQM algorithm where the information exchanged between the 5G network entities is forecasted. Subsequently, we evaluate the performance of this AQM algorithm in monolithic and disaggregated 5G networks, utilizing either exchanged or forecasted information, to measure the degree of improvement achieved.

5.3 Related Work

All AQM algorithms operate on the fundamental principle of minimizing the number of packets in the queues by discarding packets without data starving the transmission channel. Research has shown that maintaining small queue sizes can significantly reduce sojourn times, resulting in lower latency [30]. The literature contains a wide range of AQM algorithms that operate per-queue independently of any communication between multiple 5G layers that may reside in separate network entities, thus eliminating the need for coordinated decision-making. Such examples are Random Early Detection (RED) [31], Controlled Delay (CoDel) [32], Fair Queuing CoDel (FQ-CoDel) [33] and Proportional Integral Controller Enhanced (PIE) [34]. The application of RED in cellular networks, which probabilistically

discards packets at the RLC network stack layer, has been evaluated in [35]. Meanwhile, [30] presents the adoption of CoDel in cellular networks, where packets are dropped based on their sojourn time, reducing the overall latency. These algorithms function in a queue-agnostic manner, handling each queue independently without requiring additional information.

While algorithms like these are adequate, they may not be optimal, as they do not consider the network as a whole. Addressing this limitation requires the development of more sophisticated algorithms that need cross-layer communication for coordinated decision-making. Examples of such algorithms include Stochastic Fair Queuing (SFQ) [36], 5G Bandwidth Delay Product (5G-BDP), UPF-SDAP Pacer (USP), and Dynamic RLC Queue Limit (DRQL) [37]. Rather than relying on dropping packets, these algorithms limit transmission buffer sizes to manage network congestion. In [38], the successful implementation of SFQ in LTE networks, also known as Dynamic RLC Queue Management, has been reported and demonstrated promising results. In SFQ, communication between the RLC and PDCP layers is necessary. 5G-BDP, on the other hand, necessitates communication among the MAC, RLC, and SDAP layers. In this regard, the MAC layer interfaces with the RLC layer, while the SDAP layer interfaces with the MAC layer. Moreover, USP mandates communication between the SDAP and RLC layers and the User Plane Function (UPF) in the 5GC. UPF communicates with the SDAP layer, and in turn, the SDAP layer communicates with the RLC layer.

Finally, DRQL relies on the communication between the SDAP and RLC layers, as it operates by having the SDAP layer continuously querying the RLC layer for its buffer's maximum allowed capacity and occupancy. This information helps determine whether to forward packets or not. As the RLC buffers fill up, the MAC layer extracts as much data as the radio channel can support. When the MAC pulls a full RLC buffer, it becomes starved, indicating that it can handle more data and leading to an increase in the RLC buffer's maximum allowed capacity. However, if the MAC layer extracts only a portion of the data from the RLC buffer, leaving some data in the buffer indicates the necessity to reduce the maximum allowed capacity to match the radio channel capacity. While this paper concentrates on the DRQL algorithm due to its simplicity, the proposed scheme can apply to any AQM algorithm that necessitates cross-layer communication between layers located in separate network entities.

5.4 Proposed AQM Solution

5.4.1 RAN Requirements and Modifications

The integration of most AQM algorithms into 5G-RAN requires some minor architectural modifications. Despite the typical 5G-RAN architecture only allowing for RLC queues at DU entities, the exploitation of several AQM algorithms requires the existence of SDAP queues at CU entities, as Figure 5.1 depicts.

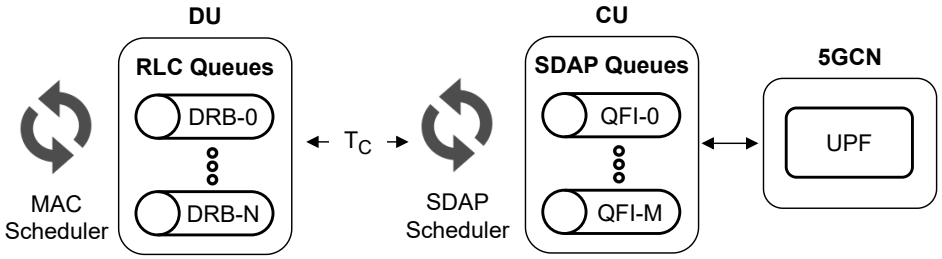


Figure 5.1: RLC and SDAP queues in disaggregated gNB.

Each CU has a set of SDAP queues mapped to different quality of service (QoS) flows, each identified by a unique QoS Flow Identifier (QFI). Each downlink packet is tagged with its appropriate QFI by UPF, according to its Packet Detection Rule (PDR), and then inserted into the respective SDAP queues at CU. The SDAP scheduler is in charge of forwarding these packets to the appropriate Data Radio Bearer (DRB) at the RLC layer of DU. Furthermore, as we have already mentioned, some AQM algorithms rely on cross-layer communication, which takes place between protocol stack layers situated on different CU and DU entities. In DRQL, for instance, communication between SDAP at CU and RLC at DU is necessary. In the case of a monolithic gNB, CU and DU entities coexist on a single host machine, facilitating almost instantaneous communication between them. In contrast, a disaggregated gNB might separate CU and DU entities onto different network nodes, unavoidably introducing communication delay between them. Hence, a disaggregated gNB can significantly impact the performance of latency-sensitive AQM algorithms.

Our proposed solution for AQM utilizes AI/ML to address the challenges posed by the communication delay in disaggregated gNB deployments. We adopt the SDN approach and O-RAN specification, according to which 5G-RAN decouples part of the control plane and moves to the RIC entities that exploit AI/ML. To avoid the delay introduced by the communication between CU and DU, the RIC entities can predict the state of DU for decision-making

at CU rather than relying on their direct, high-latency information exchange. To be more precise, RIC entities can monitor the behavior and status of the DU, such as fluctuations in its queues, and make predictions about its future state by analyzing historical patterns. Then, CU leverages the forecasts produced to make informed decisions on packet forwarding. To thoroughly assess the effectiveness of our approach, we conducted a comprehensive evaluation of the DRQL AQM algorithm in a disaggregated RAN deployment, which was previously evaluated exclusively in the context of a monolithic gNB, as documented in [37].

5.4.2 RIC Assisted DRQL

As with most AQM algorithms, DRQL establishes seamless cross-layer communication between CU and DU. For successful downlink traffic transmission from CU to DU, DRQL necessitates precise measurements of RLC queue status. These measurements provide information about the actual sizes and limits of the RLC queues, enabling efficient packet pacing and resource allocation. The efficient DRQL operation in a disaggregated gNB requires time-critical delivery of the RLC queue measurements from DU to the SDAP layer at CU. Although CU can request this information from DU using its control interface, due to communication delay, the information provided refers to a previous RLC queue status, hindering the real-time decision-making at CU. Denoting the symmetrical and biderictional communication delay between CU and DU as T_c, the information that CU requested at $t_0 - 2T_c$ and received at t_0 was generated by DU at $t_0 - T_c$, illustrated in Figure Figure 5.2. Consequently, the SDAP scheduler might make suboptimal decisions regarding packet forwarding, relying on outdated information.

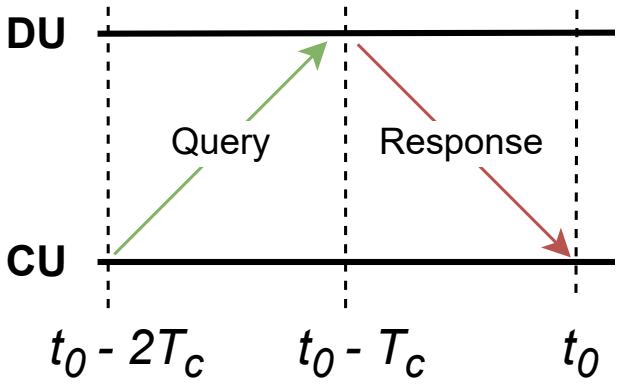


Figure 5.2: Communication latency T_c between CU and DU.

In our approach, instead of SDAP repeatedly querying measurements of RLC queue status whenever a packet needs forwarding, we employ AI/ML to predict these measurements, analyzing historical data. Periodically, DU informs the Non-RT RIC and the Near-RT RIC, of its current RLC queue measurements. It's worth noting that 5G network control loops determine the use of different RIC entities. There are near-real time loops, carried out by CU entities and the Near-RT RIC, last between 10ms and 1s, while non-real time loops, executed by the Non-RT RIC, last longer than 1s[1].

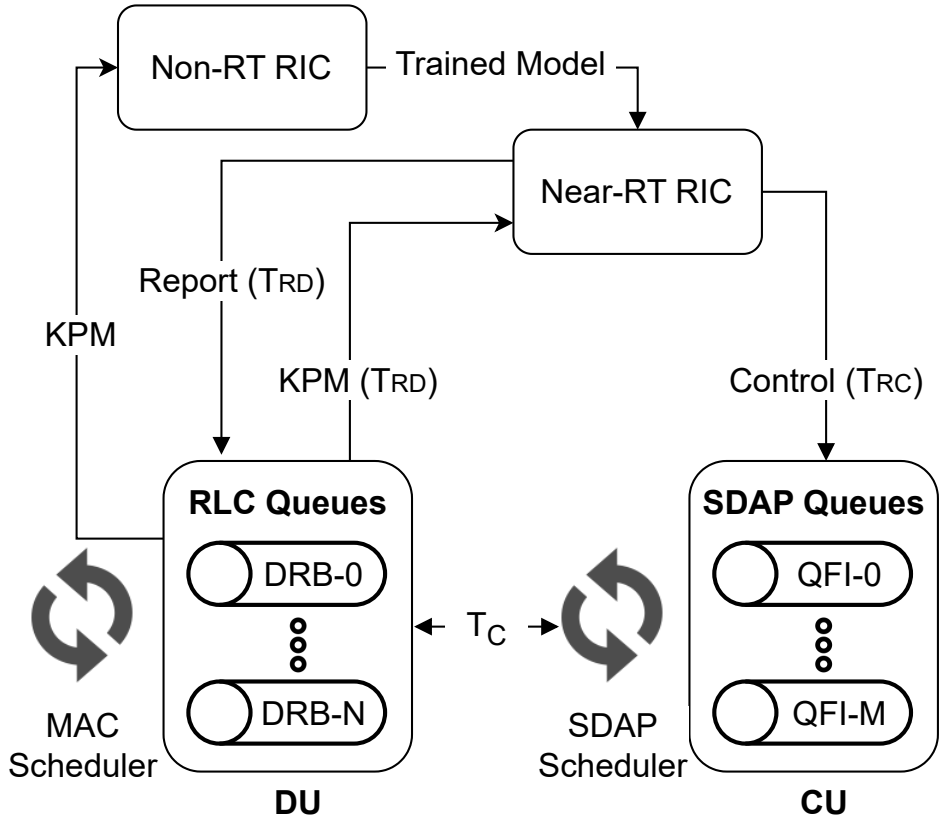


Figure 5.3: Interaction between disaggregated gNB and RIC.

The Non-RT RIC trains the AI/ML models using the collected measurements and transfers them into the Near-RT RIC. Its data collection and, therefore, training processes are performed in a monolithic network deployment, ensuring the specified, proper DRQL operation and valid monitoring of the anticipated fluctuations in the RLC queues whenever the MAC scheduler extracts data, every Transmission Time Interval (TTI). The Near-RT RIC utilizes the trained AI/ML models to make informed control decisions through its microser-

[1]These two loop types are Loop-2 and 3, respectively, since Loop-1 is the real-time control loop performed by DUs and RUs and operates on timescales of 1ms.

vices, named xApps, based on the current received measurements, as Figure 5.3 depicts. To collect the RLC queue measurements, Near-RT RIC continuously requests E2SM-KPM [39] using report messages received from the DU. The Near-RT RIC forecasts the RLC queue fluctuations and notifies the CU of the anticipated RLC queue status. The CU can use the data received from the Near-RT RIC to avoid the T_c latency.

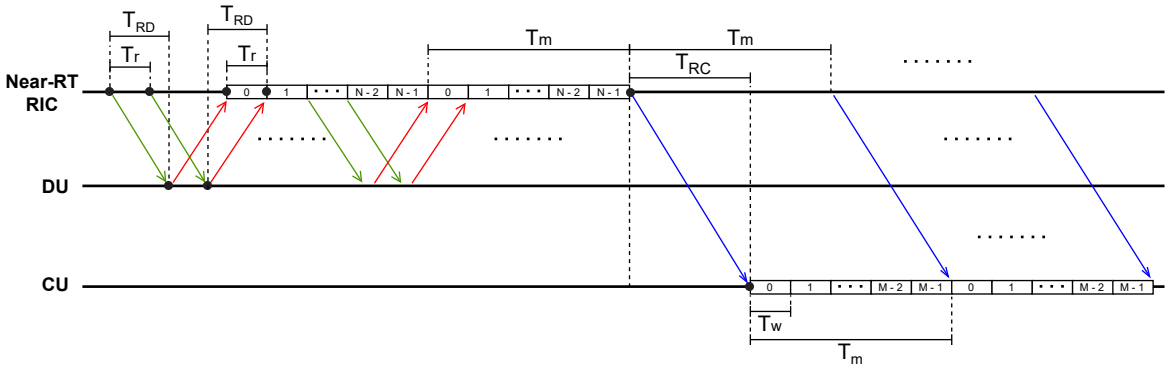


Figure 5.4: Timing of Near-RT RIC, CU and DU.

Crucial factors for efficient prediction is the communication latencies between the Near-RT RIC and the CU or DU, named T_{RC} and T_{RD} respectively, that affect model training and inference. As the left part of Figure 5.4 illustrates, we assume that Near-RT RIC sends report messages to DU every T_r to obtain the RLC queue measurements. Each T_r, with a T_{RD} latency, these report messages reach DU. When a report request is received, DU responds to Near-RT RIC with the requested information. The response takes again T_{RD} time to reach the Near-RT RIC. Given that the Near-RT RIC discovers the RLC queue measurements requested at t_0 by $t_0 + 2T_{RD}$, our model should account for the additional $2T_{RD}$ latency. It is also worth noting that the Near-RT RIC receives responses periodically at intervals of T_r.

Then, as it is illustrated in the right part of Figure 5.4, the Near-RT RIC periodically uses N responses from the DU, collected over a duration NT_r, in order to predict M values related to the RLC queue status. The inference process duration of the model, named T_m, is factored in when making predictions, since the accurate RLC queue predictions at the precise moment must be ensured. The same with the delays T_{RD} or T_{RC} introduced by the communication between Near-RT RIC and DU or CU respectively. In summary, the Near-RT RIC accounts for T_{RD}, T_{RC} and T_m, producing M predictions that reflect a span of T_m after a total elapsed time of $T_{RD} + T_m + T_{RC}$ from the most recently received RLC measurement. The CU receives these predictions with a period of T_m. Hence, the CU receives multiple predicted values,

collectively spanning a time interval of T_m, with each prediction representing a duration of T_w of the expected RLC queue status.

Finally, since the Non-RT RIC conducts model training on dataset extracted in real-time timescales, the trained models can exclusively perceive and forecast the fluctuations in the RLC queues that transpire within these particular timeframes. However, the SDAP scheduler diverges from the received predictions by not restricting packet transmission to these timescales but instead forwards packets as soon as they become available at the SDAP queues. As a result, for a point in the T_w interval between two consecutive predictions given by Near-RT RIC, the RLC queue status must be estimated by CU on its own. Since the RLC status information comprises numerical values, our proposed implementation considers a linear variability of the RLC queue status across multiple packets within the T_w time window and provides predictions with their corresponding confidence intervals. Given the received values x_n and x_{n+1} representing the RLC status at time points t_n and t_{n+1}, where $t_{n+1} - t_n = T_w$, the CU can perform linear interpolation to estimate the RLC status, x, for forwarding a packet at t within the t_n and t_{n+1} interval. This estimation is achieved through the formula $x = x_n + \frac{x_{n+1}-x_n}{y_{n+1}-y_n}t$ where x is the estimated RLC status at time t.

5.5 Evaluation Framework

This section dissects the evaluation parameters, the 5G configuration, and the measurement methodologies employed to simulate and assess our proposed solution. It's worth highlighting that the simplicity of our experimental setup is a result of constraints in hardware and computational resources. Furthermore, it's essential to emphasize that the research's primary focus is not solely on the machine learning approach but rather serves as a proof of concept, with the potential for expansion to accommodate more intricate environments.

5.5.1 5G Configuration

The OAI5G-DRQL is an extension of the original OAI5G's [40] 5G-RAN implementation, where we introduced several enhancements to the SDAP layer, including multiple SDAP queues and a Round-Robin scheduler for packet forwarding and modified the RLC layer, enabling it to dynamically adjust the limits of its queues based on the incoming packet volume. We extended the currently implemented KPM sent from DU to CU to accommodate addi-

tional RLC queue-specific information, as expected in E2SM-KPM. The Radio Frequency (RF) Simulator, provided by OAI5G, simulates the physical layer of 5G-RAN and allows testing without an RF board, while NR-UE emulates a COTS UE connected to 5G-RAN. A Pyhton3 Non-RT RIC performs data preprocessing and model training, while FlexRIC [41] is the Near-RT RIC as it connects to the 5G-RAN nodes via the E2 interface, allowing monitoring and control via report and control messages. Additionally, we deploy the 5G-CN using OAI-5G Core Network [42]. To accurately configure delays between network entities, we deploy the CU, DU, UE, Near-RT RIC, and the Non-RT RIC on a shared computational NITOS node, utilizing traffic control (`tc`). Finally, a separate node is hosting the 5G-CN.

5.5.2 ML Configuration

In the context of predicting the behavior of RLC queues in a supervised learning manner, creating a comprehensive dataset under a range of network conditions is imperative. The developed dataset should encompass bufferbloat scenarios and provide insights into the behavior of network entities when operating in the presence of an AQM algorithm. In our case, we employ `iperf3` [43] and `ping` [44] to incorporate bufferbloat into our dataset by creating simultaneous high-throughput and latency-sensitive downlink traffic with varying throughput and interval parameters for encapsulating different network application requirements. By adjusting the downlink transmission rate in `iperf3`, we replicate diverse network applications, including file downloading, video streaming, and cloud gaming. Additionally, varying the `ping` interval allows us to emulate different network requirements associated with applications like online gaming and video conferencing.

More specifically, the developed dataset contains the RLC's queue limits and actual sizes, sampled at T_r = 1ms under DRQL in monolithic network deployment, distinguished by $T_c = 0$. Employing combinations of different parameters in `iperf3` and `ping` creates distinct network downlink conditions. These combinations include UDP traffic at [1, 10, 40, 100] Mbps alongside ICMP transmission with intervals of [1, 10, 100] ms, ensuring the accuracy and relevance of the trained models' predictions across diverse applications. Therefore, the compiled dataset includes entries representing the RLC's queue limit and actual size, along with the respective timestamps of these observations.

Our preliminary studies and experiments indicate that Long Short-Term Memory (LSTM) models outperform Seasonal Autoregressive Integrated Moving Average (SARIMA) and

Random Forests for time-series forecasting of RLC's queue fluctuations in DRQL. Thus, after Non-RT RIC preprocesses the created dataset, it should develop and train a pair of time series forecasting LSTM models to predict forthcoming fluctuations of RLC's queue limits and actual sizes, respectively. Additionally, the Non-RT RIC deploys both models to the Near-RT RIC, facilitating decision-making through forecasting.

In the preprocessing stage, the Non-RT RIC scales the data in the [0,1] range using `MinMaxScaler`, provided by scikit-learn [45], and splits the dataset into 70% training, 20% validation, and 10% test sets. The feature-label pairs are derived using a rolling window approach, preserving causality by refraining from data shuffling. The input features consist of N data points, representing a time duration of $N * T_r$, while the output labels comprise M data points, equivalent to a duration of $M * T_w = T_m$, as previously noted. Therefore, for a given set of features N, the labels become M following a time interval of $T_{RD} + T_m + T_{RC}$ to account for DU to Near-RT RIC, model inference, and Near-RT RIC to CU delays. Finally, to ensure consistency and simplicity in our feature-label alignment, we define $T_w = T_r$.

The Non-RT RIC employs TensorFlow [46] with Keras [47] to handle the development and training of the LSTM models required, while KerasTuner optimizes training hyperparameters for precise forecasting results. We employ `tensorflow.keras.sequential` for both models to stack multiple layers. We specify that the input layer for the two models is an LSTM layer consisting of N nodes. KerasTuner determines the parameters for the subsequent layers, including their quantity and configurations. Additionally, we add a dropout layer, in the range of 0 to 0.5 and a step size of 0.1, that helps prevent overfitting by randomly deactivating a fraction of neurons during training. The final layer is a dense layer, which is fully connected and produces predictions aligned with the dimensions of our target labels, denoted as M in this context. We utilize the Mean Squared Error (MSE) as our chosen loss function, effectively penalizing errors and providing remarkable overall model performance. Finally, the choice of optimizer algorithm is pivotal in shaping how errors propagate through the network. The Adam optimizer is an excellent choice since it strikes an optimal balance between learning the most relevant and less frequent features.

