Reinforcement Learning Based Resource Allocation for Network Slicing in O-RAN

by

Nien Fang Cheng

Thesis submitted to the University of Ottawa
in partial Fulfillment of the requirements for the

Master of Computer Science and
Concentration Applied Artificial Intelligence
University of Ottawa

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Fifth Generation (5G) introduces technologies that expedite the adoption of mobile networks, such as densely connected devices, ultra-fast data rate, low latency and more. With those visions in 5G and 6G in the next step, the need for a higher transmission rate and lower latency is more demanding, possibly breaking Moore's law. With Artificial Intelligence (AI) techniques becoming mature in the past decade, optimizing resource allocation in the network has become a highly demanding problem for Mobile Network Operators (MNOs) to provide better Quality of Service (QoS) with less cost.

This thesis proposes a Reinforcement Learning (RL) solution on bandwidth allocation for network slicing integration in disaggregated Open Radio Access Network (O-RAN) architecture. O-RAN redefines traditional Radio Access Network (RAN) elements into smaller components with detailed functional specifications. The concept of open modularization leads to greater potential for managing resources of different network slices. In 5G mobile networks, there are three major types of network slices, Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC), and Massive Machine Type Communications (mMTC). Each network slice has different features in the 5G network; therefore, the resources can be relocated depending on different needs. The virtualization of O-RAN divides the RAN into smaller function groups. This helps the network slices to divide the shared resources further down.

Compared to traditional sequential signal processing, allocating dedicated resources for each network slice can improve the performance individually. In addition, shared resources can be customized statically based on the feature requirement of each slice. To further enhance the bandwidth utilization on the disaggregated O-RAN, a RL algorithm is proposed in this thesis on midhaul bandwidth allocation shared between Centralized Unit (CU) and

Distributed Unit (DU).

A Python-based simulator has been implemented considering several types of mobile User Equipment (UE)s for this thesis. The simulator is later integrated with the proposed Q-learning model. The RL model finds the optimization on bandwidth allocation in midhaul between Edge Open Cloud (O-Cloud)s (DUs) and Regional O-Cloud (CU). The results show up to 50% improvement in the throughput of the targeted slice, fairness to other slices, and overall bandwidth utilization on the O-Clouds. In addition, the UE QoS has a significant improvement in terms of transmission time.

Acknowledgements

First, I would like to thank my supervisor, Professor Melike Erol-Kantarci. She accommodated me into the big family of Networked systems and Communications Research Lab (NETCORE). She motivated me with her patience over the years, especially when I stumbled and almost gave up. She always inspires and encourages me to pursue it to the next level. Her commitment to life and research is an excellent example for every woman in STEM. I am very grateful to learn and work under her supervision.

Second, I would like to thank Dr. Turgay Pamuklu. I am grateful to follow his professional guidance, from technical writing to programming algorithms. His feedback and comments helped me build a solid foundation for my thesis and assisted me in completing the study.

Finally, I would like to thank all the people I met, including all the masters and PhD students who ever offered their warm hands. All the professors I took the lectures from are very knowledgeable and genuinely care about students professionally by showing and guiding them toward the path of quality research. Lastly, I would like to thank the University of Ottawa for providing a great opportunity to study in this beautiful winter city with all the great people coming from worldwide.

Dedication

This thesis is dedicated to all my loving family and friends for being supportive technically and mentally throughout the process.

*First, to my beloved **grandmother**, Chang-E Cheng-Cao, who passed away during COVID on Mar 22, 2021. She shows me, "Nothing is impossible for a willing heart." Her love is endless to our family, especially me.*

*To my beloved **husband**, Hsing-Cheng Chou. He supports all my decisions, whether good or bad and cheers me up throughout the process. It will not be possible to even start this dream without you.*

*I would also like to mention my **parents**, Tsung-Fu Cheng and Wen-Ching Chen. They are my lifelong teachers and supporters. They inspired me in science and encouraged me to follow their engineering path. They set a good example of continuing study and work-life balance for our family. To my **parents-in-law**, Tsao-Hsiung Chou and Chang-Hsing Yang, thank you both for taking me as your daughter without second thoughts and supporting us in studying and working aboard.*

*Last but not least, my two **little brothers**. Chin-Chieh Cheng, thank you for supporting me and guiding me on the last mile of the thesis process. Hung-Chun Chou, thank you for being supportive as always! I believe we can grow more together in the future.*

- Tiffany Nien Fang Cheng

Table of Contents

List of Figures

List of Tables

List of Abbreviations

2G Second Generation 60

3GPP 3rd Generation Partnership Project 1, 4, 5, 8, 22

4G Forth Generation 9, 19, 28

5G Fifth Generation iii, 1–3, 5, 19–21, 28, 29, 61, 62

6G Sixth Generation 20, 60

A3C Asynchronous Advantage Actor-Critic 13

AI Artificial Intelligence iii, 2, 3, 8, 11, 19–21, 25, 60

ANN Artificial Neural Network 12

AR Augmented Reality 6

BBU Baseband Unit 9, 10, 24

C-RAN Centralized Radio Access Network or Cloud Radio Access Network 7, 19, 24–26, 62

CAGR Compound Annual Growth Rate 1

CapEx Capital Expenditure 10

CNN Convolutional Neural Network 12

CU Centralized Unit iii, iv, 9, 10, 25, 26, 28, 40

D-RAN Distributed Radio Access Network 6, 26, 62

D2D Device to Device 20

DL Deep Learning 2, 12, 18

DQN Deep Q-learning 18, 25, 63

DRL Deep Reinforcement Learning 12, 18

DU Distributed Unit iv, 9, 10, 25, 26, 28–30, 33, 35, 36, 40

eMBB Enhanced Mobile Broadband iii, 5, 22, 25, 27, 33, 41, 45–59, 61, 62

F-RAN Fog Radio Access Network 8, 9

GAN Generative Adversarial Networks 12

GROVE Green Radio OVer Ethernet 24, 25

IoT Internet of Things 1, 5, 6

IoV Internet of Vehicles 5

KPI key performance indicator 5, 20, 22

LSTM Long Short Term Memory Networks 12

MDP Markov Decision Process 15, 29, 31, 35, 36, 61

MILP Mixed-Integer Linear Programming 24, 62

ML Machine Learning 2, 4, 12, 20

mMTC Massive Machine Type Communications iii, 5, 25

mmW Millimeter Wave 6, 28

MNO Mobile Network Operator iii, 10, 20, 24, 26, 61, 63

NFV Network function virtualization 7, 60

NLP Natural Language Processing 12

NR New Radio 1, 4, 5, 8, 19, 21

O-Cloud Open Cloud iv, 2, 4, 26, 28, 29, 31–38, 40–44, 48, 49, 52, 59, 61–63

O-CU Open Centralized Unit 9

O-DU Open Distributed Unit 9

O-RAN Open Radio Access Network iii, 1–4, 7–10, 19, 20, 22, 25–28, 34, 46, 60–62

OFDM Orthogonal Frequency-Division Multiplexing 19, 21, 22, 60

OOP Object-oriented Programming 36

OpEx Operational Expenditure 10, 20, 24, 25, 63

PPO Proximal Policy Optimization 13

QoE Quality of Experience 1, 2, 5, 52, 56, 59, 62

QoS Quality of Service iii, iv, 1, 2, 4, 5, 25, 27, 49, 52, 56, 59, 62

RAN Radio Access Network iii, 1–3, 6, 8, 19, 23–25, 60, 62

RAT Radio Access Technology 5

RB Resource Block 19, 21, 22, 60

RF Radio Frequency 43

RIC RAN Intelligent Controller 8, 10, 32, 35, 37

RL Reinforcement Learning iii, iv, 2–4, 12–14, 16, 18, 20, 26, 28, 29, 34, 36, 41, 60–63

RNN Recurrent Neural Network 12

RT Real Time 10, 28, 32

RU Radio Unit 9, 10, 28, 40, 43

SAC Soft Actor Critic 18

SARSA State–action–reward–state–action 26

SMO Service Management and Orchestration 11

SNR signal-to-noise Ratio 43

TD Temporal-Difference 16

UE User Equipment iv, 4, 26, 28, 29, 31, 32, 34–36, 41–45, 48, 49, 51–53, 58, 61, 62

URLLC Ultra-Reliable Low Latency Communications iii, 5, 22, 25, 27, 33, 35, 45, 47–52, 55–59, 61

V2X Vehicle to Everything 6

VM Virtual Machine 23

VNF Virtual Network Function 8

VR Virtual Reality 6

vRAN Virtualized Radio Access Network 7

WDM Wavelength Division Multiplexing 23

List of Symbols

j Edge O-Cloud (DU) 30

i UE 36

a range of bandwidth change 30

q action-value function 15

$\mathbf{A}$ Action space: a set of actions. 14

G Cumulative Reward 15

γ Discount factor 15, 16

ϵ behavior policy 16

α Learning rate 16

v UE movement speed 36

k network slice 30

π policy 15

$\mathbf{P_a}$ the probability of transition under action a. 14

$\mathbf{R_a}$ the immediate reward due to action a. 15

$\mathbf{S}$ State space: a set of states. 14

t time slot 30

Chapter 1

Introduction

1.1 Motivation

Fifth Generation (5G) mobile network has been publicly announced by the 3rd Generation Partnership Project (3GPP) Release 15 [18] since late 2017. Various specifications characterize a new era for the mobile network. Latest technologies are introduced to improve the Quality of Service (QoS) and Quality of Experience (QoE) by applying the concept of Internet of Things (IoT) everywhere in our daily life. The 5G Phase II was published later in Release 16[19]. This release provides much more detail and extensions to build an advanced foundation for 5G network, such as 5G New Radio (NR) and network slicing. The forecast of global Compound Annual Growth Rate (CAGR) in mobile devices is 8% and 30% on mobile speed in 5G from 2018 to 2023 [20].

5G NR gives an initial definition of disaggregated Radio Access Network (RAN) architecture. The Open Radio Access Network (O-RAN) Alliance [21] refines the NR with more openness and intelligent solution. Network slicing has been considered in 5G O-RAN

as the important key to improving the QoS by rethinking the resource allocation for each network slice based on its traffic feature.

In order to tackle the network issues of optimization problems that appear in massive, complex data flows, Artificial Intelligence (AI) plays an important role. Especially, the network can now be sliced by different QoS and QoE requirements. The analysis on optimizing resource allocation includes and is not limited to available bandwidth, CPU and power usage ranging from a global view of the network down to a functional unit in a single RAN. With consideration of many new factors, AI becomes a mandatory component as part of the 5G network. [22, 23]. Different Machine Learning (ML) techniques, such as (un)supervised learning, Deep Learning (DL), and Reinforcement Learning (RL), have been applied in different areas of practice based on different needs and demonstrate effective outcomes.

Therefore, this thesis aims to integrate a RL solution to optimize resource allocation on 5G O-RAN based on existing knowledge of functional split and network slicing.

1.2 Thesis Contribution

In this thesis, a Python-based Edge Open Cloud (O-Cloud) Network simulator is developed. The simulator includes multiple O-clouds to form a disaggregated network and user equipments moving at several speeds. A RL Q-learning model is proposed and integrated into the Python-based simulator. The main contribution of the thesis is to optimize the shared bandwidth in the edge disaggregated O-RAN. The proposed RL solution can allocate bandwidth dynamically for different network slices based on the incoming traffic situations. In addition, the QoS and QoE of end users on targeted network slice(s) are

benefited as a result. The research [24] was submitted and accepted by IEEE Consumer Communications and Networking Conference (CCNC2023) on September 2022 and presented on Jan 2023.

> **[C01] N. F. Cheng**, T. Pamuklu, and M. Erol-Kantarci, "Reinforcement Learning Based Resource Allocation for Network Slices in O-RAN Midhaul," in 2023 IEEE 20th Consumer Communications Networking Conference (CCNC), 2023, pp. 140–145. [Online]. Available: https://arxiv.org/abs/2211.07466

1.3 Organization of Thesis

After showing motivations and the contribution of the research to the dedicated areas in 5G and O-RAN with RL, Chapter 2 introduces the background knowledge of the related fields, such as RAN and AI. Those pieces of knowledge are widely used or discussed in different aspects in the later chapters. Chapter 3 covers the detail of prior works done by other researchers. Some research provided the inspiration or foundation for our work, and others offered different perspectives. The implementation of the methodology, from topology to algorithm, is explained in Chapter 4. Chapter 5 demonstrates a massive amount of results and analysis. Chapter 6 concludes the research and describes the future direction of the presented work.

Chapter 2

Background

This chapter presents general descriptions of relevant 5G technologies from 5G NR, network slicing, O-RAN, and functional split. Integration of those exciting telecommunication technologies is proposed to improve resource allocation on 5G Edge O-Cloud and ultimately provide better QoS for User Equipment (UE). Besides the communication protocols, ML techniques are also introduced in the chapter. Those ML algorithms show powerful learning ability on massive data transactions. RL is used as the main part of our proposed solution.

2.1 5G New Radio

3GPP has released standards in technical specifications and announced them publicly for every mobile generation since 1998. Early on, it provided guidelines for those well-known telecom vendors to follow globally. Nowadays, it has become an essential key for software and hardware integration between different parties, from academic researchers and private sectors to global companies. Anyone can contribute to different components in the mobile

network and later compose together in the field.

5G NR specification was released by 3GPP on 38 series [25] in the end of 2017. This specification has defined the new era of Radio Access Technology (RAT), including and not limited to the following technologies, the extension of IoT, network slicing, Internet of Vehicles (IoV) and more [26].

2.1.1 Network Slicing

Based on different requirements from end users various needs, three characteristics of network slices are targeted in the 5G mobile network [1] shown in Fig 2.1. Each network slice is targeted on a specific aspect of QoS and QoE and evaluated by different key performance indicators (KPIs). Three 5G network slices are Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC), and Massive Machine Type Communications (mMTC) [27]. They are the three main generic services supported by 5G wireless technology [27].

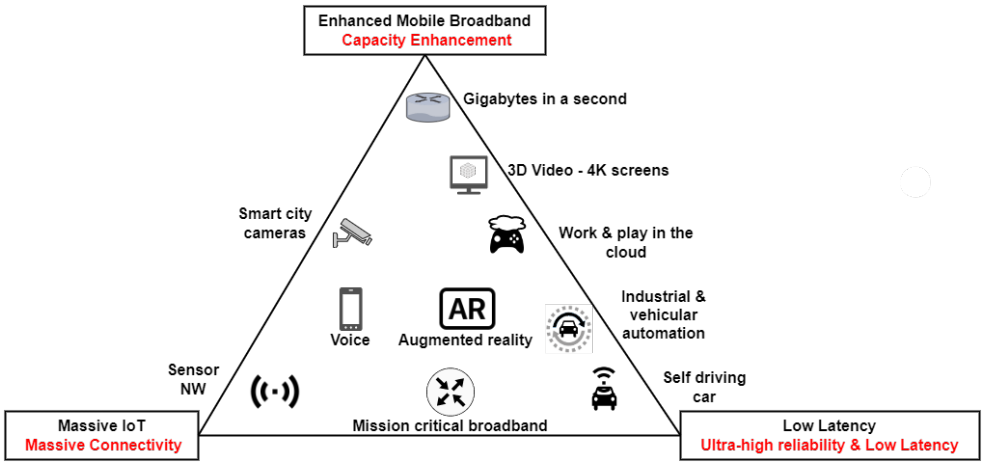

Figure 2.1: 5G services [1]

- **Enhanced Mobile Broadband (eMBB)** aims to achieve a 20 Gigabit per second [28] peak throughput rate, especially on Millimeter Wave (mmW) enabled frequency. The slice demands a high data rate from the recent data-driven 5G network, bringing better and faster user experiences than ever. The high data rate benefits various applications, such as Augmented Reality (AR), Virtual Reality (VR) or video conferencing services.

- **Ultra-Reliable Low Latency Communications (URLLC)** demand low latency and high reliability. The application of this service focuses on a real-time mission-critical area, such as autonomous driving and Vehicle to Everything (V2X) [29].

- **Massive Machine Type Communications (mMTC)** provide minimal guarantee connectivity to the high density of devices within the network range to achieve the goal of IoT.

