Multi-Agent Deep Reinforcement Learning Assisted Pre-connect Handover Management

by

Yao Wei

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

Electrical and Computer Engineering

Carleton University
Ottawa, Ontario

Abstract

Handover is an essential and significant component of mobility management in cellular networks. Handover management is more challenging for Fifth Generation (5G) networks because of ultra-reliable low latency communications (URLLC) requirements. This thesis proposes a make-before-break (MBB) adopted handover mechanism for user equipment (UE), namely, pre-connect handover (PHO). PHO aims at providing a seamless and reliable handover technique in 5G networks. PHO utilizes the Deep Q-Networks (DQN) algorithm to facilitate the sequential decision-making problem of the target base station (T-BS) selection based on the reference signal received quality (RSRQ) values and RSRQ change rates of all the candidate base stations (BSs). Furthermore, a multi-agent deep reinforcement learning (MADRL) solution is tailored to extend the DQN-assisted UE-associated PHO solution for modeling of the multi-UE environment. All the autonomous agents learn the action policy by interacting with the environment in a distributed manner. The feasibility of the PHO mechanism has been validated extensively via Network Simulator 3 (NS-3) and NS3-Gym. The performance of the DQN and MADRL-assisted PHO management solutions have been evaluated by considering various configurations. The experimental results demonstrated that the proposed PHO is not only achievable, but also that the DQN-assisted PHO technique can productively accomplish the optimal BS selection to maximize the success rate of PHO. Moreover, the MADRL-assisted PHO management solution can also be conducted and effectively applied to a realistic multi-UE environment where each UE is modeled with an agent.

Acknowledgements

First and foremost, I would like to express my deep and sincere gratitude to my supervisors, Professor Chung-Horng Lung and Professor Samuel Ajila. I appreciate their invaluable guidance, generous support, enormous knowledge and constant patience. Their insights and guidance have provided motivations and directions during the entire research process.

I am grateful to Ricardo Paredes Cabrera from Ericsson Canada Inc. His immense knowledge and insightful feedback helped me to sharpen my thinking during research.

Last but not least, no words can truly express how grateful and thankful I am to my family for their endless love, constant support, and continuous encouragement in my life.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

3GPP	Third Generation Partnership Project
5G	Fifth Generation
5GC	5G Core
A3C	Asynchronous Advantage Actor-Critic
AI	Artificial Intelligence
API	Application Programming Interface
BBM	Break-Before-Make
BS	Base Station
CHO	Conditional Handover
CN	Core Network
CQI	Channel Quality Indicator
CSI	Channel State Information
DL	Downlink
DNN	Deep Neural Network
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
EARFCN	E-UTRA Absolute Radio Frequency Channel Numbers
E-UTRA	Evolved UMTS Terrestrial Radio Access
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
EPC	Evolved Packet Core
HIT	Handover Interruption Time
IMSI	International Mobile Subscriber Identity

IMT	International Mobile Telecommunications
IPC	Inter-process communication
IQL	Independent Q-learning
ITU-R	International Telecommunication Union Radiocommunication
GPL	General Public License
KNN	K-nearest Neighbors
LSTM	Long Short-Term Memory
LTE	Long Term Evolution
MADRL	Multi-Agent Deep Reinforcement Learning
MARL	Multi-Agent Reinforcement Learning
MAS	Multi-Agent System
MBB	Make-Before-Break
MDP	Markov Decision Process
MIT	Mobility Interruption Time
ML	Machine Learning
MME	Mobility Management Entity
MR	Measurement Report
NG-RAN	Next Generation Radio Access Network
NR	New Radio
NS-3	Network Simulator 3
PDCP	Packet Data Convergence Protocol
PHO	Pre-connect Handover
PHO-SR	Pre-connect Handover Success Rate

QCI	Quality of Service Class Identifier
QoS	Quality of Service
RACH	Random Access Channel
RAN	Radio Access Network
RAPID	Random Access Preamble Identifier
RAT	Radio Access Technology
RB	Resource Block
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
RLC	Radio Link Control
RLF	Radio Link Failure
RNN	Recurrent Neural Network
RNTI	Radio Network Temporary Identifier
RRC	Radio Resource Control
RRM	Radio Resource Management
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
SDU	Service Data Units
SGW	Serving Gateway
SIM	Subscriber Identity Module
SN	Sequence Number

SNR	Signal to Noise Ratio
SVM	Support Vector Machine
S-BS	Serving Base Station
T-BS	Target Base Station
UE	User Equipment
UL	Uplink
UMTS	Universal Mobile Telecommunications System
URLLC	Ultra-Reliable Low Latency Communications
X2AP	X2 Application Protocol
XnAP	Xn Application Protocol
ZMQ	ZeroMQ

Chapter 1: Introduction

1.1 Background

The number of mobile connections in wireless communication networks is continuously growing exponentially. It is reported in the Ericsson 2021 Mobility Report [1] that mobile data traffic soared almost 300 times more than ten years ago, and that global traffic was already 3 to 4 times higher than three years earlier. Mobile subscriptions reached 8.1 billion by the end of 2021. 5G subscriptions are expected to overtake 4G Long Term Evolution (LTE) subscriptions gradually and are expected to meet the requirements of seamless connectivity, higher throughput, higher network capacity, and lower latency. Maintaining the best connection for an increasing number of mobile users in 5G cellular systems brings challenges to mobility management [2]. The handover feature is one of the critical elements of cellular networks in supporting user mobility. In cellular networks, handover occurs when a UE is active on a data session and moves from one serving cell coverage area to another.

Handovers need to be properly managed since service interruptions during handovers pose threats to quality-of-service (QoS) and degrade overall user satisfaction. Due to the growing demands of wireless mobile services and increased execution capacity of real-time applications, seeking an efficient handover technique to improve handover performance has been extensively addressed in cellular networks. Under the fast development of small cells in 5G networks, radio channel conditions change more dramatically and listing of neighboring cells changes more frequently, which leads to increased interference and more frequent handovers. More frequent handovers result in more service interruption time [3]. The high demands of mobility devices and applications

creates challenges and opportunities by driving network operators to exploit more efficient and effective technologies for handover management. Machine learning (ML) techniques have not only gained significant attention in the wireless networks research community, but also have attracted the industries' interests to create artificial intelligence (AI) networks [1].

Many researchers have explored and proposed ML-based handover management approaches. There are three main types of ML approaches: supervised learning, unsupervised learning and reinforcement learning (RL). Supervised learning uses a labeled training dataset provided from a knowledgeable external supervisor to train a model. The objective of supervised learning is to formalize a model to be used on the situations not present in the training set. Unsupervised learning typically aims to find patterns and outcomes hidden in an unlabeled dataset. RL learns with no need of pre-collected data or a pre-formulated model; instead, an agent learns the optimized policy for a solution by taking actions to maximize a numerical reward in an environment [4].

Deep reinforcement learning (DRL) is a combination of deep learning and RL. In DRL, a deep neural network (DNN) is used to approximate the agent's optimal policy and/or its optimal utility function [5]. DRL takes the advantage of RL to explicitly model the problem as a goal-directed agent interacting with an uncertain environment [4], which allows the network entities to learn and build knowledge from the environment [6]. DRL also utilizes DNN to accelerate the learning process and improve the learning performance in large-scale and complex networks [6][7].

In recent years, modern wireless networks tend to be decentralized and autonomous, where mobile UEs need to make local and autonomous decisions based on local

observations. The multi-agent reinforcement learning (MARL) technique solves sequential decision-making problems using multiple agents operating in a common environment, where each agent aims to maximize its own reward by interacting with the environment [8] [9]. MADRL extends the functions of DRL with MARL. MADRL enables multiple agents to interact with an environment to solve complex problems that the traditional DRL technique is not able to handle [10], particularly with distributed learning systems. MADRL is well suited to handover management problems in wireless networks for three major reasons:

- Firstly, the mobility model of UEs has the characteristic of being dynamic and stochastic; therefore, it is often hard to derive an accurate mathematical model to formulate the problem. A model-free DRL algorithm can solve a complex problem without a predefined model. Specifically, a UE can learn an optimal policy for T-BS selection without knowing the mobility pattern in a handover application [6].
- Secondly, the communication network is a complex real-time system. The DRL agent can conduct offline training and implement an online prediction to satisfy the real-time requirements.
- Thirdly, MADRL is a good fit for the multiple UEs handover scenario, where all the UEs act independently in the network. Each DRL agent represents an individual UE and learns the partial information of the system in a distributed manner.

1.2 Motivation

1.2.1 Problem Statement

Handover management is one of the essential features in cellular networks. It is responsible for handling the mobility of UEs in continuing active communication sessions without

disruption, ideally during a UE's movement. Any failure or error during a handover can result in dropped calls, radio link failure (RLF), and packet loss in the radio channel, which degrades customer satisfaction. With the requirement of URLLC services in 5G networks, more efficient handover techniques are required to support stable connections for UE mobility. URLLC services target 0.5ms of one-way user plane latency, and 0ms of mobility interruption time (MIT) in 5G New Radio (NR) networks [11]. Therefore, near 0ms handover execution time is required to reduce communication interruptions [12] compared with 49.5ms in LTE networks [13].

The non-optimal parameter configuration, handover trigger time, and target cell selection are possible factors for causing handover failures. Additionally, one typical deployment scenario of 5G networks is the highway scenario [11], where UEs moving with high speed may cross cell coverage areas in a few seconds, which reduces the probability of successfully making handover decisions and/or completing the handover procedure [14].

1.2.2 Research Objective

This thesis aims to design an enhanced seamless handover technique to improve handover performance and reliability. Specifically, the research focuses on increasing handover success rate and reducing packet loss during handovers. Moreover, a smart PHO management mechanism is proposed by applying the DQN algorithm and MADRL technique to pre-connection trigger time and T-BS(s) selection to improve handover performance and reliability.

1.3 Contributions

There are three main contributions in this thesis:

1. The thesis proposes an enhanced handover management approach, namely PHO, which has several advantages:
    - Firstly, PHO adopts the MBB handover scheme [15], where a UE connects to T-BS before being disconnected from the serving base station (S-BS) [12]. The MBB scheme can reduce handover interruption time (HIT) and packet loss during the handover process.
    - Secondly, in PHO, the T-BS prepares the handover in advance by pre-allocating the radio resource and sends a handover command message to UE before the handover is triggered. The preparation is implemented when radio conditions are stable and reliable, which reduces the chance of handover failure. The proposed scheme can increase the handover success rate.
    - Thirdly, once the pre-connection is established between the UE and the candidate T-BS, the downlink (DL) packets are duplicated and forwarded from the S-BS to the preconnected T-BS via X2-U interfaces. All the received packets at T-BS are stored in a capacity-adjustable queue, and sent back to UE when the handover procedure is completed. The early DL data duplicating-forwarding-buffering mechanism aims to reduce packet loss caused by degraded signal qualities on the S-BS and/or radio interference before and during handovers.
    - Fourthly, the proposed approach provides the functionality of establishing multiple pre-connections to different T-BSs, which further increases handover success rate and provides for better QoS.
2. The thesis explores the MADRL technique, specifically the multi-agent DQN algorithm [16] for PHO management. Each UE is modeled as an independent agent in

the system. All agents operate in a partially observable environment, learn in a distributed way, and no information is exchanged between each other. Each individual agent makes its own handover decisions based on candidate BSs' RSRQ values and RSRQ change rates. The goal is to find an optimal BS selection policy to maximize its own PHO success rate (PHO-SR), which reduces the interrupt time accordingly.

3. The thesis develops an offline learning and online prediction framework. Offline learning can reduce both time and space complexity [2]. The proposed PHO technique is evaluated with a NS-3 simulator and NS3-Gym toolkit. The simulated environment is adopted for offline model training. The trained model can be used to conduct an online prediction in real-time applications.

1.4 Organization

The thesis is structured as follows. Chapter 2 presents an overview of relevant materials in this study, including a detailed overview of handover concepts, conditional handover (CHO) mechanism in 5G networks [17], machine learning approaches, and research tools including NS-3 and NS3-Gym adopted for the thesis. Chapter 3 explores related state-of-the-art research in handover management solutions based on RL, DRL, or MADRL algorithms. Chapter 4 depicts the proposed multi-agent DQN-assisted PHO management solution at the system-level. Chapter 5 details the implementations with the NS-3 simulator and NS3-Gym toolkit. Chapter 6 demonstrates the experiments and summarizes the results. Finally, Chapter 7 concludes the thesis and outlines the future research directions.

Chapter 2: Background

This chapter presents the basic background and related works.

- Section 2.1 highlights the wireless radio access network.
- Section 2.2 describes the handover management concept and related techniques.
- Section 2.3 introduces two enhanced handover techniques specified by the Third Generation Partnership Project (3GPP) in 5G networks, which are conditional handover and early DL forwarding.
- Section 2.4 gives a high-level overview of the machine learning methods, and focuses on the concepts of RL, DRL, and MADRL.
- Section 2.5 introduces the research tools which are used in the experiments.

2.1 Radio Access Network

The mobile telecommunication networks consist of four main components, which are UE, radio access network (RAN), core network (CN) and transport network. RAN consists of BSs, providing wireless connectivity to UEs on the radio interfaces. Each BS contains one or more cells. The BSs monitor the UEs' air interface status, determine and execute the handover process. The handover occurs between the air interfaces of different BSs or cells [18].

In LTE networks, RAN is named as Evolved Universal Mobile Telecommunications System (UMTS) Terrestrial Radio Access Network (E-UTRAN), CN is named as Evolved Packet Core (EPC) [19]. The E-UTRAN architecture is illustrated in Figure 2.1 (a) [20]. The BS is called eNodeB (or eNB). Each eNodeB handles the radio communications between the UEs and EPC. The eNodeBs are interconnected with each other via the X2 interface and are also connected to EPC through the S1 interface. Specifically, eNodeBs

are connected to the Mobility Management Entity (MME) via the S1-MME interface and to the Serving Gateway (SGW) via the S1-U interface [20]. Both X2 and S1 interfaces can be used in handover procedures in different scenarios.

In 5G NR networks, RAN is called Next Generation Radio Access Network (NG-RAN), and CN is called 5G core (5GC). The NG-RAN architecture is illustrated in Figure 2.1 (b) [17]. The LTE and 5G NR BSs are co-existing in NG-RAN. A gNodeB (or gNB) is the 5G radio BS that connects 5G NR devices to the 5GC network on the NR radio interface. An enhanced LTE eNB (ng-eNB) is an upgraded version of eNodeB that connects LTE devices to 5GC on the LTE radio interface. All the gNodeBs and ng-eNBs are interconnected via Xn interface, and connected to 5GC via the NG interface [17].

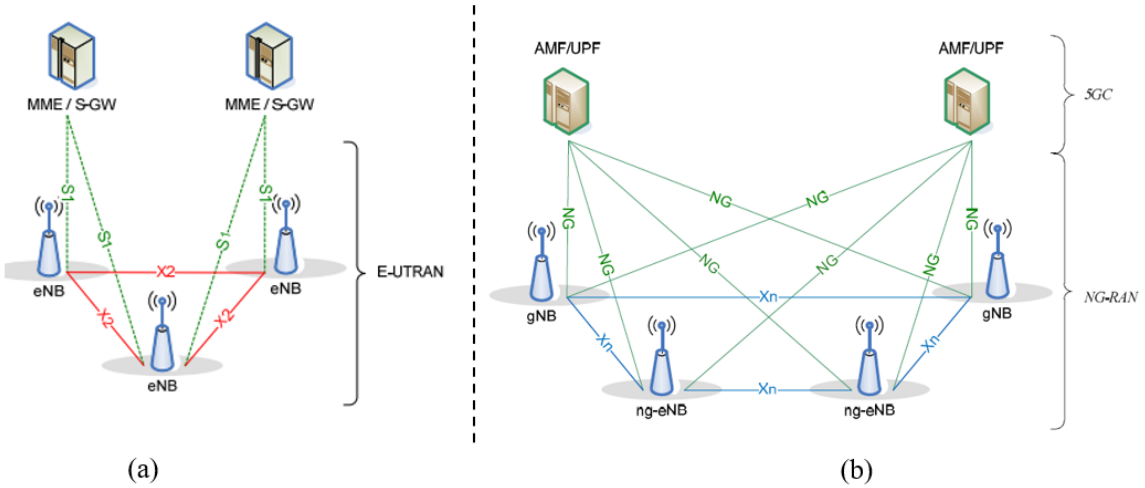


Figure 2.1 RAN Architecture: (a) LTE [20]**; (b) 5G** [17]

Mobility management is a fundamental function of mobile networks. It ensures the UEs maintain continuous service during the movements. Mobility management contains three components which are cell re-selection, handover management, and redirection [21].

The cell re-selection procedure is the process of selecting a more suitable cell for a UE to camp on [22], [23].

The X2 application protocol (X2AP) [24] handles UE mobility within E-UTRAN in LTE networks. The handover related procedures include Handover Preparation, Sequence Number (SN) Status Transfer, Early Status Transfer, UE Context Release, Handover Success Indication, Handover Cancellation, and Conditional Handover Cancellation [25]. In contrast, the Xn application (XnAP) protocol takes the relevant responsibilities of the UE mobility management in NR-RAN networks [26].

The radio resource management (RRM) is responsible for efficiently using the available air interface resources [27]. RRM in the cellular network consists of the following main functions: resource allocation, packet scheduling, link adaptation, radio admission control and handover management [28]. One of the important functionalities in RRM is to make handover decisions based on handover algorithms [27]. The specific radio resources used in handover management include DL and uplink (UL) carrier frequency, DL and UL channel bandwidth, and antenna configurations. The QoS management for all UEs is also significant during and after the handover [2]. QoS class identifier (QCI) is a mechanism specified by 3GPP to ensure carrier traffic is allocated appropriate QoS, different carrier traffic requires different QoS with different QCI values. QCI is one of the service-level QoS parameters [29]. Each QCI is associated with a priority level defined by 3GPP. The smaller priority number has the highest priority. If congestion is encountered, the lowest priority level traffic with the highest priority number would be the first to be discarded. The one-to-one mapping of standardized QCI values to standardized characteristics is listed in Appendix A [29].

2.2 Handover Management

This section describes various techniques in handover management.

- Section 2.2.1 describes handover types.
- Section 2.2.2 introduces 3GPP specified handover events and trigger conditions.
- Section 2.2.3 depicts handover management and reporting.
- Section 2.2.4 details the handover procedures.
- Section 2.2.5 introduces handover performance metrics.

2.2.1 Handover Types

2.2.1.1 Hard/Soft Handover

The hard handover algorithm is also known as break-before-make (BBM), in which the UEs detach from S-BS before establishing connections to T-BS [12]. The hard handover is standardized in LTE networks [30]. Each UE communicates only one BS at a time, which makes efficient use of network resources, and reduces the network complexity. However, the main drawback of hard handover is the temporary service disruption, increased system delay and decreased system throughput [12][15].

The soft handover algorithm is also known as MBB, in which the UEs retain the connection with S-BS when establishing the connection to T-BS [31]. Therefore, a UE can connect to S-BS and T-BS simultaneously during the handover process, which minimizes the service interruption time, improves the link gains with lower packet loss [15], and lowers the probability of handover failure [31]. However, the advantage also brings up the cost, because more radio resources are required to support one connection, which reduces the radio resources availability and degrades the network capacity. The MBB handover is

the most suitable handover type for low-latency applications [32]. The MBB handover has been specified in E-UTRAN by 3GPP [20].

2.2.1.2 Intra/Inter-RAT Handover

Handovers that occur in the same radio access technology (RAT) are called Intra-RAT handovers, which are also called horizontal handovers. In contrast, the Inter-RAT handovers occur between different radio technologies [3], which are also called vertical handovers. Both Intra-LTE and Inter-LTE handovers exist in LTE networks.

Intra-LTE handover refers to both S-BS and T-BS being served within the same LTE network. It primarily uses either the X2 or S1 interface. The X2 interface is considered faster than the S1 interface [33]. When the X2 connection is available between the S-BS and T-BS, the handover is completed without involving the EPC. During an X2-based handover, the X2 interface carries the control information transmitted between the serving eNBs and the target eNBs. However, if the X2 connection is unavailable or the X2 process has failed, then a handover can be processed through the S1 interface, with the condition that both the S-BS and T-BS are in the same MME/SGW.