It's important to emphasize that accurately forecasting the precise limits and actual sizes of RLC's queues is complex, necessitating supplementary data from all CU and DU entities. However, in our straightforward proof-of-concept implementation, minor discrepancies in our predictions only lead to marginal reductions in throughput or increases in latency due to

minimal packet drops within a single Transmission Time Interval (TTI). Furthermore, determining the model's inference duration, T_m, is a formidable task, and therefore, we calculate it as the mean inference duration obtained from multiple observations of model inference. Finally, the significance of available hardware and network infrastructure, influencing the inference duration and latency between CU and DU, along with their corresponding Near-RT RIC, respectively, becomes evident in prediction accuracy since lower inference duration and reduced latency create a shorter rolling window for predictions. Given additional computing power, we could facilitate parallel model inference, thereby reducing the forecasting horizon and enhancing result accuracy. Nevertheless, within this uncomplicated implementation, such enhancement remains unnecessary.

5.5.3 Conducted Experiments

The experiments investigate the impact of coexisting high-throughput and latency-sensitive traffic on overall network performance, particularly in achieved throughput and latency, within a 5G network environment. Downlink User Datagram Protocol (UDP) traffic is generated from 5G-CN at a rate of 50Mbps, while at the same time, latency-sensitive Internet Control Message Protocol (ICMP) packets are also transmitted downlink to the UE at 100ms intervals. The UDP traffic measures throughput, whereas ICMP traffic measures Round-Trip Time (RTT), which escalates as the packet's sojourn time increases. The UE establishes a single Packet Data Unit (PDU) session with a distinct QFI for UDP or ICMP packets, and a single RLC Acknowledge Mode DRB is responsible for transmitting and receiving both QoS flows.

We conduct all two sets of experiments in a disaggregated 5G-RAN deployment. The first set of experiments contrasts two approaches in two distinct delay settings, either with zero or non-zero delay between the CU and DU entities, T_c. In this approach, we use the original OAI5G implementation, where the SDAP forwards packets as soon as they become available to the RLC layer without querying RLC's status. It's important to note that the original OAI5G implementation utilizes a drop-tail approach for AQM in the RLC queues. This approach restricts the queue size and discards incoming packets that surpass this limit. We adjusted the RLC queue limit to increase occupancy and assess network performance with more congested queues. The second approach utilizes the OAI5G-DRQL implementation without the RIC's assistance, where CU and DU communicate directly. We tested both implementations with

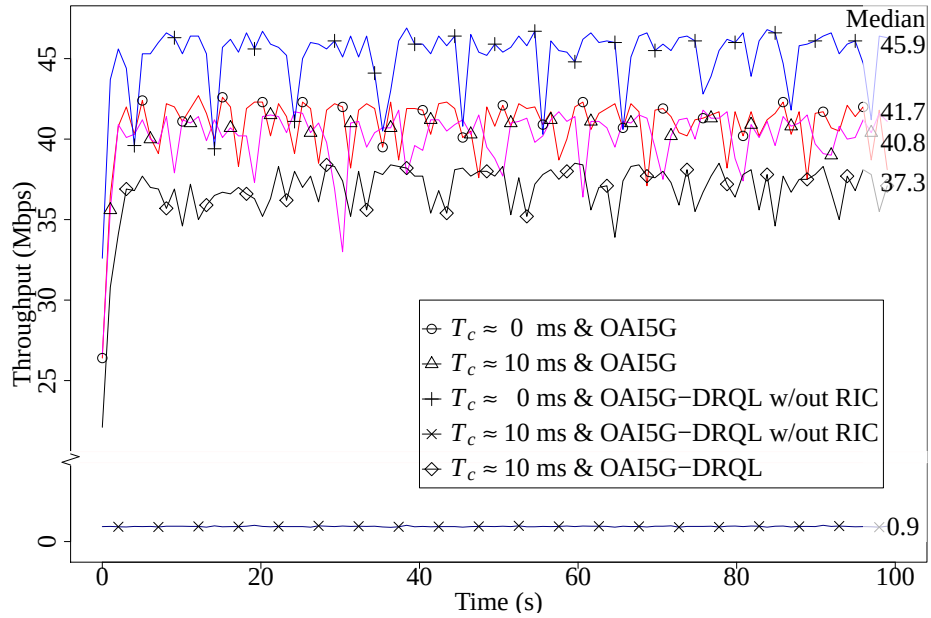


Figure 5.5: Maximum achieved downlink throughput.

$T_c \approx 0$ or $T_c \approx 10$ms.

In the second set of experiments, the Near-RT RIC utilizes the LSTM models to provide the CU with the anticipated RLC's status in advance, eliminating direct CU-DU communication for KPM retrieval. A delay of $T_{RD} \approx 10$ms is introduced between Near-RT RIC and DU, in addition to the existing $T_c \approx 10ms$ delay between CU and DU. The LSTM model analyzes 1000 data points of RLC queue statistics, which correspond to a duration of $1000T_r = 1000$ms and predicts how the RLC queues will behave during the next 400ms, given that the inference time of the model is $T_m \approx 400$ms and there is negligible latency $T_{RC} \approx 0$ between Near-RT RIC and CU. Thus, its primary objective is to generate a forecast of 400 values, following the $T_m \approx 400$ms of inference and representing a time horizon of 400ms after the initial 1000ms of historical data.

5.5.4 Experiment Results

Figure 5.5 shows a graphic representation of the throughput achieved for each set of experiments. It is evident from the results that each approach exhibits a relatively low initial throughput, which gradually stabilizes and eventually achieves a higher throughput level. The observed low initial throughput is because the RLC queues are empty at the beginning of the experiments. As the queues fill up, the MAC scheduler can pull more data from the RLC queues, resulting in higher throughput. The inherent capacity limitations of the wireless

channel are a ceiling that stabilizes the maximum achievable throughput.

In the original OAI5G, since the CU maintains a consistent packet forwarding rate, throughput remains largely unaffected by the $T_c \approx 0$ and $T_c \approx 10$ms delay settings, illustrated as the circle-pointed and triangle-pointed lines, respectively, averaging between the median of these two lines at around 41.2Mbps.

The original monolithic OAI5G-DRQL, emulated in a disaggregated environment with $T_c \approx 0$ and without employing RIC, outperforms the original OAI5G succeeding a throughput almost equal to 46Mbps, illustrated as the median of the cross-pointed line. This throughput gain is because the RLC queues fluctuate according to the radio channel capacity, which causes the MAC scheduler to process the exact amount of data required in a TTI without starving or overfilling the RLC queues.

On the other hand, the introduction of communication latency between CU and DU, equal to $T_c \approx 10$ms, causes OAI5G-DRQL without RIC assistance to suffer in terms of throughput, achieving nearly 1Mbps, represented as the median of the x-pointed line. Every packet reaching the CU undergoes the high-latency DRQL request-response process in SDAP, causing a bottleneck at CU, which leads to RLC queue starvation and underutilization of the radio channel, ultimately reducing overall throughput.

On the contrary, OAI5G-DRQL with RIC effectively avoids the latency-heavy, $T_c \approx 10$ms, communication between CU and DU, resulting in a throughput almost equal to 37Mbps, illustrated as the median of the diamond-pointed line. It's worth noting that collocating RIC and DU limits the available CPU resources, which likely reduces overall throughput.

Furthermore, RTT is also examined in the context of high throughput to assess the effectiveness of our implementation. In the original OAI5G configuration, SDAP forwards packets as they become available without employing an SDAP scheduler, which causes a significant disparity in the volume of UDP packets compared to ICMP packets in the RLC queues and causes ICMP packets to lag significantly behind the UDP packets. As illustrated in Figure 5.6 by the circular and triangular pointed lines, this imbalance results in a linear increase in Round Trip Time (RTT) over time, regardless of whether a communication delay exists between the CU and DU. In contrast to the scenario where $T_c \approx 0$ms, when $T_c \approx 10$ms, the rate of RTT growth slightly increases due to an increase in the initial acceleration of accumulation of UDP packets in the RLC queues.

When $T_c \approx 10$ms and DRQL uses RIC and when $T_c \approx 0$ms while DRQL operates

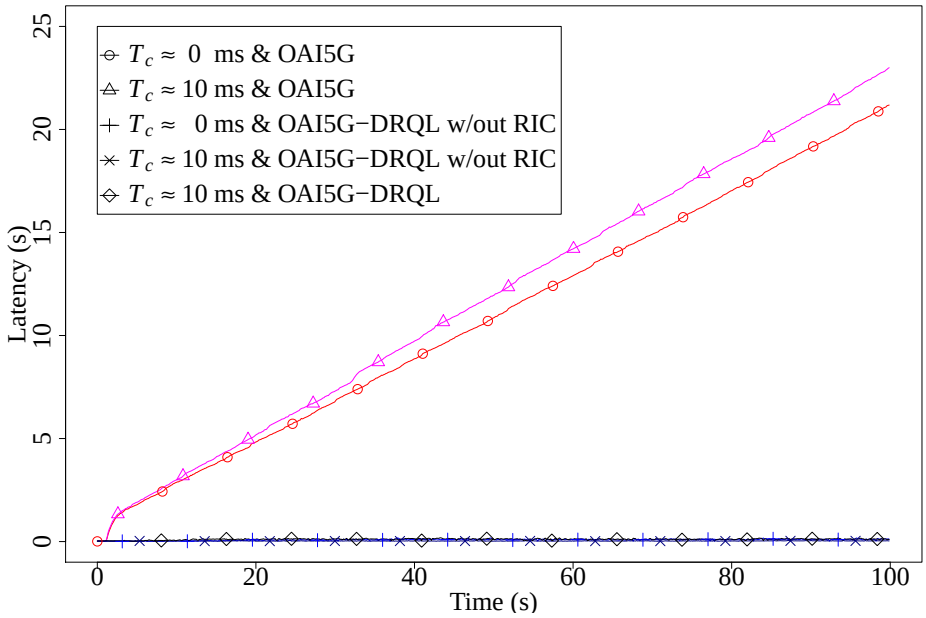


Figure 5.6: Achieved RTT.

without RIC, the DRQL does not experience network latency, which does not impede the SDAP scheduler, ensuring rapid decision-making on packet forwarding. These two cases are illustrated as the diamond-pointed and cross-pointed lines, resulting in approximately 114ms and 84ms average RTT, respectively. The additional 20ms of the average RTT in the former case corresponds to the additional $T_c \approx 10$ms it takes for an ICMP packet to travel from CU to DU and back. The latter case does not experience 0ms RTT since UDP and ICMP traffic compete for the same queues. As counterintuitive as it seems, when $T_c \approx 10$ and DRQL does not use RIC, the overall average RTT decreases to approximately 30ms, depicted as the x-pointed line. Since each traffic flow possesses a unique QFI and the SDAP scheduler operates in a Round-Robin fashion, it systematically forwards UDP and ICMP traffic interchangeably. However, despite the higher arrival rate of UDP packets at SDAP, DRQL forwards them at a notably slower pace, relative to their volume, due to the additional latency, resulting in fewer UDP packets ahead of ICMP packets in the RLC queue, and thus smaller average RTT values.

In summary, the RIC's assistance in OAI5G-DRQL enhances throughput compared to OAI5G-DRQL without RIC, approaching the monolithic deployment while maintaining relatively low RTT values. These improvements enable addressing bufferbloat with improved AQM in disaggregated high-latency deployments.

Chapter 6

Controller Placement

6.1 Controller Placement Problem Statement

Controller placement refers to the positioning of controllers within a network topology. Controllers allow the decoupling of network control from the underlying hardware, increasing management flexibility and enhancing network agility. Controller placement depends on different factors, including the underlying network topology, the traffic patterns of network devices, the device's geographical distribution, and latency and fault tolerance requirements. Optimal placement leads to enhanced network performance and increased scalability.

6.2 SDN

In SDN [48], centralized software controllers manage the behavior and configuration of the underlying network. Controllers make decisions on how to forward and handle data packets over the network topology. The SDN separates the control plane and data plane of network devices. In the traditional SDN model, network devices like switches and routers decouple the control and the data planes, as depicted in Figure 6.1 and described below.

1. The control plane makes high-level decisions about how the network should forward the data packets. Control logic is not distributed across devices but centralized in the software controller. Controllers and network devices communicate with open protocols such as OpenFlow.

2. The data plane performs the actual forwarding of the data packets according to the

decisions made by the controller. Network devices, like switches and routers, focus only on packet forwarding and don't require complex decision-making capabilities.

SDN enables network programmability, simplifying the implementation of new services, enabling dynamic configuration changes, supporting centralized network management, and optimizing the efficiency of network operations.

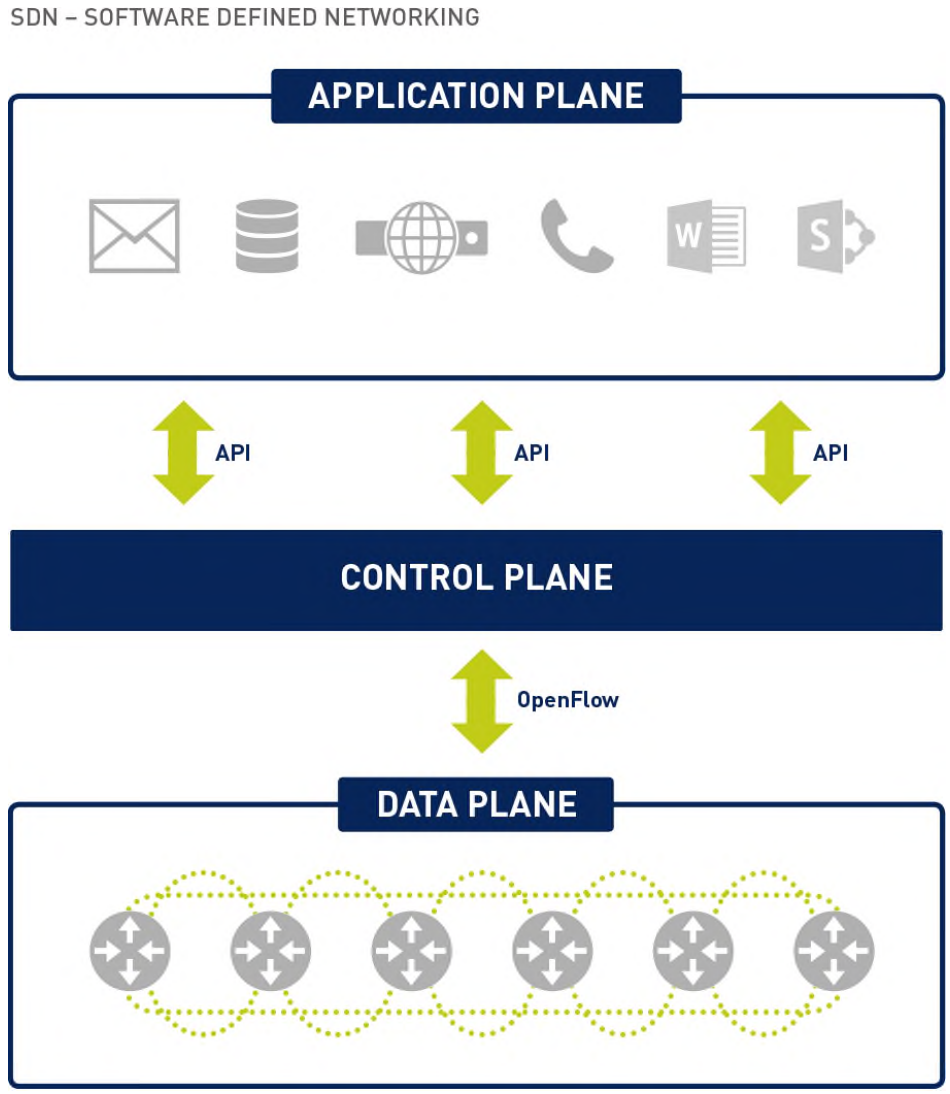


Figure 6.1: Control and Data Planes in SDN.

6.3 NFV

NFV [49] virtualizes and abstracts traditional network functions and services, turning them into software instances that run on servers or virtual machines rather than relying on dedicated hardware devices, depicted in Figure 6.2. It consists of VNFs and virtualization

infrastructure, incorporating virtualization platforms such as hypervisors and orchestration tools. Finally, NFV offers several benefits, including cost savings since dedicated hardware appliances are no longer required, faster service deployment and scalability, and increased agility in responding to changing network requirements.

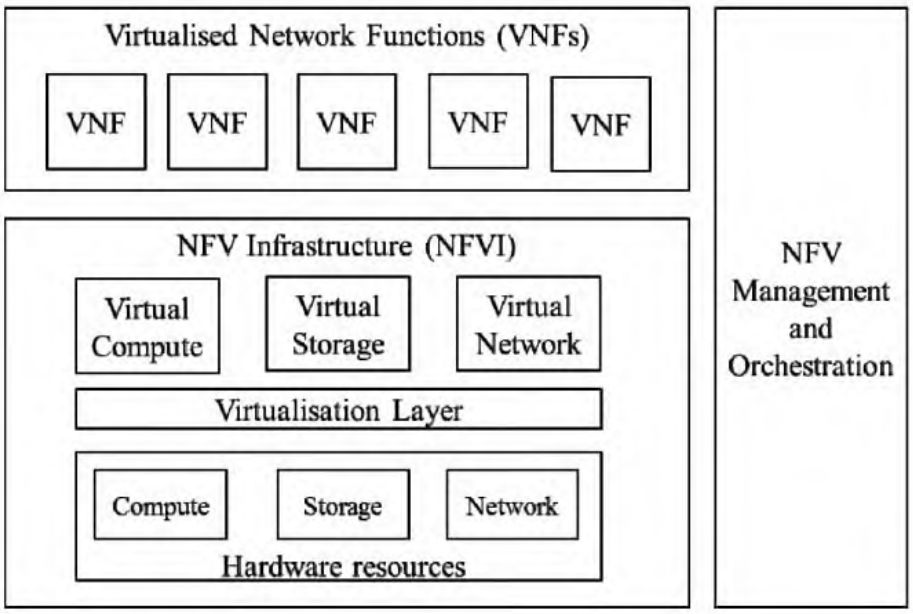


Figure 6.2: NFV with VNF and Hardware Decoupling

6.4 Previous Research

The location and orchestration of SDN controllers and VNF in NFV become critical factors in network performance since they can improve the network's service performance. In SDN, controllers control the data flow and send messages between network devices such as switches and routers. The location of these controllers impacts factors such as latency and fault tolerance, as well as network responsiveness. Typically, controllers are positioned in a way that optimizes data traffic flows, reduces congestion, and provides redundancy to guarantee network availability if some controllers fail. Figure 6.3 depicts an example of different controller placement options in SDN.

The controller placement problem has already been thoroughly investigated and addressed in SDN. It was first introduced and addressed in [50], which focused solely on two metrics, the average and the worst-case latencies. This study, however, overlooks critical metrics, such as bandwidth, failures, and those specific to the controller's application requirements.

In [51], researchers consider controller failure and determine the optimal controller placement for latency reduction in environments where controllers might fail and become unavailable. The research at [52] achieves the optimal placement for synchronization in multi-controller SDN, where controllers preserve the consistency of a network state distributed across multiple controllers through periodic synchronization. The controller locations and their resiliency to network failures affect the aforementioned synchronization.

In NFV, several factors, including the locations and topology of each VNF, are critical for resource allocation, load balancing, and performance. The [53] investigates the VNF placement within distributed edge infrastructure, intending to minimize the overall latency for users accessing their respective VNF. The [54] attempts to decrease the end-to-end communication delay and the overall deployment cost. The [55] investigates optimal decisions regarding the placement of VNFs across the physical hosts with limited computational resources. Various hosts possess varying computational and networking resources, necessitating decisions on VNF deployment across nodes according to their capacity.

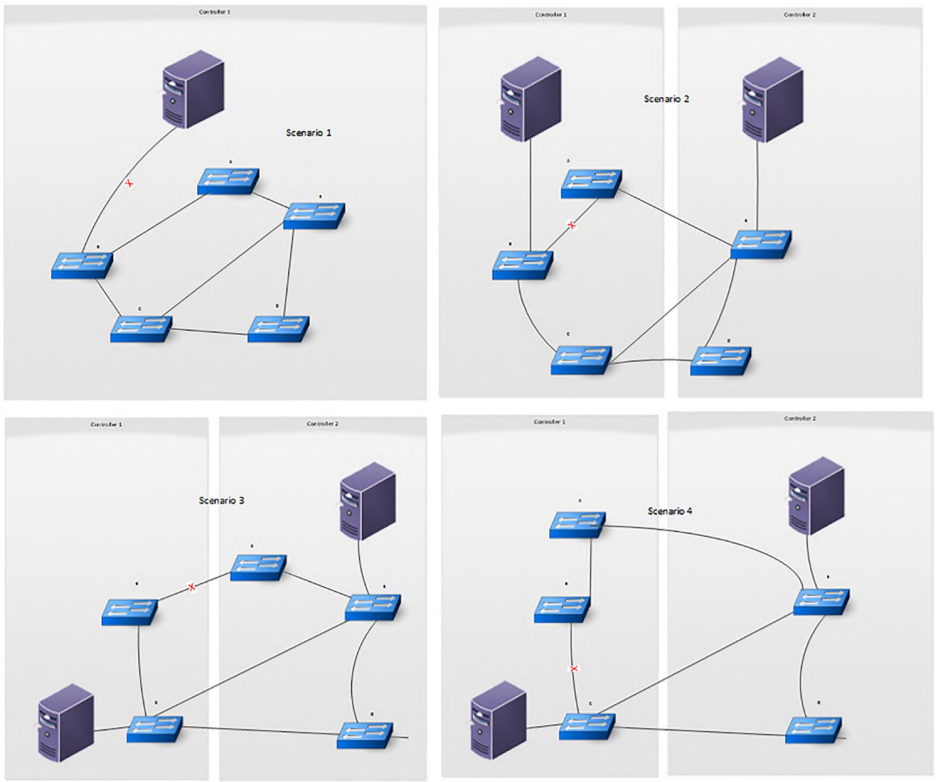


Figure 6.3: Example of Controller Placement Scenarios in SDN

6.5 Linear Programming in Controller Placement

Linear Programming is a robust mathematical approach for optimizing resource allocation in various real-world situations. At its fundamental level, linear programming attempts to identify the most suitable solution from a range of feasible alternatives by maximizing or minimizing the linear objective function in the face of a linear constraint. Whether it maximizes profit, minimizes costs, or optimizes other measurable quantities, the objective function is the end goal of optimization. The linear constraints are the constraints or restrictions that influence the decision variables, and they are often related to resources, capacity, or other operating factors. Linear Programming problems are linear, meaning that the objective functions and the constraints are linear equations and inequalities. Some algorithms for solving linear programming problems include the simplex and the interior-point methods.