2.2 Open Radio Access Network (O-RAN)

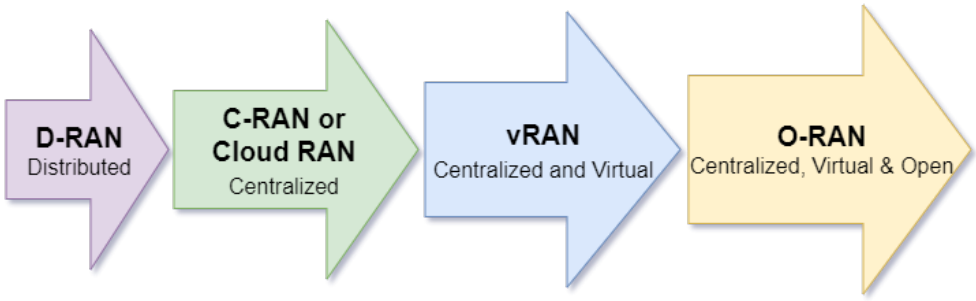

Figure 2.2: RAN Transformation [2]

The evolution of RAN has gone through the following steps shown in Fig. 2.2. In the traditional wireless network, RANs are mostly distributed and named Distributed Radio

Access Network (D-RAN). The centralized model emerged when cloud technology was recently introduced as Centralized Radio Access Network or Cloud Radio Access Network (C-RAN) [30]. Shortly, Virtualized Radio Access Network (vRAN) [31] becomes practical after the virtualization of the hypervisor is mature enough to build the end-to-end network in Network function virtualization (NFV). Soon after, O-RAN combines the cloud concept with virtualization and shares the specifications publicly for open access.

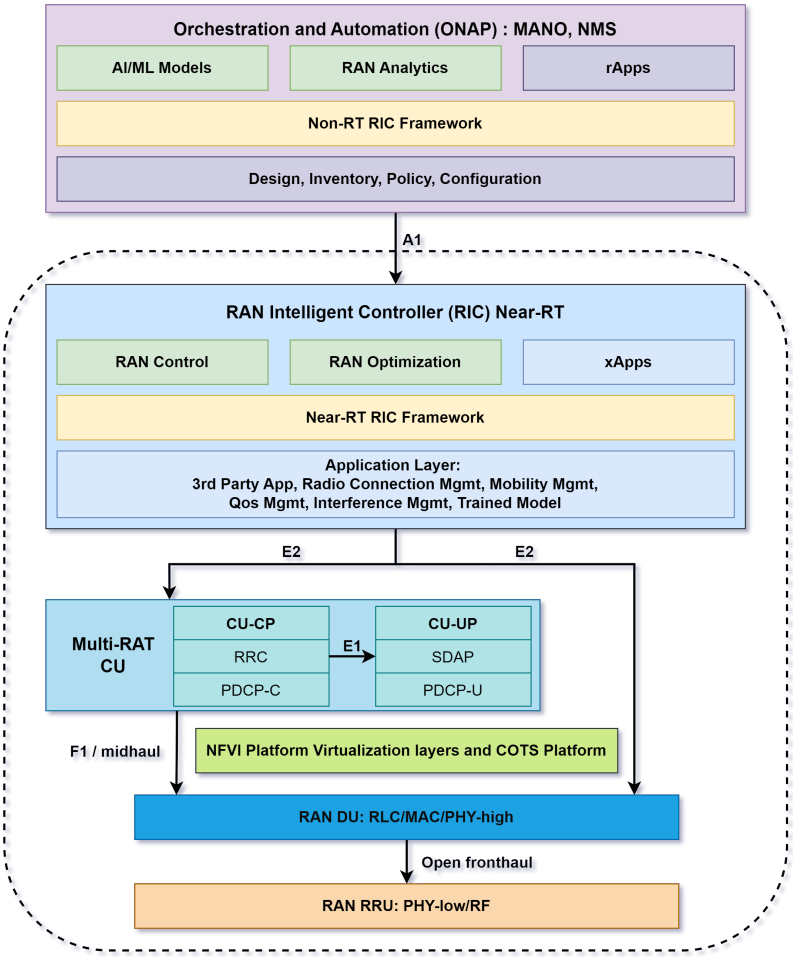

Figure 2.3: O-RAN Alliance Reference Architecture [3]

O-RAN specifications are defined and updated mainly by O-RAN Alliance [21]. The alliance was founded in 2018 by many well-known telecom giants. The implementation

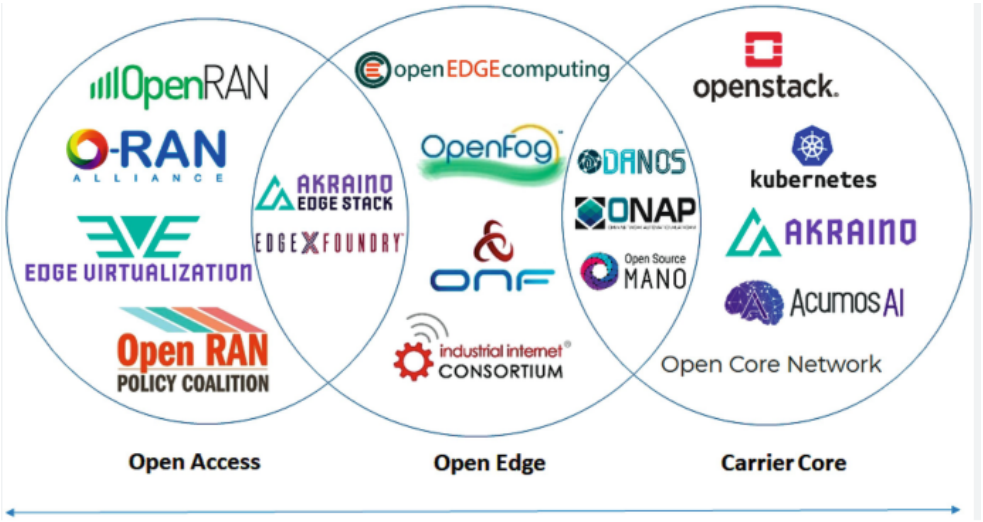

Figure 2.4: O-RAN vs F-RAN [4]

of O-RAN is based on top of 3GPP NR specifications [32]. Those specifications focus on openness and virtualization in 5G intelligent RAN [33].

The openness of O-RAN enables different parties, from academia and industry, to inter-operate their products in the fields together by following those guidelines and standards [34]. The Virtual Network Function (VNF) is highlighted as part of O-RAN implementation. Diverse functional roles are defined in detail by modularizing RAN functions into smaller software-defined components. Micro-managing these logical units becomes challenging and opens the door for AI. O-RAN specifies a functional role for this AI-enabled controller named RAN Intelligent Controller (RIC)[1] shown in Fig 2.3.

Besides O-RAN, Fog Radio Access Network (F-RAN), a parallel branch in RAN evolution, was introduced around the same time. By combining edge cloud computing and fog computing [35], Fog Radio Access Network (F-RAN) is a similar concept compared to O-RAN. Cisco has been one of the members funding the OpenFog [36] since 2016. Fig. 2.4

[1]RAN Intelligent Controller will be mentioned in more detail in Section 2.2.2.

explains the difference between the O-RAN and F-RAN. O-RAN focuses more on the access side toward end devices. F-RAN has a larger scale on the number of edge networks and plays as the gatekeeper before entering the core network.

2.2.1 Functional Split

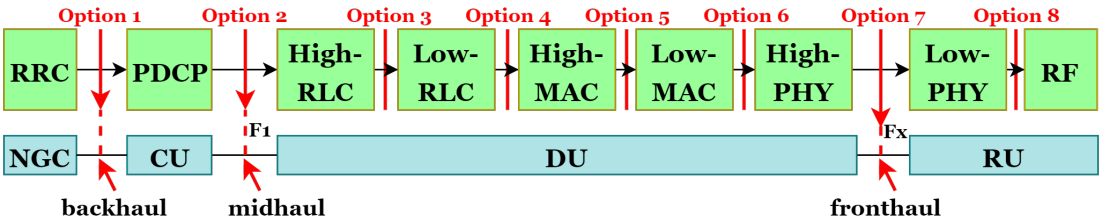

Figure 2.5: Options of Function Split in O-RAN [5]

The Baseband Unit (BBU) in Forth Generation (4G) has been divided into Centralized Unit (CU) and Distributed Unit (DU), along with Radio Unit (RU). With eight split options defined in 5G NR specification [32] shown in Fig 2.5, the modular concept provides placement freedom between CU, DU and RU shown in Fig 2.6. Bonati et al. [37] provides well-defined splitting concepts. With a detailed definition provided by O-RAN based on 5G drafts, new varieties of interfaces are introduced to connect between different stages within the disaggregated O-RAN in Fig 2.3, 2.5. An interface connecting two stages of units can operate on its own data rate and latency [37]. The fronthaul interface links between the RU and DU. O-RAN defines the interface between DU and CU and calls it midhaul or F1. The backhaul interface handles the traffic between CU and the core network in Fig 2.5.

The virtualized software-defined units in O-RAN create more freedom and combination on where to place those functional units. With the open concept in O-RAN, those logic units, CU and DU are associated as Open Centralized Unit (O-CU) and Open Distributed

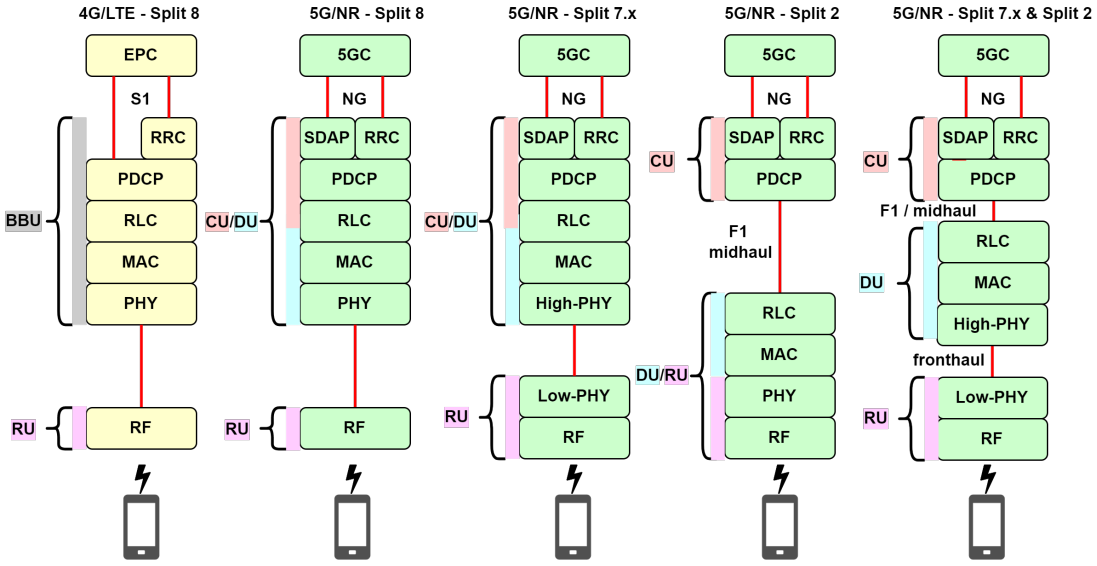

Figure 2.6: Placement of Function Splits in O-RAN [6]

Unit (O-DU). With eight major split options [38], the DU can move from the antenna station of RU to the centralized base station with CU. There are studies [38] debating the pros and cons between the decentralized and centralized models. The group placement or location of logical units is entirely based on the cost of Operational Expenditure (OpEx) and Capital Expenditure (CapEx) and hardware resources planned by Mobile Network Operator (MNO) network deployment.

2.2.2 RAN Intelligent Controller (RIC)

RIC is one of the most critical components specified in the O-RAN architecture. It manages and controls the signals coming into the BBU. Because it takes signals directly from both DU and CU shown in Fig. 2.7, the RIC is divided into near-Real Time (RT) and non-RT in order to control different types of requests based on their timing requirement.

- **Near-RT RIC** faces directly to both DU and CU independently by the E2 interface

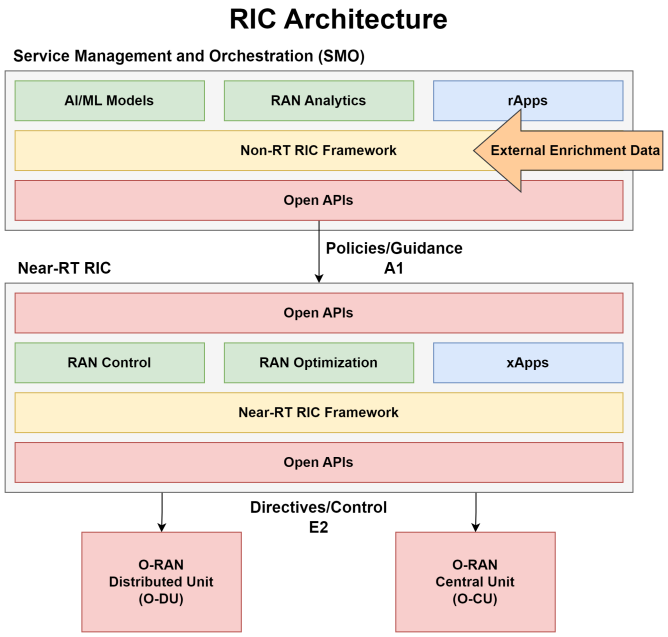

Figure 2.7: RAN Intelligent Controller Architecture [7]

shown in Fig 2.7. It aims to complete the low latency actions required by less than one second to ten milliseconds on its xApps.

- **Non-RT RIC** locates in the centralized Service Management and Orchestration (SMO). The control action can take more than one second to process. The AI models run on this layer as rApps to analyze the data and to give policy or guidance to the Near-RT RIC.

2.3 Machine Learning

Artificial Intelligence (AI) is a broad concept where humans can apply intelligence to machines with different methods and techniques. Within this decade, Machine Learning (ML) has become popular since the internet speed has increased exponentially, and memory storage is growing by following Moore's Law [39]. Storing, processing, and analyzing Big Data [40] become possible even in the mobile network. At the same time, the abilities of both hardware and software are evolving faster than ever from 3G to 5G in less than ten years. DL, RL and the combined version, Deep Reinforcement Learning (DRL), are famous families in ML to solve problems from different angles or complexity requirements (Fig 2.8).

2.3.1 Deep Learning

Deep Learning (DL) is known for process learning on multiple layers, Artificial Neural Network (ANN), to mimic the neural network [41]. The high-level features are extracted automatically after layers of learning in the neural network. The self-teaching process on historical data can be later used for recognition or prediction. Some famous neural networks are Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Generative Adversarial Networks (GAN). CNN is commonly applied in state-of-art computer vision [42]. RNN is widely used in Natural Language Processing (NLP) [43]. Long Short Term Memory Networks (LSTM)[44] is a famous time-memory variant in RNN family. The algorithm is known for solving problems with feedback as input to remember the time-relevant behaviour(s). However, due to its complexity, LSTM requires heavy computational resources so the execution time is one of the longest in ML.

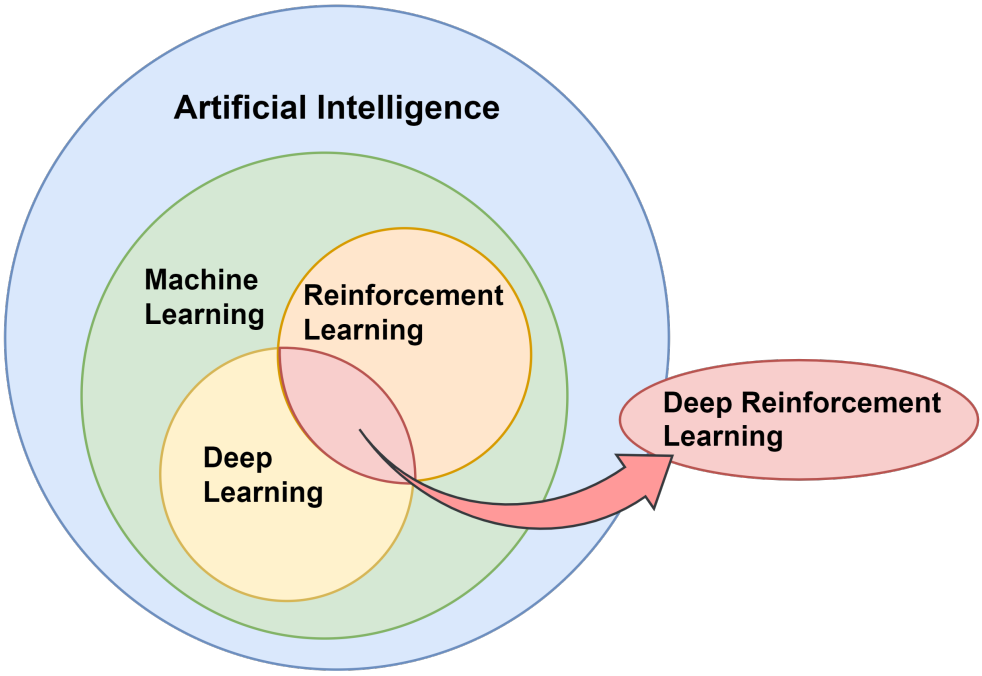

Figure 2.8: Relationship within AI [8]

2.3.2 Reinforcement Learning

Reinforcement Learning (RL) focuses on the decision-making and the consequence to the system shown in Fig 2.9. Those observations or environment changes will be recorded as part of the next state [45]. The model-free RL family branches out into policy-iteration and value-iteration methods. Proximal Policy Optimization (PPO) [46] and Asynchronous Advantage Actor-Critic (A3C) [47] are policy-oriented. Q-learning is the typical value-based RL algorithm.