Inter-LTE handover happens with other LTE nodes. Inter-LTE handover includes Inter-MME and Inter-MME/SGW. The Inter-MME handover occurs when a UE moves between two different MMEs within the same SGW. Inter-MME/SGW handover is similar to Inter-MME, the only difference is that a UE moves from one MME/SGW to another MME/SGW, within the different MMEs and SGWs.

The proposed PHO technique focuses on Intra-LTE X2-based handovers. However, the proposed method can also be applied in NR-RAN theoretically.

2.2.2 Handover Events and Trigger Conditions

A handover event is triggered when an entry condition is satisfied by the network. Such conditions are signaled by the BS in the form of threshold, hysteresis, and offset parameters. The 3GPP has specified the handover events and trigger conditions in LTE [34] and 5G networks [35], which are listed in Table 2.1. The A1-A6 events are used in Intra-RAT scenarios, whereas, B1 and B2 are used in Inter-RAT scenarios.

Table 2.1 3GPP Specified Handover Events and Trigger Conditions [34], [35]

Event	Trigger Conditions
A1	Serving cell becomes better than a threshold
A2	Serving cell becomes worse than a threshold
A3	Neighbor cell becomes offset better than serving cell
A4	Neighbor cell becomes better than a threshold
A5	Serving cell becomes worse than threshold1 and neighbor cell becomes better than threshold2
A6	Neighbor cell becomes offset better than secondary cell (This event for carrier aggregation)
B1	Inter-RAT neighbor cell becomes better than threshold
B2	Serving cell becomes worse than threshold1 and Inter-RAT neighbor cell becomes better than threshold2

2.2.3 Handover Measurements and Reporting

2.2.3.1 Handover Measurement Parameters

There are four basic measurement parameters of RRM in LTE [27], Channel Quality Indicator (CQI), Reference Signal Received Power (RSRP), RSRQ, and Carrier Received Signal Strength Indicator (RSSI).

The RSRP and RSRQ are common parameters used for cell reselection and handover decision in Intra-LTE handover [27]. A2-A4-RSRQ and A3-RSRP are two popular Intra-LTE handover algorithms [36]. Both RSRP and RSRQ are 3GPP specified cell-specific signal parameters [37]. RSRP is a signal strength parameter (in dBm) and it is defined as the average power (in Watts) of the resource elements that carry cell-specific reference signals within the considered frequency bandwidth. RSRQ is a signal quality parameter (in

dB) and it is defined as Equation 2-1, where N is the number of resource blocks (RBs) of the carrier RSSI measurement bandwidth. RSSI is the total received power, including co-channel non-serving and serving cells, adjacent channel interference and thermal noise [27].

$$RSRQ = \frac{N * RSRP}{RSSI} \qquad \textbf{Equation 2-1}$$

It can be seen that RSRQ considers the combined effect of signal strength and channel interference and noise. RSRQ performs better than RSRP in practical measurements [27]. Therefore, RSRQ is considered as a key observation in the proposed DQN-assisted PHO management, which is presented in Section 4.2.

2.2.3.2 Handover Reporting Methods

In general, there are three methods for handover reporting [3]:

- Event-triggered reporting: The UE sends a measurement report (MR) to S-BS once a configured event is triggered.
- Periodic reporting: The UE performs the MR reporting at specific time intervals.
- On-demand reporting: The UE sends the MR immediately after receiving the request from S-BS.

2.2.4 Handover Procedure

The 3GPP specified baseline handover is network-controlled and UE-assisted [20]. The handover process is composed of three phases: handover preparation, handover execution, and handover completion [17] [20]. In a cellular network, the UE performs the radio signal strength or signal quality measurement over all the surrounding BSs. If a certain predefined criteria is satisfied, A MR is sent to the S-BS [3]. The S-BS makes the handover decision

based on the RSRP and/or RSRQ values, and oversees the execution. The T-BS provides guidance for UEs on radio access, including radio resource configuration. During the handover, a UE switches the radio channels, resource, and/or cells to maintain the ongoing session.

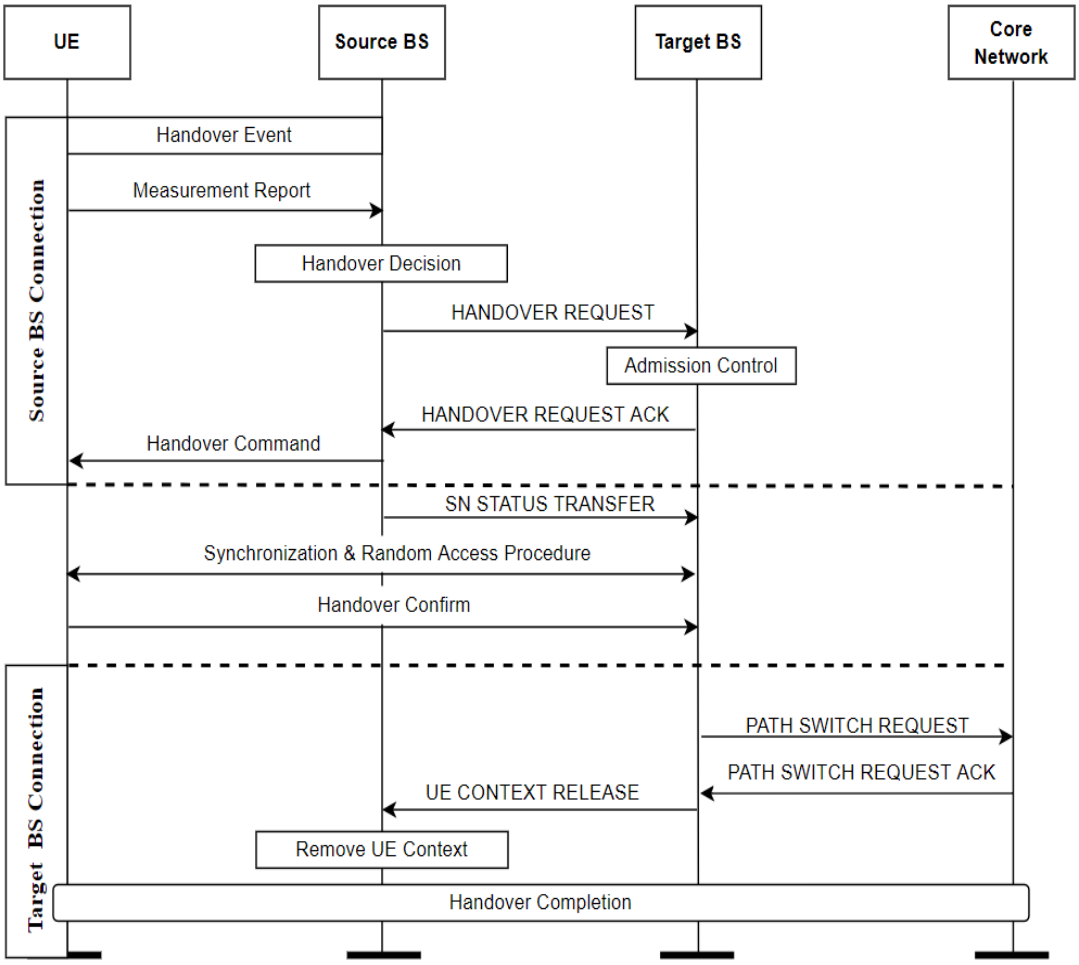


Figure 2.2 3GPP Baseline Handover Procedure (adapted from [17], [20])

As demonstrated in Figure 2.2 (adapted from [17], [20]), the handover preparation is initialized at S-BS once a handover is triggered. A Handover Request message is sent to the T-BS. Upon successful admission accepted at the T-BS, the necessary resources are allocated, and a handover command is sent to the S-BS within a Handover Request

Acknowledge message. The S-BS then sends a mobility control message to the UE. Once the UE receives the mobility control message from S-BS, the handover execution phase starts. The UE synchronizes with the T-BS and initializes a random-access procedure. The UE is attached to the T-BS until the random-access procedure is completed successfully. After the connection is established between the UE and the T-BS, the UE's data path is switched from S-BS to T-BS. Additionally, the T-BS sends a UE Context Release message to S-BS to inform the successful handover. After that, the S-BS finalized the procedure by releasing resources for the UE [17], [20].

2.2.5 Handover Performance Metrics

Various performance metrics have been used to measure the handover performance, including HIT, handover success rate, handover latency, packet loss, and ping-pong rate, etc.

- HIT is a time duration where the UE is not permitted to send data plane packets to the BS [2].
- Handover success rate is defined as the number of successful handovers divided by the total number of triggered handovers [2].
- Handover latency is defined by 3GPP [13], it is a time duration between the time of UE receiving a handover command from the S-BS and sending a radio resource control (RRC) connection reconfiguration complete message to the T-BS. In other words, it is the time taken during the execution phase.
- Ping-pong rate is defined as the number of ping-pong events during a given period of time [38]. A ping-pong event is the occurrence of a handover between a serving cell and a target cell, followed by another handover triggered by the original serving cell.

The proposed PHO approach aims to increase the handover success rate, decrease the HIT, and reduce the packet loss during the handover process.

2.3 Enhanced Handover Approaches in 5G

URLLC is one of the important usage scenarios in 5G networks defined by International Telecommunication Union Radiocommunication (ITU-R) International Mobile Telecommunications (IMT) for 2020 and beyond. The requirement of 0.5ms one-way user plane latency and 0ms MIT for URLLC mobile applications [11] is challenging. In order to satisfy the URLLC requirements, 3GPP has specified several potential handover improvement techniques [20], which include conditional handover and early data forwarding.

2.3.1 Conditional Handover

CHO is specified by 3GPP [20] as a handover improvement mechanism. It is executed by the UE when one or more handover execution conditions are satisfied. Specifically, the UE evaluates execution condition(s) for candidate cells upon receiving the CHO configuration, and executes CHO once the execution condition(s) are met for a CHO candidate T-BS. Once a handover is executed, the UE stops the evaluating [20]. The CHO supports multiple candidate T-BSs to prepare in advance in the networks.

The main idea of CHO is allowing a UE to communicate with the S-BS for the handover command receiving while the radio link is still stable and reliable. The improvements are provided by decoupling the preparation and execution phases. CHO can increase the handover success rate by avoiding a RLF, which is caused by the poor signal quality and/or cell interference [39] in the handover preparation phase, and decrease HIT

by implementing the MBB handover inherently during the handover execution phase [40]. In CHO, the handover preparation is network-controlled and the handover execution is UE-controlled [40]. The 3GPP specified CHO in LTE is depicted in Figure 2.3 [20]. The concept for CHO in 5G [17] is similar to LTE.

- Network controlled preparation in CHO is initialized at S-BS. The S-BS configures the UE with measurement configuration, which is used by the UE to trigger MR for potential CHO. Once receiving a MR from a UE, S-BS makes a decision of CHO based on the information in MR, then a request message is sent to T-BS. If the request is accepted, the S-BS receives a handover command from T-BS in the Handover Request Acknowledge message. The handover command is forwarded to the UE in a RRC Connection Reconfiguration message at step 7. However, the UE doesn't access the T-BS immediately when receiving the handover command; instead, waiting for an additional condition to be triggered.
- UE controlled execution is autonomously started by UE to T-BS once the condition is fulfilled. The number of the execution conditions can be configured up to two per candidate BS [17].

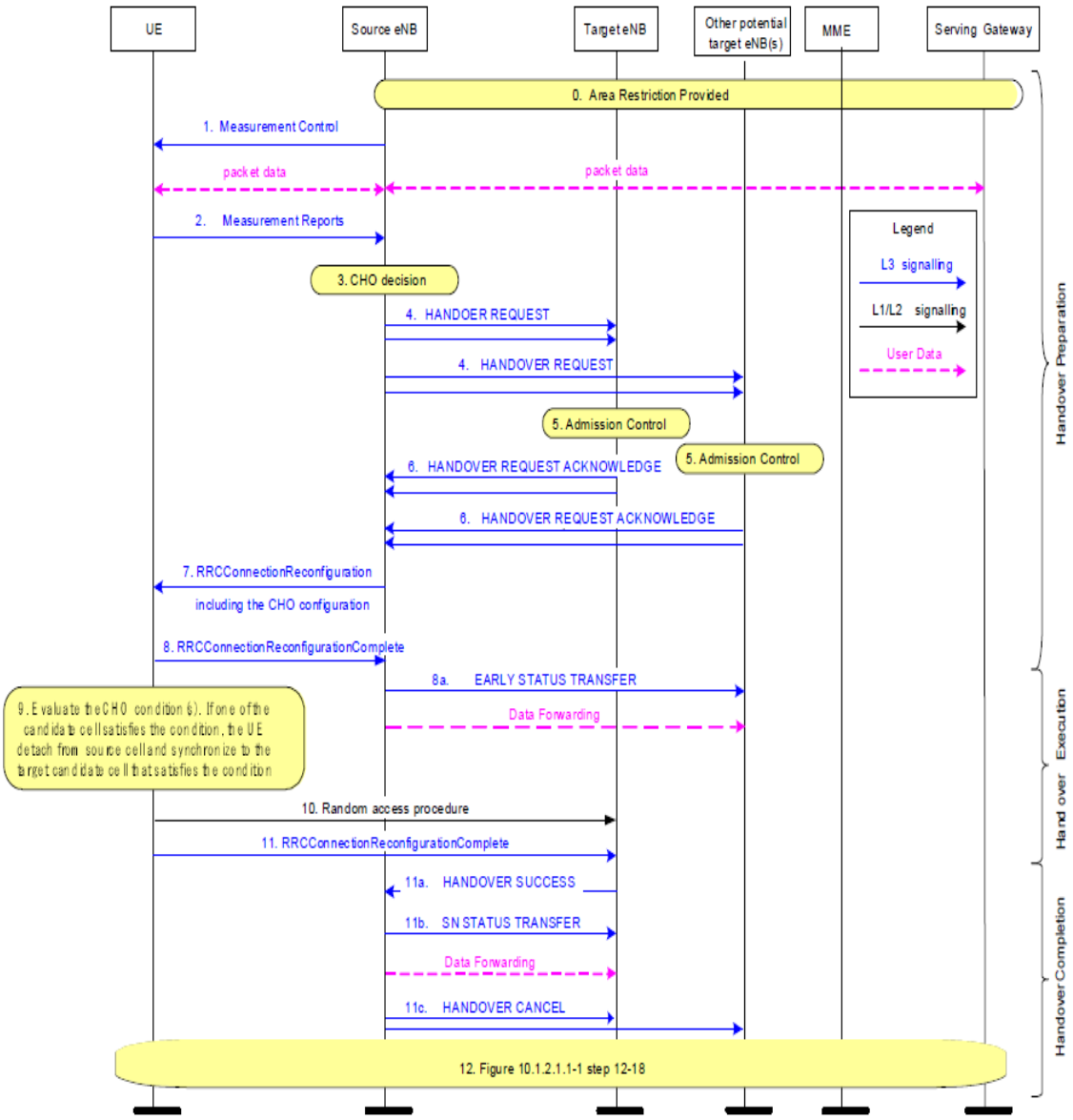


Figure 2.3 Intra-MME/SGW Conditional Handover in LTE [20]

2.3.2 Early Data Forwarding

The early data forwarding technique specified by 3GPP [17] [20] is used to reduce HIT and increase the mobility robustness [32]. In early data forwarding, the S-BS initiates data forwarding to a candidate T-BS during the handover execution phase. As shown in Figure 2.3 [20], at step 8a, S-BS initializes the early data forwarding by sending an EARLY

STATUS TRANSFER message to T-BS. Only DL Packet Data Convergence Protocol (PDCP) Service Data Units (SDUs) with SNs are supported in early data forwarding. During the early data forwarding process, the S-BS still maintains the DL data path to the UE until the handover success message is received from the T-BS [20].

2.4 ML-Assisted Handover Management

The growing demands of the wireless mobile services and complex network architecture in 5G mobile networks pose multiple network management challenges. The ML powered wireless networks are new trends from network design to infrastructure management and for user performance improvement [41]. The emerging ML-assisted techniques enable a shift from reactive-driven operations to proactive-driven operations for various network applications, including handover management.

This thesis proposes a ML-based PHO management solution. The T-BS selection for handover is a decision-making problem. The proposed ML-assisted handover management approach ensures that a decision is made for each handover in an efficient and effective manner to increase handover success rate, and reduce packet loss during the handover process.

2.4.1 Machine Learning

Machine learning is defined as a study that gives the computers the ability to learn by themselves without being explicitly programmed [42]. The ML algorithms can be classified based on how learning is performed. ML is divided into three main categories shown in Figure 2.4 [2]: supervised learning, unsupervised learning and reinforcement learning.

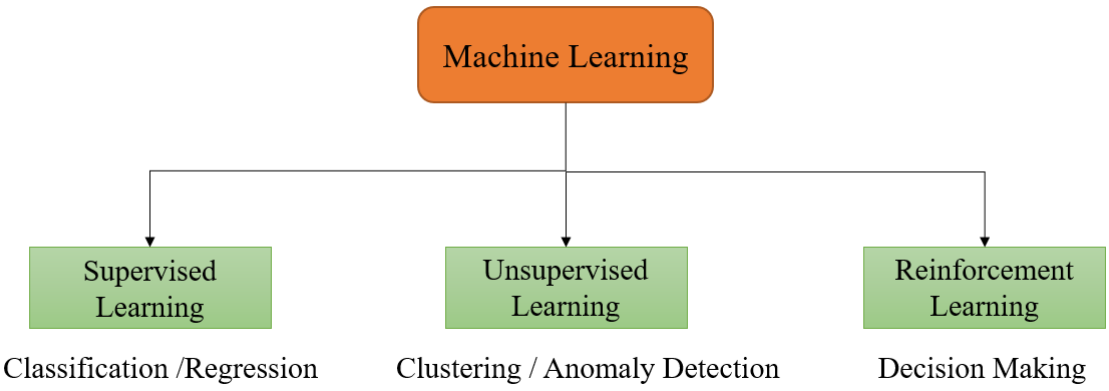


Figure 2.4 Machine Learning Categories [2]

Supervised learning requires a labeled dataset consisting of all the input and output features. It tries to learn a function that maps the inputs to the expected outputs by minimizing the bias and variance errors in the predicted results [2]. The most widely used supervised learning algorithms are: decision tree, support vector machine (SVM), linear regression, k-nearest neighbor (KNN) algorithm, random forest, and neural networks, etc. Supervised learning algorithms can help with providing user mobility information through the prediction of future location, trajectory, etc., which is needed for proactive handover optimization [43].

Unsupervised learning has the target of finding the underlying patterns and structures from unlabeled data. This approach is mainly used for clustering, anomaly detection, pattern recognition problems[2]. K-means clustering is one of the common unsupervised learning algorithms. The authors in [44] applied the K-means clustering algorithm to partitioned UEs into clusters, where the UEs have the similar mobility pattern in each cluster; then DRL was implemented to determine the optimal handover policy in each cluster.

Unlike supervised and unsupervised learning, the RL agent learns how to map the situations to actions from the feedback and experiences without any labeled or unlabeled input dataset [4]. The agent seeks the optimal action by interacting with the environment, to achieve the maximized reward.

2.4.2 Reinforcement Learning

RL is defined as an agent that learns a theoretical optimal action policy to maximize the accumulated future rewards by interacting with its environment. It is an approach for solving sequential decision-making problems [4]. Figure 2.5 [4] demonstrates the interaction between an active decision-making agent and its environment in RL.