In network optimization, particularly in SDN and distributed systems, the controller placement problem poses a significant challenge. It's a question of where the best locations for controllers should be within the network topology to optimize resource utilization, reduce latency, and improve fault tolerance. This complex problem, described as a linear programming model, shows the usage of mathematical optimization to solve real-world problems in network engineering.

The variables in the linear programming model represent the potential location of controllers at particular network nodes. The objective function, to be maximized or minimized, captures several performance metrics, including but not limited to bandwidth, utilization, latency, and fault tolerance, alongside a combination of these metrics.

Practical considerations, such as controller-node mapping, latency, fault tolerance, or bandwidth requirements, describe the linear constraints. These constraints guarantee that the placement chosen agrees with the operational and architectural limitations of the underlying network. For example, constraints may consider factors like communication distance, data transmission latency, and the network's resilience to controller or node failure.

The linear programming model solves the controller placement by providing a systematic approach that balances different goals against each other. The outcome is an informed decision about where to place controllers in the network based on factors like latency, bandwidth usage, and fault tolerance. Finally, linear programming in controller placement highlights the synergies between mathematical optimization methods and contemporary network engineering challenges.

6.6 Navigating RIC Roles in the 5G Landscape

This section outlines the utilization of Near-RT RIC or Non-RT RIC across diverse scenarios within cellular networks while specifying the parameters for setting linear objectives and constraints. Before we dive into each individual RIC use case, let's first introduce a general use case roadmap. In Figure 6.4, we depict each RIC use case outlined in the specification, explicitly indicating whether it incorporates both Near-RT RIC and Non-RT RIC, exclusively uses Non-RT RIC, or solely relies on Near-RT RIC. Since multiple deployment options are available, the RIC usage may vary. There are three distinct deployment options, including ”Non-RT and Near-RT,” which involves using both the Non-RT RIC and Near-RT RIC components; ”Non-RT Training Inference,” which utilizes only the Non-RT RIC; and ”Only Near-RT,” where the use of the Non-RT RIC is redundant. In situations where the Near-RT RIC is employed, the ”Near-RT Mapping” can be configured as either ”Single” or ”Multiple” represented as ”S” and ”M”, respectively. In accordance with the specification, when configured as ”Single,” the Near-RT RIC is designed to establish a connection for monitoring and controlling a single gNB or eNB, ensuring optimal performance with minimal latency. In ”Multiple”, the Near-RT RIC should retrieve KPM from multiple gNB or eNB entities to avoid the additional latency of fetching this information from Non-RT RIC. The designation ”Single/Multiple” or ”S/M” signifies that the Near-RT RIC should establish connections with either a single or multiple gNB or eNB entities based on the employed algorithm and the desired outcome.

We present the KPM as specified by four distinct labels, including ”None”, ”Loose, ”Medium”, and ”Strict” under the ”KPM Specification” field. The label ”None” signifies that the specification does not include any specified KPM for this particular case, whereas ”Loose” indicates that the specification provides some level of detail but does not delve deeply into specifics. The ”Medium” label indicates that the KPM is specified, but other significant details, including the exact source from where they are extracted or their granularity, are not. The ”Strict” label represents that the exact contents, sources, and granularity of the KPM are explicitly specified. Finally, we use the ”Source” field to identify the origin of the KPM for examination. When the specification lacks explicit information regarding the source of certain KPM, we label them as ”gNB.” Nevertheless, when the KPM source is explicitly defined, we use ”CU” to indicate it originates from a CU entity, ”DU” for a DU source, or ”CU/DU” when both entities contribute to the extraction of the required KPM.

Cases	Non-RT and Near-RT	Near-RT Mapping	Non-RT Training Inference	Only Near-RT	KPM Specification	Source
SLA	Non-RT Training and Near-RT Inference	S	-	-	Medium	CU/DU and gNB
Multi-Vendor Slicing	Non-RT to Near-RT Coordination Proxy	M	-	Coordination in Shared Near-RT	None	CU/DU and gNB
NSSI Resource Optimization	-	-	X	-	Loose	gNB
GoB BF	-	-	X	-	Medium and Strict	CU/DU and gNB
bMRO	-	M	X	Training and Inference	Strict	CU
SS Burst Set Configuration Optimization	Non-RT Training and Near-RT Inference	M	X	-	Strict	CU/DU
DMRS and CSI-RS Configuration Optimizations	Non-RT Training and Near-RT Inference	M	-	-	Strict	CU/DU
Optimization of SS Burst Set and DMRS and CSI-RS Configuration	Non-RT Training and Near-RT Inference	M	X	Training (Inference at gNB)	Strict (Previous Cases)	CU/DU (Previous Cases)
L1/L2 Beam Management	Non-RT Training and Near-RT Inference	S	X	Training (Inference at gNB)	Strict	CU/DU or DU Only
Non-GoB BF	Non-RT Training and Near-RT Inference	S	-	-	Strict	DU
MIMO Downlink Transmit Power Optimization MU-MIMO Pairing Enhancement MIMO Mode Selection	-	-	X	-	Strict	DU
QoE	Non-RT Training and Near-RT Inference	S/M	-	-	Medium	CU/DU and gNB
SON	Near-RT SON Functions	M	-	-	Loose	gNB
Mobility Control and Traffic Steering (HO, CA, DC)	Non-RT Training and Near-RT Inference	S/M	-	-	Medium	gNB
RF Channel Reconfiguration Carrier and Cell Switch OFF/ON	Non-RT Training and Near-RT Inference	S	X	-	Strict	CU/DU
Advanced Sleep Mode Selection	Non-RT Training and Near-RT Inference	S	X	-	Loose	CU/DU and gNB
O-Cloud Resource Energy Saving Mode	-	-	X	-	Loose	gNB
User-Centric QoE Connection Policy	Non-RT and Near-RT Proxy	S/M	-	-	None	gNB
CPM	Non-RT Training and Near-RT Inference	M	X	-	Loose	gNB
IoT	Non-RT Training and Near-RT Inference	S	X	-	Loose	gNB

Figure 6.4: RIC Use Cases Roadmap.

6.6.1 Slicing

Network slicing enables network operators to partition and allocate resources to different services. Slicing is a fundamental feature introduced in 5G, allowing for improved network performance and increased customizability, security, and versatility.

Service-Level Agreement (SLA)

A SLA specifies the QoS expected by customers from their suppliers. When an SLA exists, the provided service level is measured, awarding rewards when achieving the expected quality or penalties when not.

As specified in 3GPP, 5G should support slicing, allowing the creation and management of networks that meet custom-designed service requirements. Network slicing is an end-to-end feature that includes the CN and the RAN. Slice requirements mainly focus on throughput, latency, and energy consumption. As illustrated in Figure 6.5, the Non-RT RIC retrieves SLA target agreements from entities such as the SMO or NSSMF, monitors long-term SLA slice performance, and sends A1 policies alongside enrichment information to the Near-RT RIC to control the behavior of the slices. The Non-RT RIC can train AI/ML models and deploy them to Near-RT RIC. Additionally, slow loop optimizations can be performed by Non-RT RIC using rApps via O1. The Near-RT RIC monitors the slices with E2 reports and assures SLA requirements based on O1 configurations and A1 policies via the E2 interface with control or policy messages. Near-RT RIC makes decisions based on the execution re-

sults of AI/ML models deployed by Non-RT RIC or enforces Non-RT RIC policies. Finally, E2 nodes should support performance reports through O1 and E2 and be able to support slice assurance actions.

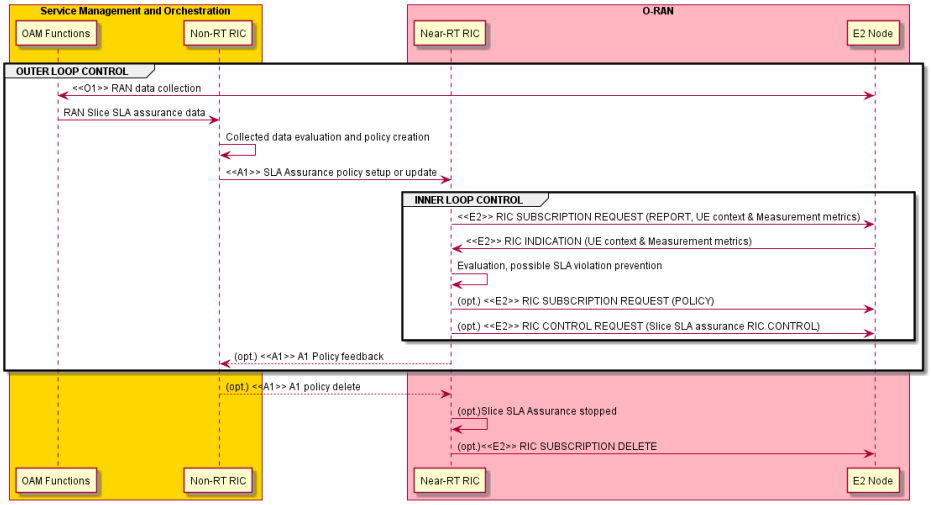


Figure 6.5: Near-RT and Non-RT in SLA.

To investigate the SLA controller placement, we should explore the communication requirements. The required KPM, retrieved with indication report messages primarily from the DU with a few exemptions by the CU, to achieve SLA assurance are in [56]. The Near-RT RIC should monitor and enforce SLA to a single E2 node no matter the behavior of nearby E2 nodes. Thus, depending on the requirements, the Near-RT RIC manages a single or multiple E2 nodes with control or policy messages. Using Near-RT RIC in slow-loop SLA is optional since the Non-RT RIC can control E2 nodes via its O1 interface. Finally, Figure 6.6 illustrates a general overview of SLA assurance using Non-RT or Near-RT RIC.

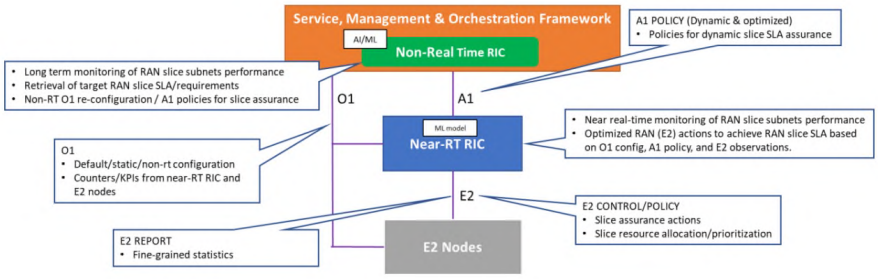


Figure 6.6: RAN Slice SLA Assurance use case overview.

Multi-Vendor Slicing

In Multi-Vendor Slicing, multiple network operators share the same RAN resources to provide their services. These operators might want to support different services with distinct service requirements. Each operator can own part of the RAN and share it with other operators while also using parts of the RAN owned by other operators. The RAN must meet its supposed slice requirements, regardless of the owner, requiring synchronization between CU and DU entities of multiple operators for resource allocation without conflicts. As displayed in Figure 6.7, there are three options to synchronize the entities forming the RAN.

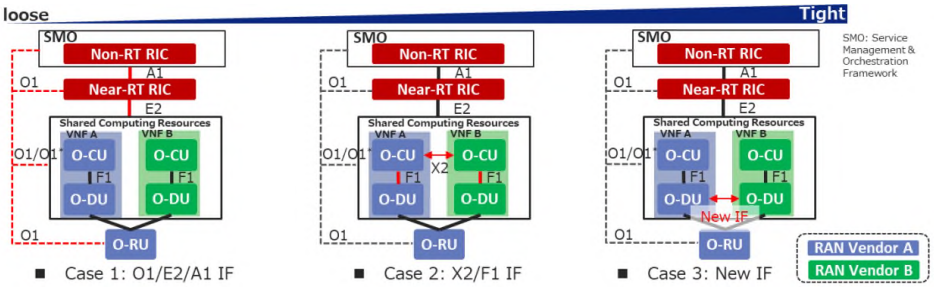


Figure 6.7: Multi-vendor Slices Coordination Scheme Options.

The interface used to perform the synchronization dictates the usage of a Near-RT RIC along with a Non-RT RIC. In moderate coordination, the synchronization is directly achieved via the X2 or F1 interfaces, making RIC usage redundant. Accordingly, tight coordination does not use any RIC either, achieving synchronization via a new network interface introduced between DU entities. Loose coordination requires the usage of both a Near-RT RIC and a Non-RT RIC. When operators connect their CU and DU entities to separate Near-RT RIC deployments, the information flows from DU and CU entities to the Non-RT RIC through the O1 interface. This information is transmitted from the Non-RT RIC to a corresponding Near-RT RIC via A1. Finally, the information arrives at the other vendor's CU and DU entities from its Near-RT RIC via the E2 interface. However, if the operators connect their CU and DU entities to the same Near-RT RIC, the information can directly flow via the E2 interface.

Even though the source of the KPM required in Multi-Vendor Slicing is specified to be from CU and DU entities, the actual content of KPM is not specified. In Multi-Vendor Slicing, the Near-RT RIC should connect to multiple CU and DU entities when the E2 latency is smaller than the latency accumulated by both O1 and A1, removing load from the O1 and A1 interfaces while avoiding their latency.

Network Slice Subnet Instance (NSSI) Resource Optimization

As networks advance, they become increasingly complex, with multiple services having different requirements competing for limited network resources. To optimize the 5G network's resource utilization, traffic demands, and network trends must be proactively predicted for resource re-allocation. The network achieves optimal resource sharing during resource surplus or scarcity with AI/ML models fed with data frequently to predict future trends.

The NSSI Resource Optimization, specified in [57], solely requires the Non-RT RIC. As illustrated in Figure 6.8, Non-RT RIC monitors the behavior of RAN via O1, makes decisions, and re-configures the NSSI attributes via O1 and cloud resources via the O2. The metrics required for decision-making originate from both the CU and the DU entities of each E2 node. The specification clearly states that only Non-RT RIC is required in this case and does not specify anything about Near-RT RIC.

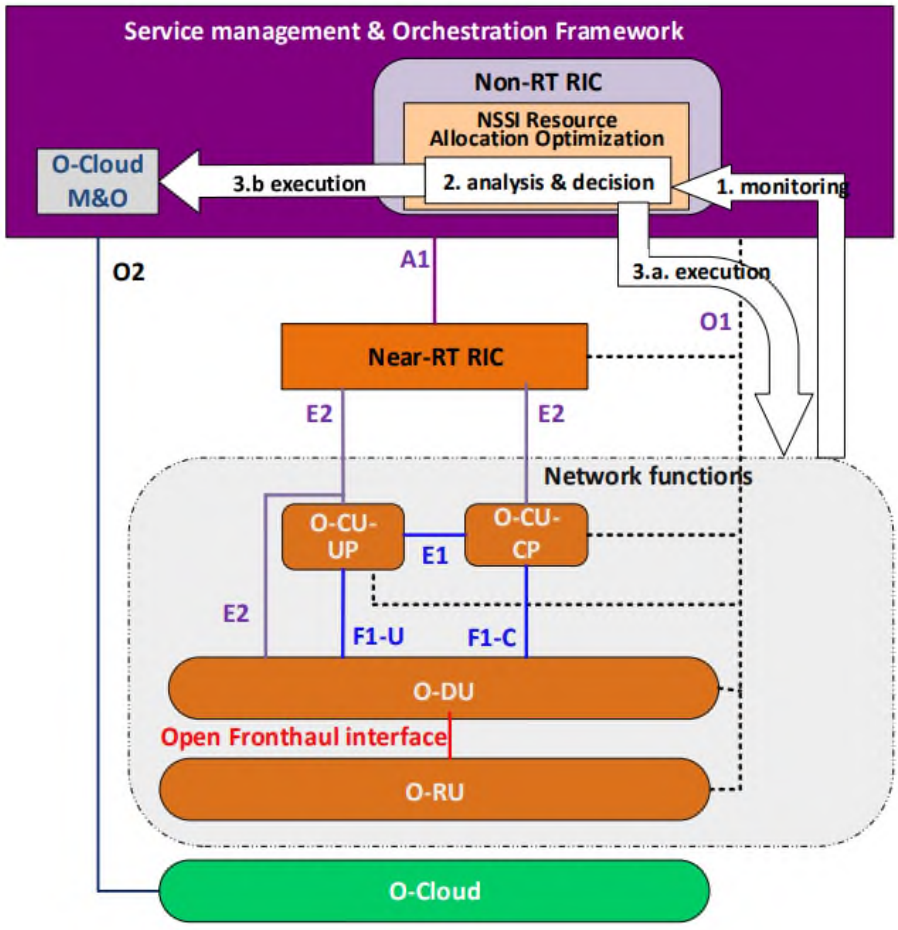


Figure 6.8: NSSI resource allocation optimization over Non-RT RIC.

6.6.2 MIMO

Modern cellular networks support many revolutionary wireless communication system technologies, including MIMO. Multiple transmitter and receiver antennas ensure high and reliable transmission and reception data rates with increased spectral efficiency. MIMO supports multiple spatial stream utilization, each taking its unique path through the airwaves.

Grid-of-Beams (GoB) Beamforming (BF)

The performance of each cell highly depends on its BF configuration. GoB is a BF configuration that tailors the cell according to the distribution of connected devices and their traffic patterns, considering the surrounding physical environment. GoB BF does not use the Near-RT RIC since it is not a time-sensitive process, requiring excessive data to work. The performance indicators specified in [58] are collected from the DU entities at Non-RT RIC almost every fifteen minutes. The Non-RT RIC can take a few hours to make decisions and send a configuration to DU and RU.

Beam-based Mobility Robustness Optimization (bMRO)

MRO is a concept introduced in LTE where the eNB configures two parameters to control handover timing. These parameters are the Time-to-Trigger (TTT), the time offset for the neighbor cell, and the Cell Individual Offset (CIO), representing the power offset from the neighbor cell's reference signal.

The 5G networks, however, differ a lot from 4G. In 5G, massive MIMO (mMIMO) is introduced where the cell is not defined by a single beam, as was in 4G, but by multiple beams. The problem arises as clear borders between cells do not exist when using multiple beams. A solution to this problem is to define a TTT/CIO pair for each beam. In bMRO, there are two different implementation options.

The first option is illustrated in Figure 6.9, and the KPM required at the Near-RT RIC are specified in [58] along with their reporting period of around a minute. Decisions made by the Near-RT RIC are trigger-based, typically occurring at intervals longer than a minute. The Near-RT RIC requires KPM from the CU and controls the RAN behavior by configurations destined for CU. The Near-RT RIC should establish connections with multiple CU entities that share neighboring cells to ensure effective and valid decision-making.