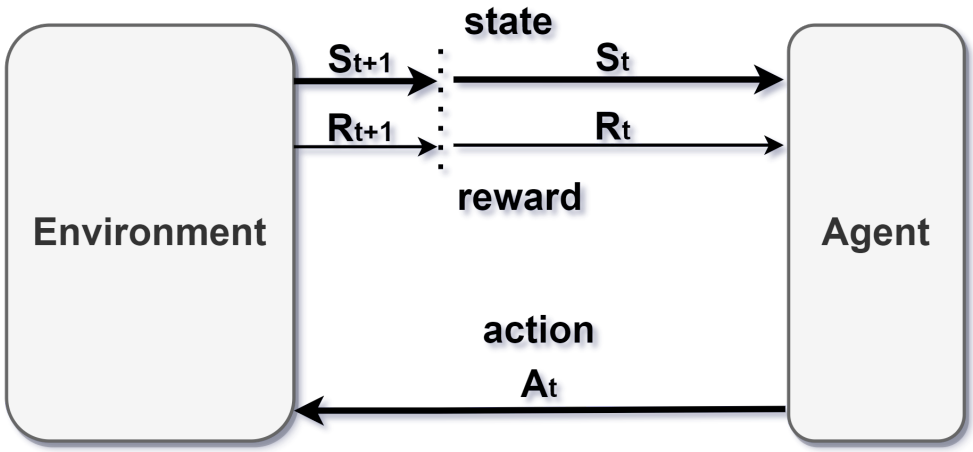

Figure 2.9: Interactions in Markov Decision Process [9]

2.3.3 Markov Decision Process

The RL environment usually follows the Markov Decision Process (MDP) shown in Fig 2.9. The agent or the so-called controller learns from the environment or the system and later makes the decisions to influence the environment [9]. Repeatedly, the agent takes an action, and the environment reacts and feeds back with a new state. Over time, the agent will be able to learn some patterns from the history series; later, those past experiences can be applied to maximize the outcome or rewards. The loop of this process has been defined mathematically in a 4-tuple (S, A, P_a, R_a)

- State space (**S**): a set of states.
- Action space (**A**): a set of actions.
- Probability ($\mathbf{P_a}$): the probability of transition (at time t) from state S to next state S_{t+1} under action a (2.1).

- Reward ($\mathbf{R_a}$): receives immediate after transitioning from state s to state s_{t+1}, due to action a.

The state-transition probability P_a represents the dynamics of a finite Markov Decision Process (MDP) in eq.(2.1). The probability is computed from the given state and action, followed by the preceding state and rewards at a particular time.

$$P_a(s_{t+1} \mid s) = Pr(s_{t+1} \mid s_t = s, a_t = a) \tag{2.1}$$

The expected reward at time t from the state-and-action pair at $t-1$ can be represented in eq.(2.2).

$$r(s, a) = \mathbf{E}\left[R_t | s_{t-1} = s, a_{t-1} = a\right] \tag{2.2}$$

Since the rewards can be predicted by the action and state pairs eq.(2.1), maximizing the cumulative reward G undoubtedly becomes the agent objective in the MDP. This goal can be achieved by finding and taking the right action. In order to receive a finite maximized cumulative reward, discount factor γ is introduced, where $0 \leq \gamma \leq 1$. Therefore, the cumulative reward G can be represented in eq.(2.3).

$$G_t = \sum_{t=0}^{\infty} \gamma^t R_a \tag{2.3}$$

As mentioned above, optimization can be done in two directions, value or policy based. The optimal policy π is defined mathematically as the expected return that is always better or equal to π' on all states. The action value function from the policy π on state s and action a can be represented by $q_\pi(s)$. The optimal action-value function looks for the

possible maximized Q-value from a policy π in eq.(2.5). The Bellman optimality equation can further transform the Q-value function into eq.(2.6).

$$q(s,a) = \max_{\pi} q_{\pi}(s,a) \tag{2.4}$$

$$= \mathbf{E}_{\pi}[R_{t+1} + \gamma G_{t+1} | s_t = s, a_t = a] \tag{2.5}$$

$$= \mathbf{E}[R_{t+1} + \gamma \max_{a'} q(s_{t+1}, a') | s_t = s, a_t = a] \tag{2.6}$$

2.3.4 Q-Learning

Q-learning is one of the popular RL algorithms and is known for being off-policy and model-free. It finds the optimized cumulative reward without the environment's transition function or the agent's policy. Instead, Q-learning stores past experiences in the lookup table. The so-called Q-table requires a good amount of memory. In other words, it takes random actions and stores the calculated Q-value based on new states and rewards in the exploring phase.

Q-value Function

Q-learning is a Temporal-Difference (TD) learning [9] and focuses on optimizing the action-value function, Q. The equation (2.7) considers the concepts of the learning rate α and the discount factor γ. With those minimal requirements, Q-value is able to converge with the ϵ-greedy-policy overtime.

$$Q(s_t, a_t) \leftarrow (1-\alpha) Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{A} Q(s_{t+1}, A) \right] \tag{2.7}$$

Algorithm 1 Q-learning

Initialization: $Q(s, a)$
Creating Q-table with all zeros $Q(s, a) = 0$
For Each episode **do**
  Initialize State S
  For Each step **do**
    Randomly pick an action A or Choose A from Q-table by ϵ-greedy
    Observe reward R, next state S'
    $Q(S, A) \leftarrow (1 - \alpha)Q(S, A) + \alpha\left[R + \gamma \max_A Q(S', A)\right]$
    $S \leftarrow S'$
  end for
end for

Therefore, the sequence of Q-learning can be described by initialling the Q-values (usually starting with all 0 for all state S and action A). From the initial state S, pick an action A with ϵ-greedy policy. Next, observe changes in the environment by computing the expected reward R and next state S. Later, update the new Q-value in eq.(2.7) into the Q-table shown in Algorithm 1 and Fig 2.10.

Q-Table

As previously mentioned, the Q-table is a lookup table that records each state-action pair. In the table, each row represents a state s, and each column is a possible action defined in action set A shown on the left bottom of Fig 2.10. Managing the Q-table's size is a design challenge due to the limitation of the memory resource in the computing hardware. In other words, the size of the Q-table faces a deployment challenge on a resource-limited network. Therefore, the state space S and action space A will need to be carefully evaluated. However, the Q-learning is still limited to high-dimensional spaces [48].

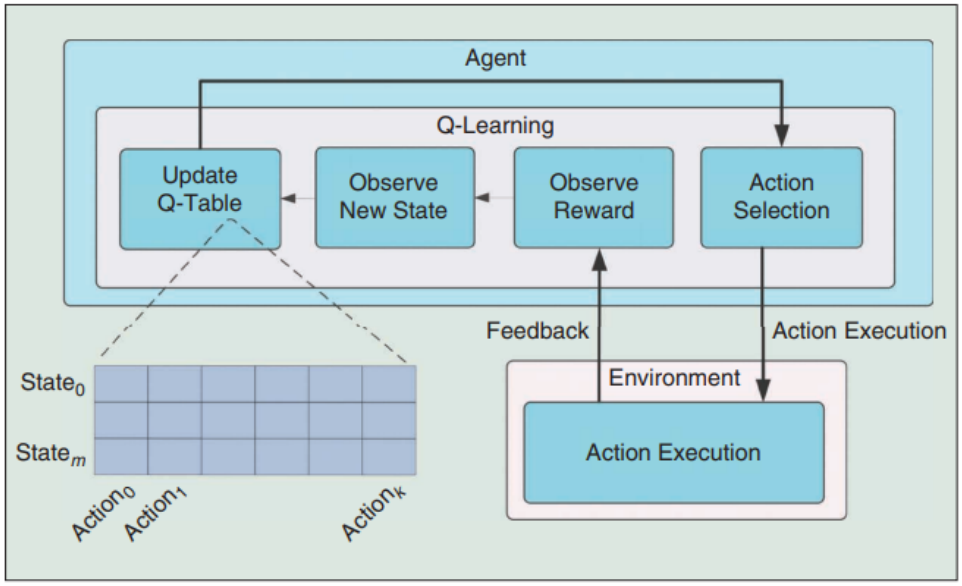

Figure 2.10: Operations of Q-learning [10]

2.3.5 Deep Reinforcement Learning

Last but not least, DRL is an extension of RL and DL by combining two learning techniques into the algorithm. Adding the DL technique into the RL significantly improves learning the high-dimensional spaces problem. There are well-known algorithms such as Deep Q-learning (DQN) and Soft Actor Critic (SAC). DQN [49, 50] is the deep extension of the Q-learning. SAC [51] is the hybrid of Q-learning and policy gradients.

Chapter 3

Literature Review

5G mobile network has allowed enormous amounts of data to stream around the network. Massive data flows are collected and analyzed during transmission. In order to optimize complex 5G networks, new data-driven applications are investing from theorems into fields quickly.

3.1 Overview

This chapter reviews the state-of-art 5G NR studies on resource allocation solutions, focusing on the AI-enabled edge network. The section 3.2 focuses on different approaches on Resource Block (RB) allocation of Orthogonal Frequency-Division Multiplexing (OFDM) in 4G and 5G mobile networks. These studies also reveal the trends in the AI-enabled 5G wireless network identified by Elsayed et al. [10]. The following section 3.3 shifts the focus into the disaggregation in the 5G RAN architecture, especially on C-RAN and then O-RAN. More angles and aspects of resource allocation can be explored by applying the

functional split concept with open-source logical units proposed by the O-RAN alliance. The booming research on the various ML techniques helps understand the relations or the policies in the 5G networks. Proposed models are now considering more than traffic-related KPIs and also OpEx for MNOs.

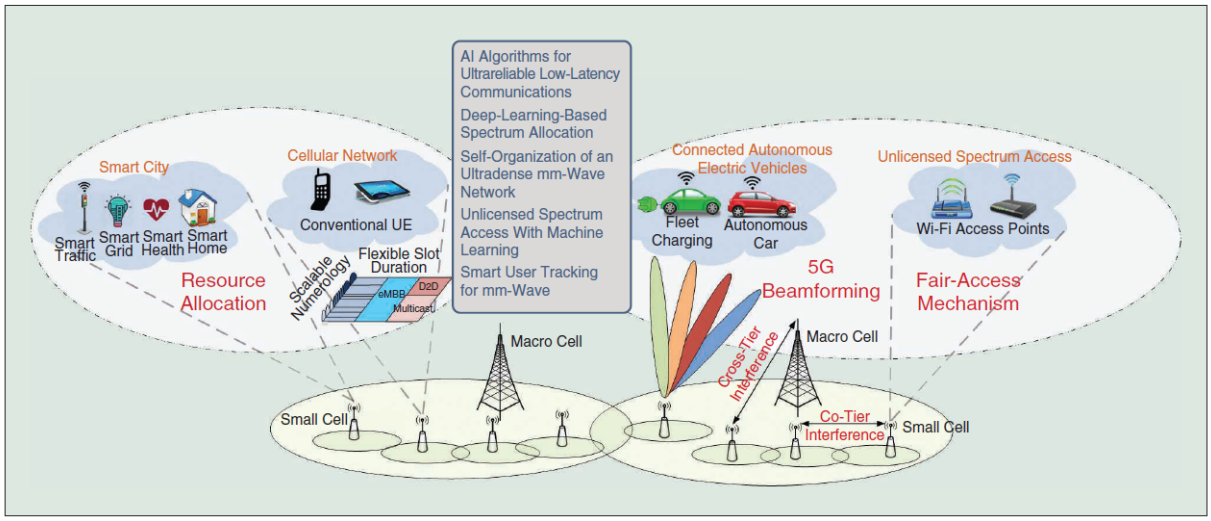

Figure 3.1: Research Areas on the AI-Enabled wireless network [10]

Elsayed et al. [10] have envisioned that the ML, especially RL, could leverage the 5G and Sixth Generation (6G) wireless network. The authors have identified the applicable areas in Fig. 3.1 for using AI techniques to solve the open issues. Resource allocation is one of the main topics in AI-enabled wireless networks. The paper [10] includes but is not limited to the following sub-topics, such as mobile broadband, Device to Device (D2D) communications and tactile internet, unmanned aerial vehicle-assisted Networks, spectrum access in unlicensed bands, energy-efficiency techniques in Fig. 3.1.

3.2 Scheduling Approach

The resource allocation in edge 5G wireless network was initially more focused on the scheduling of RB in OFDM.

3.2.1 Static Approach

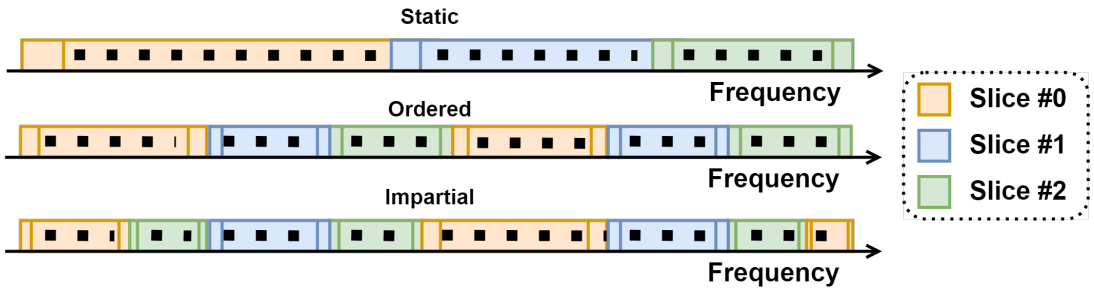

Figure 3.2: RB Allocation Schemes [11]

Nojima et al. [11] propose three different algorithms to guarantee resource allocation of frequency shown in Fig.3.2. Static allocation, allocation to ordered slices, and impartial allocation to slices are the three proposed algorithms. All three algorithms are based on ordinary packet scheduling by limiting maximum allocated RB to each slice with additional criteria. The proposed algorithms result in a high resource utilization environment compared with conventional scheduling.

3.2.2 AI Approach

Elsayed et al. [52] propose a Latency-Reliability-Throughput Q-learning algorithm to better resolve the resource allocation problem on the 5G NR network. The work focuses

on eMBB and URLLC network slices on OFDM RB allocation. The results prove that the advanced multi-agent Q-learning model can perform better than the Priority-based Proportional Fairness model and Latency-Reliability Q-learning model by demonstrating increasing eMBB throughput and degrading URLLC latency.

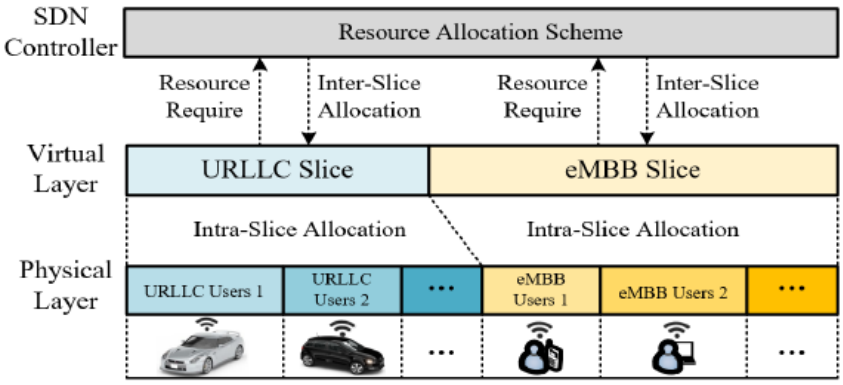

Figure 3.3: Network slicing based two-step resource allocation [12]

Zhou et al. [12] propose a Correlated Q-learning model, COQRA, on allocating inter-slice RB for eMBB and URLLC network slices shown in Fig. 3.3. The result is compared with the above baselines [52]. The performance of COQRA surpasses all KPIs, such as latency, throughput, packet loss, and convergence time, by considering network slicing.

3.3 Disaggregated Approach

After 3GPP introduces the functional-split concept, O-RAN has a practical virtualization solution combining those logical functional units. The resource allocation can be approached from a new perspective by shifting bandwidth allocation from fronthaul and RB on OFDM into other areas. The new areas include different energy sources, deployment options, midhaul interfaces, etc.