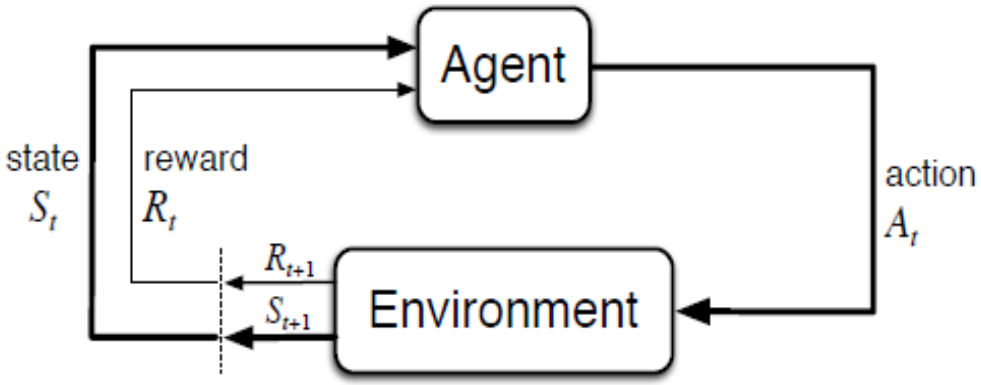


Figure 2.5 Interaction between Agent and Environment [4]

At each time step t of an episode, an agent executes an available action a_t to interact with the environment in the state s_t. The environment gives the numerical value of reward r_t as the feedback of the action. The state consists of all the necessary information for the agent to make the decision of taking the best choice of action. The action selections are determined on not only the instantaneous rewards, but also the subsequent states, and the future rewards [4].

2.4.2.1 Markov Decision Process

RL is formalized from the idea of dynamic system theory, specifically, the Markov Decision Process (MDP) approach [4]. MDP provides a mathematical framework for modeling discrete-time decision-making problems [45]. MDP can be defined by a tuple $(S, A, s_0, P, R, \gamma, H)$ [45].

- S is state space, which contains all possible states in the system.
- A is action space, which contains all possible actions the agent may take when interacting with the system.
- s_0 is the initial state of the system.
- P is the transition probability from the state s to next state s' over the actions given the current state, where $s, s' \in S$.
- R is the reward value received by the agent, which is used to evaluate the transition from the state-action pair (s, a) to state s';
- γ is a discount factor, which is a number in the interval [0,1], used to determine the discounted weight of future rewards compared to immediate rewards [46]. If $\gamma = 0$, the agent only concerns immediate rewards. γ = 1 indicates that rewards in the distant future should count just as much as the reward at the next time-step. The higher the discount factor, the slower the convergence of iterative methods. When γ approaches 1, the time for convergence approaches ∞ [47]. Tuning the discount factor implies a trade-off. The higher γ the higher possibility of ensuring average optimality for discounted-optimal policies, but the bigger the computational costs for calculating the solution.

- H is the horizon, which is the maximum number of time-steps in each episode. It may be a positive integer or ∞.

2.4.2.2 Elements of Reinforcement Learning

There are three main elements in RL: a policy, a reward, and a state-value function [4].

- A policy (π) is defined as a strategy used by the agent to take action in a given state at a given time. The policy can either be a simple lookup table or a complex strategy. The policy is the core of an agent, since it is along with the efficiency of the determining behavior. The goal of the agent is to learn an optimal policy (π^*) that maximizes the cumulative rewards.
- A reward is the feedback returned from the environment to the agent after taking an action. It is a numerical value and is used to evaluate the goodness of the action taken by the agent. In MDP, the total reward for one episode can be calculated by $R = r_1 + r_2 + \cdots + r_n$. Accordingly, the total rewards with the discounted future rewards from time t can be expressed as Equation 2-2 [4], where γ is the discount factor, k is the time steps in the future.

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{n-t} r_n = r_t + \gamma R_{t+1} = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \qquad \textbf{Equation 2-2 [4]}$$

- A value function, which is also known as state-value function, represents an estimation of how good it is for the agent to perform a certain action in a given state. The value is the maximum discounted reward value that an agent can expect to achieve in a given state. The concepts of value and value function are key to most of the RL methods [4]. The value of a state s is the expected infinite discounted sum of reward that an agent will gain if it starts in that state and under policy π. It can be defined as Equation 2-3

[48] , where γ is the discount factor, r_t is the reward at time t, and s_t is the state at time t .

$$V_\pi(s) = \mathbb{E}_\pi\{\sum_{t=0}^{\infty} \gamma^t r_t \,|s = s_t\} \qquad \textbf{Equation 2-3 [48]}$$

Accordingly, the optimal value of a state is achievable by executing the optimal policy. It is expressed as Equation 2-4 [48], where π is the policy.

$$V^*(s) = \max_\pi \mathbb{E}(\sum_{t=0}^{\infty} \gamma^t r_t | s = s_t) \qquad \textbf{Equation 2-4 [48]}$$

2.4.2.3 Model-Based and Model-Free Reinforcement Learning

Besides the three main components presented in Section 2.4.2.2, there is one more optional component in RL, which is a model of the environment. Based on the availability of the system model, there are two types of RL strategies: model-based RL and model-free RL.

Model-based RL can be split into two categories: the model is given or the model is learnt by the agent [49]. If the model is given, then the reward function and the transition process can be accessed directly by the agent or the method learns the model. However, sometimes the model cannot acquire the model directly due to the complexity of the environment, but the agent can learn a model from interactions with the environment then apply the model in policy improvement. Model-based RL assumes that the agent knows the system dynamics, that is how the system transits from one state to another one, and how the rewards are generated [5]. In a model-based system, the model of the environment is obtainable, and the interconnections of different states are known [50]. The model of the environment can predict the next state and next reward with a given state and action.

Whereas, in model-free RL, the agent learns based on explored samples without the model of the environment. Therefore, model-free RL is an explicitly trial-and-error learner [4]. Further, there are two types of methods in model-free RL, which are policy-based (also known as on-policy) and value-based (also known as off-policy).

Recall the definition of policy introduced in Section 2.4.2.2. A policy $\pi(a|s)$ is simply a function that maps states to actions in RL. Policy-based approaches learn and optimize an explicit policy during the training; the policy is updated iteratively until it reaches the maximized accumulative return [49]. The value function is not required. Policy-based methods are often adequate [4]. By contrast, value-based methods learn a value function instead of a policy, the goal of this method is to maximize the value function. However, an implicit policy can be derived directly from the value function. For example, Q-Learning, Q-Network, and DQN [51] are model-free value-based algorithms. The taxonomy of RL algorithms is summarized as Figure 2.6 (adapted from [49]).

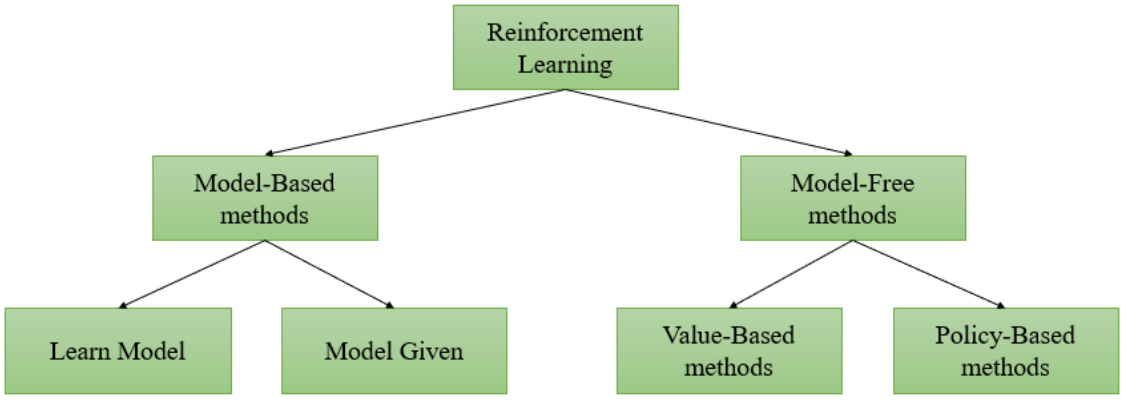


Figure 2.6 Taxonomy of Reinforcement Learning Algorithms (adapted from [49] **)**

For most real-world applications, it is challenging to have an accurate model of the environment. Therefore, the model-free RL approach is attracting more interest in the

research community. This thesis utilizes the model-free DQN algorithm for optimal T-BS selection in PHO management.

2.4.2.4 Exploration and Exploitation

In RL, the agent learns through trial-and-error interactions with a dynamic environment. There are two strategies during the learning process, which are exploration and exploitation. Exploration is the process when an agent tries to explore the environment by taking different available actions. Meanwhile, the agent also exploits the experienced optimal actions in order to achieve the maximum cumulative reward. Each action must be tried many times to gain a reliable estimate of expected reward in a stochastic scenario [4]. The trade-off between exploration and exploitation is one of the challenges in RL [46]. In order to optimize the policy, the agent has to try various actions to see the results. That might result in worse performance, because the policy for action selection might be worse than the current one. However, without exploration, it might never find any improvements.

In this research, the ϵ-greedy policy is used to balance exploration and exploitation, which is one of the most widely used strategies. ϵ-greedy algorithm selects its highest valued action with probability $1 - \epsilon + \frac{\epsilon}{k}$, and uniformly at random among all other $k - 1$ actions with probability $\frac{\epsilon}{k}$, where k is the total number of possible actions [52]. The probability ϵ is decayed in each time step.

2.4.3 Deep Reinforcement Learning

DRL [51] is a combination of RL and DNN. As demonstrated in Figure 2.7 [6] (c), DNN acts as a component of a RL agent in DRL. DRL embraces the advantage of DNN to train

the learning process in RL to accelerate the learning process and improve the learning performance in complex decision-making problems [6][7].

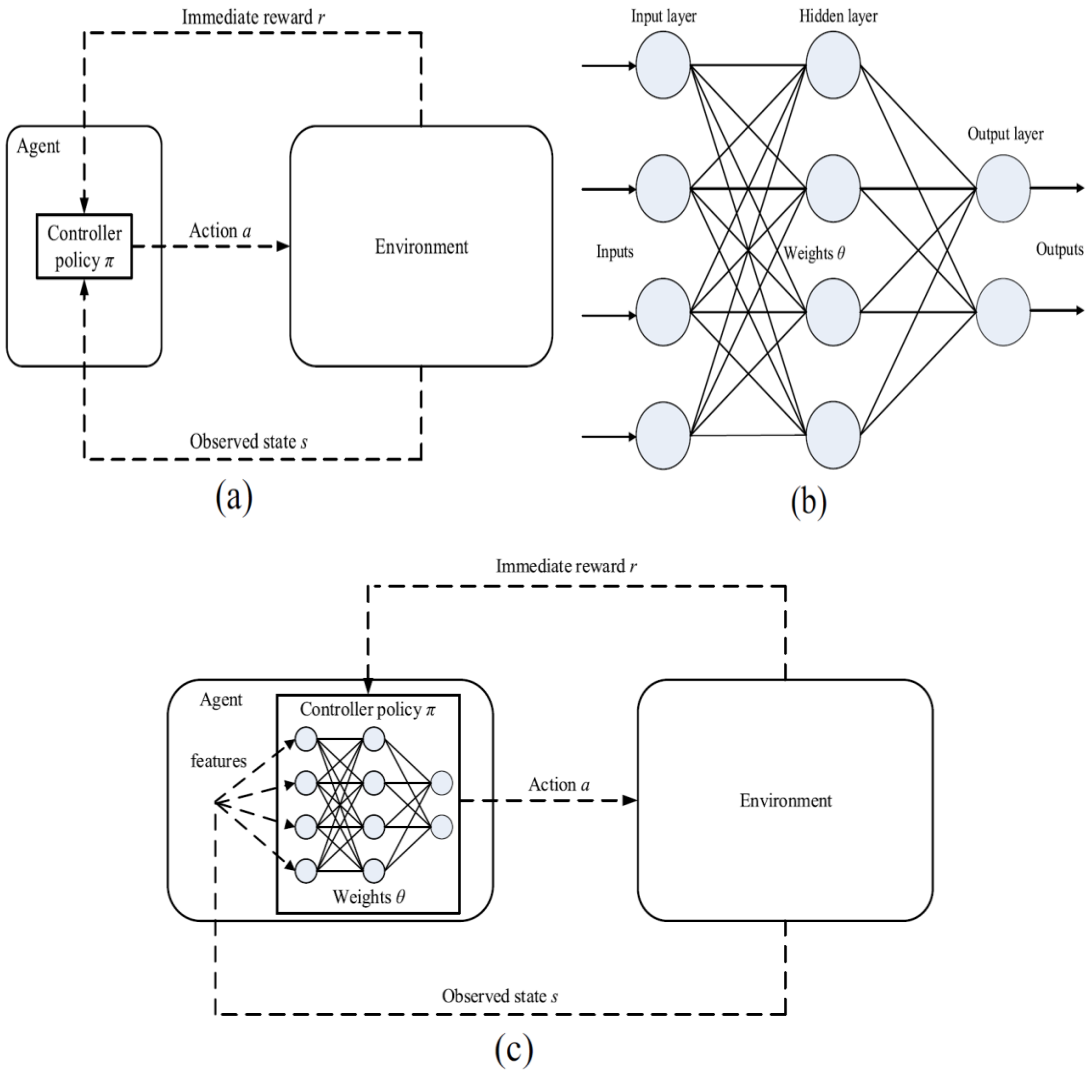


Figure 2.7 (a) RL; (b) DNN; (c) DRL [6]

There are two main advantages of the DRL algorithm: Firstly, DRL is suitable for high-dimensional state-spaces, accelerates the learning progress, overcomes the issues of memory complexity, computational complexity in RL [53]. Secondly, the neural network can take much more information as its inputs, enlarging the state-action space for better policy making [54].

2.4.3.1 Deep Q-Network

One of the most popular DRL algorithms is DQN, which is the first DRL method proposed by Google DeepMind [51]. DQN is a value-function-based DRL algorithm, which combines Q-learning with DNN. The DNN acts as a function approximator for action-value function $Q(s,a)$. DQN is based on training DNN to approximate the optimal action policy π^* and optimal action-value function $Q^*(s,a)$. As demonstrated in Figure 2.8 (adapted from [51]), the input of a multi-layer DNN is a given state s_t of an RL environment, and it outputs a vector of action-values for the given state.

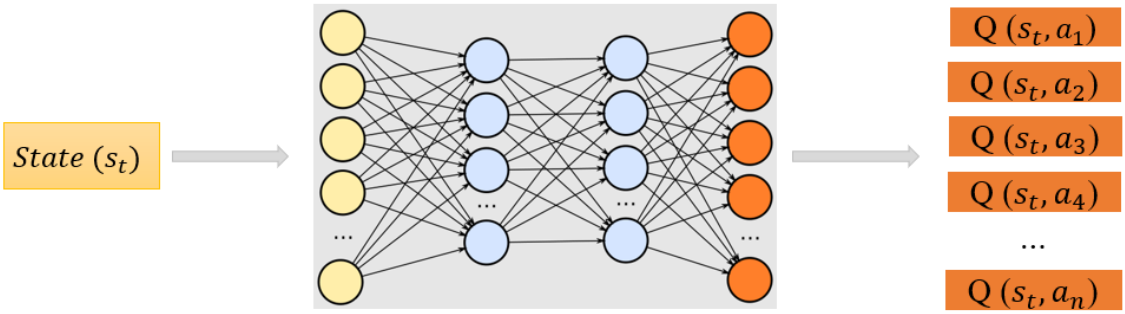


Figure 2.8 The Functionality of DNN in DQN (adapted from [51])

The action-value function $Q(s,a)$ is formulated as Equation 2-5 [51], where R_t is the total discounted reward from time step t, which is obtained by Equation 2-2. $Q(s,a)$ is the expected return starting from state s by taking action a, under the policy π.

$$Q_\pi(s,a) = \mathbb{E}[R_t \,|s_t = s,\ \ a_t = a, \pi] \qquad \textbf{Equation 2-5 } [51]$$

The parameterized action-value function can be expressed as $Q(s,a;\theta)$, where θ are the parameters of the Q-Network. DQN frames the RL learning problem as the minimization of a loss function $L(\theta)$, which is shown as Equation 2-6 [51], where θ

represents the parameters of the primary network; θ' represents the parameters of the target network.

$$L(\theta) = \mathbb{E}\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta') - Q(s, a; \theta)\right)^2\right] \quad \textbf{Equation 2-6} \text{ [51]}$$

2.4.3.2 Experience Replay and Target network

Experience replay and target network are two core techniques of the DQN algorithm, which can efficiently stabilize learning and improve performance [51].

Experience replay [55] is used to train an agent with the transitions, which are randomly sampled from a reply buffer of its previous experienced transitions. The transitions collected from the environment are stored in a fixed-size replay memory. A transition is in the form of a quadruple (s, a, r, s'), where s is the current state; a is the action; r is the reward value after executing the action a in the state s, and s' is the next state. At each time step, the current transition is added to the replay memory, and a mini-batch of the transition samples are sampled uniformly at random from the memory and used to train the parameters of neural networks [56]. The uniform sampling approach gives equal importance to all stored experiences.

The size of the replay memory is one of the hyperparameters in the experience replay technique in DQN. The memory always overwrites with the recent transitions due to fixed size. The authors in [57] state that the replay buffer size is an important task-dependent hyperparameter. The empirical results demonstrated that the agent is sensitive to the replay memory size in the complex learning system. The optimal tuned memory size can improve data efficiency and stabilize the training of a neural network. They applied a DNN with rectified linear units (ReLU) activation function for the Buffer-Q agent, which DQN can

be considered as. The empirical results demonstrated that the Buffer-Q agent with the neural network as a function approximator, learns faster with the medium replay buffer size, and fails to find the optimal solution with an extremely large replay buffer. For Buffer-Q agent, the optimal buffer size for Lunar Lander task is 10^3, but for Grid World task, the optimal buffer size is 10^4.

The experience replay technique provides several advantages. First of all, the stored experience samples can be reused multiple times, which allows the agent to learn efficiently from its experience; Secondly, it can reduce the variance and the correlation between the samples. Thirdly, it improves the stability of the network during training, and improves the sampling efficiency [56].

The target network is used to determine target Q-value, and it is structurally the same as the primary Q-network. The parameters θ' of the target network are fixed, and periodically updated from the primary Q-network ($\theta' = \theta$) every certain number of time steps, which is intended to stabilize the learning process. The target value can be achieved from Equation 2-7, where γ is the discount factor; s' is the next state; a' is the action taken to maximize the Q-value in state s'; θ' are the parameters of weights in the target network.

$$Y^{DQN} = r + \gamma \max_{a'} Q(s', a'; \theta') \qquad \textbf{Equation 2-7}$$

2.4.4 Multi-Agent Deep Reinforcement Learning

In recent years, researchers have attempted to extend the single-agent RL algorithms to multi-agent approaches. A large number of real-world problems are complex, so it cannot be solved by a single agent that interacts with the environment. Specifically, a cellular communication network is a complex system, which is a multi-agent environment by nature, and where the multi-agent system (MAS) is needed. The MAS deals with behavior

management of several independent agents, where the individual agent has its own interest and goal [58].

There are several advantages of MAS. First of all, MAS is scalable. The modularity of MAS leads to simpler programming. It is easier to add new agents to a multi-agent system compared with a monolithic system. Secondly, MAS is cost effective. The individual agent takes the partial view of the environment, implements the subtask, which reduces the complexity and computational resources compared with centralized approaches [58]. Thirdly, the redundant agents can be deployed in the system, which can tolerate failures, providing the system robustness.

MARL is a group of agents that interact with the operating environment and interact with each other to achieve the goals [10]. MADRL extends the functions of RL and MARL with deep learning [10].

A simple approach to extend single-agent RL algorithms to multi-agent algorithms is to consider each agent as an independent learner, which is called independent Q-learning (IQL) [59]. The independent learners are non-communicative agents; they cannot observe the rewards and actions of all the other agents in the MAS [60]. However, since there are no inter-communications between the independent agents, which can reduce the signaling and overhead. The biggest challenge for the independent learner is non-stationarity, as the other agents' actions toward local interests impact the environment transitions. However, the comprehensive empirical studies [60] on independent learning have shown that IQL often works well in practical applications.

This thesis proposes a multi-agent DQN-assisted PHO management solution in a multiple UEs scenario, where a single agent controlled by DQN algorithm, takes action of

T-BS selection for a UE to establish a pre-connection. However, this specific multi-agent scenario is simple, there is neither cooperative nor competitive relationship among the UEs, and no knowledge exchanged between each other. All the agents are independent and act in a distributed manner in the networks, the individual agent is only aware of its local action and reward, without any inter-agent communications.