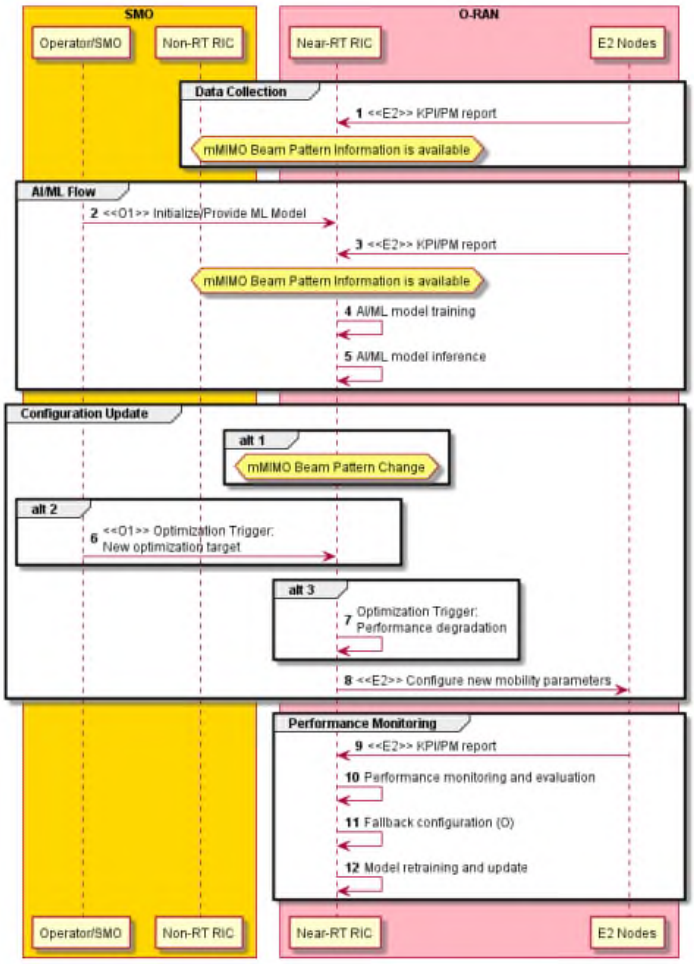


Figure 6.9: Implementation of bMRO with Near-RT RIC

In the second option, Non-RT RIC replaces the Near-RT RIC. The E2 nodes send the same performance indicators, this time to the Non-RT RIC with the same reporting periods. The Non-RT RIC makes decisions and responds accordingly in an event-driven fashion. Finally, in this option, the O1 interface is used instead of E2 to retrieve indicators and control RAN behavior.

Initial Access Optimization

The downlink reference signals should only be transmitted when necessary. The periodically transmitted signals in wireless networks are called ”always-on” reference signals. In 5G, these reference signals include Synchronization Signals (SS) or SS Burst Sets, the Channel Status Information Reference Signal (CSI-RS) Tracking Reference Signal (TRS), the Demodulation Reference Signal (DMRS), and system broadcast information.

According to the specification [58], an AI/ML optimizer optimizes SS Burst Set and CSI-

RS TRS configurations. An extended optimizer's capabilities can derive configurations of other reference signals configurations, including UE-specific CSI-RS acquisition and DMRS. The optimizer can also apply multiple optimizations simultaneously, like Single Side Band (SSB) and CSI-RS optimization, for faster beam acquisition operation. The AI/ML optimization objective is to reduce the transmission of reference signals to the minimum necessary while ensuring compliance with 3GPP standards.

The specification provides three distinct AI/ML optimization implementations, illustrated in Figure 6.10, defined by the optimization problem. In SS Burst Set optimization, the Non-RT RIC performs training and inference of AI/ML models. In DMRS and CSI-RS configuration optimizations, on the other hand, Non-RT RIC performs solely training and Near-RT RIC the inference. In these cases, the performance metrics required for optimizations originate by both CU and DU entities at specified intervals, while the configurations target DU specifically. When optimizing the SS Burst Set along with DMRS and CSI-RS simultaneously, the Non-RT RIC or the Near-RT RIC can perform model training and the CU model inference. Inference at a Near-RT RIC necessitates additional network and site-specific data provided by the Non-RT RIC. Consequently, the Near-RT RIC can establish a connection with a single E2 entity.

Beam Management

Contrary to 4G, which uses frequencies below 6GHz, 5G uses millimeter-wave (mmWave), ranging from 24GHz to 100GHz. These wavelengths support higher bandwidth, capacity, and data rates. However, they suffer from propagation loss and signal blockages due to their shorter wavelength, requiring directional-beam transmission with hybrid analog-digital BF and large-scale antenna arrays at the basestations.

Each connected UE must be aligned with at least one of its serving basestation's beams to achieve successful transmission and reception. The frequent measurement reports required for aligning a beam with a UE are taking away network resources and reducing the overall throughput. The specification [58] defines intelligent ways to handle beam selection more elegantly. These ways involve AI/ML training and inference, which rely on data extracted from DU entities. The specification outlines the data utilized for training and inference, along with the expected reporting period. Training requires data, hourly or daily, while inference at increments of one hundred milliseconds. Accordingly, CU or DU entities receive the decisions

made at increments of one hundred milliseconds. As illustrated in Figure 6.10, inference for Beam Management can either be performed on Non-RT RIC or at the E2 itself, without the use of Near-RT RIC, or with both a Near-RT RIC and a Non-RT RIC. Using a Near-RT RIC, it can be connected to only one E2 node since it does not need to know about other nodes in the network.

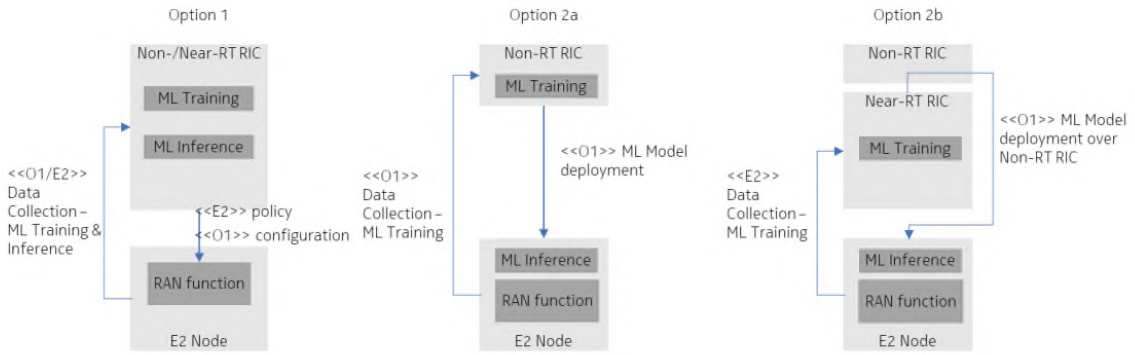


Figure 6.10: AI/ML Training and Inference.

Non-GoB

In 5G cellular networks, Non-GoB algorithms are a crucial addition in mMIMO deployments. Contrary to GoB algorithms, where predefined beams are selected, Non-GoB relies on the uplink and downlink link correspondence, computing beam weights in real time based on channel measurements. The DU calculates these weights and transmits them to each corresponding RU. Non-GoB algorithms are vendor-specific, and each algorithm might be suitable for specific environments and conditions while underperforming in others. Thus, the problem statement is to switch intelligently between Non-GOB algorithms to improve channel conditions in different complex environments.

According to the specification [58], deploying Non-GoB algorithms in 5G requires both a Near-RT RIC and a Non-RT RIC. The specification defines the contents of the performance indicators used for training and inference, along with their reporting period. The AI/ML model training is explicitly performed at Non-RT RIC and deployed to Near-RT RIC via O1 or O2 interfaces. The Non-RT RIC is also connected to an external application server, which provides additional enrichment information about each UE, forwarding this information to Near-RT RIC via A1. The Near-RT RIC collects measurements from DU entities, executes the deployed AI/ML models to make decisions, and configures the DU entities with control

or policy messages. The inference process is illustrated in Figure 6.11. Under the specified KPM and given that Near-RT RIC can acquire information about each UE through the external application server, Near-RT RIC can establish connections with a single gNB or, when disaggregated, with multiple DU entities within an E2 node.

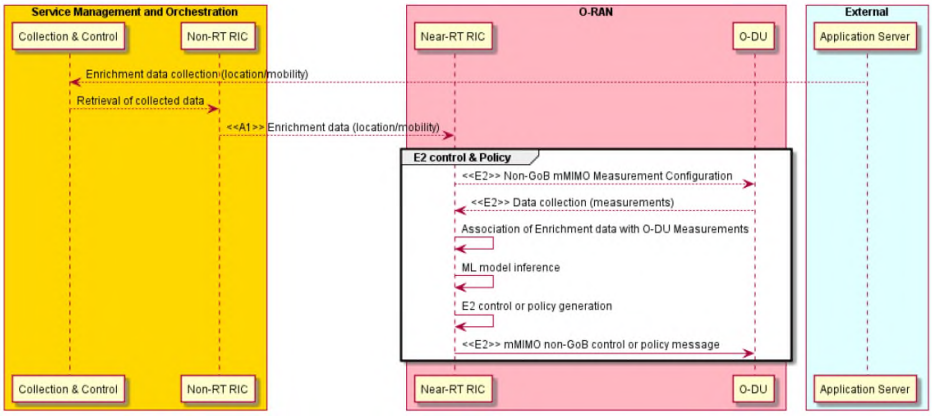


Figure 6.11: AI/ML Inference Process in Non-GoB.

MIMO Optimizations

A network's capacity depends on the technology used, the conditions within the surrounding environment, and the expected target quality when serving its users. The 5G cellular networks support many "dials" and "knobs" for MIMO optimization.

In a Multi-User MIMO (MU-MIMO) cellular network, basestations can support multiple antennas and transmit separate data streams to numerous UEs simultaneously, achieving parallel communication since each device uses its spatial stream. In Single-User MIMO (SU-MIMO), a UE utilizes multiple antennas to communicate with a base station.

Each UE shares the same transmission power, guaranteeing maximum network capacity and disregarding optimal performance and user experience. The individual UE quality degrades since this transmission power distribution does not consider the separability and path loss deltas of each UE. Downlink transmit power optimization is a method deployed using the Non-RT RIC to achieve both capacity and quality within a network.

Efficient radio resource utilization is also a crucial aspect of maintaining performance. Users who share radio resources should have a guaranteed channel orthogonality to minimize interference and maintain link signal quality. Additionally, maintaining user signal power and limiting inter-user interference with adaptive BF is necessary to sustain good quality and user

experience. In bursty traffic, RIC should classify users based on their demands to identify and pair users with low resource demand. Moreover, robust and timely channel estimation mechanisms are critical to generating accurate weights for lower volume buffer data, preventing failures in MU-MIMO pairing.

MIMO mode selection, a method of choosing between SU-MIMO and MU-MIMO, is a crucial aspect of wireless communication optimization. As networks become congested, schedulers are inclined to prioritize MU-MIMO to increase capacity and optimally utilize network resources. However, this might result in undesirably lower spectral efficiency and, thus, low throughput and poor user experience quality. Tackling this problem requires users with high throughput demands to use SU-MIMO to sustain higher channel signal quality.

According to the specification, the Non-RT RIC implements the three optimization methods as rApps. The Non-RT RIC collects measurements from DU entities at intervals ranging from one to fifteen minutes for model training and inference. Additionally, the Non-RT RIC makes decisions and configures the DU entities using the same periodicity without the involvement of Near-RT RIC. The required measurements and configurations, specified in [58], are transmitted via the O1 interface as illustrated in Figure 6.12.

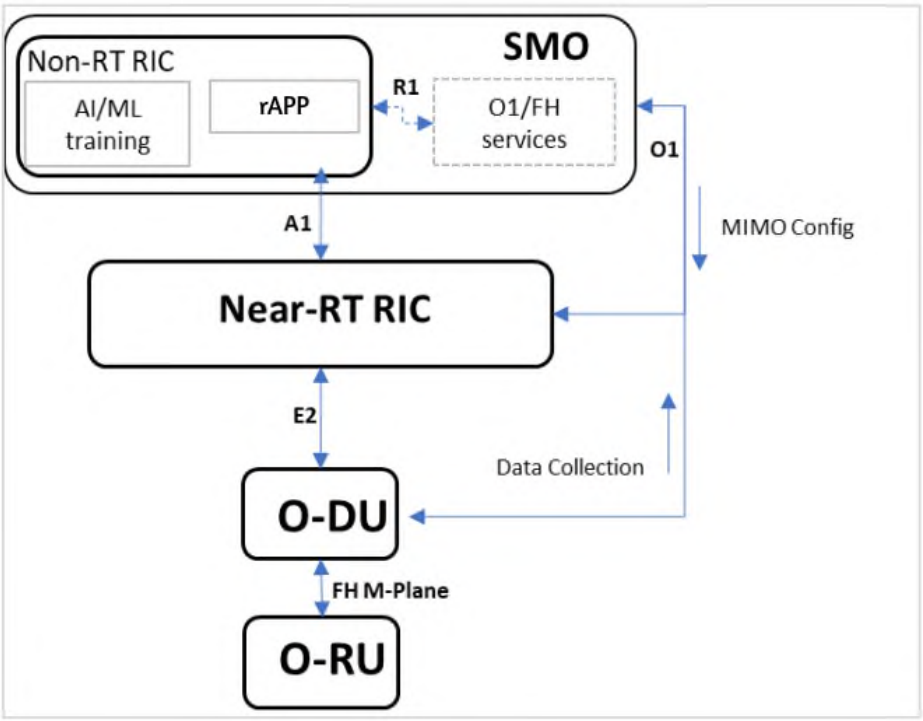


Figure 6.12: Interfaces Performing MIMO Optimizations.

6.6.3 Traffic Steering (TS) and Mobility Control

As modern cellular networks technologically advance, the traffic needs increase along with the frequency band utilization, increasing the complexity of steering traffic in a balanced distribution. Cell reselection, handovers, or modifying load calculations and cell priorities are the primary actions in TS. The Non-RT RIC and Near-RT RIC collaborate to select the appropriate serving cell via A1 policy enforcement, achieving effective mobility control within TS. As specified in [56], in TS, the Near-RT RIC retrieves performance indicators from the E2 nodes with indication report or insert messages and controls them with control or policy messages, as illustrated in Figure 6.13. The Non-RT RIC establishes static optimization targets, whereas the Near-RT RIC focuses on inference and prediction to attain predefined objectives through a series of procedures. These procedures typically include handovers, carrier aggregation (CA), and dual connectivity (DC), collectively contributing to TS. Under the specified KPM contents outlined in the specification, the Near-RT RIC retrieves data from both the CU and DU entities, whether in a disaggregated or non-disaggregated configuration. As per the procedure, the Near-RT RIC should establish connections with a single E2 or multiple E2 nodes, depending on the specific requirements.

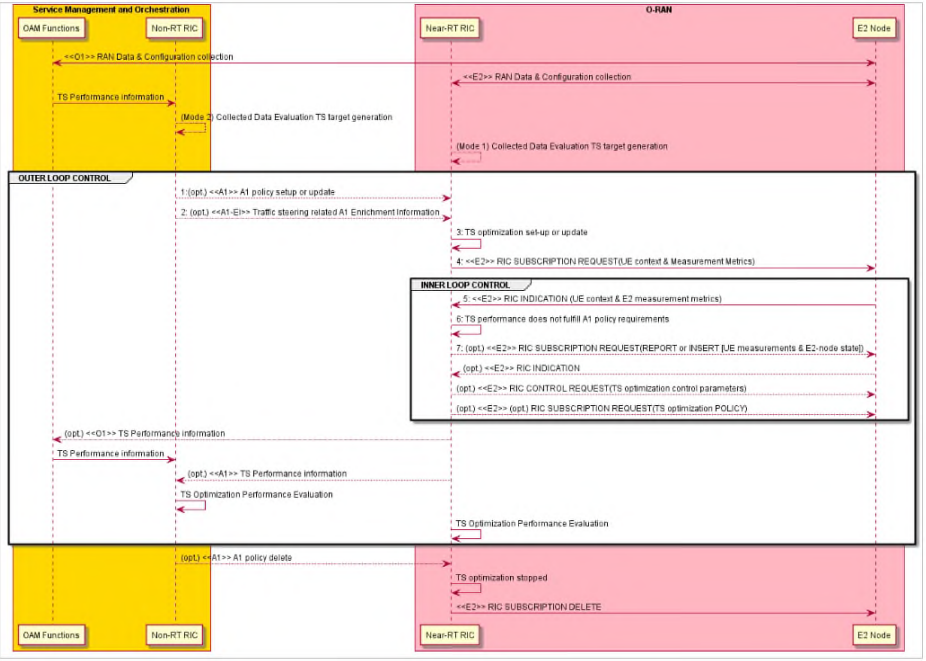


Figure 6.13: Traffic Steering with Non-RT and Near-RT RIC.

Handover

Handover transfers a UE and its data sessions from one cell or base station to another. Handovers exist to enable seamless connectivity without noticeable interruptions in services as devices move through different areas covered by various cells or base stations.

Researchers have performed thorough research on handover optimization since the release of LTE. In 5G NR, researchers introduced conditional handovers as a potential improvement over baseline handovers. Research at [59] investigates the advantages and shortcomings of conditional handovers, contrary to baseline handovers. Researchers in [60] leverage deep learning to determine the need for handovers and their execution manner and contrast their observation with alternative, already defined, conditional handover strategies. Finally, in [61], conditional handovers leverage NS-3 and AI/ML in the Near-RT RIC.

The aforementioned conditional handover methods do not consider the load of each base station when making decisions. The load distribution imbalances can lead to performance degradation, requiring the integration of load balancing [62] into the conditional handover methods. This combination is called ”Load Balancing Handover” or ”Load-Aware Handover”. In [63] and in [64], these handover types are investigated in LTE and NR, respectively.

The official specification defines an additional handover framework implementation for vehicle-to-everything (V2X) communication. The O-RAN framework should prevent handover sequences that could degrade V2X performance. Employing AI/ML algorithms for long-term analytics and real-time robust optimization can lead to optimal performance. The Non-RT RIC uses its O1 interface to maintain UE-based handover events and mobility data, identifying causes of handover anomalies and optimizing handover sequences, with resolutions stored in a database based on feedback. The Near-RT RIC monitors the real-time mobility context of each moving UE and performs real-time optimization using trained models, detecting and predicting unexpected handover events while generating optimal handover sequences to mitigate anomalies. Finally, an external V2X application server provides additional enrichment information to assist Non-RT RIC in training and Near-RT RIC inference, illustrated in Figure 6.14.

The field of research related to handover is extensive. According to the algorithm, the Near-RT RIC should operate when connected to multiple E2 nodes, as well as when connected to a single node. The specification in [65] offers general performance indicators for TS but lacks specific KPM values for the handover scenario.

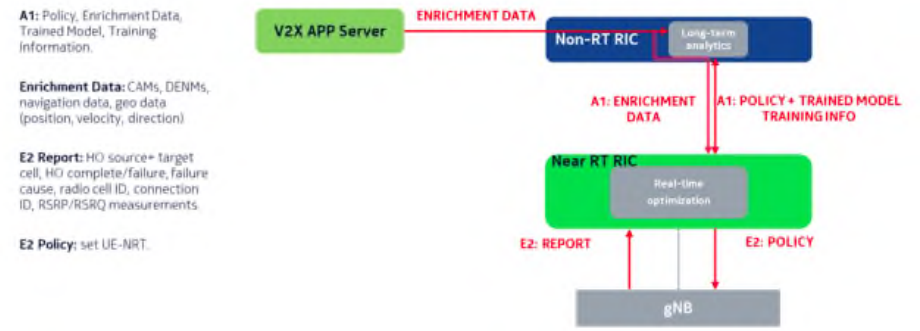


Figure 6.14: V2X with Non-RT and Near-RT RIC.

CA and DC

CA is a technique introduced in LTE to improve transmission and reception bitrate and, thus, performance. With CA, a UE can utilize aggregated resources consisting of multiple component carriers (CC) having the same or different bandwidths. The intra-band contiguous, the intra-band non-contiguous, and the inter-band non-contiguous, illustrated in Figure 6.15, constitute the three different implementation methods of CA. Intra-band contiguous describes CA of multiple CCs within the same operating frequency band. In intra-band non-contiguous, on the other hand, the multiple CCs are separated by one or more frequency bands. In inter-band non-contiguous, the different CCs belong to distinct operating frequency bands. In 5G networks, we perform CA between licensed band NR, the primary cell, and New Radio Unlicensed (NR-U), the secondary cell.

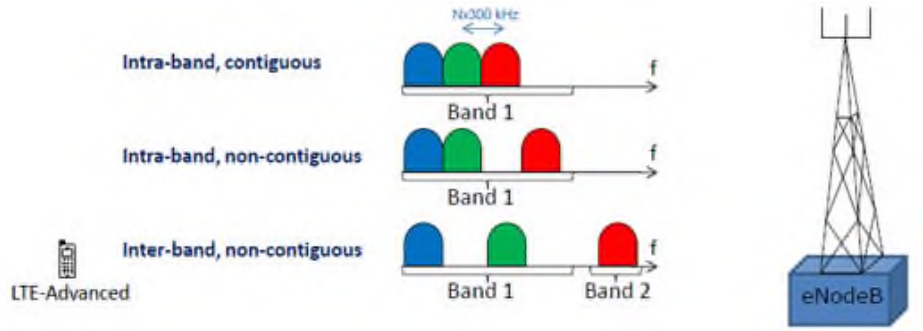


Figure 6.15: CA in LTE.

DC is also a CA-based technique introduced in LTE. A UE in LTE could establish connections with multiple eNBs for sending and receiving data. The release of 5G networks expands this feature, allowing mobile devices to utilize both LTE and NR frequency bands simultaneously for improved and reliable performance and increased coverage. The specifi-

cation [65] defines the various ways to establish dual connectivity. It can occur between the primary cell in a licensed NR band and the secondary cell in NR-U, between the primary cell in the licensed LTE band and the secondary cell in NR-U, or between the primary cell in the licensed NR band and the secondary Wi-Fi cell.

Both CA and DC use the Non-RT and the Near-RT RIC. The Non-RT RIC monitors each UE and compares its performance to the expected and required performance. The Non-RT makes decisions and forwards them as A1 Policies and additional enrichment information to the Near-RT RIC. The Near-RT RIC enforces these policies, considering the state of the RAN and the enrichment information to modify the RAN behavior and optimize the user experience. The specification lacks explicit details regarding the performance indicators necessary for decision-making in these two scenarios. In the context of CA, the Near-RT RIC can establish a connection with a single E2 node since its decisions concern the cells within that node. However, in the case of DC, it needs to connect to multiple nodes to make decisions regarding the aggregation of CCs belonging to cells from different E2 nodes.