3.3.1 Static Approach

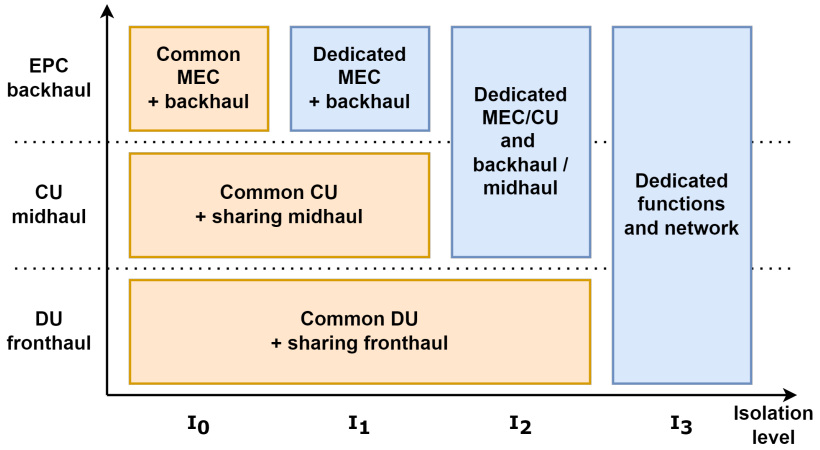

Figure 3.4: Functional splits and isolation levels [13]

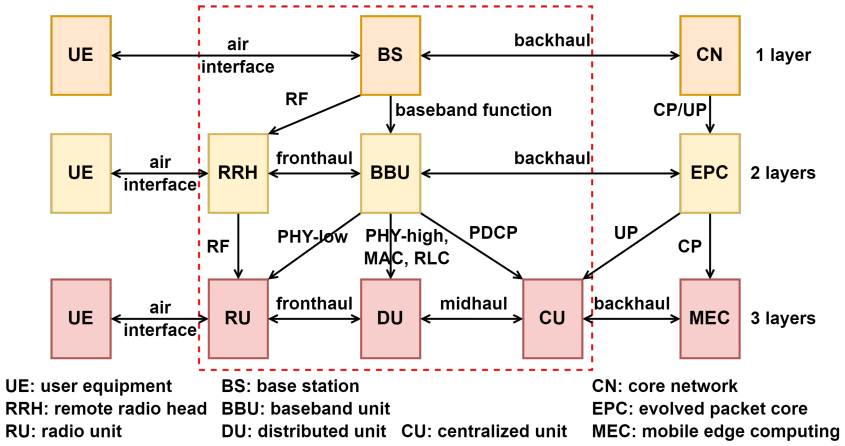

Figure 3.5: Layers in RAN architectures [13]

Yu et al. [13] explain well possible 3-layer functional split combinations in Fig. 3.4 on Wavelength Division Multiplexing (WDM) metro-aggregation networks with an extensive amount of results. The authors propose a RAN slice-aware heuristic algorithm. The results show that the higher slice isolation leads to higher resource cost, meaning more Virtual Machine (VM)s are spawned. The 3-layer RAN architecture is promoted in the research by

comparing it against the 2-layer architecture shown in Fig. 3.5. Their results show that the modern 3-layer architecture outperforms resource consolidation by the flexibility of logical unit placement.

3.3.2 Dynamic Approach

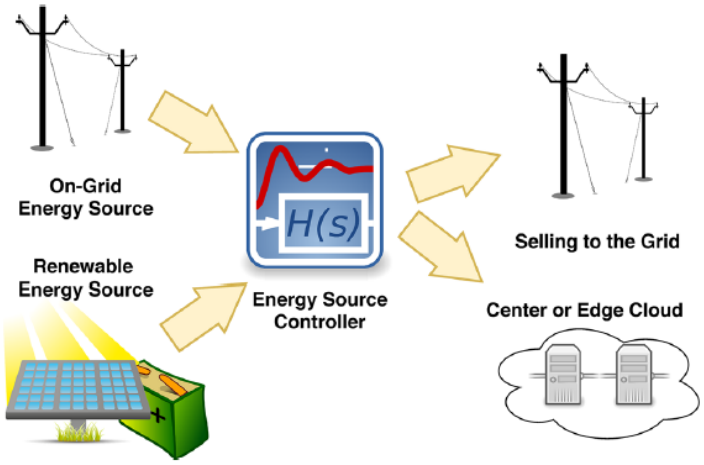

Figure 3.6: Energy Model in Green Hybrid C-RAN [14]

By taking green energy into consideration of C-RAN architecture in Fig. 3.6, Pamuklu et al. [14] provide two solutions for different scales of C-RAN, a Mixed-Integer Linear Programming (MILP) and a heuristic approach. The MILP solver aims to reduce OpEx and performs well on a small-scale RAN. The heuristic method makes the BBU functional split decision by also considering renewable energy. On a large-scale hybrid C-RAN, their fast online heuristic method can provide a more cost-effective solution for MNO in encouraging the usage of green energy.

Focusing on MILP solver, Pamuklu et al. [15] propose a novel model, Green Radio OVer Ethernet (GROVE), to solve the cost and energy problem in the C-RAN architecture.

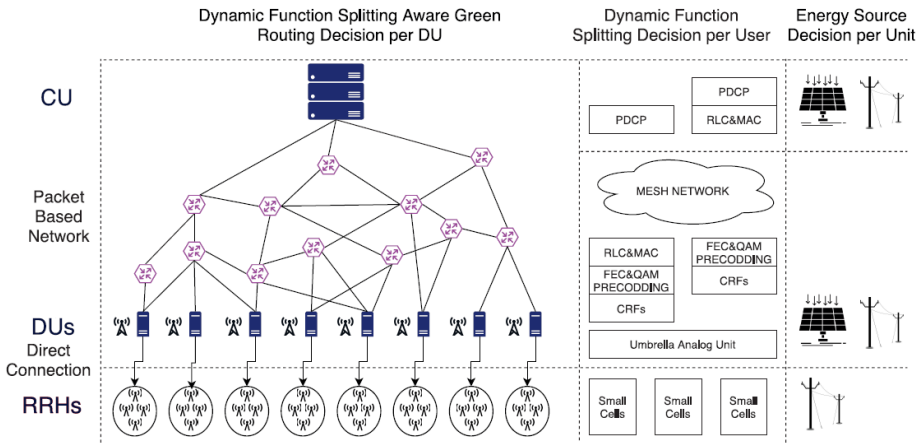

Figure 3.7: Green Radio OVer Ethernet system architecture [15]

GROVE considers the dynamic functional split on CU-DU, radio over Ethernet, and green energy sources to minimize the OpEx shown in Fig. 3.7. With known constrain on the scale of RAN by NP-Hard problem They explore model performance on different numbers of DUs. As the result shows, GROVE can outperform not only static routing but also traffic-aware models in optimizing the usage of profitable renewable energy.

3.3.3 AI Approach

In [53], Sun et al. propose a DQN learning-based algorithm to solve the bandwidth resource allocation on C-RAN. In their simulation environment, they considered a single CU-DU fronthaul-II connection based on three network slices, eMBB, URLLC, and mMTC, by sending static packet size. The results are compared to two baselines, round-robin and non-slicing allocation. The overall performance is better than baselines on QoS, packet loss, and bandwidth utilization.

Joda et al. [16] use DQN to find the cost optimization of CU-DU placement in O-RAN deployment shown in Fig. 3.8. The proposed algorithm also considers and improves the

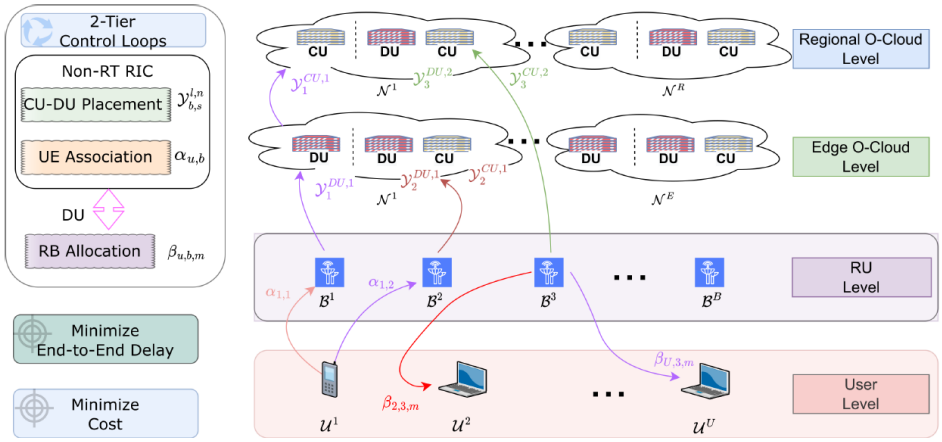

Figure 3.8: Joint User Association and CU-DU Placement in O-RAN [16]

UE delay. The authors show that the proposed model has outperformed the fixed CU-DU baselines on the edge and regional O-Clouds by reducing costs and delays.

Pamuklu et al. [54] propose a RL model on dynamic functional splitting O-RAN architecture by considering green energy. The authors promise the RL model can better manage renewable energy sources and operational costs. The model is carefully evaluated by real solar irradiation and traffic data set. Two RL algorithms, Q-learning and State–action–reward–state–action (SARSA) are implemented and compared against D-RAN and C-RAN as baselines. The result shows that the proposed model is a good cost-efficient solution for MNO. In addition, the research discusses the different scales of renewable energy sources used in the environment, such as the size of solar panels and batteries.

Chapter 4

System Model and Methodology

4.1 Problem definition

Research on the integration of functional split in O-RAN and network slices has opened a new door to optimize the network resources and provide better QoS to end users. This thesis aims to tackle the optimization of bandwidth allocation in the O-RAN cloud network by considering the main requirement differences between 5G network slices, eMBB and URLLC and regular voice slices.

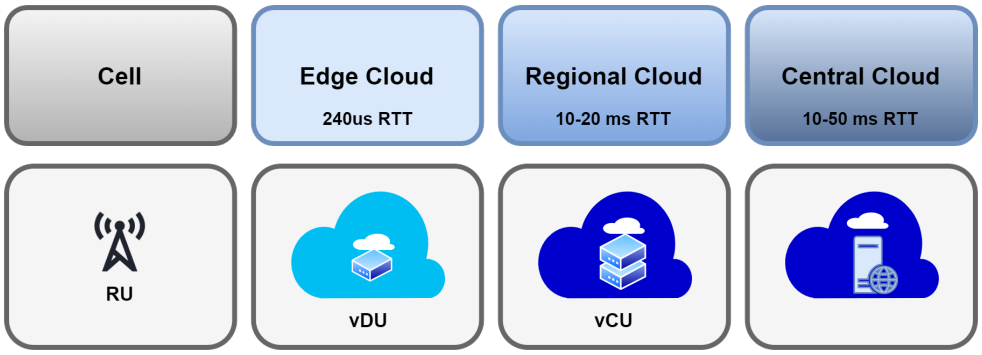

Figure 4.1: Functional split in O-RAN Cloud [17]

Prior works on resource optimization of bandwidth sharing [53, 13, 55] are based on single Edge O-Cloud (Single DU-CU) but without UE mobility. However, the 5G RU coverage is known for significantly less than the 4G mobile network. The 5G cell tower coverage is reduced to less than 300 meters for the Millimeter Wave (mmW) spectrum [56], which is the key to providing ultra-fast connections promised by 5G. In order to provide a near-realistic optimization on 5G O-RAN, we propose a RL solution considering UE mobility in multiple Edge O-Clouds with a single Regional O-Cloud topology [24]. We believe implanting the RL algorithms on non-RT controllers can improve the midhaul bandwidth optimization shared by different network slices.

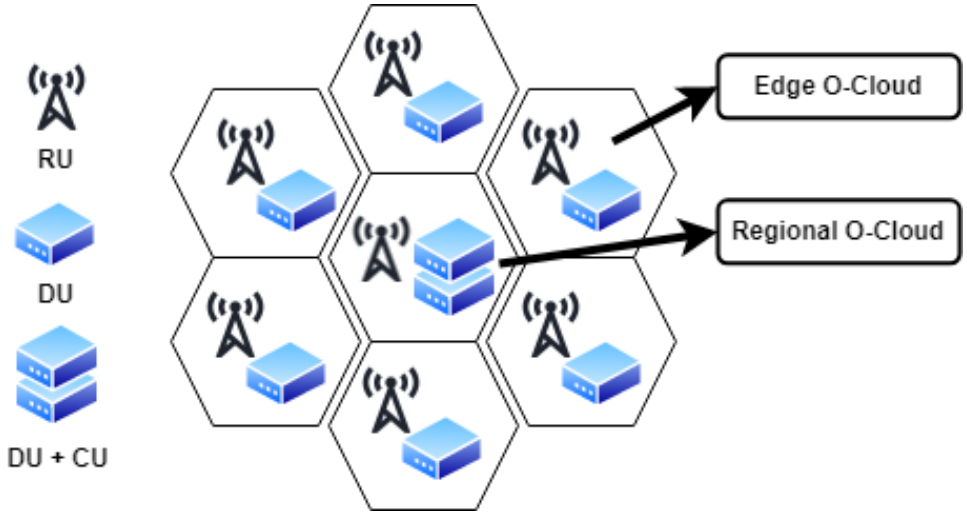

Figure 4.2: Single CU Multi DU Setup Topology.

4.2 System Model

The edge network simulator is composed of the following components, UEs, Edge O-Clouds, and a Regional O-Cloud (Fig 4.1). The functional split is implemented by placing one DU in each Edge O-Cloud with one cell site of one RU, where CU stays in the Regional O-

Cloud.

This research aims to improve the bandwidth allocation in a multi-cloud scenario. Therefore, the UE movement can be considered with more realistic moving speeds and ranges based on 5G specification[1]. The single Regional Cloud (single CU) and multi Edge Clouds (multi DUs) are designed to form a honeycomb, as illustrated in Fig 4.2, with the Regional Cloud placed in the center.

4.3 Proposed Solution

4.3.1 Q-Learning

In order to prove that RL can be the solution to the defined problem, Q-learning is selected as the starting point of the RL strategy because of its model-free and off-policy concept. The model can find the optimal action to take by looking up the Q-table for a maximized Q-value from the finite MDP. The states of UEs and DUs can be easily translated into the state space of the shared Q-table. Table 4.1 defines the notations of repeatedly referring across the entire thesis, including UE i, DU j, and more.

Table 4.1: Space Notations

Sets	**Size**	**Description**
$i \in \mathcal{I}$	I	UEs
$j \in \mathcal{J}$	J	Edge O-Clouds (DUs)
$a \in \mathcal{A}$	A	bandwidth change range
$k \in \mathcal{K}$	K	bandwidth slices
$t \in \mathcal{T}$	T	time slots
$v \in \mathcal{V}$	V	UE moving rate

Action

The action $a_{j,k}^{t}$ per second in our Q-learning model represents the changes of bandwidth $\Delta b_{j,k}$ for the network slice k at time t on DU j in Eq. (4.1). The sum of bandwidth changes from all network slices has to be zero as Eq. (4.2). It is required for the action space because the bandwidth is fixed on one physical midhaul interface and shared by all network slices. This condition furthermore helps to manage the size of the Q-table by providing the chance for order reduction into $K-1$ slices. We limit the difference within a specific range $\mathcal{A}$ to manage the number of columns further in the Q-table. In other words, the controller takes a small step action a by adding or removing a limited percentage of bandwidth from a network slice k defined in Table 4.2. This technique could also prevent the Q-table from having massive unvisited state-action pairs. The range of action decides the learning curve of the model. When the model changes less than ten percent, it will take a long time to converge. Else, when the model uses more aggressive changes, the bandwidth will all be assigned to the target slice too fast and does not learn other possibilities. Therefore, changing bandwidth by plus or minus ten percent per action helps explore possibilities and keeps the Q-table in a reasonable size.

$$A(t) = \{\{a_{j,k}^{t} \in \mathcal{A} | j \in \mathcal{J}, k \in \mathcal{K}^{-}\} \tag{4.1}$$

$$\sum_{k=0}^{K} a_{j,k}^{t} = 0, \quad \forall j \in \mathcal{J}, \forall t \in \mathcal{T} \tag{4.2}$$

Table 4.2: Range Notations

Variables	Domain	Description
$u_{j,k}^t$	$\mathcal{K}$	number of UEs use slice k in DU j
$b_{j,k}^t$	$[0, 100]\%$	allocated bandwidth for slice k of DU j
$a_{j,k}^t$	$[-10, 10]\%$	allocated bandwidth percentage change for slice k of DU j

State

$$S_j(t) = \{\{b_{j,k}^{(t-1)} | k \in \mathcal{K}\}, \{u_{j,k}^{(t-1)} | k \in \mathcal{K}\}\} \tag{4.3}$$

The state S_j of each Edge O-Cloud j is composed of two tuples in our proposed scheme, bandwidth $b_{j,k}$ in percentage and number of connected UEs $u_{j,k}$ from each slice k in the Edge O-Cloud j. Because of the state-action sequence defined in MDP, the state $S_j(t)$ at time t is the result of the previous action $A(t-1)$ taken at time $t-1$.