2.5 Research Tools

2.5.1 Network Simulator 3

The simulation tool used in the research is NS-3 [61], [62]. NS-3 is a discrete event network simulator for both IP and non-IP based networks in research and education. NS-3 provides models for LTE and 5G NR networks. NS-3 is an open-source tool licensed under GNU General Public License (GPL) v2 and is primarily used on Linux or macOS systems. The tool is designed as a set of software libraries that can be combined together and it also has the ability to link with external libraries. NS-3 is written in C++; however, users can work at the command line with C++ and/or Python software development tools.

The tracing function is one of the important mechanisms in NS-3, which can be used to gather feature information or statistical data from the network. It uses a callback-based framework of decoupling the trace sources from the trace sinks [61] and a uniform mechanism to connect the sources to sinks. Trace source entities can generate events in a simulation and provide access to underlying data. Trace sink entities are the consumers of the trace information. The users can add customized traces as needed.

2.5.2 NS3-Gym vs. NS3-AI

As ML-based applications have attracted more interest in the research community for various domains, a variety of academic tools have been developed and released, which allows researchers to work on RL-related research. OpenAI Gym [63] is a toolkit that can be used for developing and comparing RL algorithms. OpenAI Gym is an open-source Python library for developing and comparing RL algorithms by providing a set of standard application programming interfaces (APIs) to communicate between RL algorithms and environments. It can be in conjunction with numerical computation libraries, such as TensorFlow and Keras, which are used to build neural networks in this research.

NS3-Gym [64], [65] is the first toolkit for RL in networking research. It is open source under a GPL. NS3-Gym integrates OpenAI Gym into the NS-3 network simulator. Specifically, NS3-Gym simplifies feeding the RL models with the data generated in the simulator. Figure 2.9 [64] demonstrates the NS3-Gym framework.

- NS-3 provides a simulation scenario serving as an environment for an RL agent.
- OpenAI Gym framework provides a set of APIs used to access the state and execute actions in an RL environment.
- The inter-process communication (IPC) component between the environment-independent agent and the environment is based on serialized messages via ZeroMQ (ZMQ) [66] socket using the Protocol Buffers library.

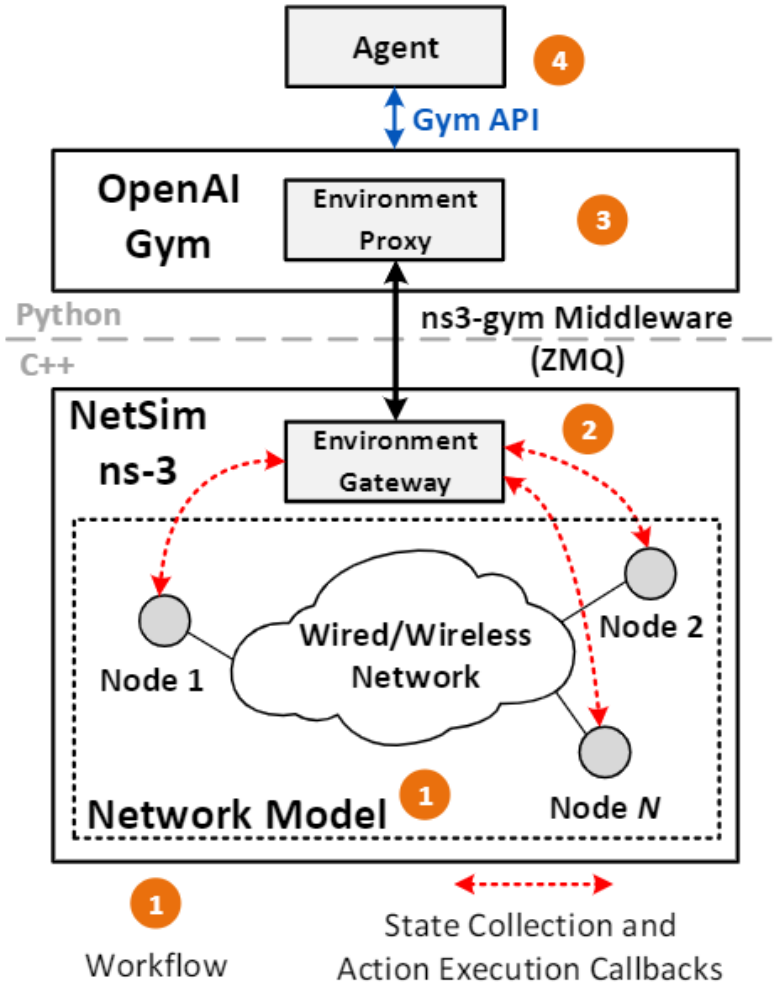


Figure 2.9 Architecture of NS3-Gym Framework [64]

NS3-AI [67] is another toolkit, which supports the integration of Python-based RL frameworks into NS-3 by using a shared memory space for communication between processes. The NS3-AI framework is shown in Figure 2.10 [67]. The authors in [67] state that the shared memory technique that allows NS3-AI to achieve a transmission 100 times faster than the ZMQ socket used in NS3-Gym on a benchmark example.

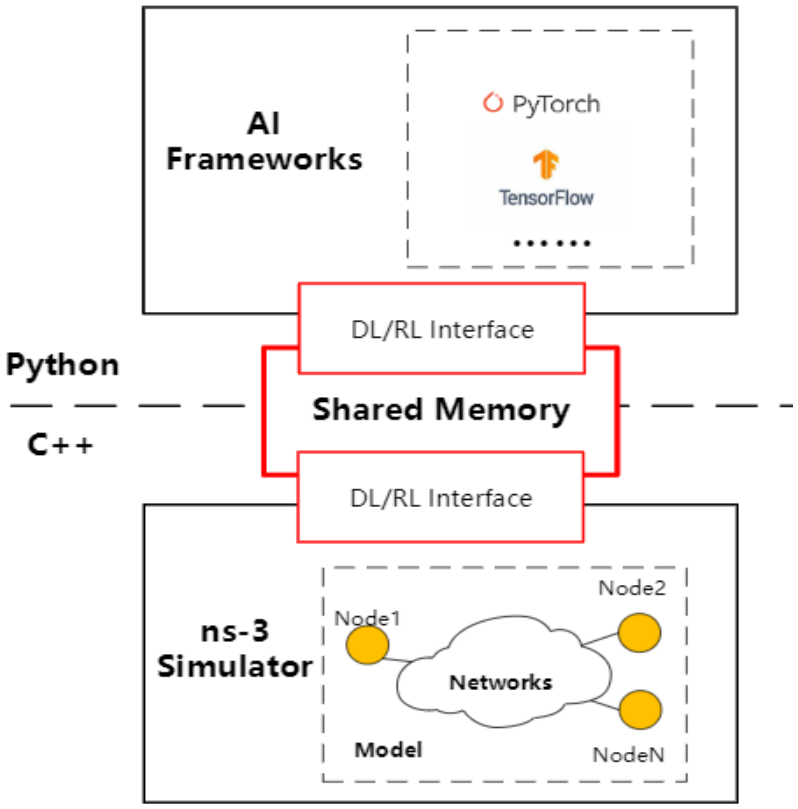


Figure 2.10 Architecture of NS3-AI Framework [67]

Even though the advantage of NS3-AI is clear. However, the shared memory has the disadvantages of data synchronization problems, limited scalability by the memory, and challenges in data management. In contrast, the Protocol Buffers library used in NS3-Gym ensures reliable and stable data transmission and can improve system robustness. In addition, from the user's perspective, the inherited Gym APIs in NS3-Gym are intuitive and practical.

In this thesis, a DQN-assisted PHO management solution is simulated and evaluated with NS-3 and NS3-Gym. More detailed implementations are introduced in Chapter 5.

Chapter 3: Review of Machine Learning for Handover Management

This chapter presents a literature review of ML-based handover management techniques. Section 3.1 gives a general description of ML for handover management. Section 3.2 depicts RL-based and DRL-based handover management solutions. Section 3.3 presents two MADRL applications for handover management. Section 3.4 presents a summary.

3.1 General Machine Learning for Handover Management

ML techniques have been applied to solving various challenges in different domains. Many research efforts on applying the ML approaches to handover management have been proposed in the literature. With the high-speed deployment requirement in 5G networks defined by 3GPP [13], the passengers in high-speed railway and vehicles demand high QoS with wireless connections.

The mobility pattern prediction of UEs brings in certain regularity in ML-based handover management. The proposed method in [68] predicted the UE movement patterns from the historical trajectories in dense networks. A Long Short-Term Memory (LSTM) network, fed by the position coordinates of each UE, was implemented to predict the UE's mobility trends. Dual connectivity was also applied to decrease interruptions caused by the hard handover. The input dataset includes the mobility patterns which capture all the UEs' two-position coordinates for 12 hours at one-minute granularity each day. However, the simulations are performed by the low-speed UEs with maximum speed of 8 m/s (28.8km/h). Moreover, the handover prediction accuracy falls to 60% when the user speed is 8 m/s. Therefore, this method is not suitable for high-speed UE scenarios.

Similarly, the authors in [69] proposed a method for predicting UEs' movement patterns from the historical trajectories to achieve mobility management in mm-wave

vehicular networks. Firstly, the channel state information (CSI) was used as the input of the Kernel-based ML algorithm for prediction of the vehicles' positions. Then, a historical handover dataset leveraging the KNN algorithm was used for handover decision-making for the UEs with the speed of 30 m/s (108 km/h). The soft handover scheme was implemented to decrease the transmission interruptions during the handover.

Some applications built the prediction model with the key objective of future signal quality conditions. A LSTM-assisted handover management scheme is proposed in [70] for RSSI prediction, which was used in handover decision-making. The historical RSSI values were collected by the vehicles. The authors in [71] adopted the RSRP values as the inputs to classify handover events either in success or failure. Firstly, the prediction of RSRP of the candidate cells was implemented by using LSTM and Recurrent Neural Network (RNN). After that, a supervised KNN classification method named neighborhood component analysis was applied for handover prediction.

All the above solutions are based on supervised learning with the following limitations. First of all, a historical dataset is required as the input, somehow, the size of the dataset influences the prediction accuracy and stability. Secondly, supervised learning cannot deal with the raw data, therefore, the data pre-processing is needed for the missing and nominal data to improve the prediction accuracy. Thirdly, most of the supervised learning approaches are not suitable for the complex system, in which the high-dimensional data may change dynamically, and not easily be collected.

3.2 Reinforcement Learning on Handover Management

Compared with supervised learning, RL has the advantage of flexibility, because it does not require a given model and can deal with the dynamic environment without the pre-

collected dataset as the inputs. Furthermore, by taking into account the power of the deep learning algorithms, the DRL is suitable for complex sequential decision-making problems. In the literature, a variety of works have been proposed to apply RL on handover management. There are two considerations for optimization in handover decision-making problems: T-BS selection and handover trigger time.

The authors in [72] proposed a Q-learning based approach to optimize the trigger time for the predictive handover with the information of pedestrian-UE's location and moving velocity as the inputs. In the proposed framework, the agent learnt the optimal handover policy by maximizing the future throughput. The RSSI value was mapped into the data rate in the reward function. Although the velocity was considered as input, it was a fixed value of 1 m/s in their experiments.

The authors in [73] proposed a Q-Learning based handover mechanism for optimal T-BS selection in a cellular drone-UEs system. The proposed frameworks aimed at a balance between maximizing the RSRP values and minimizing the number of handovers. By leveraging the RSRP values of all the surrounding BSs and the UE's trajectory information, including UE's position and moving direction, to provide an effective handover policy. Specifically, the state of the environment consisted of the drone's position, moving direction, and the current S-BS. The action is the T-BS selection. The reward function is a weighted combination of S-BS's RSRP in the future and the handover cost. Their empirical results showed that the proposed approach can significantly reduce (e.g., by 80%) the number of handovers.

Both applications conducted the Q-learning algorithm for decision making. Q-learning is impractical in large and complex networks. Firstly, Q-learning lacks scalability,

because it only works in the environment with discrete, finite, and low-dimensional state and action space. Secondly, the known over-estimation issue with Q-learning influences the learning performance. Additionally, the UEs' trajectory information was known for both of the solutions. However, it is challenging or impractical to dynamically collect all the UEs' locations and velocities information in reality.

Many of the successes in DRL, which is powered by the function approximation properties of neural networks, have been achieved in high-dimensional and complex problems. The authors in [74] proposed a DRL assisted proactive handover decision solution to optimizing handover trigger time by using time-consecutive camera images of the UEs as the input. The camera images were mapped to action values. Specifically, the proposed DRL approach was used to predict the cumulative sum of the future data rates, then decision-making was based on the predicted values. However, the solution depends on the resolution of the camera and external conditions to adequately capture the images.

The authors in [75] applied double deep reinforcement learning for T-BS selection to minimize the frequency of handovers in mm-wave networks, and subsequently to maximize the system throughput. The BS selection was based on the UE that received the signal-to-noise ratio (SNR) values from all surrounding BSs. An offline learning framework was proposed to alleviate the negative impact of online learning in terms of computational costs. The UE's velocity was configured as 8 m/s (28.8 km/h) in their simulations.

3.3 Multi-Agent Deep Reinforcement Learning on Handover Management

MADRL-based techniques for handover management provides the ability to adapt a UE association reinforcement learning scenario to a multiple UEs circumstance. MADRL has

the advantages of scalability and flexibility, because in most multi-agent systems, a new agent can be easily added into or removed from the system [16].

The authors in [44] utilized DQN to optimize the handover process to reach a balance between the handover rate and the system throughput. In this approach, DRL was used to make handover decisions with the input of RSRQ measurements from UEs. Firstly, all the UEs were clustered based on their mobility patterns with K-Means Clustering, which is one popular unsupervised learning algorithm. Then, the multi-user handover process in each cluster was optimized in a distributed manner by using the Asynchronous Advantage Actor-Critic (A3C) RL framework. Then DNN was applied to approximate the Q-function and to generate the policy in A3C. The reward was defined as the weighted sum of the average handover rate and throughput. The state consisted of the UE's received RSRQ values of all the BSs, and the index of the current S-BS.

A MADRL framework for distributed handover management called RHando was presented in [76]. In their proposed scheme, each UE was modeled as an agent. Each DQN controlled agent aimed at learning an optimal policy to perform handover in order to maximize the network throughput. The fully distributed solution reduced the signaling and computation overhead. The state space included the information of UE's velocity, Received Signal Strength (RSS) values of the surrounding BSs, previous data rate, and network sum-rate. The UEs learned how to perform association requests that limit handovers and avoid collisions across service requests. The collisions occurred when the number of UEs requesting a handover within a given BS was greater than the capacity of the connections at the BS.

3.4 Analysis of Existing ML-Assisted Handover Management Schemes

The previous sections present the existing ML approaches to handover management. As described, each of those approaches has some strengths and limitations.

The supervised learning based handover management solutions are summarized in Table 3.1, The MBB handover scheme is applied in [69] [70], which decreases the transmission interruption during the handover, therefore, subsequently improves the QoS. Research methods in [70]–[72] support the high-speed scenario, which is one of the deployment requirements in 5G networks. However, all the above solutions are based on supervised learning with some limitations, which are highlighted in Section 3.1.

Table 3.1 Summary of Supervised Learning-Based Approaches to Handover Management

Related Works	ML Method		Input	Historical Dataset Needed	Multiple UEs Scenario	High Speed Scenario	MBB Handover
[68]	Supervised Learning	LSTM	UEs' trajectories	Yes	Yes	No	Yes
[69]	Supervised Learning	KNN	CSI values & handover dataset	Yes	Yes	Yes	Yes
[70]	Supervised Learning	LSTM	Dataset of RSSI values	Yes	Yes	Yes	No
[71]	Supervised Learning	RNN & LSTM	Dataset of RSRP values	Yes	Yes	Yes	No

Table 3.2 presents a summary of the aforementioned RL-based handover schemes. All the solutions have the same defined actions, which is the T-BS selection. Different reward functions depend on the different objectives. Compared with Table 3.1, the RL-based approaches have the advantages of no historical dataset needed as the input. Additionally, DRL has the power of dealing with complex and high-dimensional systems. Furthermore, MADRL-based techniques can manage the multiple UEs scenario. However,

neither [44] nor [76] takes into account the advantage of the MBB handover scheme, and supports the high-speed UE scenario and multi-connectivity between UEs and BSs.

Table 3.2 Summary of RL-Based Approaches to Handover Management

Related Works	ML Method	Multiple UEs Scenario	High Speed Scenario	MBB Handover	Action	State	Reward
[72]	RL	No	No	No	T-BS selection	UE's position & velocity, S-BS index	UE's throughput
[73]	RL	No	No	No	T-BS selection	UE's position & moving direction, S-BS index	Handover cost and the S-BS's RSRP
[74]	DRL	No	No	No	T-BS selection	time-consecutive received power values	BS's data rate
[75]	DRL	No	No	No	T-BS selection	SNR values, S-BS index	System throughput and the number of handovers
[44]	MADRL	Yes	No	No	T-BS selection	RSRQ values, S-BS index	Average DL throughput and the number of handovers
[76]	MADRL	Yes	No	No	T-BS selection	RSS values, UE's data rate, network sum-rate	Network throughput and the number of handovers

This thesis proposes a DQN-based enhanced handover technique by devising the MBB scheme and multiple T-BSs pre-connection. MADRL approach is conducted for T-BS selection in multiple UEs handover management scenarios by taking into account that MADRL has the advantages of higher scalability and more flexibility. Additionally, the proposed solutions are evaluated in the highway scenario to meet the 5G deployment requirements [11].

Chapter 4: System Model and Design

This research focuses on a solution that fulfills the following requirements:

1. Design of an enhanced handover mechanism, namely PHO, which considers the following functionalities:
    a) Support for the MBB handover scheme to reduce HIT.
    b) Support for pre-connections to multiple T-BSs to ensure QoS.
    c) Support for pre-connections to candidate T-BS(s) in advance before handover triggered.
    d) Design of early DL data duplicating-forwarding-buffering mechanism to reduce packet loss during the handover process.
2. Application of the DQN-assisted UE-associated PHO solution for optimized T-BS selection to maximize the PHO success rate.
3. Development of a MADRL-based PHO management solution to handle the multiple UEs scenario.
4. Support for the high-speed UE mobility scenario.
5. Support for offline learning and online prediction.

The rest of the chapter is as follows: Section 4.1 describes the detailed process of the proposed PHO mechanism. Section 4.2 discusses DQN-assisted UE-associated PHO management. Section 4.3 presents a MADRL-based approach to PHO for a multiple UEs scenario.

4.1 Pre-connect Handover

The proposed PHO is an enhanced handover technique based on the 3GPP baseline handover process presented in Figure 2.2. During a UE's movement, the PHO can be

triggered based on all the surrounding BS's RSRQ conditions. It is a network-controlled and UE-assisted handover solution. The detailed process is depicted in Figure 4.1.