6.6.4 Energy Saving

Energy saving in 5G is a critical aspect of next-generation wireless technology, as it addresses the increasing demand for high-speed data transmission while minimizing the environmental impact and reducing operational costs. The energy saving issue can be measured using two metrics, the energy efficiency [66] and the energy consumption [67].

Radio Frequency (RF) Channel Reconfiguration

Modern cellular networks can increase capacity and throughput with mMIMO antennas and BF. In mMIMO, TX and RX arrays refer to arrays of transmitters (TX) and receivers (RX) used in communication systems. In cases with low network usage and the number of connected devices falling below the expected threshold, the network can conserve energy and reduce DU energy consumption by deactivating specific TX/RX arrays. Switching off an RU decreases the spatial streams and Synchronization Signal Blocks (SSBs). SSBs are a specific signal structure, primarily used for synchronization and cell discovery purposes, transmitted from each RU to devices.

The main focus of RF channel reconfiguration is to optimize TX/RX array selection. During TX/RX array reselection, adjustments may be necessary, including changes to the

maximum number of spatial streams, the quantity of SSB Beams, or the transmit power of the RU antenna. According to [68], RF channel reconfiguration uses either the Non-RT RIC exclusively or in conjunction with the Near-RT RIC. The AI/ML model training is explicitly performed at Non-RT RIC, while the inference is performed at Non-RT RIC when the Near-RT RIC is not used or at Near-RT RIC when used. The required performance indicators for training and inference are specified and retrieved by both CU and DU entities at defined intervals. The content and timing of decision output messages are also specified and intended for DU entities. Finally, a Near-RT RIC, if used, can connect to a single E2 entity as it doesn't need knowledge of the entire network to execute RF Channel Reconfiguration.

Carrier and Cell Switch Off/On

Modern 5G networks utilize one or more frequency layers, called carriers, to provide coverage in the same area. Switching these carriers on or off increases or decreases network capacity, respectively. When a carrier is off, the cell forces each UE connected to this specific cell to transfer to another. The issue arises when energy saving and performance should be both optimal. Fewer carriers result in less overall energy consumption. However, each cell that remains on must serve more UEs, decreasing performance.

Carrier and Cell Switch Off/On is a complex problem that requires AI/ML. To make correct decisions that reduce the overall energy while maintaining the performance, we should know the whole network's energy consumption and performance in the Non-RT RIC or Near-RT RIC as specified in [68].

Performing Carrier and Cell Switch Off/On without a Near-RT RIC is illustrated in 6.16. The Non-RT RIC utilizes its O1 interface to receive performance indicators and send configurations to the RAN. The specification defines the performance indicators alongside their reporting period. These indicators are extracted both from CU and DU. The specification also defines the contents of the decisions made by the Non-RT RIC, along with the expected inference duration of the AI/ML models. Finally, in this case, AI/ML training and inference are performed exclusively in Non-RT RIC.

Near-RT RIC can also perform Carrier and Cell Switch Off/On, as illustrated in Figure 6.17. The Non-RT RIC is responsible for periodically receiving performance indicators from the O1 interface and training AI/ML models deployed to Near-RT RIC via O1 or O2 for inference. Near-RT RIC can receive optimization targets through two configuration channels, via

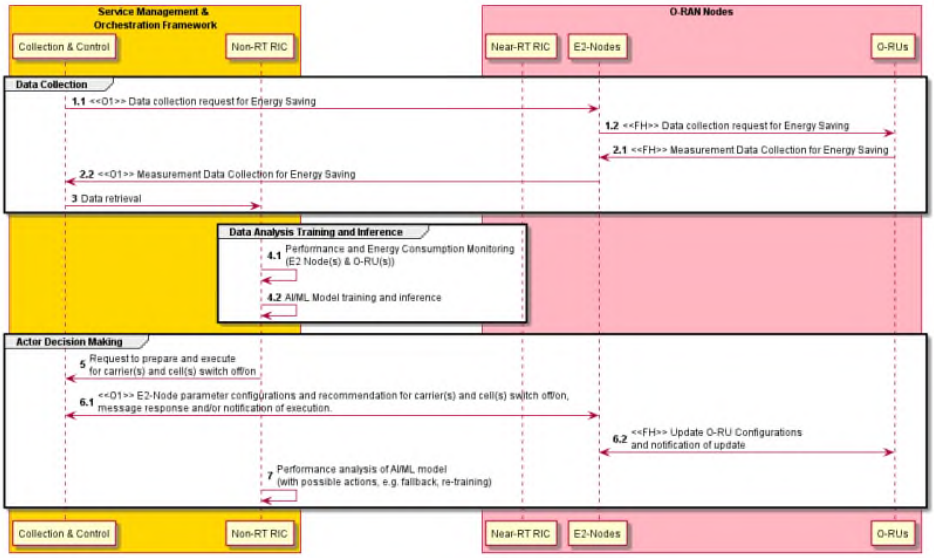


Figure 6.16: Non-RT RIC AI/ML inference in Carrier and Cell Switch Off/On

O1 and policies via A1 interfaces. The KPM, extracted exclusively from the DU, alongside their designated reporting intervals, are explicitly outlined in the specification. Furthermore, the decision-making process and configuration schedule are also clearly defined, with decisions capable of impacting either the CU or DU. In this scenario, the Near-RT RIC should establish connections with multiple DU entities and a single CU, resulting in the Near-RT RIC connecting to a solitary E2 node.

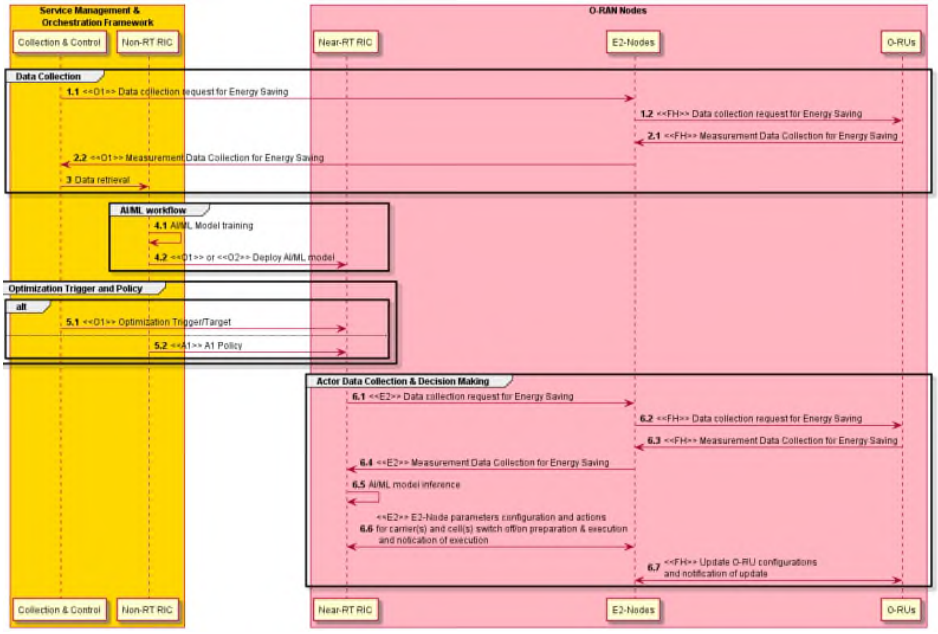


Figure 6.17: Inference via Near-RT RIC.

Advanced Sleep Mode Selection

Contrary to implementing advanced sleep mode in monolithic network architectures, where the RU capabilities are available within the E2 node, allowing for autonomous execution of sleep modes, enforcing it in disaggregated deployments is more complex since it requires DU and CU to understand the sleep mode capabilities of the RU. Additionally, the Near-RT RIC and the Non-RT RIC need to be aware of DU and CU capabilities to formulate policies regarding sleep modes.

In advanced sleep mode selection, we can use Non-RT RIC for AI/ML model training and Non-RT RIC or Near-RT RIC for inference. The Near-RT RIC, when used, should establish connections with multiple DU entities within the same RAN, as it exclusively retrieves KPM from these DU entities, which it also configures. The specification [68] does not explicitly define the KPM contents nor their reporting granularity.

O-Cloud Resource Energy Saving Mode

As operators adopt virtualized architectures, there is a need for a solution for energy saving in the O-Cloud. In the 7.2x split, CUs and DUs perform signal processing, consuming energy to maintain a certain level of system availability even with low or no traffic load on VNFs or Cloud-Native Network Functions (CNFs). The aim is to enable energy savings in the O-Cloud by reducing the power consumption of various O-Cloud components without impairing the network performance. The official specification specifies two approaches to save O-Cloud resources, including the *Node Shutdown* and *Central Processing Unit (CPU) Energy Saving Mode*. Both approaches use solely the Non-RT RIC to control the behavior of O-Cloud via its O2 interface.

In *Node Shutdown*, O-Cloud shuts down physical nodes to save resources at low load and idle times. To shut down an O-Cloud node, we must evacuate the node from its VNFs and CNFs to free it up. The Non-RT RIC monitors the traffic load in CU and DU via O1 and sends the reported data to O-Cloud via O2. The O-Cloud examines the data it receives from the Non-RT RIC and decides whether to drain a node. The Non-RT RIC decides when to shut down an O-Cloud node and transmits this instruction to O-Cloud via O2, from which the O-Cloud replies after executing node shutdown.

In *CPU Energy Saving Mode*, the Non-RT RIC and SMO control the various CPU energy-saving modes. Controlling these modes, which correspond to different CPU energy-saving

states, implemented by the vendor or standardized, achieves CPU power management. The operators can now, for instance, control the P-State and C-State of the CPU. P-States provide a way to scale the CPU frequency and voltage to reduce the power consumption of the CPU. C-States are idle states that the CPU can have when it is not actively processing tasks to save energy. The O-Cloud can adapt fast and autonomously to achieve minimal energy consumption. The Non-RT RIC collects KPM from CU and DU entities via O1 and guides the O-Cloud behavior via O2.

The specification [68] describes thoroughly *Node Shutdown* and *CPU Energy Saving Mode* approaches. However, it does not mention the required KPM values, reporting and control granularity, and whether CU or DU are the sources.

6.6.5 QoE

QoE is a measurement that quantifies the overall user satisfaction, perceived quality of communication, and data services provided by the cellular network. In 5G, the main objective is to ensure QoE optimization is supported within the O-RAN architecture to achieve user application requirements. The sections below investigate the different approaches for achieving QoE based on [65] and [56].

User-Centric QoE Connection Policy

In this approach, as illustrated in Figure 6.18, external applications communicate with the Non-RT RIC and dynamically request specific RAN behavior, eliminating the need for preloading static configuration data and QoS profiles into E2 nodes. These requests are transmitted from the Non-RT RIC to the Near-RT RIC as policy directives, with the Near-RT RIC tasked with their enforcement.

User-Centric QoE Performance Policy in Non-RT RIC

We can achieve optimization of the QoE through frequent performance feedback and accurate RAN behavior guidance. In Figure 6.19i, the Non-RT RIC manages RAN QoE policies based on input from external systems. These policies are translated and communicated via the A1 interface to the Near-RT RIC. An alternative approach in 6.19ii involves application layer measurement reporting from the UE to the RAN, forwarded to a QoE handling node at SMO,

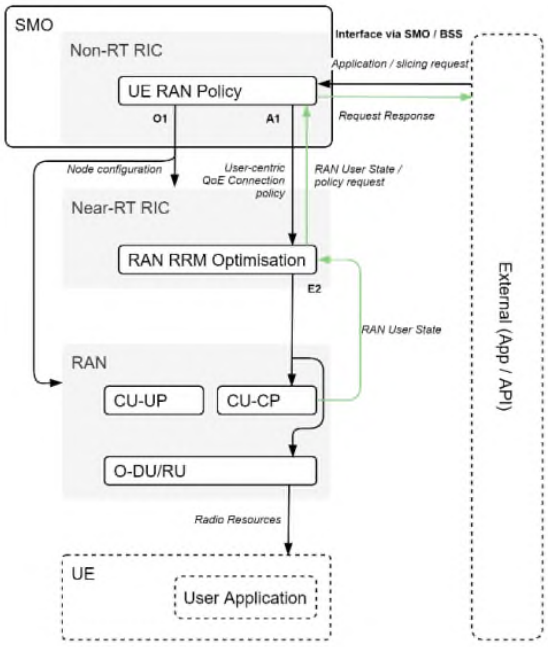


Figure 6.18: User-Centric QoE Policy Approach

providing input to the Non-RT RIC with reduced latency. In both approaches, the Near-RT enforces the policies it receives on specific E2 nodes for real-time user management.

User-Centric QoE Performance Policy in Near-RT RIC

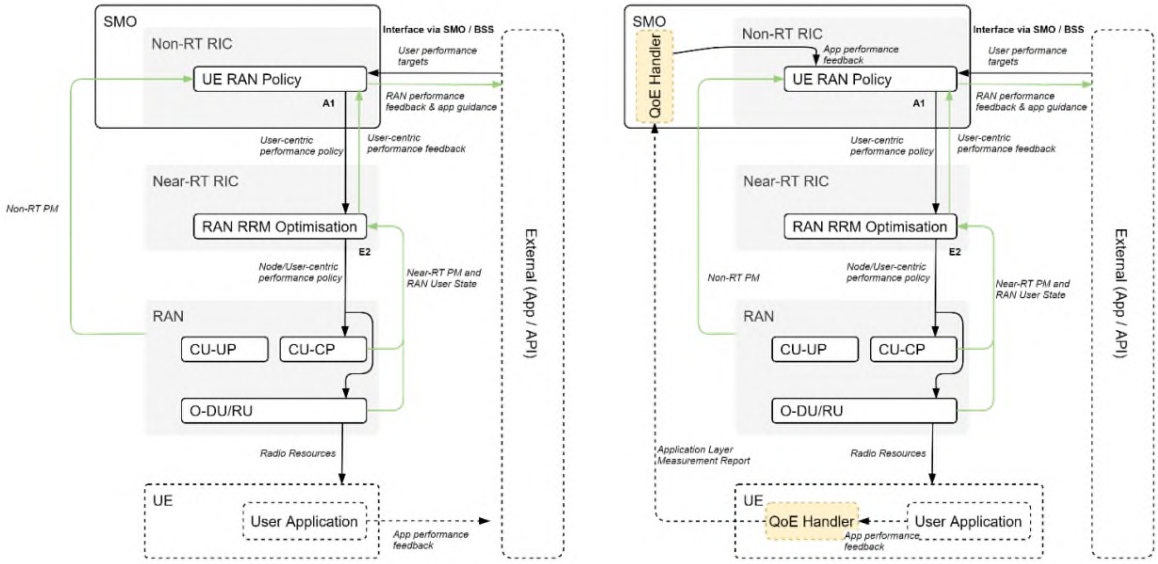
This approach is the same as *User-Centric QoE Performance Policy in Non-RT RIC*, but the QoE policy configuration is moved to the Near-RT RIC as illustrated in Figures 6.20i and 6.20ii.

AI/ML QoE Enhancements

The Non-RT RIC is responsible for constructing, training, and deploying AI/ML models to the Near-RT RIC. The E2 nodes send data through the O1 interface to the SMO, which forwards it to the Non-RT RIC for model training. The Near-RT RIC retrieves performance indicators by the E2 nodes, is responsible for model inference, and enforces its decisions via its E2 interface, as illustrated in Figure 6.21.

Radio Performance Analytics

This case involves an external application server requesting RAN performance from Near-RT RIC. The Near-RT RIC is a middleware between the external application and the E2



(i) Feedback Loop with External Application. (ii) Feedback Loop with QoE Handler.

Figure 6.19: User-Centric QoE Performance Policy with Non-RT RIC.

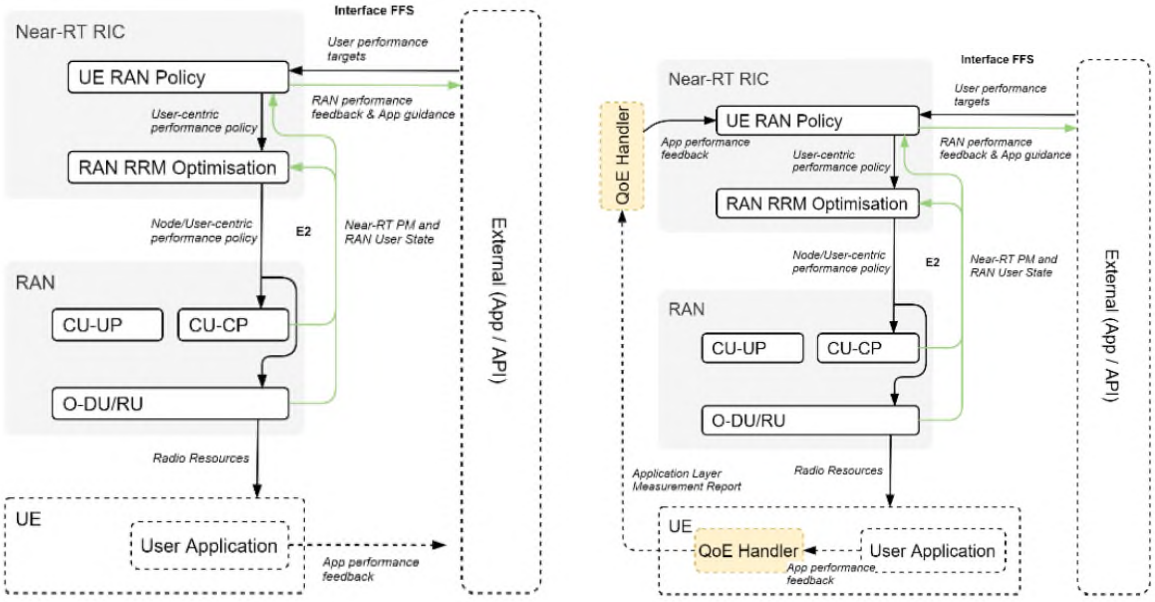
RAN. In response to application requests, Near-RT RIC subscribes to measurements through the E2 interface and runs a radio performance prediction model to generate predicted performance for a specific UE or cell.

In Figure 6.22i, Near-RT RIC exposes radio performance prediction information through Local NEF to External Applications. In Figure 6.22ii, Near-RT RIC exposes radio performance prediction information directly to the Multi-Access Edge Computing (MEC) Application deployed in the MEC Application server. The 3GPP defined NEF in [69] and ETSI MEC specified MEC in [70].

6.6.6 Congestion Prediction and Management (CPM)

CPM is a mechanism that proactively detects congestion and manages the network to avoid it, introduced in 5G networks. When congestion exists, significant problems arise as quality decreases and user experience suffers. Even though congestion patterns are complex and unpredictable, detecting them eradicates the possible performance decrease. The specification at [65] defines Near-RT RIC and Non-RT RIC utilization for proactive detection and avoidance of network congestion.

We can implement CPM either at the Non-RT RIC exclusively or at both Near-RT RIC and Non-RT RIC. Non-RT RIC alone can directly help mitigate congestion by



(i) Feedback Loop with External Application. (ii) Feedback Loop with QoE Handler.

Figure 6.20: User-Centric QoE Performance Policy with Near-RT RIC.