Reward

The reward in our design is composed of two portions, negative and positive. Hence, they can be calculated and weighted independently.

$$N_j(t) = -\sum_{k \in \mathcal{K}} \begin{cases} 1 & \text{if } b_{j,k}^t < L \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

$$P_j(t) = \sum_{k \in \mathcal{K}} (s_{1,k} \Delta \mathcal{F}_{j,k}^t + s_{2,k} \mathcal{B}_{j,k}^t + s_{3,k} \mathcal{D}_{j,k}^t) \tag{4.5}$$

$$R_j(t) = w_p P_j(t) + w_n N_j(t) \tag{4.6}$$

The negative part Eq.(4.4) reserves a minimal amount of bandwidth L for each network slice k. Therefore, the shared midhual interface will have a small portion of bandwidth allocated for each slice to handle the newly incoming requests at time $t+1$. This could help avoid the allocation of all bandwidth for the more favorite or high demands network slice(s) at time t, while the least favorite network slice has no bandwidth allocated by the non-RT RIC. In other words, a UE joining on a less favorite network slice won't be stuck in the queue forever due to all the bandwidth being occupied by the more favorite network slice(s).

The positive reward Eq.(4.5) is calculated by three components shown in Table 4.3. The intent of the reward is to encourage the bandwidth allocation to favorite a targeted slice. $\mathcal{D}_{i,k}^{t}$ calculates the average UE data rate using slice k at time t. $\mathcal{B}_{j,k}^{t}$ is the throughput of a network slice k used on Edge O-Cloud j. Last, bandwidth utilization is assessed for a network slice k on $\mathcal{F}_{j,k}^{t}$. The scale factor s is implemented to balance each feature in our reward function into the same scope of scale. Consequently, these three components are equally important, and one at a smaller scale will not be diluted or ignored.

The total reward function Eq.(4.6) is composed of the positive portion Eq.(4.5) minus the negative part Eq.(4.4). The weight factor w is also considered and assigned to each reward portion. Therefore, we can control which factor is more important than the other later.

Shareable Q-table

Q-table is a lookup chart for the agent to store the Q-value. Those Q-values are calculated from the reward in the exploring phase. After the Q-table is filled, the agent can search for the optimal action to take based on the mapped Q-value from the state in the testing

Table 4.3: Reward Notations

Runtime Data	Domain	Description
$\mathcal{D}_{i,k}^{t}$	$\mathbb{N}^{+}$	The average data rate of slice k of UE i
$\mathcal{B}_{j,k}^{t}$	$\mathbb{N}^{+}$	Throughput of slice k of DU j
$\mathcal{F}_{j,k}^{t}$	$\mathbb{N}^{+}$	Free bandwidth of slice k of DU j

or the actual practice. However, there are some known drawbacks to having or using a Q-table. For instance, the Q-table demands memory storage heavily; otherwise, it runs out of bounds quickly. Besides, if the learning rate is aggressive, it is possible that the table is empty in most parts. In other words, the table might have most portions unvisited. Therefore, several methods of reduction of the order are applied to keep the Q-table size reasonable and functional in our proposed solution.

First, we propose sharing a Q-table between Edge O-Clouds. The Q-table can be defined with common states and actions for the DUs from different Edge O-Clouds to update and look up accordingly. The more powerful centralized Regional O-Cloud stores and manages the Q-table. This assumption can be made because similar network environment settings are applied to Edge O-Clouds within the Regional O-Cloud.

$$B_j = \{\sum b_{j,k} | k \in \mathcal{K}^{-}\} \tag{4.7}$$

$$b_{j,k} = B_j - \{\sum_{k=1}^{k-1} b_{j,k} | k \in \mathcal{K}^{-}\} \tag{4.8}$$

The second method is reducing order in the state dimensions on network slices. Firstly, three considered network slices, eMBB, URLLC, and voice, share the same transmission

midhaul interface on an Edge O-Cloud j in Eq.(4.7). Thus, the bandwidth of the last slice can be calculated easily in Eq.(4.8). Secondly, the dimension of the user count can be reduced. The technique of using ranking or the step function is introduced by counting tens or hundreds of users per step based on the user scale on each network slice.

Last, the bandwidth in both state and action can be represented in percentage (%) to address the different bandwidth scales used between the Edge O-Clouds. For example, a smaller station might have 100 Gigabytes of bandwidth in total to share, while the other station has 10 Terabytes of bandwidth.

4.4 System Implementation

With levels of customization provided by Python packages, we are able to implement the main components and events of the disaggregated Edge O-RAN architecture with given speeds of various UE movements.

4.4.1 Simulation Platform

This thesis implements a Python-based simulator to mimic the O-RAN environment, including UE mobility in Fig 5.1. A Python-based RL algorithm interacts directly with this simulation environment. To achieve the integration between the simulator and the Q-learning model, the following open-source Python packages are selected.

SimPy

The simulation environment is based on a Python framework, SimPy[57]. The framework is designed for handling **process-based discrete-event**, which is ideal for handling incoming UE requests sequentially on Edge O-Clouds. The event following the time sequence is implemented with a Python generator, which works well on recording events in a time manner. In addition, several shared resources are provided for implementing the queuing system and bandwidth sharing. **Priority Resource** is selected for queuing UE connection requests by considering different priorities for each network slice and releasing them when the transmission finishes. For example, an UE requesting a URLLC connection can connect to the DU before other request types. **Container**, another shared resource type, is selected for managing continuously shared bandwidth, where in-use midhaul bandwidth and users can be monitored. The framework also provides monitoring and step-in functions to inject action or read the status of an UE or an O-Cloud at any time. The RIC later heavily uses the monitoring function and acts as the agent in Q-learning to observe the state and reward from the environment.

OpenAI

OpenAI [58] creates the MDP environment where the agent takes action based on the states of the environment, and the environment feeds the new states and the reward back to the agent. In order to customize Q-learning parameters, the **Gym**[59] package from the OpenAI is selected as it is also written in Python. The GYM package provides the customization from initial states, actions, and rewards to recreate our own MDP Q-learning model.

Expendable Model

SimPy, Gym, and OpenAI packages can stack and interact well with each other as they all follow Object-oriented Programming (OOP) practice written in Python. SimPy creates the network environment. Gym defines states, actions, and rewards at each timestamp. OpenAI provides RL exercise environment and also the baselines of other RL algorithms, which could be integrated in the future. As stated above, the network environment is running in a group of 7 routers by forming a hex-formed network. Expanding the size of the network is possible because those network variables are the collection of input to create an object of the logical unit in our simulation.

4.4.2 MDP Implementation

As described in Algorithm 2, the Q learning model is trained for N episodes, and each episode contains T seconds. On initialization, each network slice k of each DU[1] j starts with $\beta_{j,k}$ percent of the bandwidth. Each UE i is assigned to request a fixed data amount y_i from a network slice k, and a movement type defined in Table 5.2. After the simulation begins, several events happen simultaneously on each second (t).

State

Each UE i moves v_i meters every second based on its movement type. The UE i transmits at d_i GB per second assigned by the connected Edge O-Cloud j nearby[2]. The summation of connected UEs' data rate $\sum d_i$ of the DU j represents the throughput $\mathcal{B}_j$. The number

[1] One DU is located in the center of an Edge O-Cloud

[2] Details of UE mobility and data rate will be discussed in section 5.1.2

Algorithm 2 Q-learning Based Solution

Initialization:
Each DU j starts with bandwidth $b_{j,k}$ for each k slice
Each UE i is randomly assigned with moving speed v_i m/s and network slice k
Each UE i connects to nearby DU j and requests y_i GB data
For Each environment step at time t **do**
    UE (i) moves at v_i m/s, and receives d_i G/s
    Controller observes state s_t
    x $\leftarrow$ uniform random number between 0 and 1
    if $x \geq \epsilon$ **then**
        Select $A_j^t = \left[a_{k1,j}^t, a_{k2,j}^t ...\right] = \arg\max_a Q_(s_t, a_t)$ for each DU j
    else
        Select random action A_j^t for each DU j
    end if
    Execute A_j^t for each DU j
    Observe the next state s_{t+1} and reward r_{t+1}
    Update Value Action: $Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t)$
    $+\alpha\left[r_{t+1} + \gamma \max_A Q(s_{t+1}, A)\right]$
end for

of connected UEs is recorded per network slice k per DU j. The state $S_j(t)$ is composed of the throughput and the user counts of the DU j.

Action

On the Regional O-Cloud, the Non-RT RIC will check the incoming midhaul traffic request for each DU j as a state $S_j(t)$. The RIC has two options of actions to take for exploration and exploitation.

Table 4.4: Notation of Simulation Parameters

Given Data	Domain	Description
v_i	$\mathbb{N}^+$	movement speed of UE i
β_k^I	$\mathbb{N}^+$	initial bandwidth for slice k
β_k^M	$\mathbb{N}^+$	maximum bandwidth for slice k
$l_{j,k}$	$\mathbb{N}^+$	floor value of bandwidth threshold for slice k
$u_{j,k}^t$	$\mathbb{N}^+$	number of UEs for slice k of DU j

- **Random action** is mainly for exploring other possibilities of reward based on the current state.
- **Greedy action** is based on the epsilon-greedy approach (ϵ-greedy). The controller looks up from the Q-table to find the best action for the Edge O-Cloud. The action is chosen based on the maximized Q-value from the historical actions recorded in the Q-table.

In the early phase of the training, the random action is selected most of the time to fill up the Q-table. The controller leans toward greedy action later when most scenarios are learnt.

Reward

The action $a_{j,k}$ impacts the Edge O-Cloud environment by increasing or decreasing the bandwidth $b_{j,k}$ of a network slice k. After the action is taken and the reward $R_j(t)$ is received, the Regional O-Cloud calculates the new Q-value using the Bellman Optimality

Equation (Eq. 2.7) and updates the Q-table accordingly.

Chapter 5

Results

5.1 Simulation Setting

In order to better explain the result later and prove the model is an effective solution, this section demonstrates the requirement and numbers used in the experiment.

5.1.1 O-Cloud Implementation

An Edge O-Cloud has K network slices. Each slice is assigned to $b_{j,k}$ of bandwidth in percentage. The implemented network slices are eMBB, URLLC and Voice.

The topology is composed of a 7-cell honeycomb hex with six Edge O-Clouds in a small hex shape around a Regional O-Cloud in the center with Non-RT RIC shown in Fig. 5.1. The Edge O-Cloud is composed of a RU-DU pair. The Regional O-Cloud adds a CU along with the RU-DU pair. Therefore, the topology has a total of 7 RU, 7 DU, and 1 CU as a centralized network topology.

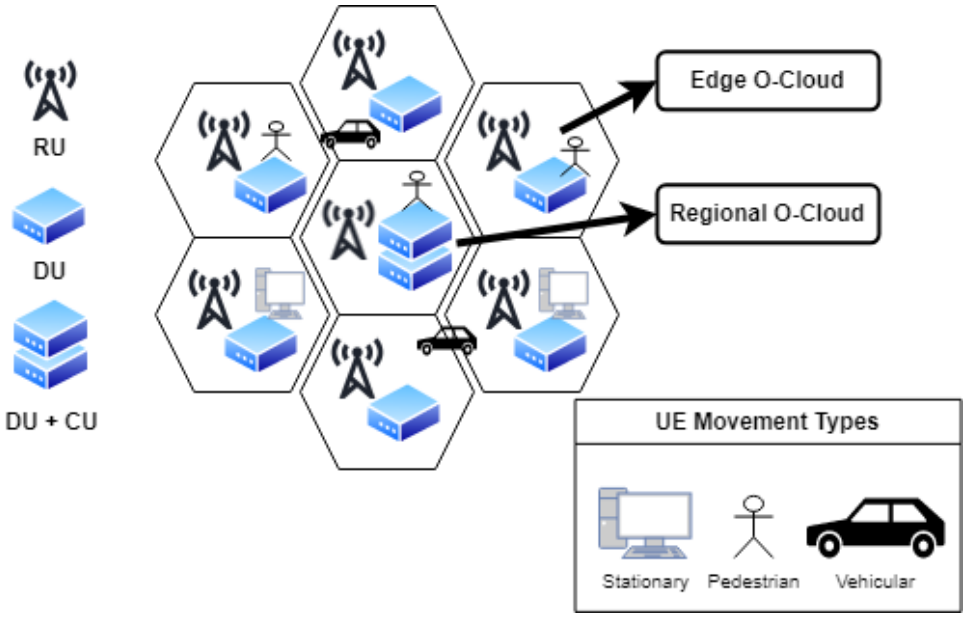

Figure 5.1: Joint O-Cloud and UE Setup

Different network slices have specific traffic requirements. Table 5.1 sets the initial traffic rate of a new UE when the UE connects to the nearby Edge O-Cloud based on its labelled network slice. The bottom part of Table 5.1 defines the max rate an UE can receive from the Edge O-Cloud. The eMBB slice targets a high peak rate, the initial rate of an eMBB user is higher than other types of users, and also the max rate is not limited; while other slices are not peak rate demanding, the initial rates are lower. Same for the total data per request, the eMBB users request more data than others. In general, the model will increase the transmission rate per UE by one GB per second if there is available bandwidth to assign.

5.1.2 UE Implementation

Poisson distribution is used on the time domain to create a new UE event to introduce randomness into the RL algorithm. An UE will be randomly assigned to a network slice

Table 5.1: Rates of Network Slices

Variables	**Unit**
eMBB Initial Rate	1 GB/s
URLLC Initial Rate	100 MB/s
Voice Initial Rate	100 MB/s
eMBB Max Rate	N/A
URLLC Max Rat	5 GB/s
Voice Max Rate	1 GB/s

and a type of movement independently. During the simulation, an UE moves freely within the ORAN network and requests data of a given network slice from a nearby Edge O-Cloud. The nearby Edge O-Cloud is selected for establishing the connection based on its distance rank and connected user count.

UE mobility

The UEs are initially randomly distributed with Gaussian distribution as their initial locations. UEs will go to their new places every second based on their given speed defined in Table 5.2.

The following moving speeds for each UE type in Table 5.2 are given by [1]. The unit presented in the requirement [1] is in kilometer per hour. During the implementation, the moving rate is converted into meter per second. The convention is for better alignment to the time scale used in action and reward, which are all taken or calculated every second.

Table 5.2: UE Mobility Types and Speed

UE Types	**Speed (km/hr)**
Stationary	0
Pedestrian	0-10
Vehicular	10-120

UE data rate

While UE moves in the O-Clouds, the UE data rate changes not only by the assigned bandwidth but also by its distance against the center of the connected Edge O-Cloud where the RU is located. The received data rate and distance relationship is impacted by the Radio Frequency (RF) Power Density (5.1) in [60] and the Shannon-Hartley Theorem (5.2) in [61].

$$S = \frac{P * G}{4 * \pi * R^2} \tag{5.1}$$

$$C = Blog_2(1 + \frac{S}{N}) \tag{5.2}$$

The received signal power S is calculated by the transmitter power P; the power gain G over the distance R^2. The power density is in a three-dimensional sphere shape ($4\pi R^2$). The channel capacity C in (5.2) is the theoretical UE data rate. B is the channel's bandwidth in hertz. S/N is the signal-to-noise Ratio (SNR). S is the received signal power, and N is noise power. Therefore, the data rate d_i of the UE i based on its distance R to the connected O-Cloud can be easily calculated. In other words, UE data rate degrades when the UE moves away from the connected Edge O-Cloud. The consideration of this data rate degradation can be further extended to the handover problem between O-Clouds in the future.

Additionally, A greedy approach is taken to static bandwidth allocation per UE to increase the model's performance. In other words, if an Edge O-Cloud has more free bandwidth allocated to a slice k, the controller will increase the UE data rate $v_{i,k}$ by 1 Gigabyte every second until the UE i finishes the transmission of y_i GB from initial requested data.

Table 5.3: Q-learning Settings

Parameters	
Learning Rate (α)	0.1
Discount Factor (γ)	0.9
Epsilon Decay Factor	0.01
Min Epsilon ($\epsilon_m in$)	0.01

5.1.3 Q-Learning Implementation

$$\varepsilon = max(\varepsilon_{min}, e^{-(decay * episode)}) \tag{5.3}$$

$$s.t.\ \ \varepsilon_{min} < \varepsilon < 1 \tag{5.4}$$

The Q-learning settings are shown in Table 5.3. The model uses a slow learning rate (α) of 0.1 and a typical discount factor (γ) of 0.9 for the Q-learning equation (2.7). A dynamic ϵ-greedy approach is implemented in Eq.(5.3) for calculating the exploration probability (ϵ). The larger value between a given minimum ϵ or ϵ value of the episode is used. The ϵ value per episode is calculated by taking the exponential value of the negative decay factor multiplied by the i episode. Therefore, the used ϵ value will fall between the range of 1 and ϵ_{min} (Eq.(5.4)). This dynamic exploration method will start the training with more

explorations and end with more exploitation.