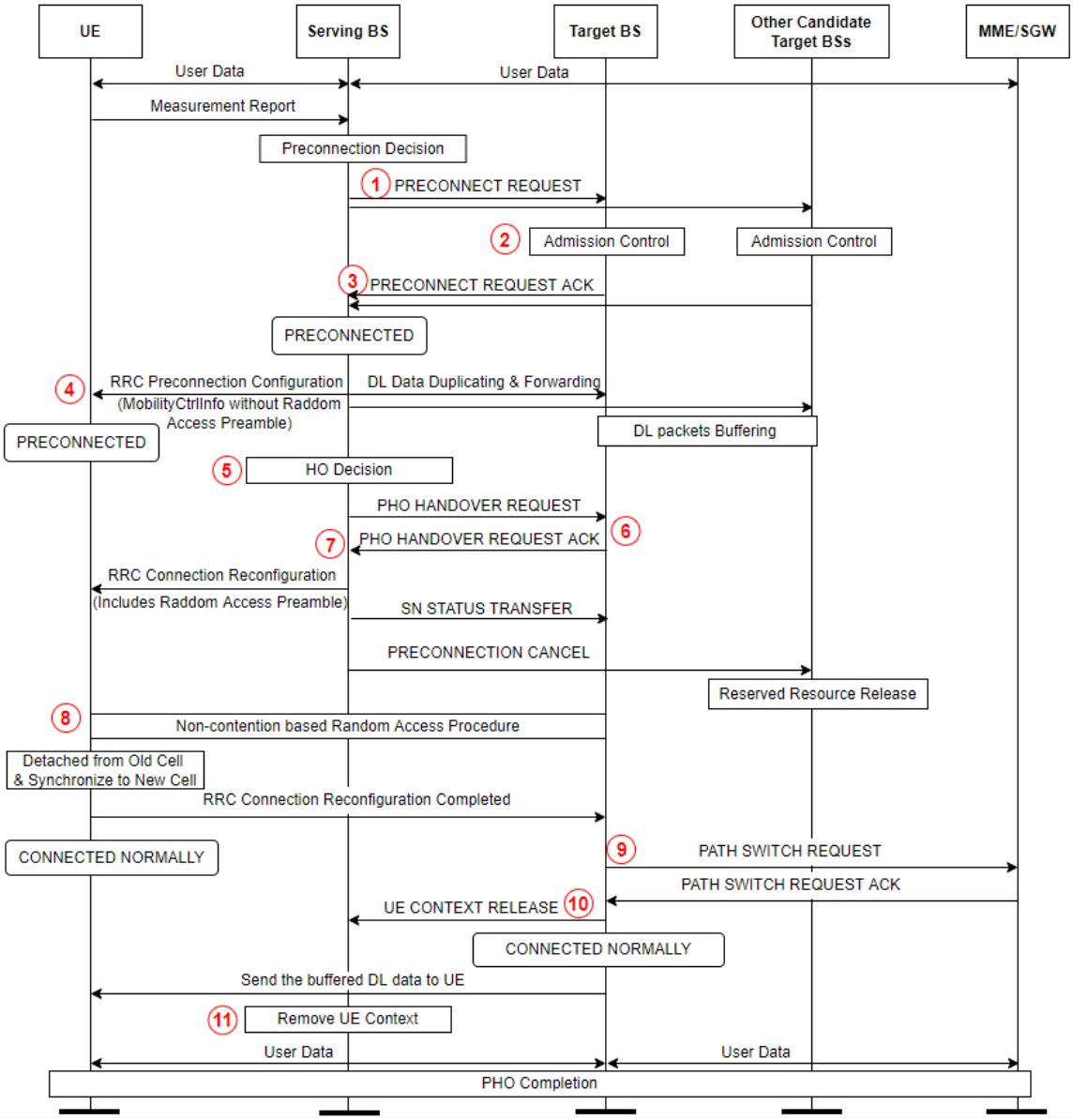


Figure 4.1 Pre-connect Handover Process

The main steps are detailed as follows:

Step 1. The S-BS initiates the process by sending a Preconnect Request message to one or more T-BSs through the X2 interface.

Step 2. Upon receiving the Preconnect Request message from S-BS, T-BS can either accept or reject the request under its own admission control. If the request is accepted, the T-BS takes the following actions:

1) Allocate a new Radio Network Temporary Identifier (RNTI) for UE. The RNTI is a 16-bit number, it can be considered as a UE identifier for traffic between the UE and the lower layer of a BS [20].
2) Map the International Mobile Subscriber Identity (IMSI) to RNTI. An IMSI is a unique 15-digit number provisioned in the Subscriber Identity Module (SIM) card, and used by mobile network operators to identify individual subscribers in a cellular network [77].
3) Reserve a data radio bearer. The data radio bearer is the service provided by Layer 2 for data transmissions between a UE and RAN [78].
4) Prepare a handover command message with mobility control information, including target cell identifier, RNTI, carrier DL and UL frequency, and carrier DL and UL bandwidth. The target cell identifier is a unique number used to identify each BS or a sector of BS.
5) Encode the handover command message as RRC context and send it within the Preconnect Request Acknowledge message to S-BS via the X2 interface.

Step 3. Once receiving the Preconnect Request Acknowledge message from T-BS, the S-BS sends the handover command to UE through the RRC Preconnection Configuration message and switches the state to PRECONNECTED. After that, the S-BS starts the early DL forwarding process to the pre-connected T-BS. The received DL packets are

buffered in a capacity-adjustable queue at T-BS. The queue capacity is defined as the number of PDCP SDUs.

Step 4. The UE receives RRC Preconnection Configuration message and switches the state to PRECONNECTED, which indicates the pre-connection is established successfully. The UE holds the received handover command without taking any action.

Step 5. When the trigger condition of the handover event is satisfied, the pre-connect handover is triggered on S-BS by sending a Preconnect Handover Request message to the pre-connected T-BS. The handover events and trigger conditions are listed in Table 2.1 [34], [35].

Step 6. Upon receiving the Preconnect Handover Request message from the S-BS, the T-BS checks the availability of the pre-allocated resources for the UE. If the resources are still reserved, then the T-BS allocates a random-access preamble identifier (RAPID) [79], which is used by a UE to access the BS on the random-access channel (RACH). T-BS switches the state to PRECONNECT_HANDOVER_JOINING and sends a Preconnect Handover Request Acknowledge message including the RAPID to S-BS to accept the request.

Step 7. The S-BS receives the Preconnect Handover Request Acknowledge message, then switches the state to PRECONNECT_HANDOVER_LEAVING, and sends a RRC Connection Reconfiguration message to UE to modify an RRC connection of RBs to perform the handover. The S-BS also sends the SN Status Transfer message to the T-BS through the X2 interface to convey the UL PDCP SN receiver status and the DL PDCP SN transmitter status of the E-UTRAN Radio Access Bearer [20]. In addition, if multiple pre-connections have been established with other candidate T-BSs, then S-BS

sends a Pre-connection Cancel message to notify all other candidate T-BSs to release the reserved resources.

Step 8. After receiving the RRC Connection Reconfiguration message from S-BS, the UE extracts the RAPID, starts the random access procedure with T-BS. If the random access procedure is completed successfully, the UE sends an RRC Connection Reconfiguration Complete message to T-BS to notify of the success, and switches the state to PRECONNECT_NORMALLY. The successful outcome indicates the handover completion in RAN networks.

Step 9. Upon receiving the RRC Connection Reconfiguration Complete message from the UE, T-BS sends a Path Switch Request message through the S1 interface to inform MME that UE has switched the connection to T-BS, and request the path switch on the CN.

Step 10. The T-BS receives the Path Switch Request Acknowledge message, which means the data plane has been switched to T-BS by the CN. The T-BS starts sending the buffered DL PDCP SDUs to the UE, and informs the successful handover to the S-BS by sending the UE Context Release message.

Step 11. Finally, upon receiving the UE Context Release message, the S-BS releases the radio and control plane resources associated with the UE.

4.2 DQN-Assisted PHO Management

Inspired by the huge success of DRL in resolving complicated control problems, this research designs a DQN-assisted PHO management solution. Each UE-associated agent interacts with the environment to learn the optimized policy for T-BS(s) selection with the

goal of maximizing the PHO-SR. PHO-SR is defined as Equation 4-1, where the total number of triggered handovers includes PHOs and 3GPP baseline handovers.

$$\text{PHO-SR} = \frac{\text{Number of Successful PHOs}}{\text{Total number of triggered handovers}} \qquad \textbf{Equation 4-1}$$

A system level architecture of DQN-assisted PHO management is depicted in Figure 4.2, which contains two main components: environment and DQN model.

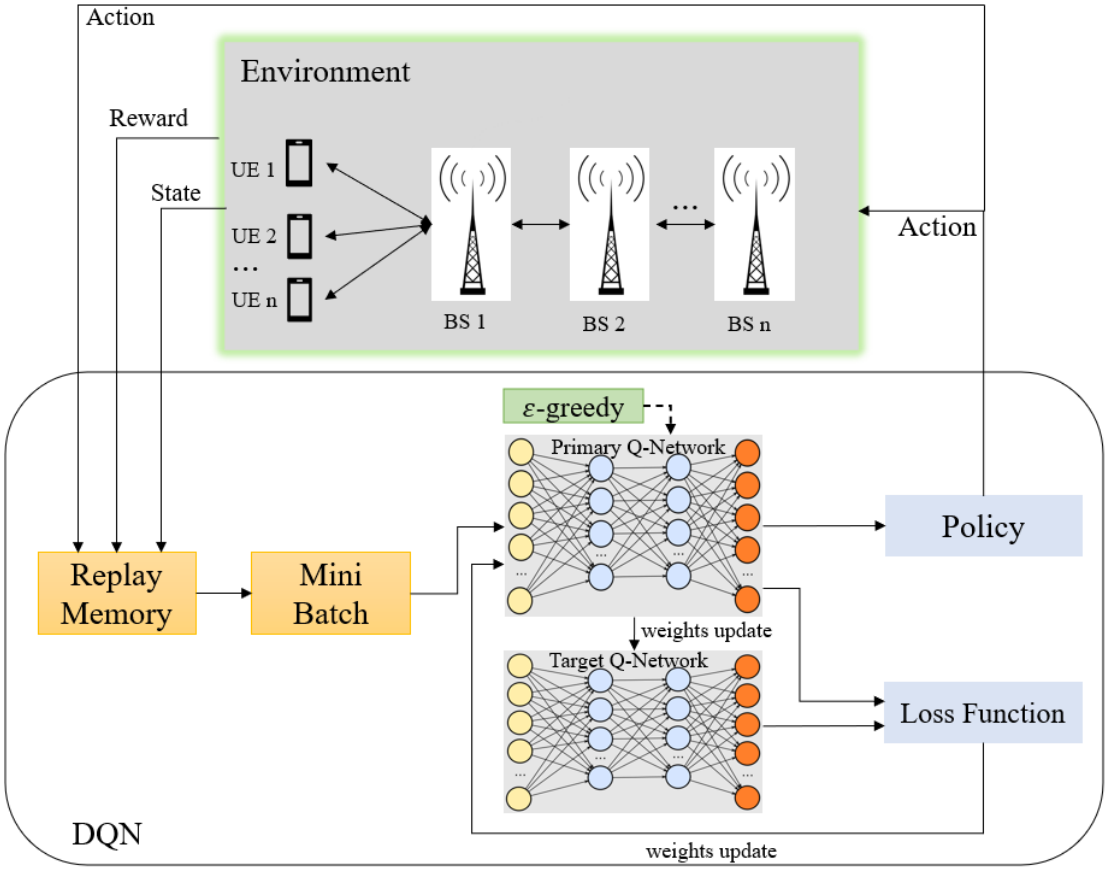


Figure 4.2 System Architecture for DQN-Assisted PHO Management

- Section 4.2.1 details the customized environment for PHO management, including the state, action, and reward.
- Section 4.2.2 presents the proposed algorithm of the DQN-assisted PHO mechanism.

4.2.1 PHO Management Environment

As shown in Figure 2.5, the environment in RL consists of three main components, which are state space (S), action space (A) and reward (R).

4.2.1.1 State Space

The state space S is a set of all the states that an agent can transit to. To learn the optimal strategy, the agent continuously collects information from the environment. Because it is impractical in reality to dynamically collect and manage UEs' geographical locations and speeds information, the proposed DQN-assisted PHO solution correlates a UE's relative location and speed to the RSRQs and RSRQ change rates of all surrounding BSs.

The state vector is defined as Equation 4-2, where S_t is the state at time step t, n is the number of BSs in the system, RSRQ_n is the UE measured RSRQ value of BS_n.

$$S_t = \{\{\text{UE's QCI}, \text{servingBS}\}, \{\text{RSRQ}_1, \dots \text{RSRQ}_n\}, \{\text{RSRQ Change Rate}_1, \dots \text{RSRQ Change Rate}_n\}\} \quad \textbf{Equation 4-2}$$

- The UE's QCI value and S-BS information are observed directly from the environment.
- The RSRQ values of all the surrounding BSs represent a UE's relative location, which are collected at each time step t.
- The RSRQ change rate values can be calculated by Equation 4-3. The time step t is a fixed value in the environment.

$$\text{RSRQ Change Rate} = \frac{\text{RSRQ}_{s_{t+1}} - \text{RSRQ}_{s_t}}{\text{time step t}} \quad \textbf{Equation 4-3}$$

4.2.1.2 Action Space

Action space A is a set of all possible actions the agent can take in a certain environment. At every time step t, the agent is allowed to choose an action a from the action space A,

where $a \in A$. In the proposed DQN-assisted PHO solution, the action space comprises all the BSs in the system, including the S-BS and the candidate T-BSs. An action a is defined as the selected T-BS for the pre-connection before the A2-A4-RSRQ handover event occurs. The one-hot encoding technique [80] is used to represent the BS selection. One-hot encoding is a representation of integer variables as a binary vector. For example, if the system supports up to six BSs for the handover decision making, and BS3 is selected, then the one-hot vector is expressed as $[0,0,1,0,0,0]$.

4.2.1.3 Reward

The reward function is used to obtain an optimized policy for the agent to take an action by maximizing the cumulative reward values in the long run. This proposed solution aims at maximizing the PHO-SR. In a dynamic network, the handover result can only be obtained once it occurs. Hence, we do not know what delay to expect after taking an action, which results in a delay in observation. To avoid the delayed reward [81], an immediate award is defined by associating the award with the RSRQ values. The reward function is defined as Equation 4-4, which rewards the selected BS with the maximum RSRQ and the RSRQ change rate for a UE, while penalizing all the other conditions.

$$\text{Reward} = \begin{cases} 2 & \text{selected BS has maximum RSRQ and maximum RSRQ change rate} \\ 1 & \text{RSRQ of selected BS } \geq \text{ RSRQ of serving BS} \\ -1 & \text{otherwise} \end{cases} \qquad \textbf{Equation 4-4}$$

An example of topology is given in Figure 4.3. A UE is originally attached to BS1, and moves from BS1 to BS6, with a constant moving speed of 110 km/h. The location vector of UE and each BS is represented in the format of x, y, and z coordinates. The distance between each two neighboring BSs is 400 meters. During a UE's movement, the

handovers sequentially occur in the following order: BS1-> BS2, BS2-> BS3, BS3-> BS4, BS4-> BS4, and BS5->BS6.

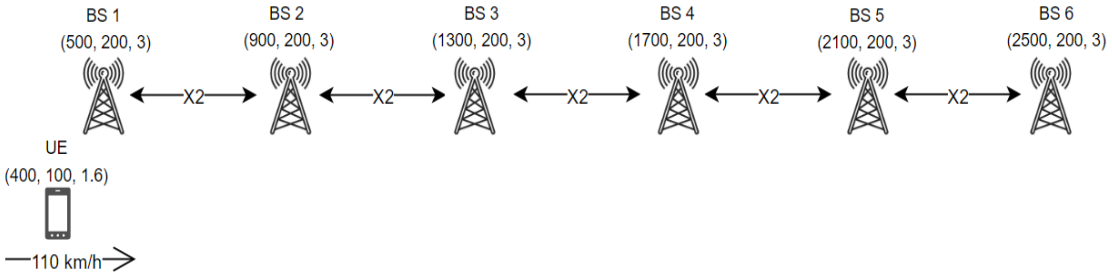


Figure 4.3 Example Topology Adopted in Explanations of PHO Environment

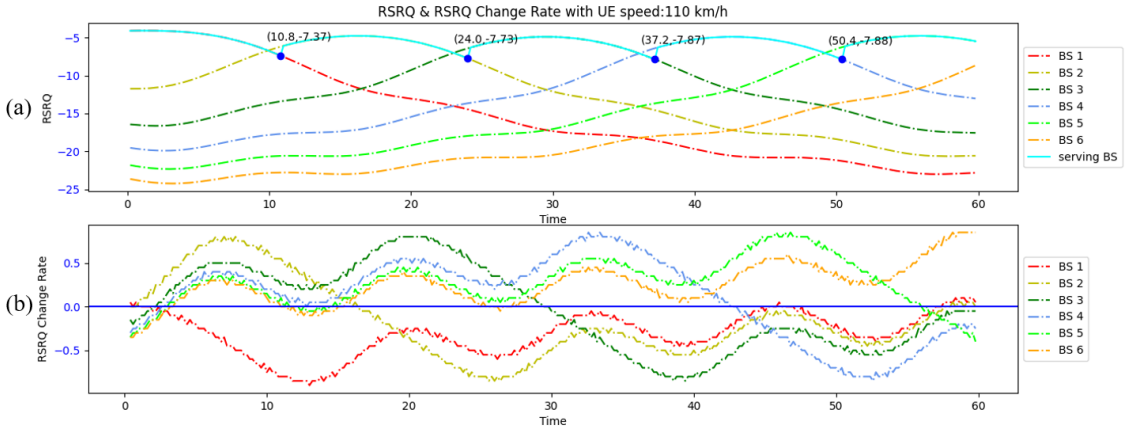


Figure 4.4 Plots of RSRQ and RSRQ Change Rate

The UE measured RSRQ values are periodically collected from the NS-3 simulator, which is illustrated in Figure 4.4 (a). The values of the RSRQ change rate are calculated by Equation 4-3, as shown in Figure 4.4 (b). The blue dots indicate that the 3GPP baseline handover occurs during the UE's trajectory, which are at 10.8s, 24.0s, 37.2s and 50.4s. It can be seen that, before the first handover is triggered at 10.8s, UE departs from BS1 and approaches BS2. BS1 has the highest RSRQ value, both RSRQ value and RSRQ change rate decrease; in contrast, BS2 has the highest RSRQ change rate, both RSRQ value and RSRQ change rate increase. Therefore, BS2 is the best T-BS for handover. Similarly, the T-BSs for the sequential handovers are BS3, BS4, and BS5.

By applying the proposed DQN-assisted PHO solution to the same scenario, the T-BS selection and PHO trigger time can be obtained as shown in Figure 4.5. The comparisons are summarized in Table 4.1. The PHO is triggered earlier than the 3GPP baseline handover.

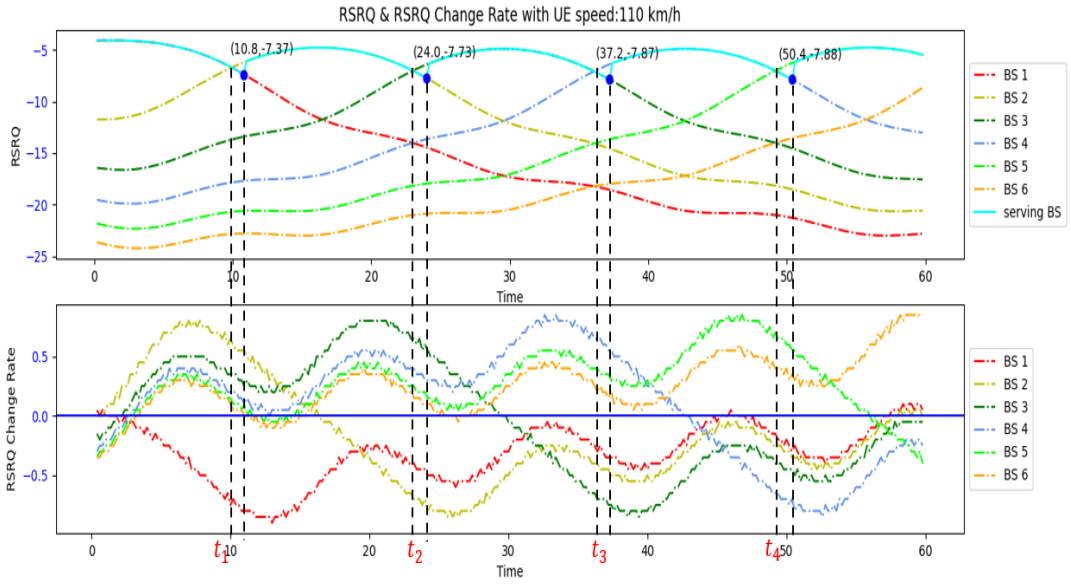


Figure 4.5 T-BS Selections in DQN-Assisted PHO

Table 4.1 Handover Trigger Time - 3GPP Baseline Handover vs. PHO

Handover Occurrence	3GPP Baseline Handover Trigger Time (s)	S-BS Index	T-BS Index		PHO Trigger Time	Optimized T-BS Index Selected by Agent
1^{st}	10.8	1	2		$[t_1, 10.8]$	2
2^{nd}	24.0	2	3		$[t_2, 24.0]$	3
3^{rd}	37.2	3	4		$[t_3, 37.2]$	4
4^{th}	50.4	4	5		$[t_4, 50.4]$	5

4.2.2 DQN-Assisted PHO Management

4.2.2.1 Deep Neural Network Model

As illustrated in Figure 2.7 [6] (c), a DNN is the brain of an RL agent. DNN is typically a neural network with two or more hidden layers [6]. Each layer contains nodes that are fully

connected to the next layer. The nodes contain a nonlinear activation function. The activation functions in neural networks are significant, as they can help in the learning by making a non-linear and complicated mapping between the inputs and outputs [82]. The prediction accuracy of DNN depends in part on the number of layers and the type of activation function.