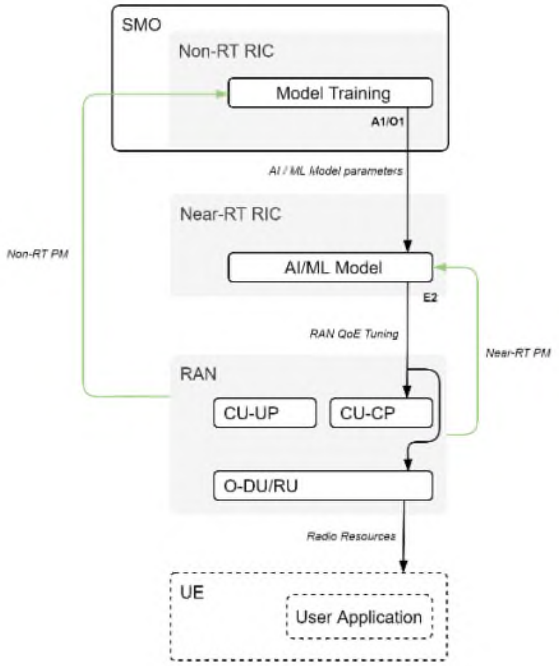
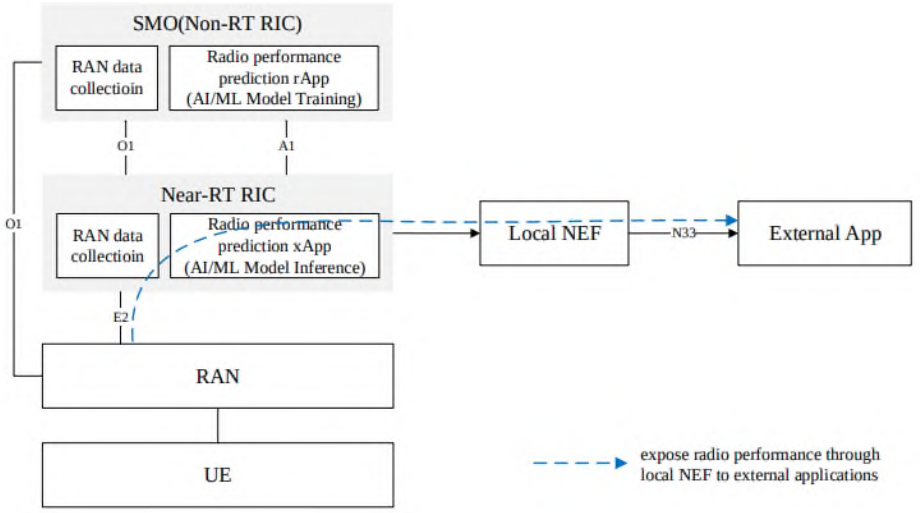
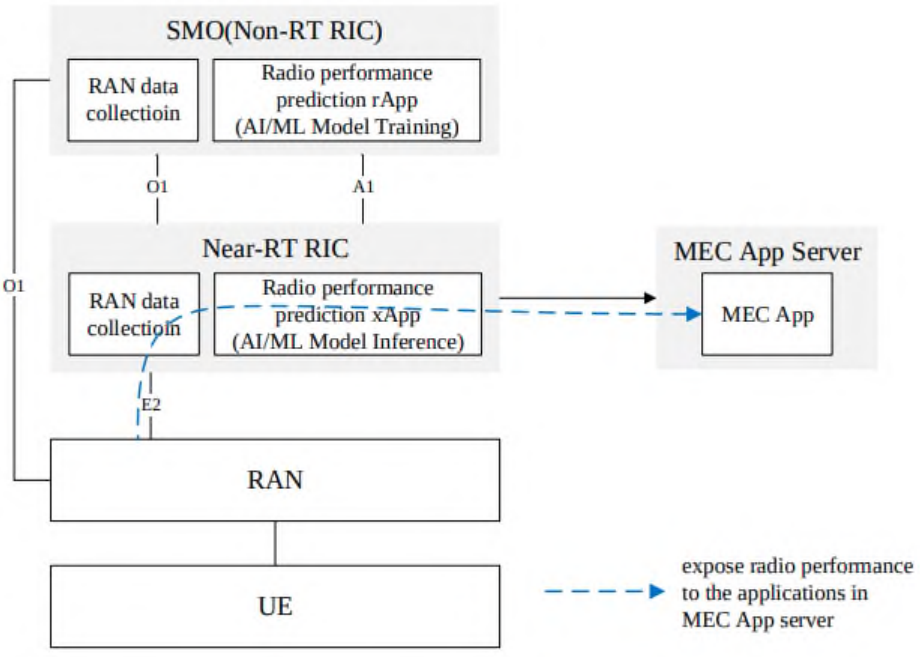


Figure 6.21: AI/ML QoE Enhancements.



(i) RAN to Near-RT RIC to Local NEF to External Application.



(ii) RAN to Near-RT RIC to MEC Application.

Figure 6.22: Near-RT RIC Exposure of Radio Performance.

1. Splitting cells
2. Adding carriers
3. Switching to higher order MIMO
4. Switching to some users to Wi-Fi.

Otherwise, we can accommodate CPM using a Near-RT RIC and a Non-RT RIC. The AI/ML inference is moved from the Non-RT RIC to the Near-RT RIC via the A1 interface. The Near-RT RIC can avoid congestion by

1. Switching to DC
2. Debarring of user access
3. Enabling load sharing

and control the RAN via its E2 interface. Two additional mitigation methods, 6A and 6B are present in Figure 6.23.

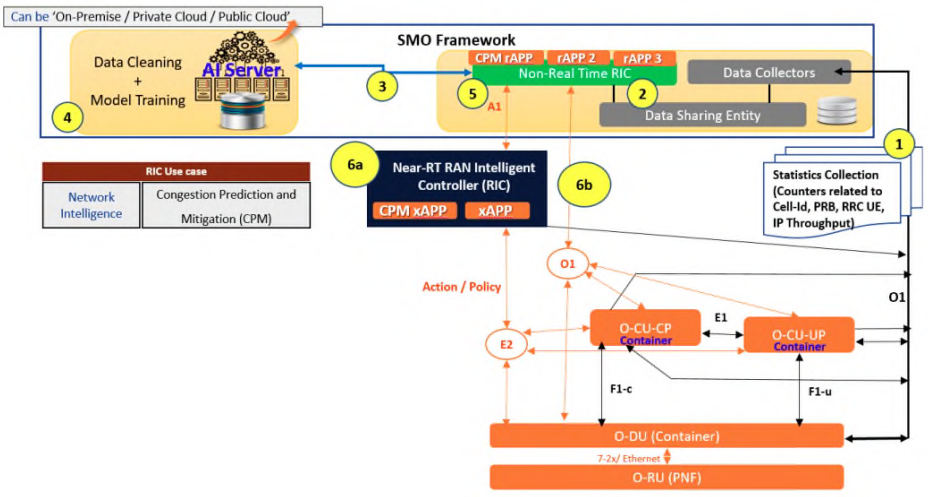
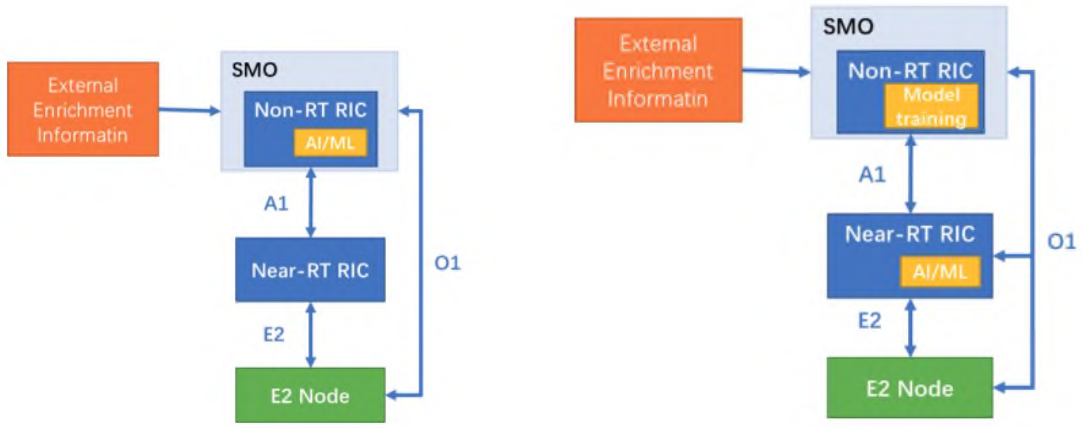


Figure 6.23: Proposed CPM solution.

The specification presents a general overview of indicators required to make decisions for CPM without explicitly specifying their contents and origin. According to CPM goals and KPM, when implementing it in Near-RT RIC, the Near-RT RIC needs comprehensive network awareness to predict and potentially mitigate congestion through load sharing and must possess information regarding the operational status of multiple E2 nodes.



(i) IoT Optimizations on Non-RT RIC. (ii) IoT Optimizations on Near-RT RIC.

Figure 6.24: IoT Optimizations.

6.6.7 IoT

A significant advancement in modern networks is their capacity to accommodate IoT devices. Networks that support IoT should offer exceptional reliability and efficiency. Optimizations performed by the Near-RT or Non-RT RIC on the RAN are essential to achieve the IoT's high demands. The specific optimizations and the requisite KPM are contingent upon the type of optimization performed.

As illustrated in Figure 6.24i, the Non-RT RIC collects performance indicators from the E2 node and additional enrichment information from an external enrichment information server. The Non-RT RIC executes the AI/ML models using the data received and makes decisions transmitted to the E2 node via the O1 interface.

In 6.24ii, on the other hand, the AI/ML models are trained in the Non-RT RIC and deployed at the Near-RT RIC. The Near-RT RIC is responsible for optimizing its E2 nodes either by monitoring their behavior via its E2 interface and making decisions or by receiving and enforcing A1 policies provided by Non-RT RIC.

The specification provides a general overview of the KPM needed for this optimization process but lacks precise details. The specification mentions a few optimizations performed in IoT, including PDCP, PDU session or/and QoS flow duplication, Ethernet Header Compression (EHC), Accurate reference timing provisioning, QoS and scheduling enhancements, and different prioritized transmission multiplexing. For these optimizations, the Near-RT RIC can connect to a single E2 node since it does not need knowledge of the whole network for

decision-making. More specifically, Near-RT RIC optimally should only communicate with the CU entity of the E2 node since it does not require any information from DU.

6.6.8 Self Organizing Networks (SON)

Modern 5G networks support SON functionalities [71], which enables automated configurations, optimizations, protection, and healing functions. These functionalities reduce the cost of running a mobile network, improve network performance, and eliminate manual configuration. SON is an automated process that optimizes network functionalities to achieve specified goals or increase overall network performance.

As specified in [65], the Near-RT RIC alongside Non-RT RIC or solely the Non-RT RIC can support SON functions. As illustrated in Figure 6.25, hardware-intensive SON functions are implemented in Non-RT RIC as rApps or at management entities at SMO. The Near-RT RIC deploys time-sensitive SON functions as xApps.

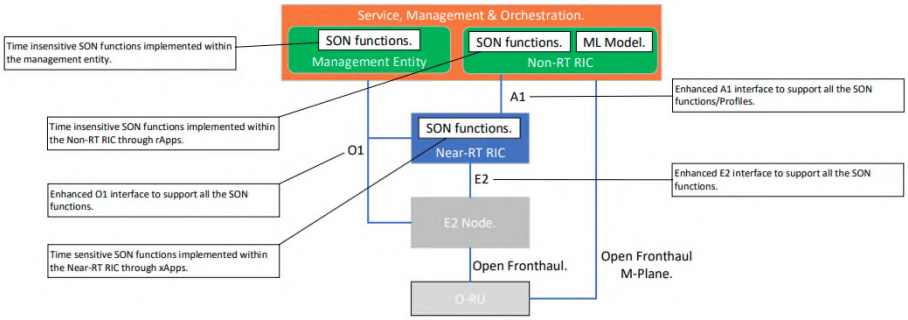


Figure 6.25: SON Function Deployment

The performance indicators used in SON are not specified since they depend on the SON function. Supporting SON functionalities leads to Physical Cell Identity (PCI) conflicts. Each E2 node has its own PCI, used by mobile devices to distinguish it between neighboring cells and establish connections. As we deploy more E2 nodes, race conditions for PCI acquisition arise and pose a significant issue. A PCI conflict exists when two or more E2 nodes share the same PCI. A Near-RT RIC connected to multiple E2 nodes can resolve race conditions and avoid PCI conflicts.

6.7 Modeling 5G in Linear Programming

To formulate and solve controller placement in 5G we model the overall network architecture in linear programming equations.

6.7.1 Constant Variables

These variables represent statically defined parameters about the network.

$$y_i = \begin{cases} 1 & \text{if E2 node deployed at node } i, \\ 0 & \text{otherwise.} \end{cases}$$

6.7.2 Decision Variables

The linear solution sets the value of each variable within the set below. These variables are essential in optimizing controller placement and node-to-controller and controller-to-controller mapping.

$$C_i = \begin{cases} 1 & \text{if Near-RT RIC deployed at node } i, \\ 0 & \text{otherwise.} \end{cases}$$

$$G_i = \begin{cases} 1 & \text{if Non-RT RIC deployed at node } i, \\ 0 & \text{otherwise.} \end{cases}$$

$$c_{ij} = \begin{cases} 1 & \text{if Near-RT RIC deployed at } i \text{ mapped to Non-RT RIC at } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{if E2 node } i \text{ mapped to Near-RT RIC at } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if E2 node } i \text{ mapped to Non-RT RIC at } j, \\ 0 & \text{otherwise.} \end{cases}$$

6.7.3 General Mapping Constraints

Firstly, we represent as N the set of available computational nodes in the network. The two conditions required to map a Near-RT RIC to a Non-RT RIC are that it must exist and cannot connect to more than one Non-RT RIC.

$$\forall i \in N \quad \sum_{j}^{N} C_{ij} = C_i,$$

Accordingly, to map an E2 node to a Near-RT RIC, the E2 node must exist and connect a single Near-RT RIC.

$$\forall i \in N \quad \sum_{j}^{N} y_{ij} = y_i,$$

Mapping an E2 node to a Non-RT RIC requires the existence of the E2 node, which connects to one Non-RT RIC.

$$\forall i \in N \quad \sum_{j}^{N} x_{ij} = y_i,$$

When an E2 node, deployed at node i, is mapped to a Near-RT at node j, a Near-RT RIC must have been deployed at node j.

$$\forall i, j \in N \quad y_{ij} = C_j,$$

When an E2 node, deployed at node i, is mapped to a Non-RT at node j, a Non-RT RIC must have been deployed at node j.

$$\forall i, j \in N \quad x_{ij} = G_j,$$

When Near-RT RIC, deployed at node i, is mapped to a Non-RT at node j, a Non-RT RIC must have been deployed at node j.

$$\forall i, j \in N \quad c_{ij} = G_j,$$

Since we do not consider the linear problem in RAN Sharing, we deploy only one Non-RT RIC.

$$\sum_{i}^{N} G_i = 1,$$

6.8 Evaluation Framework

6.8.1 Introduction

As we assess the controller placement problem, we secure genuine datasets that comprehensively describe the landscape of cellular networks, including 4G and 5G mobile networks. We use two different datasets, consisting of the official coverage map of Switzerland, which is publicly available from the European Commission, and a carefully curated supplementary dataset provided by OpenCellID. Once meticulously processed, the datasets determine the location of the 4G and 5G towers to compute the linear solution, resulting in the deduction of the RIC positions. Determining RIC locations prepares the ground for the accurate simulation environment in NS-3. We use the locations on the datatsets to deduce the simulated network topology while the linear solution determines the controller placement. The following sections describe each sequential step towards controller placement evaluation.

6.8.2 Processing Datasets

The dataset by the European Commission covers the entirety of the coverage spectrum in Switzerland, including all 4G and all 5G cells scattered across the geography of Switzerland, and is illustrated in Figure 6.26. However, a single mobile cell tower can support multiple cells, one for each RU. Particularly given that our focus is on the mapping between controllers and mobile cell towers, rather than the radio units themselves, K-Means comes into play to locate the approximate positions of cell towers. A single run of K-Means deducts gNB positions, which is the center of the RU cluster. In disaggregated deployments, if we run K-Means twice, it helps us figure out DU positions by finding the center of the RU clusters, and then it tells us where the CU positions are in the DU clusters. So, if we run K-means twice within the first set, we'll get either gNB positions or the combination of CU and DU positions, depending on how deep the algorithm is running.

The OpenCellID dataset provides a comprehensive view of cell towers ranging from 1G to 4G, covering the entire spectrum of existing towers globally, allowing for the flexibility to limit the scope to particular cities. Given our goal of combining the European Commission and OpenCellID datasets, we focus on Switzerland, specifically Zurich, as illustrated in Figure 6.27. Taking advantage of the capabilities of [72] and [73], we can extract the geographical boundaries of Zurich, thus refining the dataset to include the contents enclosed in

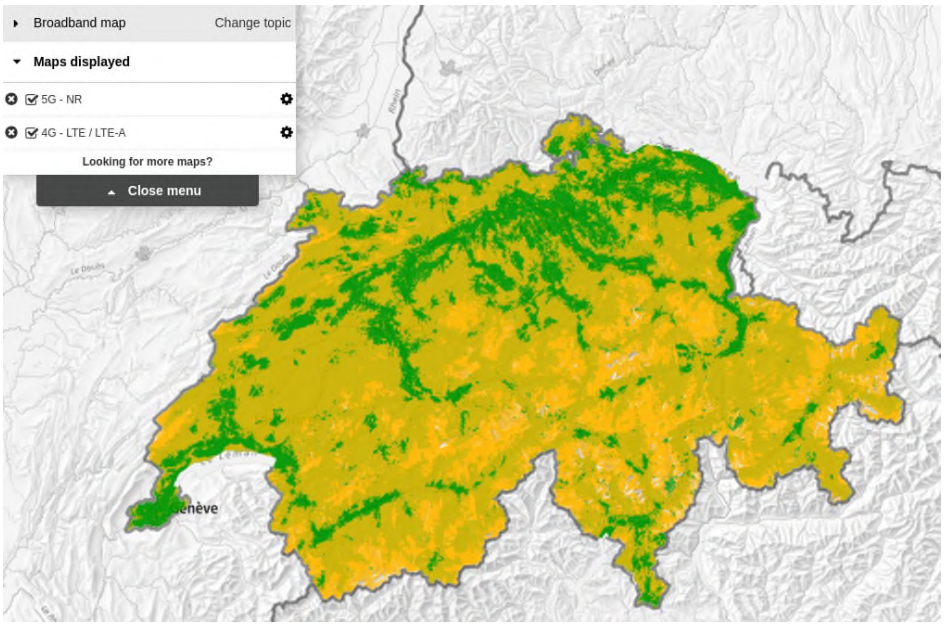


Figure 6.26: European Commission Dataset of Switzerland 4G and 5G Cells.

the red area in Figure 6.28.

Finally, we combine the two previously mentioned datasets into one. However, due to the limitations of our computational resources, we decided to sample the cell tower dataset as seen in 6.29. This intentional decision helps to reduce the computational load during processing. In addition, within this framework, we simplify our approach by assigning a single LTE tower while the other towers take on the NR designation. This aspect will be elaborated upon in the forthcoming section regarding the actual simulation.

6.8.3 Simulation Environment

To run the simulation, we start by analyzing the dataset in NS-3 that we have selected. This analysis includes the exact location of 5G mobile towers according to the dataset's requirements. Since the operational flow of the simulation is that all UE devices start in LTE cells and then migrate to NR via handovers, a 4G mobile tower is strategically placed at the center of the simulation topology. This strategic positioning is in line with the connectivity needs of the scenario.

Once we implement the cellular network in NS-3, the next step is establishing a connection between each E2 node and a Near-RT RIC. The total Near-RT RIC instances and their respective positions are not specified. The linear controller placement solution deduces this information, including the Non-RT RIC location. We use Python3 and the PuLP library to

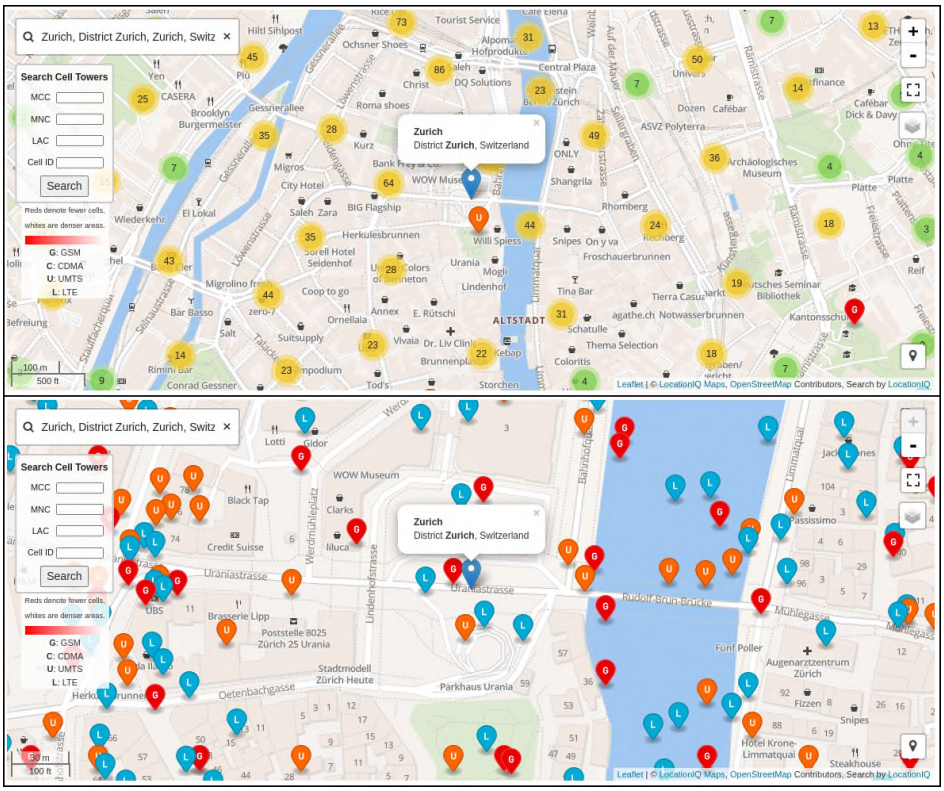


Figure 6.27: OpenCellID Switzerland Zurich.