5.1.4 Testsets

Table 5.4: Testsets

Testset	**Mid**	**High**
eMBB UEs	[100,200,300,400,500]	
URLLC UEs	500	1000
Voice UEs	1000	500

In order to show that the Q-learning model can utilize the bandwidth in various scenarios, the counts of UEs increases in the network between different experiment sets shown in Table 5.4. The increase of UE can show the model performance from idle to busy network activities. Considering experimental and controlled groups in mind, the UEs of the targeted network slice, eMBB, increase by 100 UEs from 100 UEs to 500 UEs between different experiment sets. The controlled network slices, URLLC, and voice slice have a fixed number of UEs in the network per scenario. The UE profile with 500 URLLC UEs and 1000 voice UEs is named the **Mid (URLLC) traffic** profile. Since the Voice network slice has much fewer traffic demands than the other slices, switching the UE counts between the two controlled network slices, URLLC and Voice, can significantly increase the traffic stress in the network. The UE profile of the High-Traffic network is named the **High (URLLC) traffic** profile. In order to evaluate the model and the performance on different traffic stress levels, the research conducts a simulation run of 1000 episodes on all five sets of eMBB UEs for both Mid and High profiles.

5.1.5 Baselines

Compared to our Q-learning model, two sets of bandwidth allocation baselines are created in Table 5.5. The **Balanced** profile allocates the bandwidth fairly between network slices and assigns ten percent more bandwidth to the eMBB slice because eMBB is known for more traffic demanding. The other baseline, **eMBB-Focus**, gives 90% bandwidth to the eMBB slice, and the other two network slices share the rest of 10% bandwidth equally to meet the minimal requirement.

Table 5.5: Baseline Profiles

Bandwidth (%)	**eMBB**	**URLLC**	**Voice**
Balanced[1]	40	30	30
eMBB-Focus[2]	90	5	5

5.2 Performance Evaluation

Fig 5.2 shows the learning curves of different eMBB scenarios. The learning curve is plotted by accumulating the rewards over time for 1000 episodes. The cumulative reward is high because a single reward falls on the scale between [1-10]. During the training phase, the cumulative reward is recorded after each iteration and takes an average of 50 iterations per data point. Five learning curves are plotted to represent each eMBB user's scenario. The optimal Q-value is found around 400 episodes shown in Fig 5.2, where the learning curves are converged. The convergence of cumulative reward proves that the model can learn from the simulated O-RAN environment. Therefore, the model is ready to proceed

[1]Balanced baseline is also referred as base40 or eMBB40
[2]eMBB-Focus baseline is also referred as base90 or eMBB90

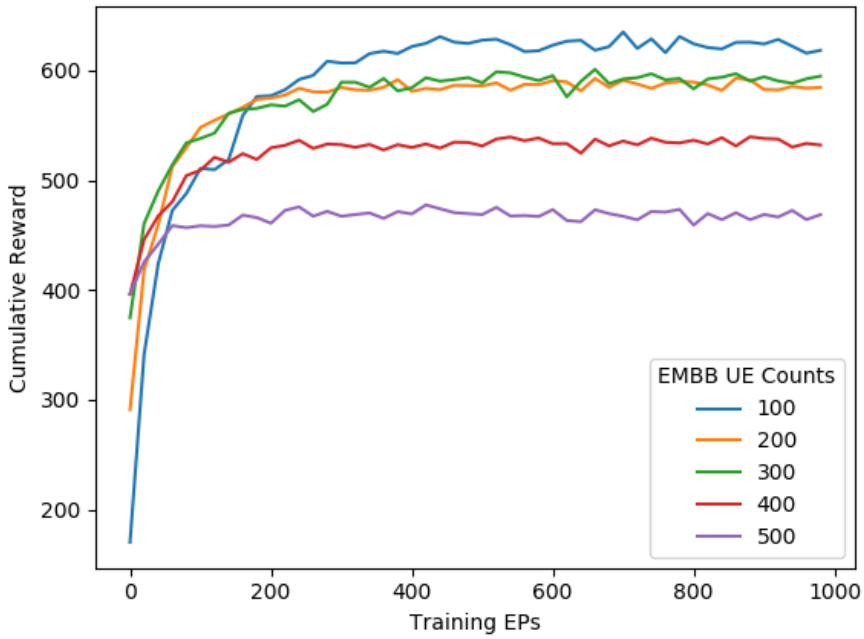

Figure 5.2: Reward convergence during training process

to the test phase. The test phase compares the Q-learning result against the baselines of the same eMBB user counts in the same scenario. As mentioned in Table 5.5, the baselines have static bandwidth allocated to each network slice. The simulation duration is set to 120 seconds for collecting results from baselines and the Q-learning model.

In order to evaluate if the proposed Q-learning model is able to adjust to the changing traffic flows in the simulation network environment, Fig 5.3 is a snapshot from the running simulation generated from one of the edge O-Clouds. The figure shows the allocated bandwidth between different slices, the actual bandwidth usage or demands of each slice, and the number of users per slice. The stats of eMBB slice is in blue on the left column. The URLLC is in yellow in the middle column. The voice slice is in green on the right column. The first row shows the bandwidth capacity allocated to each network slice. The

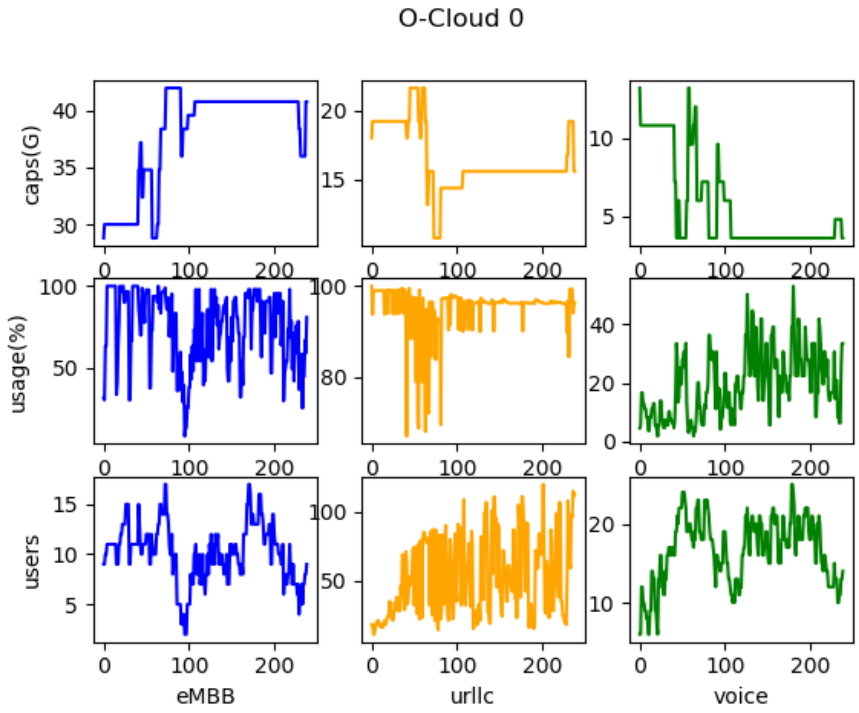

Figure 5.3: Bandwidth Allocation and O-Cloud States

eMBB slice gets more bandwidth allocated over time, while the URLLC and the voice slices receive less bandwidth over time. The second row of the figure shows the percentage of usage based on the allocated bandwidth. The bandwidth increase doesn't reduce the eMBB usage percentage much, meaning throughput increases for each UE. The third row is the number of users connected to the O-Cloud for each slice, which reflects the traffic demands of each slice. The usage percentage increases or decreases accordingly to the change in user counts.

After the evaluation of the convergence of the model and the dynamic changes captured in the Edge O-clouds states, we can claim the model is working as expected and ready to proceed to the next stage.

5.3 Results and Analysis

In the following section, we will discuss the results collected after the simulations in both Mid-Traffic and High-Traffic scenarios and compare our scheme with two baselines. The analysis also covers discussion on both perspectives of Edge O-Cloud performance and the UE QoS.

5.3.1 Midhaul Bandwidth Usage

Mid-Traffic Result

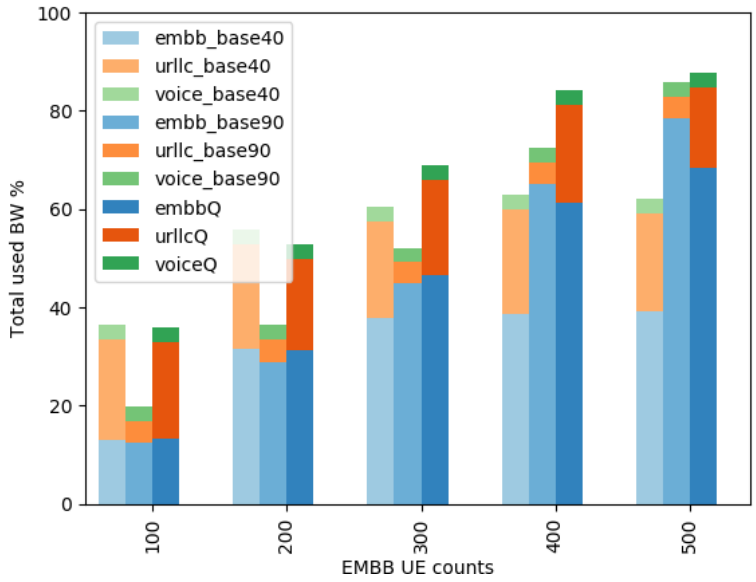

Figure 5.4: Bandwidth Usage in Mid-Traffic Profile

In the Mid-Traffic scenario, the controlled parameters are 1000 UEs using the Voice slice and 500 URLLC UEs set in Table 5.4. None of these two network slices is expected to

[1]base40 is Balanced baseline, and base90 is eMBB-Focus baseline.

consume more bandwidth than eMBB slice. In the following figures, we have a standard colour schema for each network slice. eMBB is in blue, URLLC uses orange, and the voice slice shows in green. In bandwidth usage Fig 5.4, the results are grouped by the number of eMBB users from left to right. In each group, the 3 bars from left to right represent the balanced baseline result, the eMBB Focus result, and the Q-learning result.

The **Balanced** baseline shows that the static allocation limits the eMBB bandwidth to the given value while a huge chunk of bandwidth in other slices is unused. Fig. 5.4 illustrates that the balanced baseline provides sufficient bandwidth for users of 100, 200, and 300 eMBB in the Mid-Traffic profile. However, this static balanced bandwidth approach is inadequate for higher users of eMBB. eMBB throughput has already reached the limit of 40% bandwidth and cannot increase more.

After seeing eMBB slice is suffered from the limitation of static bandwidth in a balanced baseline, the **eMBB-Focus** baseline improves and assigns 90% of the bandwidth to eMBB since eMBB requests are expected to be bandwidth eager in the network. As the result of this baseline scenario in Fig. 5.4, the eMBB slice receives enough bandwidth while users increase in eMBB slices. However, the trade-off is that the other network slices are not able to request more bandwidth from the network, although the eMBB slice has unused bandwidth in less traffic stress scenarios on the left side of Fig. 5.4.

On the other hand, the **Q-learning model** can dynamically assign extra unused portions from other network slices to the eMBB slice while the eMBB request increases from 100 users to 500 users. At the same time, the other slices can receive sufficient bandwidth allocated while unused bandwidth is available in the network. On the max 500 eMBB users scenario, the Q-learning model is able to provide fair bandwidth distribution to each network slice by sacrificing slightly eMBB bandwidth and sharing it with other

slices compared to eMBB-Focus baseline. Therefore, we can see that the eMBB slice has a slightly lower percentage of allocated bandwidth on the right (Q-learning) bar than the middle (eMBB-Focus) bar on the rightmost group in Fig. 5.4. Contrarily, the URLLC slice receives triple the bandwidth from the Q-learning model over eMBB-Focus baseline.

High-Traffic Result

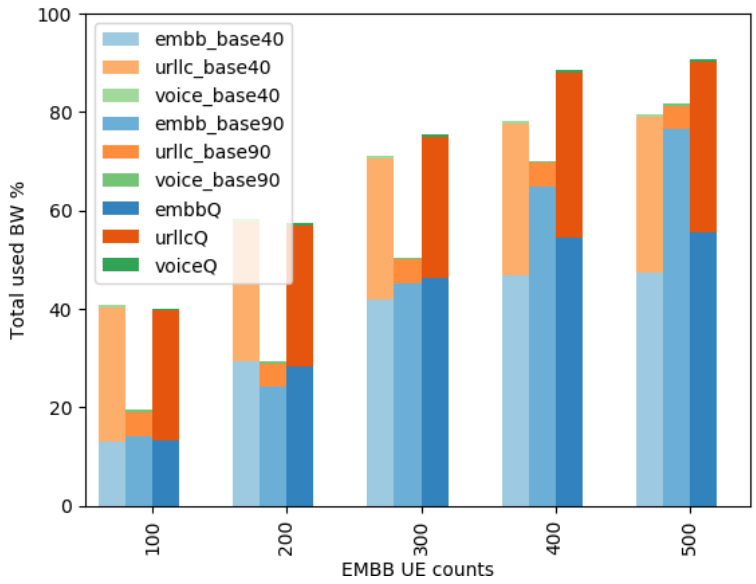

Figure 5.5: Bandwidth Usage in High Traffic

Compared to the Mid-Traffic profile, the High-Traffic profile doubles URLLC UEs and reduces voice UEs described in Table 5.4. When the eMBB UEs are low in counts, the balanced baseline, base40, is able to manage the increase of the URLLC demands as well as the Q-learning model. However, the eMBB-Focus baseline, base90, provides a saturated bandwidth allocation for URLLC slice. As expected, the 5% bandwidth allocated to URLLC is fully in use by higher URLLC users in the high-traffic profile with only 100 eMBB UEs shown on the left side of Fig. 5.5. Moving toward the right side of Fig. 5.5,

the balanced baseline is inadequate to provide sufficient bandwidth for eMBB slice. The eMBB-Focus baseline never makes more effort to share bandwidth to URLLC slice. The Q-learning model attempts to find the fairness of bandwidth sharing between the URLLC and eMBB slices by sacrificing some more eMBB bandwidth than the Mid-Traffic profile.

Overall, the results from the High-Traffic profile highlight the known differences between the two baselines and the Q-learning model even more. The Q-learning model tends to give weight to the fairness factor while maintaining high bandwidth allocation on the target eMBB slice. This fairness behavior is suspected of being encouraged by Eq.(4.4), discouraging bandwidth allocation from being lower than the minimum threshold. However, the weight factor of negative reward is left as-is in Eq.(4.6) in the Q-learning model. In the later section, the fairness of Q-learning is going to be evaluated with more discussion on whether other performance factors are impacted or not. In other words, the sacrifice of eMBB bandwidth should be rewarded or adjusted by the weight factor. Generally, the unused bandwidth is well utilized with the Q-learning model by 10% or more, and the target slice and other slices all receive optimal throughput.

5.3.2 QoS Result and Analysis

After careful examination of the utilization on the Edge O-Cloud side, the evaluation of the QoS and QoE is provided from the client side. UE data rate and transmission time (time to finish) on static data requests are two good indicators to represent the UE QoS and QoE provided by our proposed Q-learning solution. In the following section, we illustrate the results of the comparison between the two baselines, the Balanced and eMBB-Focus baselines, and the proposed Q-learning method in terms of UE experience.

UE data rate

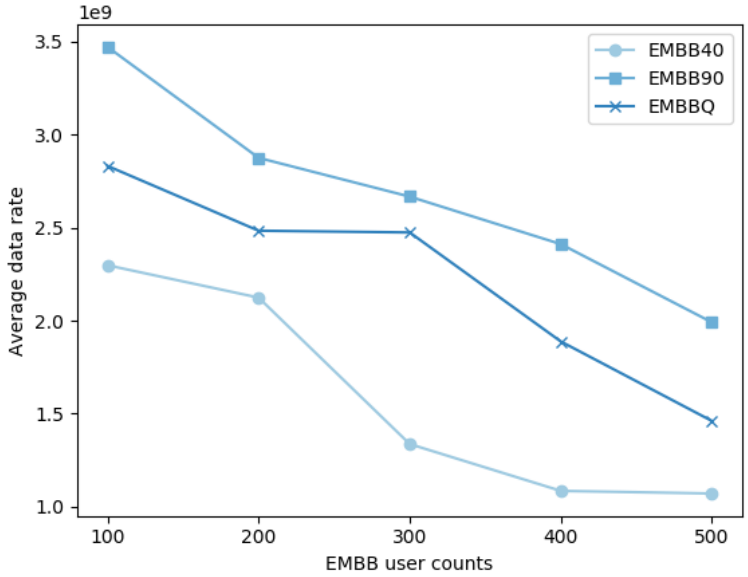

Figure 5.6: UE Average data rate in Mid Traffic

Initially, the UE data rate is calculated and presented by taking the average of data rate from UEs in the eMBB slice in Fig. 5.6. The three blue lines represent the average eMBB data rates of different models. The light blue and round dot is the Balanced baseline, eMBB40, the blue line with the "x" symbol is the Q-learning result, and the blue line with the square symbol is the eMBB-Focus baseline, eMBB90. The eMBB average data rate of the Q-learning result is higher than the balanced baseline and lower than the eMBB-Focus baseline. This shows the same result as the bandwidth usage discussed in the above section 5.3.1. However, the average result does not present substantial information on the peak rate nor the distribution of UE data rate. With greedy peak rate implementation, the Q-learning model will try to increase the UE data rate when there is available bandwidth. This greedy implementation will help improve the peak rate, especially for the eMBB slice.