ReLU is one of the most widely used activation functions in hidden layers [82]. However, ReLU has a potential disadvantage during optimization, known as the Dying ReLU issue [83]. As shown in Figure 4.6 [84] (a), the gradient is 0 when the unit deals with the negative inputs, which could lead to the problem that the gradient-based optimization algorithm will not adjust the weights of a unit that never activates initially.

The Leaky ReLU activation function is a solution to the Dying ReLU problem [85]. As is shown in Figure 4.6 [84] (b), Leaky ReLU assigns a non-zero slope to the negative inputs, which enables the negative part of feature information to be retained. The parameter α was assigned as 0.01 in the original paper [85].

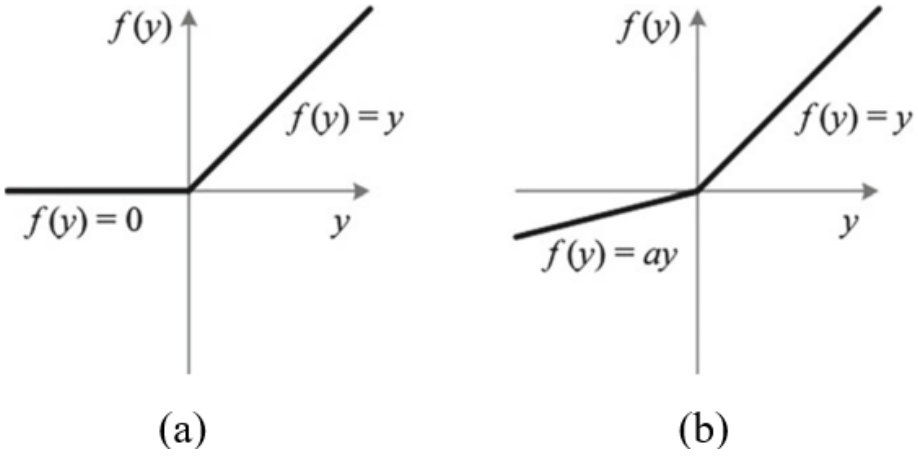


Figure 4.6 Comparison between (a) ReLU and (b)Leaky ReLU [84]

The Softmax activation function is used to compute the probability distribution from a real numbers vector, and it is commonly used in the output layers [86]. The output values of Softmax are in the range of 0 to 1, and the sum of all the outputs is 1. The Softmax function is also used in the case of RL output probabilities related to different actions to be taken, which is suitable for the BS selection in the proposed PHO mechanism.

The optimizer used in the DNN model is Adam. Adam is a first-order gradient-based stochastic optimization algorithm, with the advantages of computationally efficient and little memory requirement [87].

A four-layer DNN model applied in the DQN-assisted PHO is summarized in Figure 4.7. The figure depicts all the layers, input shape, output shape and number of parameters. The parameters are used in the inter-connections of the DNN, and are updated by gradient descent during training.

- Input Layer: The input shape (None, 2, 6) can be interpreted to (any batch size, 2: two input features {RSRQs, RSRQ change rates}, 6: the number of candidate BSs). Defining a size parameter as none means the size is not predetermined.
- Hidden Layer: The Leaky ReLU activation function is used. Because the RSRQs are negative values, typically in the range of -19 to -3dB [22]. While the RSRQ change rates can either be positive or negative.
- Flatten layer: It is used to reshape the two-dimensional data to the one-dimensional array for inputting it to the output layer. The Flatten layer is required because the output shape of the hidden layer is (None, 2, 32), but the output layer requires a single dimensional input.

- Output layer: The output shape (None, 6) means (any batch size, 6: the number of candidate BSs) The Softmax is used as the activation function, with the output values as the probabilities of all the six BSs, and the BS with the highest probability value will be selected as the action.

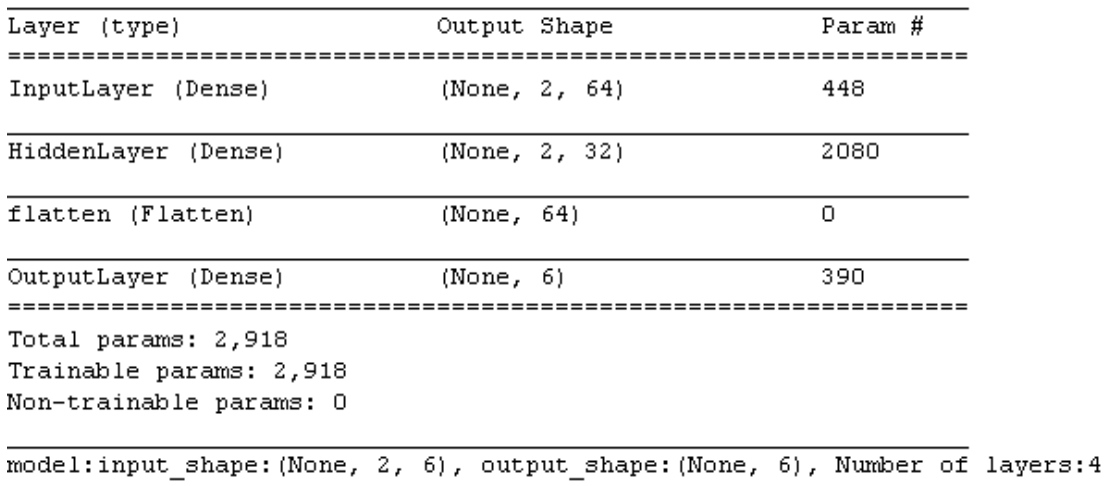


Figure 4.7 DNN Model Applied in DQN-Assisted PHO

4.2.2.2 DQN-Assisted UE-Associated PHO Algorithm

The algorithm of the DQN-assisted UE-associated PHO mechanism is presented as Algorithm 1. All the parameters are listed in Table 4.2.

Table 4.2 Parameters used in DQN-Assisted UE-Associated PHO Algorithm

Parameter's Name	Description	Value
$N_{episode}$	the number of episodes for training	2,000
T	maximum number of time steps in one episode	60
M_{ER}	size of experience replay memory	50,000
$minM_{ER}$	minimum size of experience replay memory to start training	100
minibatch	minibatch size for training	64
C	the number of step period used to update target network	3
ϵ	initialized exploration rate	1
ϵ_{decay}	ϵ decay rate	0.999
ϵ_{min}	minimum epsilon value	0.001
γ	discount factor	0.99

Algorithm 1: DQN-Assisted UE-Associated PHO (adapted from [51]**)**

Input: UE dynamically measured surrounding BSs' RSRQs and calculated RSRQ change rates

Output: The offline trained model file, which is UE-associated.

Parameters: $N_{episode}$, M_{ER}, $minM_{ER}$, minibatch, ϵ_{min}, ϵ_{decay}, γ, T, C

Initialize Experience Replay Memory M_{ER},
Initialize Primary Q-network (as shown in Figure 4.7) with random weights θ
Initialize Target Q-network (as shown in Figure 4.7) with weights $\theta' = \theta$
Initialize $\epsilon = 1$ for ϵ-greedy policy
for episode = 1 to $N_{episode}$ do
  Receive initial state by resetting the customized PHO environment
  for time step t = 1 to T do
    Following ϵ-greedy policy, select action a, which represents the index of T-BS for PHO. k is the total number of possible actions (candidate BSs).

$$a = \begin{cases} \text{a random action} & \text{with probability } \frac{\epsilon}{k} \\ \text{argmax } Q^*(s, a;\ \theta) & \text{otherwise} \end{cases}$$

    Execute action a to observe the reward r the next state s'
    Store transition (s, a, r, s') in M_{ER}
    If buffer size >= minM_{ER}
      Sample random minibatch of transitions (s_j, a_j, r_j, s_{j+1}) from M_{ER}

$$\text{target}_{Q} = \begin{cases} r_j & \text{if episode is terminated at step j} + 1 \\ r_j + \gamma \max\limits_{a'} Q(s_{j+1}, a'; \theta') & \text{otherwise} \end{cases}$$

      Perform a gradient descent to minimize loss function:

$$L(\theta) = \left(\text{target}_{Q} - Q(s_j, a_j; \theta)\right)^2$$

      Update target parameter ($\theta' = \theta$) in every C step
    end if
  end for
  if $\epsilon > \epsilon_{min}$:
    $\epsilon *= \epsilon_{decay}$
    $\epsilon = \max(\epsilon_{min}, \epsilon)$
  end if
  Set up filter for saving the trained model
end for: // DQN-assisted UE-associated PHO offline training for optimal T-BS selection

4.2.3 Offline Learning and Online Prediction

The DRL-based algorithms sometimes are time-consuming. Offline learning uses a simulator of the environment as a cheap way to get training samples for safe and fast learning [46]. A framework of offline learning and online prediction is adopted to alleviate the negative impact of online learning in terms of computational cost and to ensure a fast and responsive prediction in the real-time system.

4.2.3.1 DQN-Based Offline Learning Framework

This section gives the detailed design of how the T-BS prediction problem is formulated into a sequential decision-making solution. Figure 4.8 demonstrates the framework of DQN offline learning.

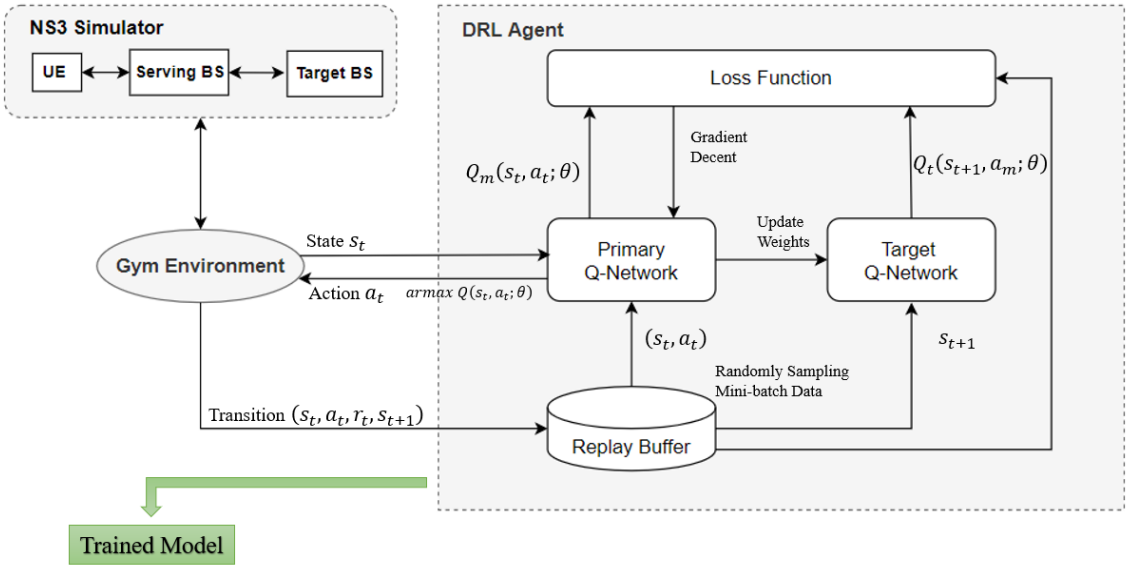


Figure 4.8 DQN Offline Learning Framework

The input of the DQN is a state vector, which is the aforementioned two-dimensional parameter: the RSRQs and RSRQ change rates. The parameters of the DQN are trained by the collected transitions (s_t, a_t, r_t, s_{t+1}) from the environment through the repeated

episodes. The loss function defined in Equation 2-6 is used to minimize the difference between the observed output from the primary Q-Network and the desired output from the target Q-Network. The back propagation computes the gradient of the loss function with respect to the weights of the primary Q-Network.

A filter is set up to save the trained model. In the proposed solution, the model trained in the last episode is saved. Alternatively, some other criteria can be used for model saving. For example: a reward threshold.

4.2.3.2 Online Prediction for Target Base Station Selection

The trained DQN model can be applied to predict the T-BS in an online manner, so that the system can make a real-time decision on pre-connection. The proposed online prediction is depicted in Figure 4.9. The input of the system is the state vector. The output of the DQN algorithm is the index of the T-BS which has the maximum Q-value among all the BSs. If the predicted T-BS is different from the S-BS, the PHO process is initialized at the S-BS as shown in Figure 4.1.

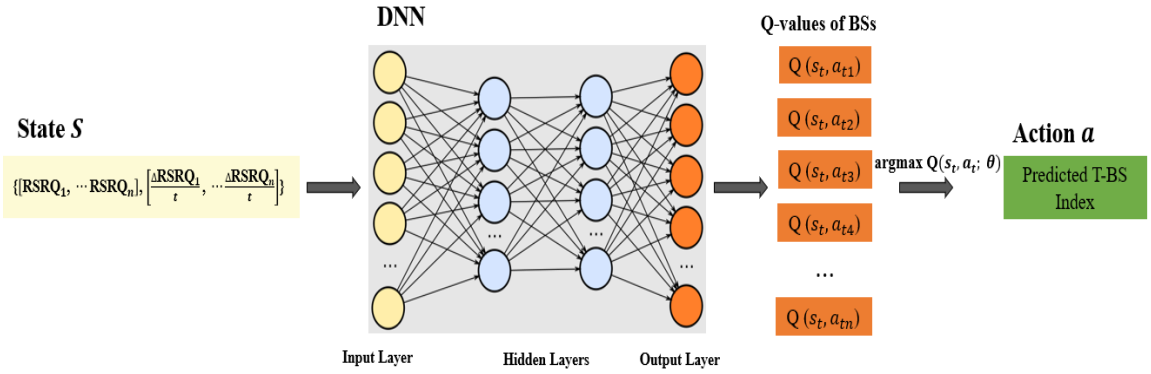


Figure 4.9 Online Prediction Using Trained DQN Model

In addition, the proposed PHO provides the ability for multiple pre-connections to more than one candidate T-BS simultaneously. This is designed to further improve the

handover reliability and ensure QoS. The number of T-BSs to pre-connect for a UE can be deduced from the QCI priority level. As introduced in Section 2.1, the lower QCI priority level service can have multiple pre-connected T-BSs. For example, UE's service with QCI priority level 1 could have one or more T-BSs with pre-allocated resources; while service with priority level 8 may have no preconnected T-BS, and only use standard 3GPP handover procedure.

4.3 Multi-Agent Deep Reinforcement Learning Assisted PHO Management

MAS provides mechanisms for complex systems management involving multiple agents and coordination of independent agents' behaviors. MAS allows a constraint satisfaction problem to be subcontracted to different agents with their own interests and goals. The simplest MAS is a multi-agent system with a non-communicating scenario [58].

MADRL is a good fit for the multiple UEs PHO scenario, where all the UEs act independently in the network. Each DRL agent represents an individual UE and learns partial information from the system in a distributed way. This thesis extends the single agent DQN-assisted UE-associated PHO solution to the simplest MADRL-assisted solution for the multiple UEs scenario. All the agents operate in a partially observable environment and no information is exchanged between each other.

The credit assignment problem may arise when considering extending the single agent RL model to a multi-agent ML application. The credit assignment issue occurs when the same global reward is given to all the agents as feedback without distinguishing their individual contributions [88]. This issue may encourage the lazy agents in the system, and affect individual learning performance. Since the proposed multi-agent PHO solution focuses on the non-communicating and independent agents' application, one possible

solution is the local reward strategy, which is generated based on the individual agent's behavior [88] [89].

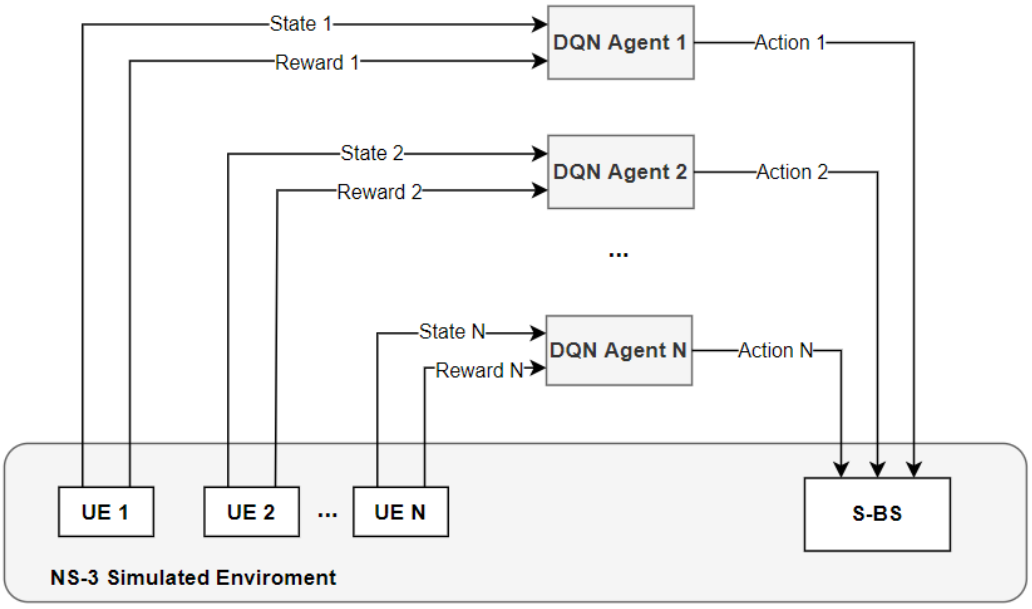


Figure 4.10 Multi-Agent System for PHO Management

As shown in Figure 4.10, each independent UE-associated agent is controlled by the DQN algorithm, and the local reward strategy provides different reward values as the feedback to the agent. An individual agent learns the optimal policy for T-BS selection with the goal of maximizing its own PHO-SR. The DQN algorithm can run on a high-performance computing server which can be set up on a base station. For the multi-agent DQN-based solution, the parallel training for agents can be considered to accelerate the learning process.

The pseudo code of the proposed multi-agent DQN-assisted PHO is designed as Algorithm 2. It can be considered as an extension of Algorithm 1, which is presented in Section 4.2.2.2, by adding a for loop to iterate over a multi-UE application. The parameters listed in Table 4.2. Additionally, N_{UE} is the number of UEs.

Algorithm 2: Multi-Agent DQN-Assisted PHO (adapted from [51]**)**

Input: UEs dynamically measured surrounding BSs' RSRQs and calculated RSRQ change rates

Output: The offline trained models (model per agent).

Parameters: $N_{episode}$, M_{ER}, $minM_{ER}$, minibatch, ϵ_{min}, ϵ_{decay}, γ, T, C

for UE = 1 to N_{UE} do:
  Initialize Experience Replay Memory M_{ER},
  Initialize Primary Q-network (as shown in Figure 4.7) with random weights θ
  Initialize Target Q-network (as shown in Figure 4.7) with weights $\theta' = \theta$
  Initialize $\epsilon = 1$ for ϵ-greedy policy
end for
for episode = 1 to $N_{episode}$ do
  Receive initial state of all the UEs by resetting the customized PHO environment
  for time step t = 1 to T do:
    for UE = 1 to N_{UE} do:
      Following ϵ-greedy policy, select action a, which represents the index of T-BS for PHO. k is the total number of possible actions (candidate BSs).

$$a = \begin{cases} \text{a random action} & \text{with probability } \frac{\epsilon}{k} \\ \text{argmax } Q^*(s, a;\ \theta) & \text{otherwise} \end{cases}$$

      Execute action a to observe the reward r the next state s'
      Store transition (s, a, r, s') in the M_{ER}
      If buffer size >= $minM_{ER}$
        Sample random minibatch of transitions (s_j, a_j, r_j, s_{j+1}) from M_{ER}

$$\text{target}_{\text{Q}} = \begin{cases} r_j & \text{if episode is terminated at step j + 1} \\ r_j + \gamma \max\limits_{a'} Q(s_{j+1}, a'; \theta') & \text{otherwise} \end{cases}$$

        Perform a gradient descent to minimize loss function:

$$L(\theta) = \left(\text{target}_{\text{Q}} - Q(s_j, a_j; \theta)\right)^2$$

        Update target parameter ($\theta' = \theta$) in every C step
      end if
    end for
  end for
  if $\epsilon > \epsilon_{min}$:
    $\epsilon *= \epsilon_{decay}$
    $\epsilon = \max(\epsilon_{min}, \epsilon)$
  end if
  for UE = 1 to N_{UE} do:
    Set up filter for saving the trained model for individual agent
  end for
end for // multi-agent DQN-assisted PHO offline training for optimal T-BS selection

Chapter 5: Implementation with NS-3

Implementation is critical in evaluating the proposed PHO. Implementation of the proposed PHO solution has been conducted using NS-3 and it involves various components, which warrants further explanations. Section 5.1 presents an overview of the system. Section 5.2 details the PHO solution and all the customized traces used in the PHO solution on NS3. Section 5.3 describes the DQN-assisted UE-associated PHO management, and MADRL-based implementation with Keras and TensorFlow.