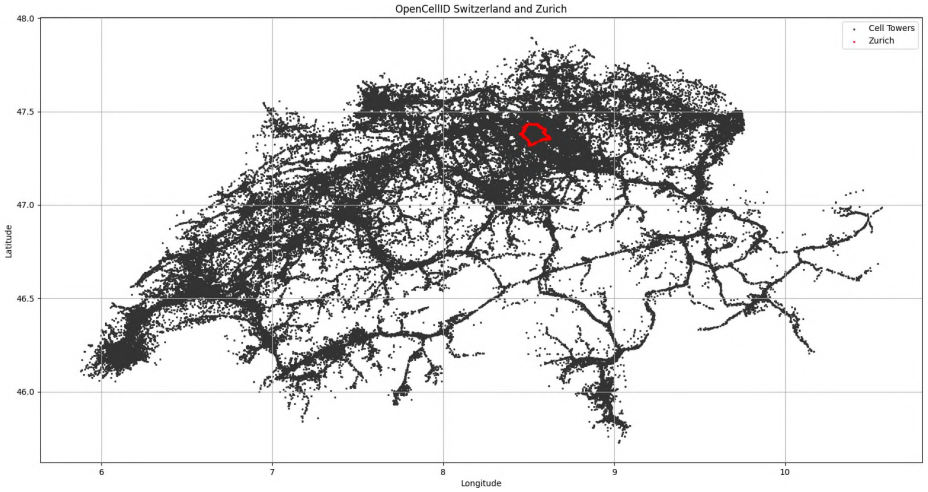


Figure 6.28: OpenCellID Switzerland Zurich.

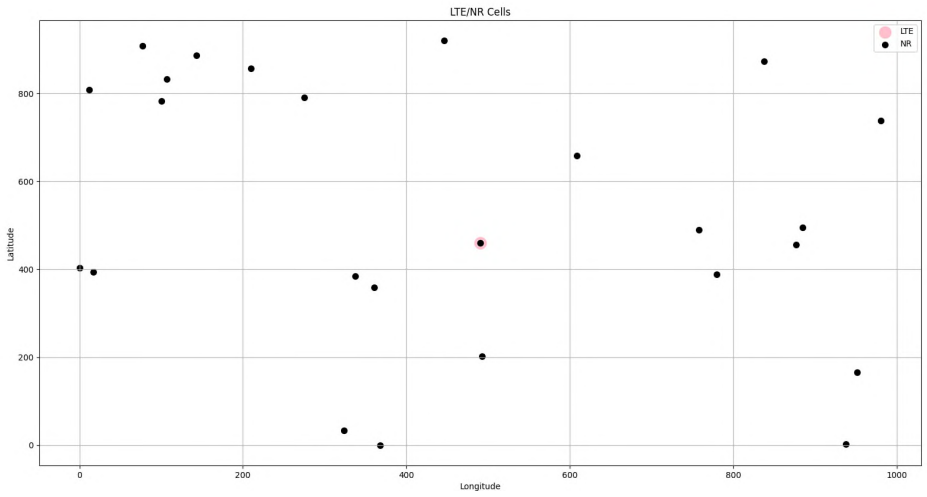


Figure 6.29: Sampled Combination of European Commission and OpenCellID Datasets.

model the cellular network in linear programming and to derive the linear solution based on the specified objective.

6.8.4 Proposed Experiments

The experiments focus on simulating two different scenarios in a similar network topology. Each scenario requires a different controller placement configuration to achieve the best possible performance. In these simulations, we highlight the importance of optimal controller placement, emphasizing its importance in improving the performance of the scenarios discussed in the following sections.

CA

In the first scenario, the network necessitates increased data rates, necessitating CA. In this situation, a set of UEs connected to gNBs move through the network's boundaries in different directions. Each UE can perform handovers according to a threshold based on the Signal-To-Interference-Plus-Noise ratio (SINR) received from the current base station relative to neighboring base stations.

In this scenario, the gNB plays a critical role by sending an indication report to the relevant Near-RT RIC. This report contains critical information, such as the current number of used and available but unused component carriers and the number of connected devices. Based on this data, the Near-RT RIC makes a critical decision. It decides whether to combine multiple carriers to increase bandwidth, a strategic decision to improve network performance,

as illustrated in Figure 6.30.

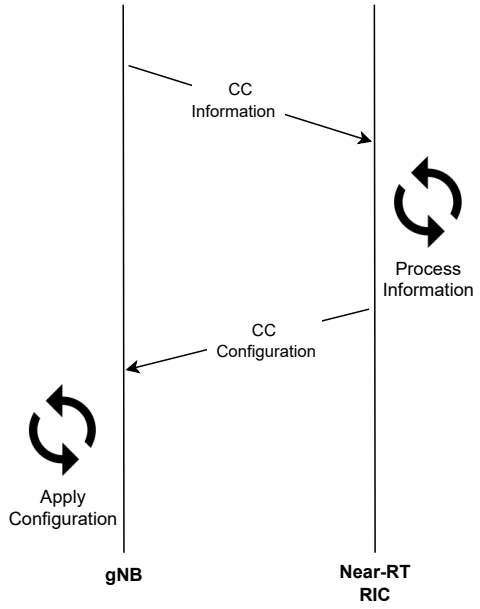
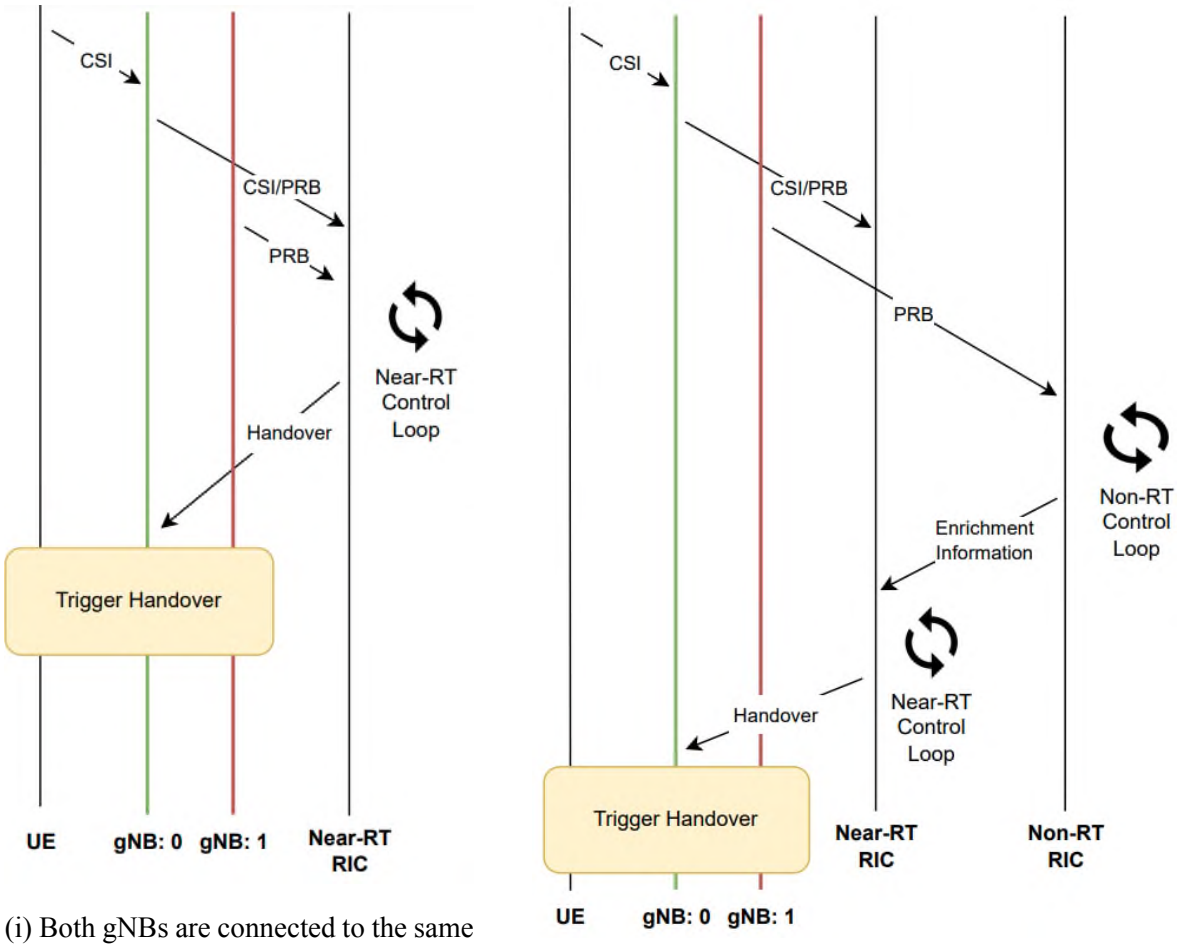


Figure 6.30: Report and Control Messages in CA.

Conditional Handovers

The second scenario examines conditional handovers inside the cellular network. One of the drawbacks of handing off handover decision-making to the UE is that it cannot make decisions based on the complete network state. For example, when a UE performs a handover based on the SINR alone, it does not know the cell tower's load it's switching to. This lack of knowledge led to the proposal of load-balancing conditional handover algorithms that avoid these suboptimal handover decisions. For these algorithms, we strategically move the decision-making process away from the UE and into the Next-RT RIC. Centralizing this responsibility allows the Next-RT RIC to gradually understand the total network status, leading to better and more efficient handover decisions in the long run.

The algorithm suggested in this scenario makes informed decisions for each UE by considering both signal strength and the load of the cell that the UE connects to and the neighboring mobile base stations. We extract the signal strength of each UE from the CSI report messages, which are messages sent periodically by each UE to its base station. The CSI report contains information about what SINR the UE experiences from its and the neighboring cells. Thus, this CSI report is also a list of possible neighbor cells to handover. The load of each base station is computed based on the usage of physical resource blocks (PRB) within



(i) Both gNBs are connected to the same Near-RT RIC

(ii) Each gNB is connected to a different Near-RT RIC

Figure 6.31: Load-Balancing Conditional Handover in Near-RT RIC

the base station. The base station forwards all CSI reports and PRB statistics to the Near-RT RIC.

Upon receipt of the potential neighbor's list from the CSI report, the Near-RT RIC obtains the PRB utilization data from each neighbor. If the adjacent cell is directly associated with the Near-RT RIC, the PRB data is included in the subsequent KPM report as long as it is not already present. Conversely, if the adjacent cell is not directly associated with the Near-RT RIC, the required PRB data is obtained from the Non-RT RIC as Enrichment Information. Once the Near-RT RIC has collected all neighbors' PRB utilization data, it will determine the UE's optimal base station. Depending on the decision, the Near-RT RIC either sends a handover control message to the relevant base station or performs no actions if the UE is already optimally paired with the correct base station. Figure 6.31i and Figure 6.31ii depict the described process.

6.8.5 Linear Solution

To assess the controller placement performance in CA and handover cases, we approach the linear optimization problem separately for each situation. As the optimal solution for one situation may not remain optimal in another, we can compare network behavior under optimal and sub-optimal controller placement.

In CA, the primary objective is to reduce the latency between a base station and a Near-RT RIC. In this case, information from multiple adjacent base stations is unnecessary, as the data from a single base station is sufficient to determine its future configuration by the Near-RT RIC, meaning that the linear problem has the following optimization objective.

$$\text{minimize} \quad \sum_{i,j \in N} d(i,j) y_{ij}$$

The $d(i,j)$ is a function that represents a proportional measurement of the distance between point i and j. The solution to the linear objective above gives the controller placement displayed in Figure 6.32.

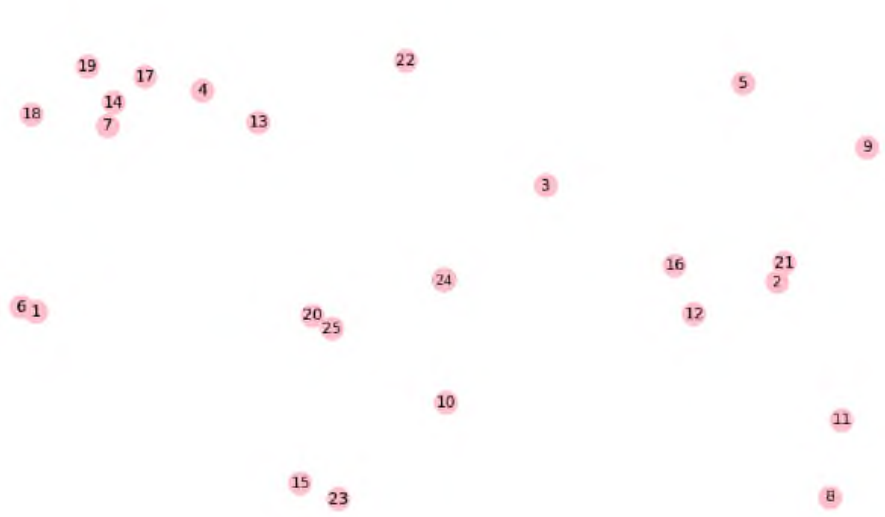


Figure 6.32: Same Nodes Contain Both E2 and Near-RT RIC

As observed, in the optimal solution for the current optimization objective, each node contains an E2 entity and its corresponding Near-RT RIC, minimizing latency between a base station and its Near-RT RIC is minimized.

The primary goal of the scenario involving handovers is to reduce the latency for each Near-RT RIC to communicate with base stations in close vicinity. When a Near-RT RIC does

not directly connect to a base station, the Non-RT RIC transmits the base station's information as enrichment data. In this scenario, it is necessary to fine-tune the linear solution to minimize the overall network delay while considering the fundamental requirement of obtaining states across all network nodes to enable decision-making, setting the following linear objective.

$$\text{minimize} \quad \sum_{i,j,k,l \in N} d(i,j)y_{ij} + d(k,j)y_{kj} + d(j,l)C_{jl}$$

Figure 6.33 illustrates the linear solution to the controller placement problem in cellular networks that require handovers. It is crucial to point out the coexistence of an E2 entity and a Near-RT RIC in both 14 and 24 nodes. Additionally, node 24 has the Non-RT RIC, which facilitates the transmission of enrichment information from node 24 to node 14.

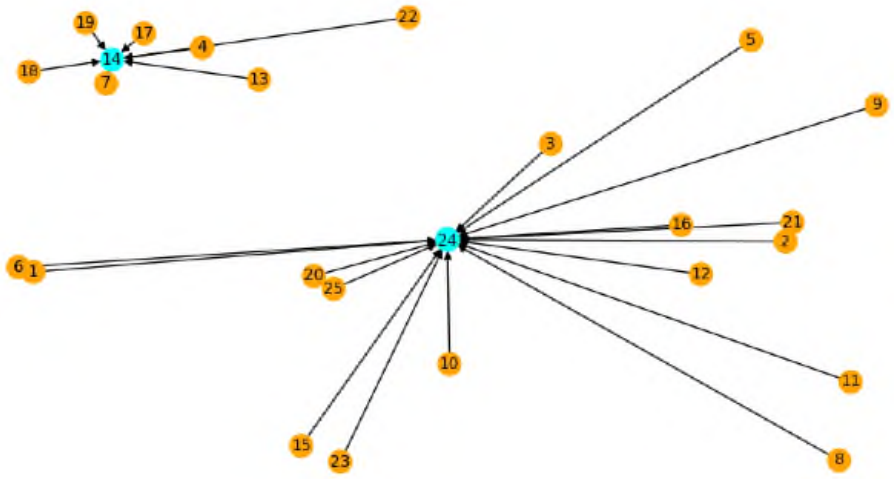
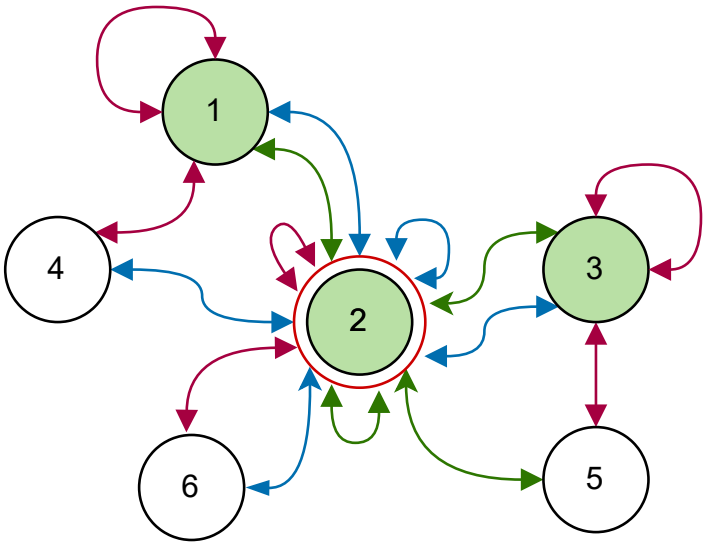


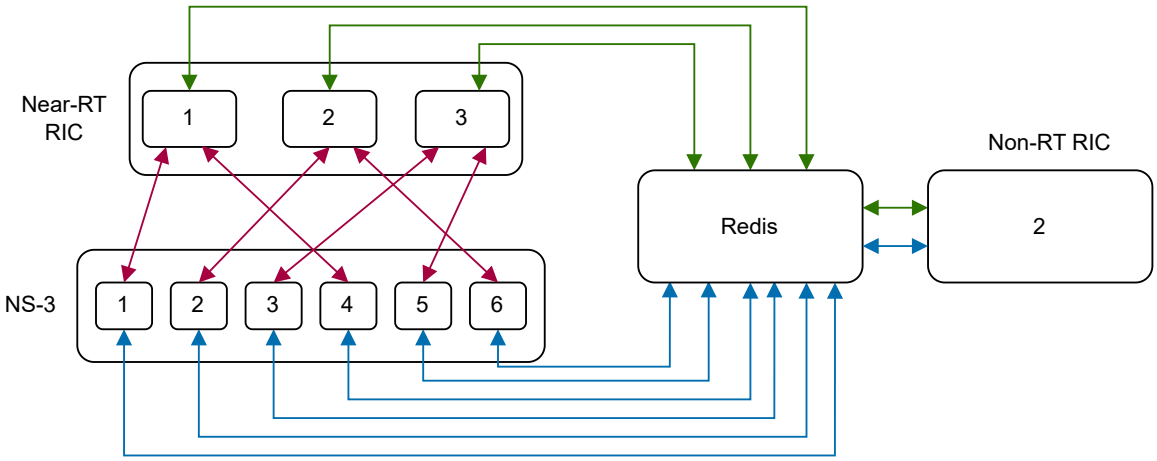
Figure 6.33: Nodes Containing E2 entities connected to their Near-RT RIC

6.8.6 Evaluation Framework

The NS-3 supports gNB and eNB nodes to share an IP address while utilizing distinct port numbers. The NS-3 parses the network topology and constructs a network containing only gNB nodes to simulate 5G. The center node within the topology is an eNB node, required by NS-3 for each UE to perform its initial connection. Each UE re-establishes a connection with a nearby gNB after the eNB connection. After creating the network topology, we solve the linear controller placement problem to deduce where to deploy the Near-RT RIC entities and the Non-RT RIC. Since the Colosseum Near-RT RIC connects with all deployed gNB and eNB nodes with the NS-3's IP, we modify it to support connections with individual nodes



(i) Network Topology



(ii) Simulation Environment

Figure 6.34: Network Topology and Simulation Environment

based on their port number. To simulate multiple Near-RT RIC entities, we deploy a single modified Near-RT RIC containing clusters of connections with gNB nodes. We compute the propagation delay between a node and its Near-RT RIC based on distance and queuing delays. The Non-RT RIC uses Redis for message passing to establish communication channels with Near-RT RIC entities and gNB nodes.

Figure 6.34 illustrates the implementation of an example network topology 6.34i to the actual simulation environment 6.34ii. Each connection has its delay settings, and we represent the E2, O1, and A1 interfaces with red, blue, and green colors. Finally, since NS-3 does not experience time in real-time, we should record every timestamp and measure time in NS-3.

6.8.7 Evaluation Results

In this section, we evaluate the performance of the optimal controller placement for each case and compare it to the performance achieved between the two controller placement solutions. Since the main focus of the experiments is to evaluate the efficiency of the linear solution and not the algorithms themselves, the lifetime of each KPM is measured. The measurement of each KPM starts when the KPM is about to be sent to the Near-RT or Non-RT RIC until an E2 entity receives the decision made using this KPM.

CA

In the CA scenario, we consider two different delay measurements, including the average delay since the E2 nodes send the indication message to Near-RT RIC and the average delay since the connection and disconnection of a UE. The former is the measurement taken regarding the lifetime of each KPM before the E2 node sends an indication message to the Near-RT RIC or the Non-RT RIC. The latter is the lifetime of the actual KPM at the base station since its generation. In both cases, the lifetime ends after the Near-RT RIC decides its actions when the E2 receives the control message sent by the Near-RT RIC. Both delay measurements are illustrated in Figure 6.35 under the two distinct controller placement solutions.

Figure 6.35: Delay Measurements in CA

As suspected, the linear objective that results in a distributed controller placement outperforms the centralized controller placement since the Near-RT RIC does not need to communicate with multiple base stations. The distributed controller placement avoids the unnecessary propagation delay due to high distances and the reliance on enrichment information from Non-RT RIC, which is slow due to the Non-RT control loop.

In the CA scenario, an additional informative measurement is the percentage of incorrect decisions that the Near-RT RIC makes. These incorrect decisions can occur due to inconsis-

tencies in information shared between base stations and their corresponding Near-RT RIC due to delays in the network. To monitor inconsistent decisions, the Near-RT RIC records the KPM it based its decision on, encapsulates it in a control message along with its actions, and sends it to its connected E2 nodes to modify their current configuration. The E2 compares its current state with the encapsulated KPM, included in the control message, received from the Near-RT RIC. When these metrics differ, an inconsistency occurs. As presented in Figure 6.36, the distributed controller placement results in more accurate decisions contrary to the centralized placement. Whenever network delays exist, each Near-RT might consider expired metrics for decision-making.