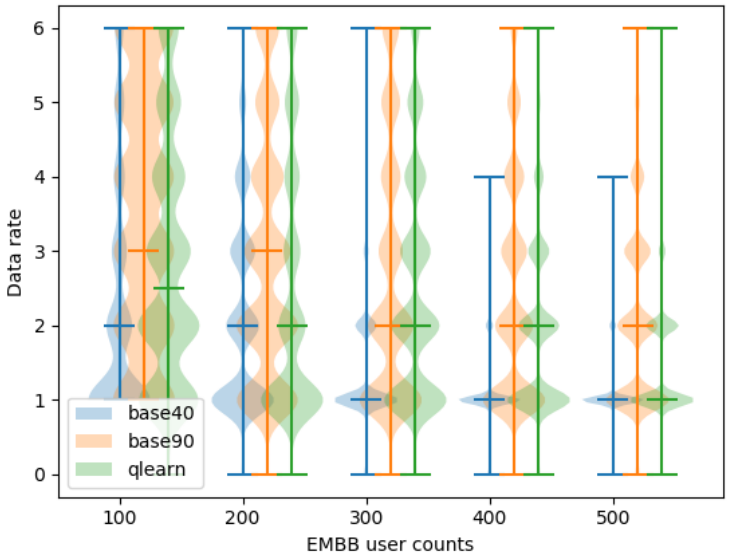

Figure 5.7: Average Data Rate in Mid Traffic

In order to present the eMBB data rate distribution visually into one combined figure, the **violin plot** [62] is chosen to represent the density of the data rate as it is a combination of the box plot [63] and kernel density plot [64]. The box plot is usually used to show the maxima, minima, and majority of the data distribution. The kernel density plot is used for displaying the probability distribution. Therefore, the violin plot can well demonstrate the maxima and minima from the box plot and sub-populations with density in the distribution from the rotated kernel density plot of our created scenarios.

Five groups in Fig. 5.7 from left to right represent the increase of eMBB users in our designated experiment. The three different colored bars in each group have the following labels, base40, base90, and qlearn. The label of base40 in the blue bar represents the result for the balance baseline as it assigns eMBB slice with 40% bandwidth. The orange bar, base90, represents the result of the eMBB-Focus baseline. The green violin bar is the Q-learning result.

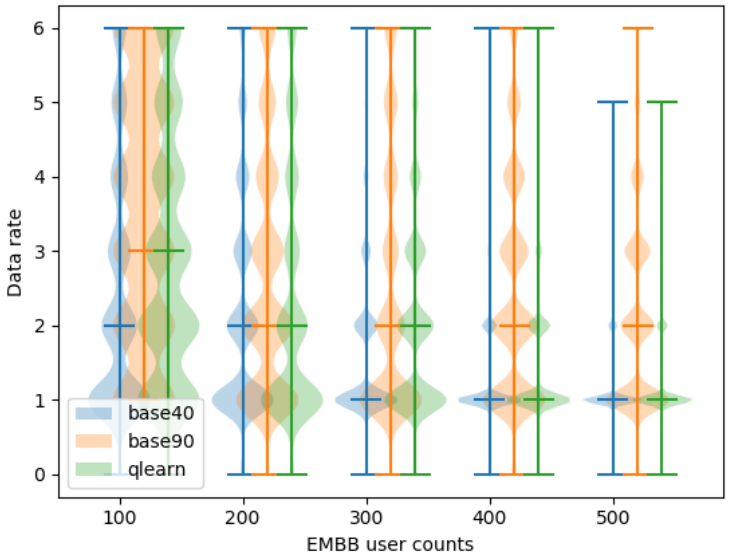

Figure 5.8: Average Data Rate in High Traffic

Fig. 5.7 shows the network's user data rate distribution in **Mid-Traffic** profile. The blue violin plot from the balanced baseline shows less population on the higher data rate than both eMBB-Focus baseline and Q-learning result. On the other hand, the violin bars of the Q-learning model and eMBB-Focus baseline have similar outcomes on both the max data rate (peak rate) and sub-populations. As analysis shown in the above UE data rate section, eMBB-Focus baseline, base90, has a slightly larger population on the high data rate than the Q-learning model because the Q-learning model shares some bandwidth with the URLLC slice. One thing to notice from Fig. 5.7 is that the tiny horizontal bar in the plot is the data distribution's median. The eMBB-Focus baseline always has the highest median compared to the other two models. The Q-learning model's median varies between the two baselines' medians

The **High-Traffic** result is shown in Fig. 5.8. The results show the same trend that the balanced baseline has most of its data rate populations on the lower side. The eMBB-

Focus baseline shows its ability to provide good QoS to eMBB users by holding the peak rate up to 500 eMBB users and the highest median among the other two models. With higher URLLC demands, the peak rate and the median rate on Q-learning now drop when eMBB users increase.

Transmission Time

The transmission time is another indicator of the QoE. In general, if one model can finish the transmission faster for the same amount of data requested by an UE, the model will provide a better-downloading experience to the user. In the following result, the transmission time is individually compared on different slices and also summed up into the total average. From left to right, we have the same five groups by increasing 100 eMBB users starting at 100 users, as described in Table 5.4. Each slice uses the same color shown in the above figures. eMBB is in blue, URLLC is in orange, and the voice is in green. Purple is newly introduced to represent the average total transmission time. The same color bars from light to dark represent the balanced baseline, the eMBB-Focus baseline, and the Q-learning result. In order to compare the result accurately, the data request uses different static values depending on its network slice type.

In **Mid-Traffic** profile, the duration of eMBB requests in blue increases when more eMBB users join the network on the top row in Fig. 5.9. The transmission time increases slower on the eMBB-Focus baseline and Q-learning model than on the Balanced baseline when increasing eMBB users between testsets. As stated earlier, the Q-learning is known to share the more unused bandwidth to the URLLC slice than eMBB. Because Q-learning has less bandwidth allocated for eMBB, the transmission time on the Q-learning model is slightly higher than the eMBB-Focus baseline.

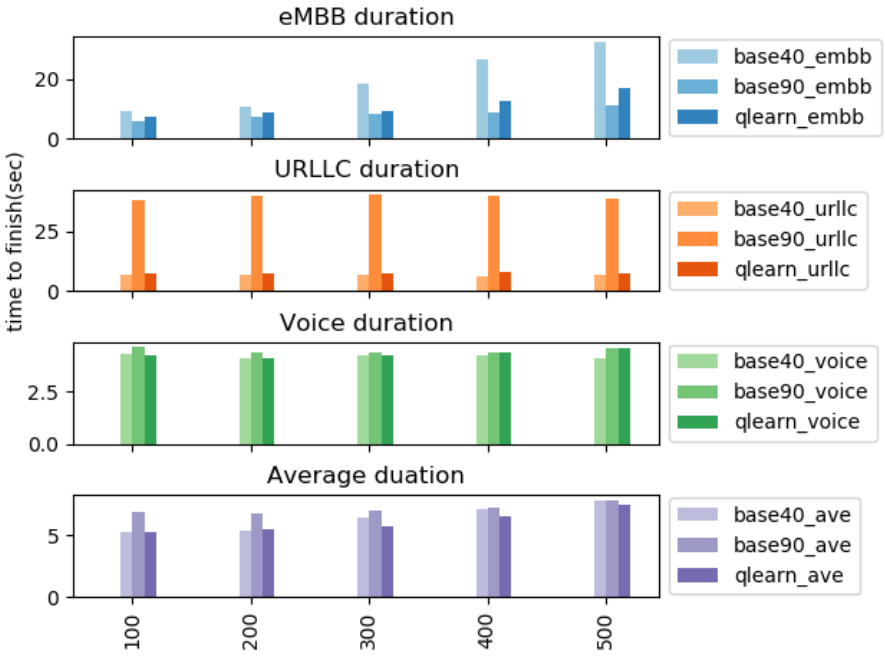

Figure 5.9: UE Transmission Time in Mid Traffic

Comparing the transmission time of the URLLC slice shown in orange, the balanced baseline is actually taking a shorter time to finish than Q-learning in Fig. 5.9. On the Opposite, the eMBB-Focus baseline has the significantly worst performance on completing the transmission for the same data amount requested by the URLLC UEs compared to the other two models. This undesired performance is due to the minimum bandwidth of five percent allocated for the URLLC slice.

The green bars on the third row in Fig. 5.9 show the transmission time of the voice slice. The baselines and Q-learning models have similar results on the transmission time of voice traffic. The bottom row in purple of Fig. 5.9 is the total average transmission regardless of network slices. The Q-learning model is able to finish the transmission faster than the other two baselines, especially when it comes to 500 eMBB users on the right of the Fig. 5.9.

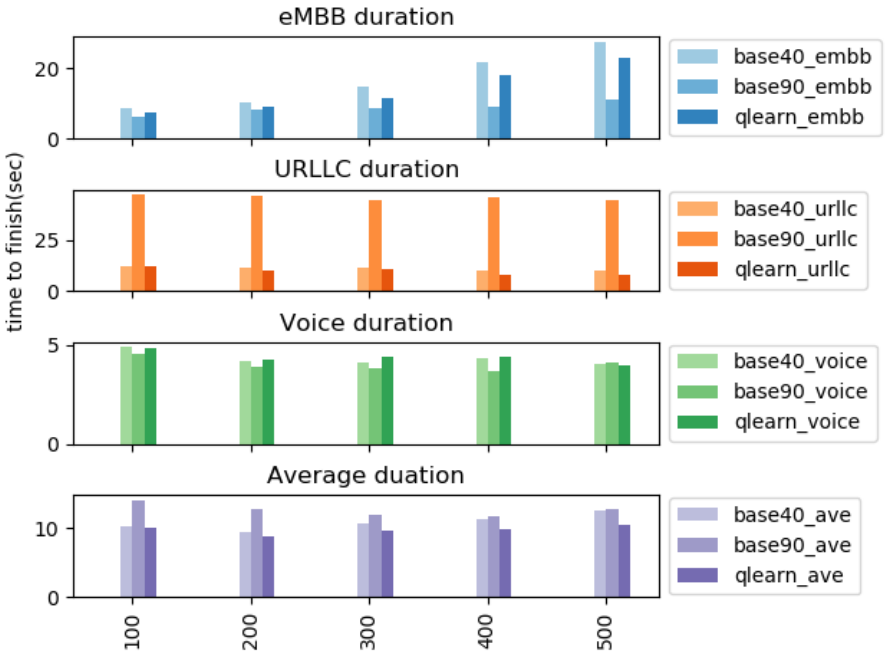

Figure 5.10: UE Transmission Time in High Traffic

In the **High-Traffic** profile, we demonstrate the transmission time from the eMBB slice to the total average time in Fig. 5.10 by the same order in Fig. 5.9. As expected, eMBB-Focus baseline has the shortest transmission time due to the most bandwidth allocated for eMBB usage. The eMBB transmission time increases significantly for the balanced baseline. In the Q-learning model in dark blue, the eMBB transmission time increases slower than the Balanced baseline but faster than eMBB-Focus baseline. Contrarily, the eMBB-Focus baseline has a significantly high URLLC transmission time with only 100 eMBB UEs in the network. The balanced baseline has a static URLLC transmission time between different numbers of eMBB UEs. The Q-learning model is able to provide the shortest URLLC transmission time. This result illustrates the trade-off between eMBB throughput and the URLLC transmission time. The green bars represent the transmission time of the voice slice. The models have similar results in general. Lastly, the total average

duration in purple indicates that the balanced baseline increases slowly over time. The eMBB-Focus baseline performs the worst with fewer eMBB demands. The Q-learning model again provides the shortest transmission time.

In summary, the Q-learning model demonstrates that it provides the best QoS and QoE. The peak rate and data rate distribution are comparable to the eMBB-Focus baseline. In addition, the Q-learning model provides a shorter transmission time. Last but not least, the fairness of the Q-learning significantly benefits the URLLC slice on transmission time. In other words, the impact of the fairness generated by the Q-learning model can neglect the loss of the eMBB throughput on the Edge O-Cloud.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

RAN has evolved alongside different generations of the mobile network, from the Second Generation (2G) to the approaching 6G mobile network. Nowadays, resource allocation in edge networks can be interpolated into diverse areas when the functional units within RAN are divided into smaller sources inside NFV. With the help of AI and growing computational power, machines can analyze and learn massive data and understand it in an unexplored way. Complex RL algorithms can extract new patterns and undiscovered policies.

This thesis focuses on finding the optimized bandwidth allocation for different network slices on the disaggregated O-RAN architecture. First, we started by reviewing the prior research in the related fields. As Elsayed et al. [10] envisioned, many areas in RAN can introduce AI techniques to provide better optimization solutions than traditional methods. In the early stage, the research focuses on the fronthaul resource allocation in the wireless network by optimizing RB in time or frequency domains in OFDM model. While the

resource allocation problem in fronthaul is still prevalent, we discovered limited research focusing on the midhaul area. Therefore, we decide to step in and contribute to this field.

In the thesis, we propose a novel Q-learning model to dynamically allocate the shared midhaul bandwidth between network slices with the existing O-RAN functional split concept. To prove our proposed scheme, a Python-based simulator is developed for customizing 5G edge mobile network and RL environment. Compared to prior works, we consider the UE movement with various speeds defined in [1]. In order to prevent UE from running out of bounds and adding complexity to our work, we propose to have a multi O-Clouds topology, which forms a beehive form of 7 O-Clouds by 6 Edge O-Clouds surrounding a centralized Regional O-Cloud. The simulator is written in Python, as most of the latest RL libraries are also developed in Python. The customized RL environment is implemented in the OpenAI and gym python packages. The proposed Q-learning model carefully follows the rules of MDP. The Q-table contains the UE counts and the bandwidth percentage of network slices as state, and the change of bandwidth in percentage as action. The reward is formulated with free bandwidth, utilization, and minimum threshold concepts.

In the result section, the validation shows that the proposed Q-learning model provides a better optimization of bandwidth allocation for both MNOs and UEs compared to the two baselines. The baselines cover a balanced bandwidth allocation and preferred bandwidth allocation, namely eMBB Focus baseline. Experiments are conducted on different test sets of traffic profiles with the Q-learning model and two baselines. On the Edge O-Cloud side, the Q-learning model is able to provide eMBB slice with high throughput in both mid and high (URLLC) traffic. However, the Q-learning model is noticed to sacrifice a small throughput on eMBB to URLLC slice compared to the eMBB Focus baseline. The fairness behavior is revealed on both mid and high traffic profiles. On the other hand,

our Q-learning model can provide a similar result on eMBB UE datarate as eMBB Focus baseline. In addition, our Q-learning model outperforms the transmission time of different slices and the total average duration of both baselines. Therefore, we can claim that the trade-off on throughput can be neglected as the impact on QoS is not noticeable.

Overall, the exciting results prove the RL is an ideal direction to follow on the bandwidth allocation problem. The simulator has the potential to work with other RL algorithms and also expand the scale of networks or O-Clouds. The journey of RAN goes from D-RAN, C-RAN to now O-RAN. I believe the evolution of RAN will not stop here. I am glad that I could participate in this evolution and make a small contribution to bringing people better QoS and QoE in this data-driven era.

6.2 Future Work

More aspects of bandwidth optimization can be attributed to the midhaul disaggregated O-RAN architectures. More could be identified after considering the following directions.

- **UE Handover**: Due to 5G cell having shorter coverage than older generations, the handover will happen more often when an UE moves within the designated beehive topology between the Edge O-Cloud. In order to increase the realism of the proposed scheme, it's highly suggested to add the extra application layer to deal with the handover problem.

- **Dynamic baselines**: The research only provides the comparison against static bandwidth by considering balanced or eMBB-Focus scenarios. Other baselines such as the Round-Robin scheduling method [53], or MILP [15] could be implemented for future comparison.

- **Deep dive into RL**: After showcasing the RL is powerful for optimizing the bandwidth utilization with the most basic but memory-demanding Q-learning algorithm in network sliced O-Clouds. Other members in the RL family can be explored and provide an equal or better result. For example, DQN is more memory efficient in high-dimension space, and more factors can be added into consideration and increase the complexity of the model. Although it might require a more powerful computational unit. Thanks to the freedom of virtualization providing, the computing unit or memory unit can be added or removed easily from an O-Cloud.

- **OpEx resources**: With more powerful RL algorithms introduced in the near future, more factors related to OpEx can be added to the model. Throughput is not the only factor of MNO network deployment. In order to provide a high peak rate on a cost-effective network, CPU and power usage and more resources should be considered in the future scheme when building a disaggregated virtualization network.