5.1 System Overview

The implementation of the proposed DQN-assisted PHO solution with NS-3 comprises three main components: NS-3 simulator, DQN Agent and OpenAI Gym. The proposed system architecture is shown in Figure 5.1.

- NS-3 simulator: The NS-3 provides the network functionalities in a simulation scenario. The simulator provides an environment for an agent. The proposed PHO process presented in Figure 4.1 is implemented with NS-3 in C++.
- DQN agent: The single DQN agent algorithm, which is given in Section 4.2.2.2, is implemented with TensorFlow and Keras libraries in Python.
- NS3-Gym: It integrates the OpenAI Gym framework and the NS-3 simulator. The OpenAI Gym utilizes the interface between an agent and the simulator. The communication between the agent and the environment is based on serialized messages via ZMQ [66] socket using the Protocol Buffers library. The NS3-Gym middleware consists of two components: Environment Proxy and Environment Gateway. The environment proxy, written in Python, is inherited from the Gym APIs. The

environment gateway, written in C++, is a part of the NS-3 simulator environment. The callback functions are applied for the state collection and the action execution.

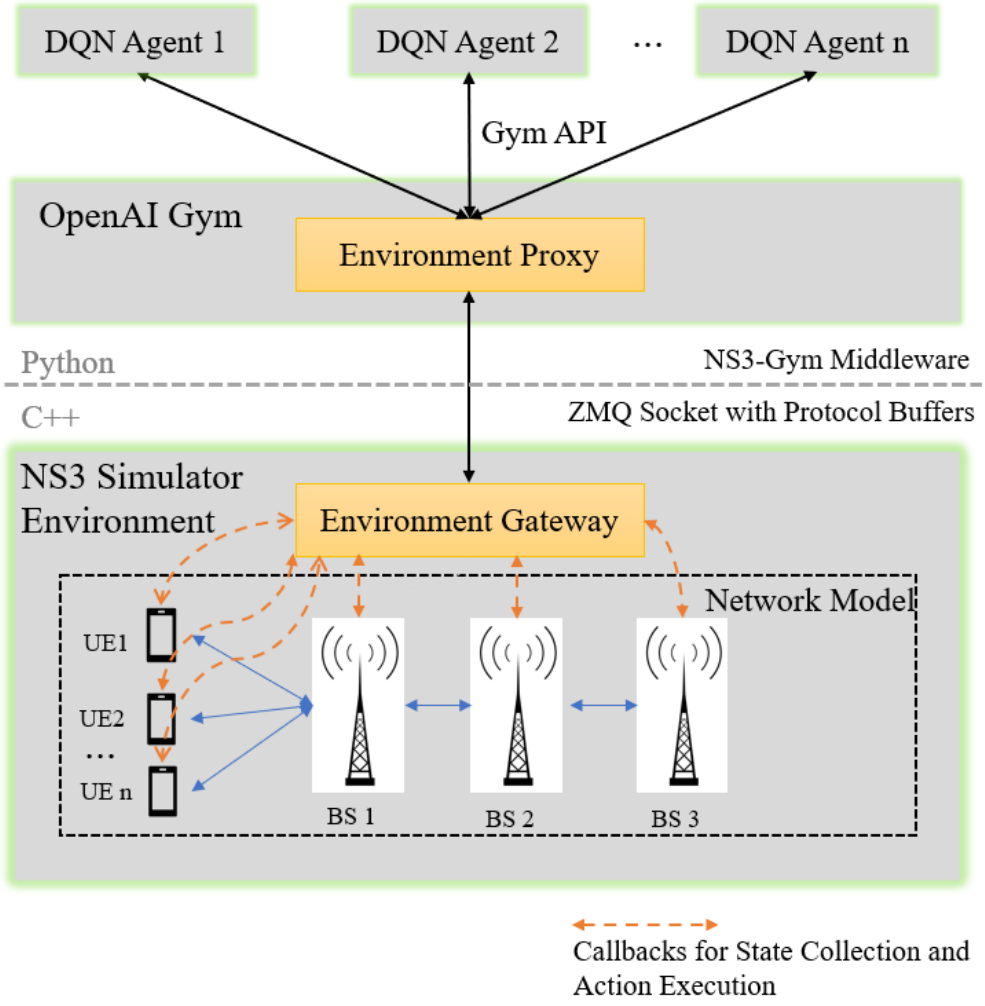


Figure 5.1 Simulation Architecture of Multi-Agent DQN-Assisted PHO Management

5.2 Pre-connect Handover Implementation

Recall the PHO process presented in Figure 4.1. This thesis implements the proposed X2-based PHO solution in a NS-3 simulated LTE network. The implementation is mainly in EPC and RRC modules.

The NS-3 LTE-EPC control model is shown in Figure 5.2 [90]. The control interfaces are S1-AP, S11 interface and X2-AP interface. The PHO is mainly scattered in the X2-AP interface, the EpcX2Application module, and the LteEnbRrc module.

The EpcX2 entity is installed inside an eNodeB and provides the functionality for the X2 interface, including the X2 control plane interface X2-C and the user plane interface X2-U. In the proposed PHO mechanism, all the control messages are transmitted on the X2-C interface, the signaling message between the S-BS and T-BS are through the X2-C socket of the EpcX2 module. The early DL forwarding from S-BS to T-BS is via the X2-U socket.

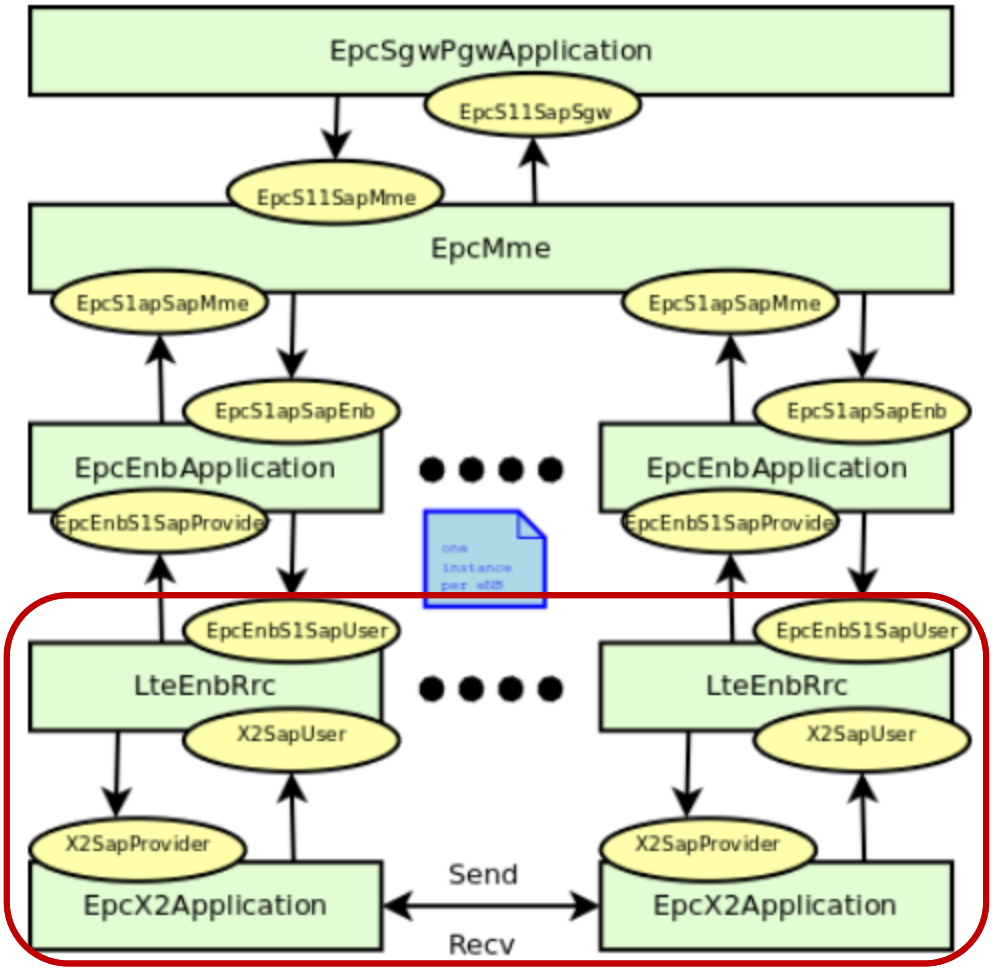


Figure 5.2 EPC Control Model in NS-3 [90]

The RRC is a network layer protocol that is defined by 3GPP in [34] [35]. It is used in the air interface between a UE and a BS. The major functions of the RRC protocol include: connection establishment and release, system information broadcasting, radio admission control, radio bearer establishment, configuration and release, UE RRC measurement model, handover, etc. There are two different RRC protocol models supported in NS-3 for message transmissions, which are real and ideal. The real mode is adopted in the proposed PHO solution.

The RRC entities in NS-3 include LteUeRrc and LteEnbRrc, respectively, at the UE and the eNodeB. The LteUeRrc handles the UE's measurement functionalities in the handover process, including measurement configuration, measurement performing, measurement report triggering, and measurement reporting. LteEnbRrc is the LTE Radio Resource Control entity on eNodeB. All the BS-managed radio resource related functionalities in PHO management are in the LteEnbRrc module. For example, the radio bearer and the radio resource block management, the RNTI assignment, the resource pre-allocation in handover command, including the DL and UL carrier frequency, DL and UL channel bandwidth, RACH preamble, etc.

As mentioned in Section 4.1, a capacity-adjustable queue is required at T-BS to buffer the received early forwarding DL packets for UE. An attribute, namely QueueCapacity, is added in the LteEnbRrc module, and it can be dynamically configured.

The A2 and A4 events are used in the proposed PHO as the handover trigger conditions. The NS-3 provided A2-A4-RSRQ algorithm utilizes the RSRQ measurements acquired from A2 and A4 events. The algorithm is summarized in Figure 5.3 [90]. There

are two attributes in the algorithm: the *ServingCellThreshold* for Event A2 and the *NeighbourCellOffset* for Event A4.

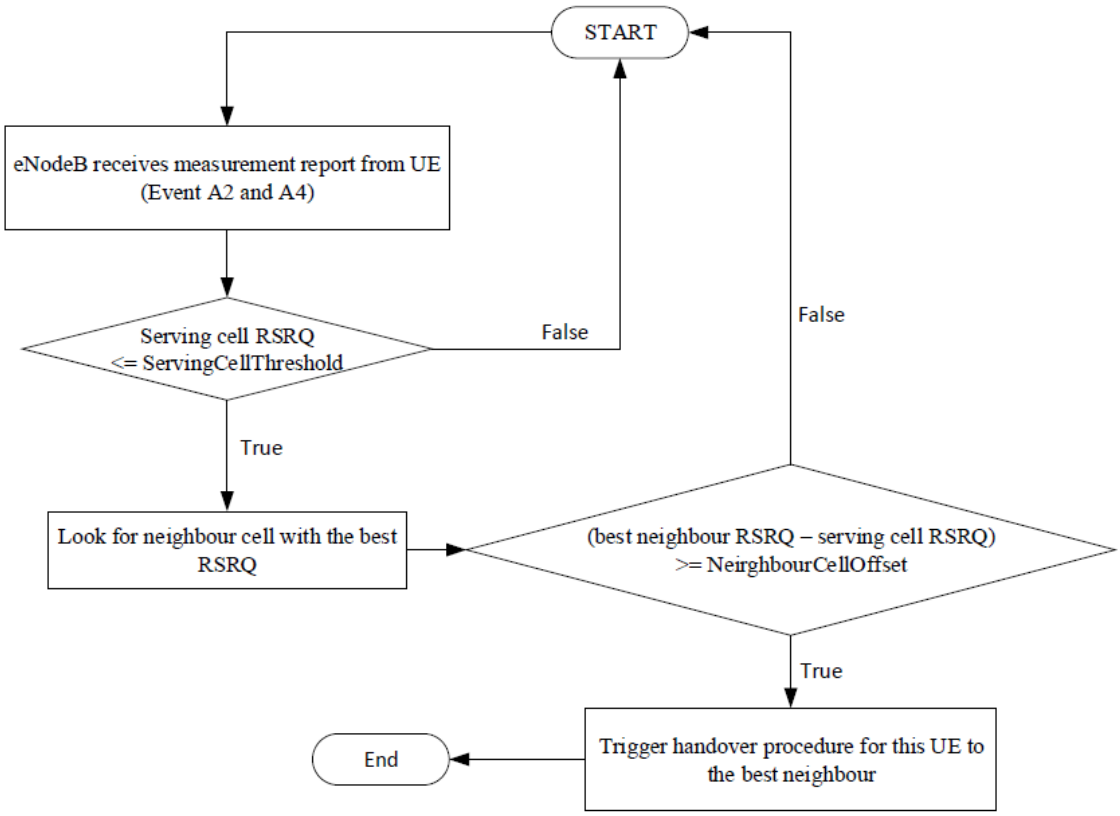


Figure 5.3 A2-A4-RSRQ Handover Algorithm in NS-3 [90]

The tracing utility tool with the callback functionality in NS-3 is significant. As introduced in Section 2.5.1, the tracing system is built on the concept of independent tracing sources acting as the provider, and tracing sinks acting as the consumer. It applies a uniform mechanism for connecting the sources to the sinks [61][91]. The tracing tool can be used to retrieve information from a simulation. The customized traces listed in Table 5.1 are implemented in the DQN-based PHO management solution, which are used for the communication between the simulation environment and the agent.

Table 5.1 Traces used for DQN-Assisted PHO Management

NS3 Model	Availability in NS3	Trace Source Name	Trace Fired upon Condition	Functionality
LteUeRRC	Existing	ConnectionEstablished	Successful connection establishment event between UE and BS	NS-3 sends the UE QoS information to the environment
LteUeRRC	Existing	3GPPHandoverEndOk	Successful outcome of a 3GPP baseline handover between UE and T-BS	NS-3 sends the 3GPP handover success result to the environment
LteUeRRC	Added	PHOEndOk	Successful outcome of a PHO between UE and T-BS	NS-3 sends the PHO handover success result to the environment
LteUePhy	Added	ReportRsrqAllBSs	Reports UE measurements of RSRQs (dB) for all the surrounding BSs	UE measured RSRQ values are generated in NS-3, and are sent to the Gym environment.
GymEnv	Added	EnvNextBSIndex	The agent selects the T-BS index for PHO.	Agent sends the selected T-BS index to simulated environment

Additionally, the traces in Table 5.2 are used to support the early data duplicating-forwarding-buffering mechanism in PHO. Because the proposed PHO is based on the MBB scheme, there is no packet loss during the simulated PHO process, but the early DL duplicating-forwarding-buffering mechanism can prevent packet loss if there are radio link failures and error occurrences during handovers.

Table 5.2 Traces used for Early DL Duplicating-Forwarding-Buffering

NS3 Model	Availability in NS3	Trace Source Name	Trace Fired upon Condition
LteUeRRC	Added	UeRxDlPacketStats	The statistics of UE received DL packets, including the number of lost and duplicated packets.
LteEnbRrc	Added	QueueCapacity	The capacity of the queue is configured at preconnected T-BS for early DL data buffering in PHO.
LteEnbRrc	Added	ForwardDLDataToPreconnectedBS	Once S-BS receives the Pre-connect Request Acknowledge message from T-BS, The S-BS starts forwarding DL packets to T-BS.
LteEnbRrc	Added	SendPreconnectBufferedDLDataToUe	Pre-connected T-BS receives a Path Switch Request Acknowledge message from CN. It indicates the data plane switched to T-BS from CN. T-BS sends the buffered data to UE.

5.3 DQN Agent Implementation with Keras and Tensorflow

The MRs are periodically generated by the UEs in the NS-3 simulator. The MRs are further processed in the environment by extracting the RSRQ values, calculating the RSRQ change rates, and formatting them into a state vector. As shown in Figure 5.1, the formatted state information flows from the NS-3 simulated environment through the NS3-Gym middleware to the agent. The agent takes the state vector as the input, and performs the training process with the goal of the optimal policy of T-BS selection by maximizing the long-term rewards in a trial-and-error manner.

The TensorFlow and Keras libraries are imported into Python to utilize DNN in DQN. TensorFlow is a ML software library released by Google [92]. Keras is a deep learning API focusing on deep learning and running on top of the TensorFlow platform [93].

Recall the system framework in Figure 5.1 that supports the independent multi-agent application for the multi-UE scenario. There are three additional components that need to be added for MADRL-based implementation. Firstly, the multiple UE entity nodes need to be added to the simulation environment, and each node simulates an agent. Secondly, the local reward is required for the individual agent in the environment. Thirdly, the serialized structured data in the Protocol Buffers message of the NS3-Gym middleware needs to be modified to support the local reward feedback. Furthermore, the Protocol Buffers message is required to be re-compiled and the NS3-Gym Python module needs to be reinstalled to activate the local reward functionality.

Chapter 6: Experiments and Results

The feasibility of the proposed PHO mechanism, the performance of the DQN and MADRL-assisted PHO management solutions are evaluated in this chapter. Each experiment is described with an objective, adopted topology, essential parameters, and results.

- Section 6.1 describes the system formation, including simulation software version, operating system, and hardware. In addition, the simulation parameters with NS-3 and the hyperparameters used in DQN are listed.
- Section 6.2 depicts the feasibility evaluation results of the proposed PHO mechanism simulated on NS-3.
- Section 6.3 analyses the performance of the proposed DQN-assisted PHO management in both single-agent and multi-agent experiments.
- Section 6.4 analyses the impact of some hyperparameters, including discount factor, replay buffer capacity, and ReLU vs. Leaky ReLU activation functions.
- Section 6.5 presents a summary.

6.1 System Information and Simulation Parameters

To evaluate the feasibility of the designed PHO technique and the proposed DQN-assisted PHO management, all the evaluation experiments were conducted with the NS-3 simulator and a modified NS3-Gym tool specific to MAS. As aforementioned in Section 5.3, a modification of the NS3-Gym tool is needed to support the reward value of the individual agent in an MAS. The information of software and hardware is summarized in Table 6.1.

Table 6.1 Software and Hardware Information

NS-3 Version	ns-3.33
NS3-Gym Version	1.0.0
OS	Ubuntu 20.04.3 LTS 64-bit
Processor	AMD Ryzen 7 3700X 8-core @ 3.60 GHz
RAM	64.0 GB

The proposed PHO handover mechanism is simulated and evaluated with NS-3. Table 6.2 lists the relevant configurations in the simulation. All the listed parameters are held constant except for the parameters of interest: the number of UEs and UE's mobility model.