Figure 6.36: Inconsistencies in CA

Conditional Handover

In the conditional handover scenario, our evaluation of controller placement performance considers three distinct measurement values. When the Near-RT RIC makes a decision, and the base station receives the control message, each KPM considered for the decision is associated with a timestamp indicating its transmission time to either the Near-RT or Non-RT RIC. The difference between the simulation time that a handover was triggered and the timestamp of each KPM used in the decision-making process is called staleness. To evaluate the linear solution to the controller placement problem in this scenario, the oldest, latest, and average KPM staleness values, as illustrated in Figure 6.37, are compared between the distributed and centralized controller placement. It is important to note that we decreased the Non-RT control loop from the 1s, mentioned in the specification, to 100ms due to hardware limitations and slow simulation times in NS-3.

As observed, the average staleness of the oldest metrics in centralized controller placement slightly outperforms the distributed one because some KPM, regardless of the controller

placement, are still provided by the Non-RT RIC as enrichment information, casting both controller placement solutions almost similar. However, the average staleness of the latest metrics in centralized controller placement is slightly worse than the distributed one since the distributed placement results in minimal propagation delay between the Near-RT RIC and its E2 nodes. Finally, we prove that the linear solution optimizes the controller placement objective, minimizing the average network delay. As presented, the average network staleness decreases in the centralized controller placement compared to the distributed one, enabling controllers to manage the freshest metrics overall, leading to improved decision-making across the board.

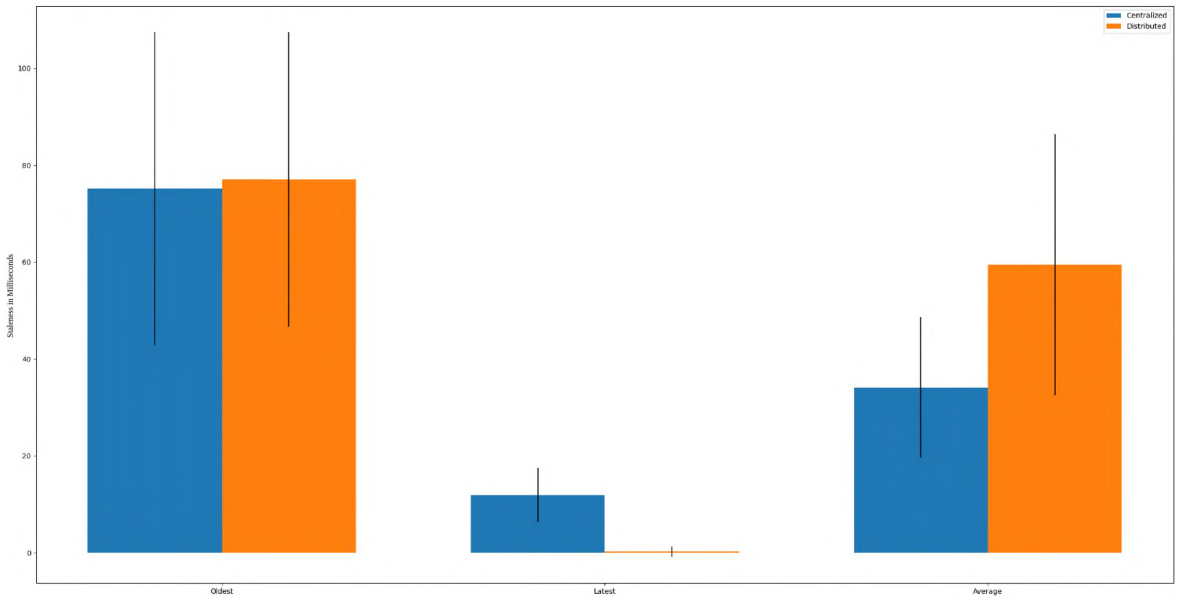


Figure 6.37: Delay Measurements in HO

6.9 Extending the Linear Model

As networks advance, the complexity of RIC usage increases, particularly with the utilization of xApps and rApps, which involve the training and inference of machine learning models. Since each network node has limited computational capacity and bandwidth, controller placement transforms into a problem regarding the deployment of applications, being xApps if deployed on Near-RT RIC or rApps if on Non-RT RIC, and ML model training and inference.

6.9.1 Additional Constant Variables

Boolean Variables

These variables, described in this section, can be either one or zero and are defined by the network engineers or vendors based on their desired network capabilities.

$$A_{ij} = \begin{cases} 1 & \text{if node deployed at node } i \text{ uses application } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$B_{ij} = \begin{cases} 1 & \text{if application } i \text{ uses model } j, \\ 0 & \text{otherwise.} \end{cases}$$

Non-Boolean Variables

These variables, described in this section, are positive numbers determined by the hardware and network capabilities and software complexity.

$M_i \geq 0$: Computational capacity of node i.

$F_i \geq 0$: Computational complexity to train model i.

$I_i \geq 0$: Computational complexity for inference of model i.

$e_{ij} \geq 0$: Bandwidth required between E2 node at i and Near-RT RIC at j.

$o_{ij} \geq 0$: Bandwidth required between E2 node at i and Non-RT RIC at j.

$g_{ij} \geq 0$: Bandwidth required between Near-RT RIC at i and Non-RT RIC j.

6.9.2 Additional Decision Variables

$$T_{ij} = \begin{cases} 1 & \text{if Non-RT RIC deployed at node } i \text{ trains model } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$t_{ij} = \begin{cases} 1 & \text{if Near-RT RIC deployed at node } i \text{ trains model } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$a_{ij} = \begin{cases} 1 & \text{if node } i \text{ has an application } j \text{ deployed,} \\ 0 & \text{otherwise.} \end{cases}$$

$$b_{ij} = \begin{cases} 1 & \text{if node } i \text{ deploys model } j \text{ ,} \\ 0 & \text{otherwise.} \end{cases}$$

6.9.3 Additional Constraints

Hardware Utilization

When the Non-RT RIC trains a model, it can be deployed to a Near-RT RIC, eliminating the need for Near-RT RIC to train the model independently. However, if the Non-RT RIC does not train the model, each interested Near-RT RIC must train the model itself.

$$\forall i, j \in N \quad T_{ij} \geq 1 - \sum_{k}^{N} t_{ki},$$

If a Near-RT RIC and a Non-RT RIC coexist on the same computational node, we should not calculate the training process twice.

$$\forall i, j \in N \quad T_{ij} + t_{ij} = 1,$$

When training a model in Non-RT RIC, the Non-RT RIC must exist.

$$\forall i, j \in N \quad T_{ij} \leq G_i,$$

Accordingly, the Near-RT RIC must exist to train a model on it.

$$\forall i, j \in N \quad T_{ij} \leq C_i,$$

An application can be deployed as an xApp on Near-RT RIC, an rApp on Non-RT RIC, or as a part of the RAN when used by the E2 node.

$$\forall i, j, k \in N \quad a_{ij} Aik \leq y_{ki} + x_{ki} + y_k,$$

If an E2 wants to use an application, a Near-RT or a Non-RT RIC should deploy the application at least once.

$$\forall j, k \in N \quad \sum_{i}^{N} a_{ij} \geq A_{kj},$$

Additionally, if an application uses a model, a Near-RT or a Non-RT RIC should deploy the model at least once.

$$\forall i, j, k \in N \quad a_{ij} B_{jk} \leq b_{ik},$$

This particular Near-RT RIC performs training, or the Non-RT RIC trains and deploys the model to either itself, the Near-RT RIC, or the E2 node.

$$\forall i, j, m \in N, \forall k \in M, \forall l \in L \quad y_{ij} A_{ik} B_{kl} \leq t_{jl} + T_{ml}$$

where N is the set of available nodes in the network., M are the available applications and L the available models.

The final constraint maintains the computational load within a node's hardware capacity.

$$\forall i \in N \quad M_i \geq \sum_{j}^{M} ((T_{ij} + t_{ij}) F_j + b_{ij} I_j)$$

where N and M are the set of available nodes and the set of models respectively.

Network Utilization

The link capacities between Near-RT RIC and E2 (6.1), Non-RT RIC and E2 (6.2), and Near-RT RIC and Non-RT RIC (6.3) should not surpass their maximum bandwidth capacity.

$$\forall i, j \in N \quad y_{ij} e_{ij} \leq b(i, j) \tag{6.1}$$

$$\forall i, j \in N \quad x_{ij} o_{ij} \leq b(i, j) \tag{6.2}$$

$$\forall i, j \in N \quad c_{ij} g_{ij} \leq b(i, j) \tag{6.3}$$

where $b(i, j)$ represents a function that provides the maximum permissible bandwidth between nodes located at positions i and j.

Chapter 7

Reflections and Future Directions

This concluding chapter reexamines the objective of this thesis with a focus on presenting the significance and contribution of this research study.

7.1 Summary

This thesis is a journey through the optimization process on mobile cellular networks, focusing on 5G and beyond. It offers a general overview of the architecture and functionality of the different cellular network generations, and it presents the importance of high performance along with the mathematical models required to achieve it. Specifically, it introduces AI/ML methods in AQM and LP in Controller Placement. It also dives deep into the history of AQM research, contributing by pioneering an approach for AQM in disaggregated 5G network deployments. Finally, this thesis explores all possible controller usages in 4G, 5G, and beyond cellular networks while it proposes and evaluates the controller placement in such networks.

7.2 Conclusion

The AQM optimization showcased remarkable performance enhancements within disaggregated high-latency network deployments, enabling a more open and flexible implementation approach of 5G's RAN. Additionally, optimal controller placement underscored the critical significance of strategically positioning controllers according to the modern requirements of cellular networks.

7.3 Future Work

We selected the DRQL AQM algorithm due to its simplicity; however, for a more comprehensive assessment of the proposed scheme's effectiveness in handling AQM in disaggregated networks, future work should include testing other algorithms like SFQ, 5G-BDP, and USP. Testing such algorithms would offer additional insights into the scheme's capacity to tackle AQM challenges in disaggregated networks. Moreover, to further validate the effectiveness of our proposed scheme, it should also be evaluated under real-life scenarios using traffic generated by multiple users. To explore more alternatives, we should also compare the accuracy of other AI/ML models that use different KPM from different protocol stack layers based on the AQM algorithm implemented and the network's requirements. Finally, we will examine how straightforward the development of AI/ML models can be in the previously mentioned scenarios and the scheme's ability to make accurate predictions in more complex environments.

In controller placement, we should also simulate the hardware and network utilization to verify the correctness of the constraints. We should also accommodate the network model to capture the RAN Sharing scenario, where a Near-RT RIC establishes communication with its E2 nodes through multiple interconnected Non-RT RIC entities.

Bibliography

[1] Cellular networks. https://en.wikipedia.org/wiki/Cellular_network.

[2] N. Makris, C. Zarafetas, P. Basaras, T. Korakis, N. Nikaein, and L. Tassiulas. Cloud-Based Convergence of Heterogeneous RANs in 5G Disaggregated Architectures. In *Proc. IEEE ICC*, 2018.

[3] Interface between the Control Plane and the User Plane nodes (3GPP TS 29.244 version 16.5.0 Release 16).

[4] Andres Garcia-Saavedra and Xavier Costa-Pérez. O-ran: Disrupting the virtualized ran ecosystem. *IEEE Communications Standards Magazine*, 5(4):96–103, 2021.

[5] Bharath Balasubramanian, E. Scott Daniels, Matti Hiltunen, Rittwik Jana, Kaustubh Joshi, Rajarajan Sivaraj, Tuyen X. Tran, and Chengwei Wang. Ric: A ran intelligent controller platform for ai-enabled cellular networks. *IEEE Internet Computing*, 25(2):7–17, 2021.

[6] Supervised Learning. https://www.javatpoint.com/supervised-machine-learning.

[7] Unsupervised Learning. https://towardsdatascience.com/supervised-vs-unsupervised-learning-in-2-minutes-72dad148f242.

[8] Reinforcement Learning. https://becominghuman.ai/the-very-basics-of-reinforcement-learning-154f28a79071?gi=66448831373a.

[9] NITLAB. https://nitlab.inf.uth.gr.

[10] NITOS). http://nitos.inf.uth.gr.

[11] CERTH. https://www.certh.gr.

[12] OneLab. https://onelab.eu.

[13] Florian Kaltenberger, Aloizio P. Silva, Abhimanyu Gosain, Luhan Wang, and Tien-Thinh Nguyen. Openairinterface: Democratizing innovation in the 5g era. *Computer Networks*, 176:107284, 2020.

[14] NS-3). https://www.nsnam.org.

[15] Andrea Lacava, Michele Polese, Rajarajan Sivaraj, Rahul Soundrarajan, Bhawani Shanker Bhati, Tarunjeet Singh, Tommaso Zugno, Francesca Cuomo, and Tommaso Melodia. Programmable and customized intelligence for traffic steering in 5g networks using open ran architectures. *IEEE Transactions on Mobile Computing*, pages 1–16, 2023.

[16] NS-3 mmWave. https://github.com/wineslab/ns-o-ran-ns3-mmwave.

[17] NS-3 E2 O-RAN. https://github.com/o-ran-sc/sim-ns3-o-ran-e2.

[18] Colosseum Near-Real-Time RIC. https://github.com/wineslab/colosseum-near-rt-ric.

[19] E2-SIM. https://github.com/wineslab/ns-o-ran-e2-sim.

[20] NumPy. https://numpy.org.

[21] Pandas. https://pandas.pydata.org.

[22] Tensorflow. https://www.tensorflow.org/.

[23] Tensorflow. https://en.wikipedia.org/wiki/TensorFlow.

[24] TensorFlow Keras Guide. https://www.tensorflow.org/guide/keras.

[25] Keras tuner documentation. https://keras.io/keras_tuner/.

[26] Docker. https://www.docker.com/.

[27] Redis. https://redis.io/.

[28] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi. 5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View. *IEEE Access*, 6:55765–55779, 2018.

[29] M. Irazabal, E. Lopez-Aguilera, I. Demirkol, R. Schmidt, and N. Nikaein. Preventing RLC Buffer Sojourn Delays in 5G. *IEEE Access*, 9:39466–39488, 2021.

[30] M. Irazabal, E. Lopez-Aguilera, and I. Demirkol. Active Queue Management as Quality of Service Enabler for 5G Networks. In *Proc. EuCNC*, 2019.

[31] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.

[32] K. M. Nichols, V. Jacobson, A. McGregor, and J. R. Iyengar. Controlled Delay Active Queue Management. *RFC*, 8289:1–25, 2018.

[33] T. Høiland-Jørgensen, P. E. McKenney, D. Täht, J. Gettys, and E. Dumazet. The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm. *RFC*, 8290:1–25, 2018.

[34] R. Pan, P. Natarajan, C. Piglione, M. S. Prabhu, V. Subramanian, F. Baker, and B. VerSteeg. PIE: A lightweight control scheme to address the bufferbloat problem. In *Proc. IEEE High Performance Switching and Routing (HPSR)*, 2013.

[35] A. K. Paul, H. Kawakami, A. Tachibana, and T. Hasegawa. An AQM based congestion control for eNB RLC in 4G/LTE network. In *Proc. IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2016.

[36] P.E. McKenney. Stochastic fairness queueing. In *Proc. IEEE INFOCOM*, 1990.

[37] M. Irazabal, E. Lopez-Aguilera, I. Demirkol, and N. Nikaein. Dynamic Buffer Sizing and Pacing as Enablers of 5G Low-Latency Services. *IEEE Transactions on Mobile Computing*, 21(3):926–939, 2022.

[38] R. Kumar, A. Francini, S. Panwar, and S. Sharma. Dynamic control of RLC buffer size for latency minimization in mobile RAN. In *Proc. IEEE WCNC*, 2018.

[39] O-RAN Near-Real-time RAN Intelligent Controller E2 Service Model (E2SM) KPM 2.0. O-RAN Working Group 3, July 2021.

[40] Open Air Interface (OAI). https://openairinterface.org/.

[41] R. Schmidt, M. Irazabal, and N. Nikaein. FlexRIC: An SDK for next-Generation SD-RANs. In *Proc. CoNEXT*, 2021.

[42] Open Air Core Network 5G. https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed.

[43] iPerf. https://iperf.fr/.

[44] Ping. https://linux.die.net/man/8/ping.

[45] Fabian Pedregosa et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

[46] Martin Ababi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, 2016.

[47] Keras. https://keras.io/.

[48] Software-Defined Networking. https://www.vmware.com/topics/glossary/content/software-defined-networking.html.

[49] Network Functions Virtualization. https://www.sdxcentral.com/networking/nfv/definitions/whats-network-functions-virtualization-nfv.

[50] Brandon Heller, Rob Sherwood, and Nick McKeown. The controller placement problem. In *Proc. First Workshop on Hot Topics in Software Defined Networks*, pages 7–12, 2012.

[51] Bala Prakasa Rao Killi and Seela Veerabhadreswara Rao. Controller placement with planning for failures in software defined networks. In *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6, 2016.

[52] Tamal Das and Mohan Gurusamy. Controller placement for resilient network state synchronization in multi-controller sdn. *IEEE Communications Letters*, 24(6):1299–1303, 2020.

[53] Richard Cziva, Christos Anagnostopoulos, and Dimitrios P. Pezaros. Dynamic, latency-optimal vnf placement at the network edge. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 693–701, 2018.

[54] Aris Leivadeas, George Kesidis, Mohamed Ibnkahla, and Ioannis Lambadaris. Vnf placement optimization at the edge and cloud †. *Future Internet*, 11(3), 2019.

[55] Satyam Agarwal, Francesco Malandrino, Carla-Fabiana Chiasserini, and S. De. Joint vnf placement and cpu allocation in 5g. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 1943–1951, 2018.

[56] O-RAN Use Cases and Requirements 3.0 (O-RAN.WG3.UCR-R003-v03.00). https://orandownloadsweb.azurewebsites.net/specifications.

[57] O-RAN Slicing Architecture 10.0 (O-RAN.WG1.Slicing-Architecture-R003-v10.00). https://orandownloadsweb.azurewebsites.net/specifications.

[58] O-RAN Massive MIMO Use Cases Technical Report 1.0 (O-RAN.WG1.mMIMO-Use-Cases-TR-v01.00)). https://orandownloadsweb.azurewebsites.net/specifications.

[59] Henrik Martikainen, Ingo Viering, Andreas Lobinger, and Tommi Jokela. On the basics of conditional handover for 5g mobility. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–7, 2018.

[60] Changsung Lee, Hyoungjun Cho, Sooeun Song, and Jong-Moon Chung. Prediction-based conditional handover for 5g mm-wave networks: A deep-learning approach. *IEEE Vehicular Technology Magazine*, 15(1):54–62, 2020.

[61] Baud Haryo Prananto, Iskandar, and Adit Kurniawan. A new method to improve frequent-handover problem in high-mobility communications using ric and machine learning. *IEEE Access*, 11:72281–72294, 2023.

[62] Andreas Lobinger, Szymon Stefanski, Thomas Jansen, and Irina Balan. Load balancing in downlink lte self-optimizing networks. In *2010 IEEE 71st Vehicular Technology Conference*, pages 1–5, 2010.

[63] Andreas Lobinger, Szymon Stefanski, Thomas Jansen, and Irina Balan. Coordinating handover parameter optimization and load balancing in lte self-optimizing networks. In *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, pages 1–5, 2011.

[64] Sara Khosravi, Hossein S. Ghadikolaei, and Marina Petrova. Learning-based load balancing handover in mobile millimeter wave networks. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–7, 2020.

[65] O-RAN Use Cases Analysis Report 11.0 (O-RAN.WG1.Use-Cases-Analysis-Report-R003-v11.00). https://orandownloadsweb.azurewebsites.net/specifications.

[66] 3GPP. Vocabulary for 3GPP Specifications. Technical Report 21.905, 3rd Generation Partnership Project (3GPP), 2015.

[67] ETSI. Metrics and measurement method for energy efficiency of wireless access network equipment; Part 1: Power consumption - static measurement method. Technical Specification ES 202 706-1, European Telecommunications Standards Institute (ETSI), 2022.

[68] O-RAN Network Energy Saving Use Cases Technical Report 2.0 (O-RAN.WG1.Network-Energy-Savings-Technical-Report-R003-v02.00). https://orandownloadsweb.azurewebsites.net/specifications.

[69] 3GPP. 5G System: Network Exposure Function Northbound APIs: Stage 3 (3GPP TS 29.522 version 15.3.0 Release 15). Technical Report TS 29.522, 3GPP, 2019.

[70] ETSI. Multi-access Edge Computing (MEC): Framework and Reference Architecture. Technical Specification GS MEC 003, ETSI, 2019.

[71] 3GPP. Management and orchestration; Self-Organizing Networks (SON) for 5G networks. Technical Specification 28.313, 3rd Generation Partnership Project (3GPP), December 2020. Release 16.

[72] Nominatim openstreetmap search. https://nominatim.openstreetmap.org/ui/search.html.

[73] Openstreetmap polygons. https://polygons.openstreetmap.fr/index.py.