Bibliography

[1] "Minimum requirements related to technical performance for imt-2020 radio interface(s)," ITU-R, TR IMT-2020.TECH PERF REQ, Nov 2020.

[2] K. Faisal, "C-RAN vs Cloud RAN vs vRAN vs O-RAN vs traditional RAN guide!" Apr 2021. [Online]. Available: https://telcocloudbridge.com/blog/c-ran-vs-cloud-ran-vs-vran-vs-o-ran/

[3] "O-RAN whitepaper - building the next generation ran, october 2018," O-RAN Alliance, Oct 2018. [Online]. Available: https://www.o-ran.org/resources.

[4] A. Bhat, N. Gupta, J. Thaliath, R. Banerji, V. Sapru, and S. Singh, "Role of Open-Source in 6G Wireless Networks," in *6G Mobile Wireless Networks*. Springer International Publishing, March 2021, pp. 379–392. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-72777-2_17

[5] O. Andersson, "Functional splits: The foundation of an Open 5G RAN," in *5G Technology World*. Benetel, May 2021. [Online]. Available: https://www.5gtechnologyworld.com/functional-splits-the-foundation-of-an-open-5g-ran/

[6] Z. Ghadialy, "5G RAN functional splits," *3G4G Blog*, Mar 2021. [Online]. Available: https://blog.3g4g.co.uk/2021/03/5g-ran-functional-splits.html

[7] "What is a RAN intelligent controller (RIC)?" *Juniper Networks*. [Online]. Available: https://www.juniper.net/us/en/research-topics/what-is-ric.html

[8] H. Ji, O. Alfarraj, and A. Tolba, "Artificial Intelligence-Empowered Edge of Vehicles: Architecture, Enabling Technologies, and Applications," *IEEE Access*, vol. 8, pp. 61 020–61 034, 2020.

[9] R. S. Sutton, F. Bach, and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press Ltd, 2018, ISBN: 978-0262039246.

[10] M. Elsayed and M. Erol-Kantarci, "AI-Enabled Future Wireless Networks: Challenges, Opportunities, and Open Issues," *IEEE Vehicular Technology Magazine*, vol. 14, pp. 70–77, 9 2019.

[11] D. Nojima, Y. Katsumata, T. Shimojo, Y. Morihiro, T. Asai, A. Yamada, and S. Iwashina, "Resource Isolation in RAN Part While Utilizing Ordinary Scheduling Algorithm for Network Slicing," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, 2018, pp. 1–5.

[12] H. Zhou, M. Elsayed, and M. Erol-Kantarci, "Ran resource slicing in 5G using multi-agent correlated q-learning," in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2021, pp. 1179–1184.

[13] H. Yu, F. Musumeci, J. Zhang, M. Tornatore, and Y. Ji, "Isolation-Aware 5G RAN Slice Mapping over WDM Metro-Aggregation Networks," in *Journal of Lightwave Technology*, vol. 38, no. 6. IEEE, 2020, pp. 1125–1137.

[14] T. Pamuklu, C. Cavdar, and C. Ersoy, "Renewable Energy Assisted Function Splitting in Cloud Radio Access Networks," in *Mobile Networks and Applications*, vol. 25, no. 5. Springer, Oct. 2020, pp. 2012–2023.

[15] T. Pamuklu and C. Ersoy, "GROVE: A Cost-Efficient green radio over ethernet architecture for next generation radio access networks," in *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 1. IEEE, Mar. 2021, pp. 84–93.

[16] R. Joda, T. Pamuklu, P. E. Iturria-Rivera, and M. Erol-Kantarci, "Deep Reinforcement Learning-Based Joint User Association and CU–DU Placement in O-RAN," in *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, 2022, pp. 4097–4110.

[17] T. Uitto, "Making Sense of O-RAN and vRAN - Part Two," *Nokia Blog*, Nov 2020, accessed: 2022-09-22. [Online]. Available: https://www.nokia.com/blog/making-sense-of-o-ran-and-vran-part-two/

[18] "Technical specifications and technical reports for a 5G based 3GPP system," 3GPP, TS 21.205, 2018.

[19] "Summary of release 16 work items," 3GPP, TR 21.916, 2021.

[20] "Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper," in *Cisco Mobile Visual Networking Index (VNI)*. Cisco, Jan 2022. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[21] "O-RAN Alliance: About Us," *O-RAN Alliance*, Feb 2018. [Online]. Available: https://www.o-ran.org/about

[22] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, "Intelligent 5G: When Cellular Networks Meet Artificial Intelligence," in *IEEE Wireless Communications*, vol. 24, no. 5, 2017, pp. 175–183.

[23] A. Platek and J. You, "What is the relationship between AI and 5G?" *Ericsson Blog*, Jan 2022. [Online]. Available: https://www.ericsson.com/en/blog/2022/1/whats-the-relationship-between-ai-5g

[24] N. F. Cheng, T. Pamuklu, and M. Erol-Kantarci, "Reinforcement Learning Based Resource Allocation for Network Slices in O-RAN Midhaul," in *2023 IEEE 20th Consumer Communications Networking Conference (CCNC)*, 2023, pp. 140–145. [Online]. Available: https://arxiv.org/abs/2211.07466

[25] "NR; Base Station (BS) radio transmission and reception," 3GPP, TS 38.101, 2017.

[26] A. Gupta and R. K. Jha, "A Survey of 5G Network: Architecture and Emerging Technologies," in *IEEE Access*, vol. 3, 2015, pp. 1206–1232.

[27] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View," in *IEEE Access*, vol. 6, 2018, pp. 55 765–55 779.

[28] E. Mohyeldin, "Minimum Technical Performance Requirements for IMT-2020 radio interface(s) ITU-R Workshop on IMT-2020 terrestrial radio interfaces," 2017. [Online]. Available: https://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2020/Documents/S01-1_Requirements%20for%20IMT-2020_Rev.pdf

[29] X. Ge, "Ultra-Reliable Low-Latency Communications in Autonomous Vehicular Networks," in *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, 2019, pp. 5005–5016.

[30] A. Söderlund, G. Foglander, R. Laczko, and S. B. Muiño, "Exploring new centralized RAN and fronthaul opportunities," *Ericsson Blog*, May 2021. [Online]. Available: https://www.ericsson.com/en/blog/2021/5/exploring-new-centralized-ran-and-fronthaul-opportunities

[31] S. Hsu, "What is vRAN?" in *Understanding virtualization*. Red Hat, 2018. [Online]. Available: https://www.redhat.com/en/topics/virtualization/what-is-vran

[32] "NR; NR and NG-RAN Overall description; Stage-2," 3GPP, TS 38.300, 2017. [Online]. Available: http://www.3gpp.org/DynaReport/38300.htm

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

[34] P. E. Iturria-Rivera, H. Zhang, H. Zhou, S. Mollahasani, and M. Erol-Kantarci, "Multi-Agent Team Learning in Virtualized Open Radio Access Networks (O-RAN)," *Sensors*, vol. 22, no. 14, p. 5375, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/14/5375

[35] M. Moffett, "Edge and fog computing: Cutting through the haze (part 1)," *Cisco Blogs*, Oct 2020. [Online]. Available: https://blogs.cisco.com/government/edge-and-fog-computing-cutting-through-the-haze-part-1

[36] "OpenFog," *OPC Foundation*, Oct 2016. [Online]. Available: https://opcfoundation.org/markets-collaboration/openfog/

[37] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead," *Computer Networks*, vol. 182, p. 107516, 12 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620311786

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

[39] R. R. Schaller, "Moore's law: past, present and future," *IEEE Spectrum*, vol. 34, no. 6, pp. 52–59, 1997.

[40] V. Marx, "The big challenges of big data," *Nature*, vol. 498, no. 7453, p. 255–260, Jun 2013. [Online]. Available: https://www.nature.com/articles/498255a

[41] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, "A Survey on Deep Learning: Algorithms, Techniques, and Applications," *ACM Comput. Surv.*, vol. 51, no. 5, 2018. [Online]. Available: https://doi.org/10.1145/3234150

[42] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 818–833.

[43] E. Cambria and B. White, "Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]," *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, 2014.

[44] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, 2017, pp. 2222–2232.

[45] S. Csaba, *Algorithms of Reinforcement Learning*. Morgan and Claypool, 2010.

[46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *OpenAI*, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[47] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 1928–1937.

[48] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An Introduction to Deep Reinforcement Learning," *Foundations and Trends in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018. [Online]. Available: https://doi.org/10.1561/2200000071

[49] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing Atari with Deep Reinforcement Learning," *DeepMind Technologies*, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602

[50] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos, "Distributional Reinforcement Learning with Quantile Regression," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, no. 10. AAAI Press, 2018, pp. 2892—-2901. [Online]. Available: http://arxiv.org/abs/1710.10044

[51] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proceedings of 6th International Conference on Learning Representations (ICLR 2018)*, 2018. [Online]. Available: http://arxiv.org/abs/1801.01290

[52] M. Elsayed and M. Erol-Kantarci, "AI-Enabled radio resource allocation in 5G for URLLC and eMBB users," in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019, pp. 590–595.

[53] Y. Sun, Y. Wang, H. Yu, B. Guo, and X. Zhang, "A learning-based bandwidth resource allocation method in sliced 5G C-RAN," in *2019 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2019, pp. 1–5.

[54] T. Pamuklu, M. Erol-Kantarci, and C. Ersoy, "Reinforcement learning based dynamic function splitting in disaggregated green open RANs," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[55] H. Zhang, H. Zhou, and M. Erol-Kantarci, "Team Learning-Based Resource Allocation for Open Radio Access Network (O-RAN)," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 4938–4943.

[56] T. Bai and R. W. Heath, "Coverage and Rate Analysis for Millimeter-Wave Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 1100–1114, 2015.

[57] "Discrete event simulation for Python," SimPy, 2020. [Online]. Available: https://simpy.readthedocs.io/en/latest/

[58] "Build next-gen apps with OpenAI's powerful models," OpenAI, 2022. [Online]. Available: https://openai.com/api/

[59] "Gym is a standard API for reinforcement learning, and a diverse collection of reference environments," *Gym Documentation*. [Online]. Available: https://www.gymlibrary.dev/

[60] J. S. Seybold, "12.5.1," in *Introduction to RF propagation*. Wiley, 2005, p. 292.

[61] C. Shannon, "Communication in the presence of noise," in *Proceedings of the IRE*, vol. 37, no. 1, 1949, pp. 10–21.

[62] "Violin plot," National Institute of Standards and Technology, 2019. [Online]. Available: https://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/violplot.htm

[63] "Box plot," National Institute of Standards and Technology, 2015. [Online]. Available: https://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/boxplot.htm

[64] "Kernel density plot," National Institute of Standards and Technology, 2018. [Online]. Available: https://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/kernplot.htm

[65] Y. Yao, H. Zhou, and M. Erol-Kantarci, "Deep Reinforcement Learning-based Radio Resource Allocation and Beam Management under Location Uncertainty in 5G mm Wave Networks," in *2022 IEEE Symposium on Computers and Communications (ISCC)*, 2022, pp. 1–6.

[66] S. Mollahasani, T. Pamuklu, R. Wilson, and M. Erol-Kantarci, "Energy-Aware Dynamic DU Selection and NF Relocation in O-RAN Using Actor–Critic Learning," *Sensors*, vol. 22, no. 13, p. 5029, 2022.

[67] "Cloud architecture and deployment scenarios for O-RAN virtualized RAN - v02.02," O-RAN ALLIANCE, Tech. Rep., 2021.

[68] M. Dryjański, Łukasz Kułacz, and A. Kliks, "Toward Modular and Flexible Open RAN Implementations in 6G Networks: Traffic Steering Use Case and O-RAN xApps," *Sensors*, vol. 21, p. 8173, 12 2021.

[69] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, "Intelligence and Learning in O-RAN for Data-Driven NextG Cellular Networks," *IEEE Communications Magazine*, vol. 59, pp. 21–27, 10 2021.

[70] E. Sarikaya and E. Onur, "Placement of 5g ran slices in multi-tier o-ran 5g networks with flexible functional splits," in *2021 17th International Conference on Network and Service Management (CNSM)*, 2021, pp. 274–282.

[71] Y. Wang, G. Feng, J. Wang, F. Wei, Y. Sun, and S. Qin, "Self-healing of radio access network slices," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[72] J.-W. Cho, P. Yang, T. Q. Quek, and J.-H. Kim, "Service-aware resource allocation design of uav ran slicing," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 801–805.

[73] Q.-T. Luu, S. Kerboeuf, and M. Kieffer, "Uncertainty-aware resource provisioning for network slicing," in *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, 2021, pp. 79–93.

[74] M. Maule, P.-V. Mekikis, K. Ramantas, J. Vardakas, and C. Verikoukis, "Real-time dynamic network slicing for the 5g radio access network," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[75] X. Wang and T. Zhang, "Reinforcement Learning Based Resource Allocation for Network Slicing in 5G C-RAN," *2019 Computing, Communications and IoT Applications, ComComAp 2019*, pp. 106–111, 2019.

[76] H. Yu, F. Musumeci, J. Zhang, M. Tornatore, L. Bai, and Y. Ji, "Dynamic 5G RAN slice adjustment and migration based on traffic prediction in WDM metro-aggregation networks," in *Journal of Optical Communications and Networking*, vol. 12, no. 12, 2020, pp. 403–413.

[77] H. Hirayama, Y. Tsukamoto, S. Nanba, and K. Nishimura, "Ran slicing in multi-cu/du architecture for 5g services," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–5.

[78] Y.-J. Liu, G. Feng, J. Wang, Y. Sun, and S. Qin, "Access Control for RAN Slicing based on Federated Deep Reinforcement Learning," in *ICC 2021 - IEEE International Conference on Communications*, no. Grant 62071091. IEEE, Jun 2021, pp. 1–6.

[79] S. S. Shinde, D. Marabissi, and D. Tarchi, "A network operator-biased approach for multi-service network function placement in a 5G network slicing architecture," *Computer Networks*, vol. 201, no. November, p. 108598, 2021. [Online]. Available: https://doi.org/10.1016/j.comnet.2021.108598https://linkinghub.elsevier.com/retrieve/pii/S1389128621004989

[80] S. II Moon, H. Hirayama, Y. Tsukamoto, S. Nanba, and H. Shinbo, "Ensemble Learning Method-Based Slice Admission Control for Adaptive RAN," in *2020 IEEE Globecom Workshops (GC Wkshps.* IEEE, dec 2020, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9367536/

[81] C. Bektas, D. Overbeck, and C. Wietfeld, "SAMUS: Slice-Aware Machine Learning-based Ultra-Reliable Scheduling," in *ICC 2021 - IEEE International Conference on Communications.* IEEE, 2021, pp. 1–6.

[82] A. Umesh, "Study on new radio access technology: Radio access architecture and interfaces (release 14)," 3GPP, TR 38.801, Apr 2017.

[83] "Management and orchestration; 5G end to end key performance indicators (kpi)," 3GPP, TS 28.554, 2019.

[84] "RL Baselines Made Easy," Stable Baselines, 2021. [Online]. Available: https://stable-baselines.readthedocs.io/en/master/index.html

[85] T. Pamuklu, S. Mollahasani, and M. Erol-Kantarci, "Energy-efficient and delay-guaranteed joint resource allocation and DU selection in O-RAN," in *2021 IEEE 4th 5G World Forum (5GWF).* IEEE, oct 2021.

Appendix A: Notations

Table A.1: All Notations

Sets	**Size**	**Description**
$i \in \mathcal{I}$	I	UEs
$j \in \mathcal{J}$	J	Edge O-Clouds (DUs)
$a \in \mathcal{A}$	A	bandwidth change range
$k \in \mathcal{K}$	K	bandwidth slices
$t \in \mathcal{T}$	T	time slots
$v \in \mathcal{V}$	V	UE movement
Variables	**Domain**	**Description**
$b^t_{j,k}$	$[0, 100]$	allocated bandwidth for slice k of DU j
$a^t_{j,k}$	$[-10, 10]$	allocated bandwidth percentage change for slice k of DU j
Given Data	**Domain**	**Description**
v_i	$\mathbb{N}^+$	movement speed of UE i
β^I_k	$\mathbb{N}^+$	initial bandwidth for slice k
β^M_k	$\mathbb{N}^+$	maximum bandwidth for slice k
$l_{j,k}$	$\mathbb{N}^+$	floor value of bandwidth threshold for slice k
$u^t_{j,k}$	$\mathbb{N}^+$	number of UEs for slice k of DU j
Runtime Data	**Domain**	**Description**
$\mathcal{D}^t_{i,k}$	$\mathbb{N}^+$	average data rate of slice k of UE i
$\mathcal{B}^t_{j,k}$	$\mathbb{N}^+$	throughput of slice k of DU j
$\mathcal{F}^t_{j,k}$	$\mathbb{N}^+$	Free bandwidth of slice k of DU j