Table 6.2 NS-3 Simulation Parameters

Parameter		Setting
Simulation Duration (Seconds)		60
Number of BSs		6
Number of UEs		1-3
UE's Mobility Model		RandomWalk2D; ConstantVelocity
RRC Mode		Real
EPS Bearer To Radio Link Control (RLC) Mapping Mode		RLC_UM
Antenna Transmission Mode		SISO
eNodeB Power Transmission (dBm)		50
LTE-FDD Resource	**DL EARFCN**	100
	UL EARFCN	18100
	DL Bandwidth (Number of RBs)	100
	UL Bandwidth (Number of RBs)	100
Scheduler		RrFfMacScheduler (Round Robin)
Pathloss Model		FriisPropagationLossModel
Handover Algorithm		A2-A4-RSRQ
A2-A4-RSRQ Parameters	**ServingCellThreshold**	30
	NeighbourCellOffset	2

The carrier frequency in UL and DL is designated by the Evolved Universal Terrestrial Radio Access (E-UTRA) Absolute Radio Frequency Channel Number (EARFCN) in the range of 0 to 65535 [94]. The relation between EARFCN and the carrier

frequency (MHz) for UL is given by Equation 6-1, where N_{UL} is UL EARFCN. Accordingly, Equation 6-2 is for DL, where N_{DL} is DL EARFCN. The factors of F_{UL_low}, $N_{Offs-UL}$, F_{DL_low}, and $N_{Offs-DL}$ are dependent on the operating band, and can be obtained in a lookup table specified by 3GPP [94]. The lookup table is given in Appendix B [94]. The specific carrier frequency listed in Table 6.2 is 1930 MHz for UL, and 2120 MHz for DL, which belong to LTE Band 1.

$$F_{UL} = F_{UL\_low} + 0.1 * (N_{UL} - N_{Offs-UL}) \quad \textbf{Equation 6-1}$$

$$F_{DL} = F_{DL\_low} + 0.1 * (N_{DL} - N_{Offs-DL}) \quad \textbf{Equation 6-2}$$

The radio channel bandwidth is configured as the number of RBs. Table 6.3 shows the mapping of the number of RBs against channel bandwidth used in NS-3.

Table 6.3 Mapping of Number of RBs - Channel Bandwidth

Number of RBs	Channel Bandwidth (MHz)
6	1.4
10	3
25	5
50	10
75	15
100	20

The propagation loss models determine the wireless signal strength at the receivers. The NS-3 simulator presently supports up to 16 different loss models in its library [90] . The model used in this research is Friis path loss model [95]. Frris assumes a free space scenario. The formula is defined as Equation 6-3, where P_r is reception power (W); P_t is transmission power (W); G_r is reception gain; G_t is transmission gain; λ is wavelength (m); d is distance (m); L is system loss; C = 299792458 m/s is the speed of light in vacuum; and f is the frequency (Hz).

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2 L} = \frac{P_t G_t G_r C^2}{(4\pi df)^2 L} \qquad \textbf{Equation 6-3}$$

Furthermore, the DQN algorithm is applied to solve the decision-making problem of optimized T-BS selection in PHO management. The DQN agent is trained with the listed hyperparameters adopted from Table 6.4.

Table 6.4 Training Hyperparameter - DQN

Parameter	Value
Number of Episodes	2,000
Optimizer	Adam
Learning Rate of DNN	0.01
Activation Function in Hidden Layer	Leaky ReLU
Activation Function in Output Layer	Softmax
Exploration Decay Rate	0.999
Minimum Epsilon Value	0.001
Discount Factor	0.99
Capacity of Experience Replay Memory	50,000
Minimum Number of Transition Samples in Replay Memory to Start Training	100
Mini-batch Size (The number of Samples)	64

6.2 Experimental Network Topologies

Two network topologies were adopted in various experiments: free space topology and highway topology.

The free space topology is depicted in Figure 6.1, which is a 6-BS system. All the entities are positioned having centers at coordinates (x, y, z) with the assumption that the UE's height is 1.6 meters and the BSs' height are 3 meters. Since the simulation duration in one episode is a fixed value, the total number of handover occurrences during a UE's movement depends on the UE's velocity and moving direction. The single agent UE-associated DQN solution is evaluated with this topology.

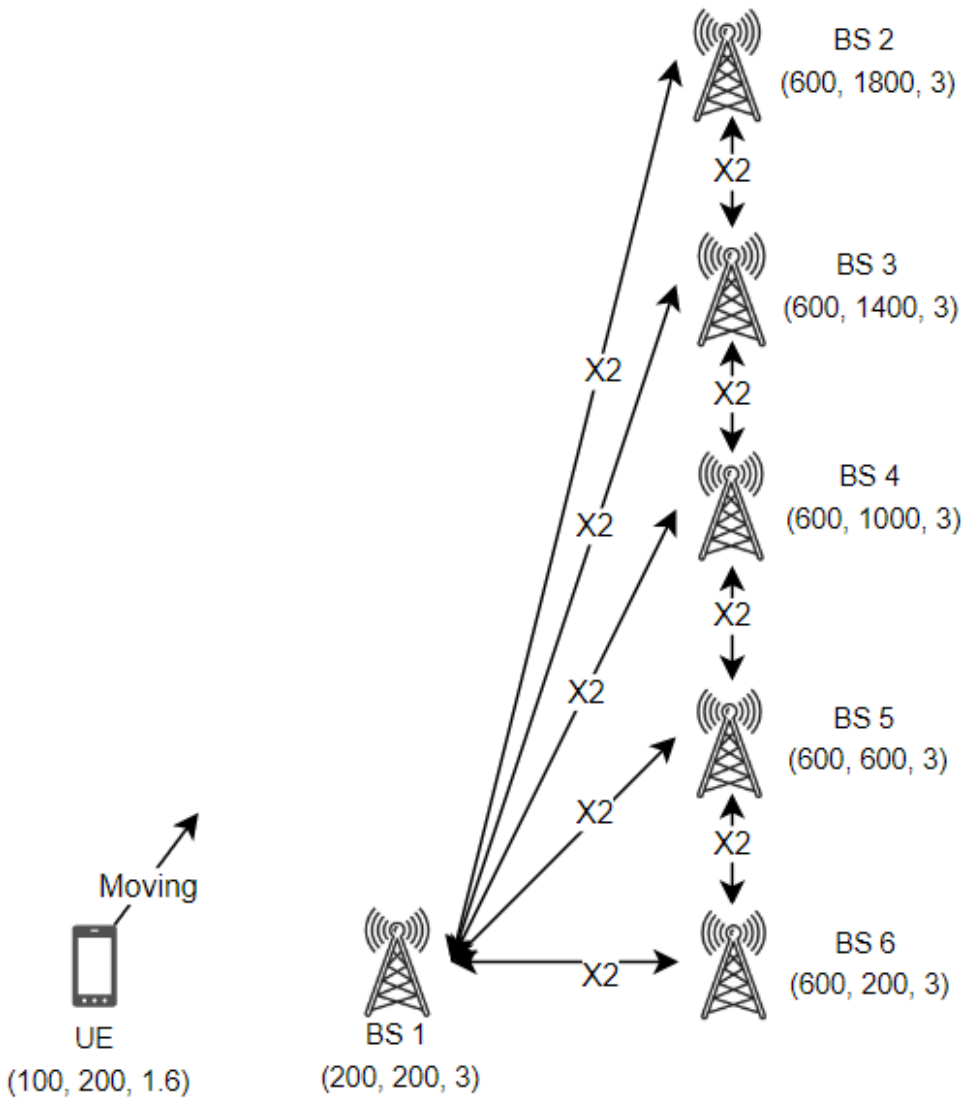


Figure 6.1 Free Space Topology Adopted in Experiments

The highway scenario is one of the typical deployment scenarios required in 5G networks [11]. The impact of UE's velocity for handover performance was evaluated in [96], in which the UE's velocity was in the range of 30 to 100 m/s. The empirical results show that the UE with the higher speed received a worse signal power and quality. The high velocity increases the possibility of handover failure, which counteracts the seamless data service and deteriorates QoS. The highway topology with coordinates used in the

experiments is shown in Figure 6.2. The proposed DQN-assisted and MADRL-based PHO solutions were evaluated with highway topology.

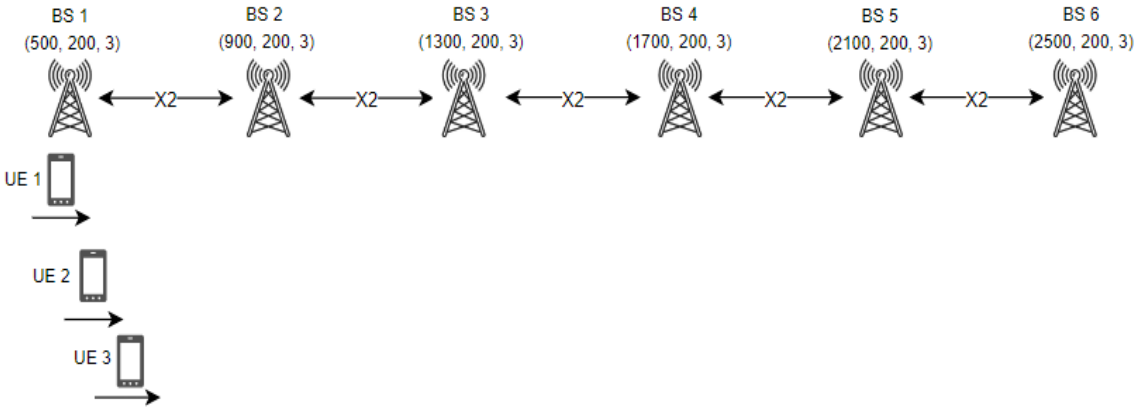


Figure 6.2 Highway Topology Adopted in Experiments

6.3 Evaluation of PHO Mechanism

❖ **Experiment Objective**:

This experiment aims to verify the feasibility of the proposed PHO mechanism described in Section 4.1 and Figure 4.1. In this experiment, the PHO is triggered manually at specific time slots without any ML approach involved. The following three main scenarios are simulated:

- Pre-connection of single candidate T-BS in PHO
- Pre-connections of multiple candidate T-BSs in PHO
- PHO cancellation with pre-connection and pre-allocated radio resource release.

❖ **Topology and Parameters:**

The adopted topology is shown in Figure 6.2. The simulation parameters are listed in Table 6.2. A single UE is configured as the ConstantVelocityMobilityModel with the velocity of 27.78 m/s (100 km/h) and departure position at coordinates (400, 100, 1.6).

As introduced in Section 4.1, the proposed early DL data duplicating-forwarding-buffering mechanism embraces a capacity-adjustable queue at T-BS for packet buffering. The queue capacity is defined as the number of PDCP SDUs, which is configured as 10 in this experiment.

❖ **Results:**

Firstly, the proposed PHO with a single candidate T-BS scenario was simulated and verified. The trace logs generated by NS-3 are illustrated in Figure 6.3. Some important steps are highlighted as follows:

```
1  /NodeList/9/DeviceList/0/LteUeRrc/ConnectionEstablished: UE 1 connects to S-BS 1, at time 0.0212143s
2  /NodeList/3/DeviceList/0/LteEnbRrc/ConnectionEstablished : S-BS 1 successfully connects to UE 1, at time 0.0217143 s
3  /NodeList/3/DeviceList/0/LteEnbRrc/SendPreconnectionRequest: S-BS 1 sends Preconnect Request to T-BS 2 for UE 1, at time 10 s
4  /NodeList/4/DeviceList/0/LteEnbRrc/ReceivePreconnectionRequest: T-BS 2 receives Preconnect Request from S-BS 1 for UE 1, at time 10 s
5  /NodeList/3/DeviceList/0/LteEnbRrc/ForwardDLDataToPreconnectedBS: S-BS 1 starts forwarding DL data to T-BS 2 for UE 1, at time 10 s
6  /NodeList/9/DeviceList/0/LteUeRrc/PreconnectionEstablished: UE 1 preconnects to T-BS 2, at time 10.0005 s
7  /NodeList/3/DeviceList/0/LteEnbRrc/PreconnectHandoverStart: S-BS 1 initializes Preconnect Handover Request to T-BS 2 for UE 1, at time 11.4805 s
8  /NodeList/9/DeviceList/0/LteUeRrc/PreconnectHandoverStart: UE 1 starts Preconnect Handover from S-BS 1 to T-BS 2, at time 11.481 s
9  /NodeList/9/DeviceList/0/LteUeRrc/RandomAccessSuccessful: UE 1 notifies random access successful to T-BS 2, at time 11.4852 s
10 /NodeList/9/DeviceList/0/LteUeRrc/PreconnectHandoverEndOk: UE 1 successfully completes Preconnect Handover with T-BS 2 in RAN network, at time 11.4852 s
11 /NodeList/9/DeviceList/0/LteUeRrc/ConnectionEstablished: UE 1 connects to S-BS 2, at time 11.4852s
12 /NodeList/4/DeviceList/0/LteEnbRrc/PreconnectHoPathSwitchRequestAck: T-BS 2 receives Path Switch Request Ack from MME for preconnected UE 1,
13                                                                      sends UE Context Release to S-BS 1,
14                                                                      starts transmitting the buffered DL data to UE , at time 11.5058 s
15 /NodeList/4/DeviceList/0/LteEnbRrc/PathSwitchRequestToMME: T-BS 2 sends Path Switch Request to MME for preconnected UE 1, at time 11.4857 s
16 /NodeList/4/DeviceList/0/LteEnbRrc/PreconnectHandoverEndOk: T-BS 2 successfully completes Preconnect Handover with UE 1, at time 11.5058 s
17 /NodeList/4/DeviceList/0/LteEnbRrc/SendPreconnectBufferedDLDataToUe: T-BS 2 starts sending buffered DL data to UE 1, at time 11.5058 s
18 /NodeList/3/DeviceList/0/LteEnbRrc/RemoveUe: S-BS 1 receives UE Context Release from T-BS 2. Remove UE 1, at time 11.5058 s
19 /NodeList/9/DeviceList/0/LteUeRrc/UeRxDlPacketStats UE 1: receives buffered DL data from T-BS 2. Packet Loss: 0. Packet Duplication: 10, at time 11.5352 s
```

Figure 6.3 NS-3 Tracing Logs of Simulated PHO Process: Single Pre-connection

- Line 3 shows that S-BS1 sends a Pre-connect Request message to T-BS2 via the X2-C interface.
- Line 4 shows that T-BS2 receives request message, then takes the following actions: accepts the request, pre-allocates the radio resources, prepare the handover command, encode handover command in a Pre-connect Request Acknowledge message, and send the acknowledge message to S-BS1.

- Line 5 indicates that S-BS1 receives the Pre-connect Request Acknowledge message including the handover command from T-BS2. S-BS1 sends an RRC Pre-connection Configuration message including the handover command towards UE. In addition, S-BS1 initializes the early data forwarding to T-BS2 on the X2-U interface.
- Line 6 shows that the UE receives the handover command from S-BS1. The pre-connection is established between UE and T-BS2.
- Lines 7 to 16 demonstrate the handover process.
- Line 17 indicates the data plane has switched to T-BS2. T-BS2 starts transmitting the buffered data to UE.
- The packet statistic results at line 19 shows that there is no packet loss during the PHO simulation.

```
1  /NodeList/9/DeviceList/0/LteUeRrc/ConnectionEstablished: UE 1 connects to S-BS 1, at time 0.0212143s
2  /NodeList/3/DeviceList/0/LteEnbRrc/ConnectionEstablished : S-BS 1 successfully connects to UE 1, at time 0.0217143 s
3  /NodeList/3/DeviceList/0/LteEnbRrc/SendPreconnectionRequest: S-BS 1 sends Preconnect Request to T-BS 2 for UE 1, at time 10 s
4  /NodeList/3/DeviceList/0/LteEnbRrc/SendPreconnectionRequest: S-BS 1 sends Preconnect Request to T-BS 3 for UE 1, at time 10 s
5  /NodeList/4/DeviceList/0/LteEnbRrc/ReceivePreconnectionRequest: T-BS 2 receives Preconnect Request from S-BS 1 for UE 1, at time 10 s
6  /NodeList/5/DeviceList/0/LteEnbRrc/ReceivePreconnectionRequest: T-BS 3 receives Preconnect Request from S-BS 1 for UE 1, at time 10 s
7  /NodeList/3/DeviceList/0/LteEnbRrc/ForwardDLDataToPreconnectedBS: S-BS 1 starts forwarding DL data to T-BS 2 for UE 1, at time 10 s
8  /NodeList/3/DeviceList/0/LteEnbRrc/ForwardDLDataToPreconnectedBS: S-BS 1 starts forwarding DL data to T-BS 3 for UE 1, at time 10 s
9  /NodeList/9/DeviceList/0/LteUeRrc/PreconnectionEstablished: UE 1 preconnects to T-BS 2, at time 10.0005 s
10 /NodeList/9/DeviceList/0/LteUeRrc/PreconnectionEstablished: UE 1 preconnects to T-BS 3, at time 10.0005 s
11 /NodeList/3/DeviceList/0/LteEnbRrc/PreconnectHandoverStart: S-BS 1 initializes Preconnect Handover Request to T-BS 2 for UE 1, at time 11.4805 s
12 /NodeList/9/DeviceList/0/LteUeRrc/PreconnectHandoverStart: UE 1 starts Preconnect Handover from S-BS 1 to T-BS 2, at time 11.481 s
13 /NodeList/9/DeviceList/0/LteUeRrc/RandomAccessSuccessful: UE 1 notifies random access successful to BS 2, at time 11.4852 s
14 /NodeList/9/DeviceList/0/LteUeRrc/PreconnectHandoverEndOk: UE 1 successfully completes Preconnect Handover with T-BS 2 in RAN network, at time 11.4852 s
15 /NodeList/9/DeviceList/0/LteUeRrc/ConnectionEstablished: UE 1 connects to S-BS 2, at time 11.4852s
16 /NodeList/4/DeviceList/0/LteEnbRrc/PathSwitchRequestToMME: T-BS 2 sends Path Switch Request to MME for preconnected UE 1, at time 11.4857 s
17 /NodeList/4/DeviceList/0/LteEnbRrc/PreconnectHandoverEndOk: T-BS 2 successfully completes Preconnect Handover with UE 1, at time 11.5058 s
18 /NodeList/4/DeviceList/0/LteEnbRrc/PreconnectHoPathSwitchRequestAck: T-BS 2 receives Path Switch Request Ack from MME for preconnected UE 1,
19                                                                      sends UE Context Release to S-BS 1,
20                                                                      starts transmitting the buffered DL data to UE , at time 11.5058 s
21 /NodeList/4/DeviceList/0/LteEnbRrc/SendPreconnectBufferedDLDataToUe: T-BS 2 starts sending buffered DL data to UE 1, at time 11.5058 s
22 /NodeList/3/DeviceList/0/LteEnbRrc/RemoveUe: S-BS 1 receives UE Context Release from T-BS 2. Remove UE 1, at time 11.5058 s
23 /NodeList/5/DeviceList/0/LteEnbRrc/PreconnectionRelease: Other candidate T-BS 3 releases preconnection to UE 1, at time 11.5058 s
24 /NodeList/9/DeviceList/0/LteUeRrc/UeRxDlPacketStats: UE 1 receives buffered DL data from T-BS 2. Packet Loss: 0. Packet Duplication: 10. at time 11.5352 s
```

Figure 6.4 NS-3 Tracing Logs of Simulated PHO Process: Dual Pre-connections

Secondly, the pre-connection with multiple candidate T-BSs in PHO was verified. The logs obtained from NS-3 are shown in Figure 6.4. The pre-connection establishing process is the same as the single T-BS pre-connection scenario in Figure 6.3. In this dual

pre-connection example, there are two pre-connected T-BSs, specifically T-BS2 and T-BS3. Lines 3 to 8 demonstrate the signaling communication between S-BS and T-BSs on X2-C interfaces. S-BS1 sends a Pre-connect Request message to both T-BS2 and T-BS3. Once the candidate T-BSs accept the request, S-BS1 starts forwarding DL data to both T-BS2 and T-BS3 simultaneously, which are shown at line 7 and 8. Additionally, it can be seen that the pre-connections are established between UE and T-BS2 and T-BS3 at line 9 and 10. During UE's movement, the actual handover occurs with T-BS2 at line 12. Based on the proposed PHO process, once the UE completes the handover with a T-BS successfully, the established pre-connection and the reserved pre-allocated resources at other candidate T-BSs should be released, as shown at line 23. Finally, the packet statistic results at line 24 indicate that there is no packet loss during the PHO simulation.

```
1  /NodeList/9/DeviceList/0/LteUeRrc/ConnectionEstablished: UE 1 connects to S-BS 1, at time 0.0212143s
2  /NodeList/3/DeviceList/0/LteEnbRrc/ConnectionEstablished : S-BS 1 successfully connects to UE 1, at time 0.0217143 s
3  /NodeList/3/DeviceList/0/LteEnbRrc/SendPreconnectionRequest: S-BS 1 sends Preconnect Request to T-BS 3 for UE 1, at time 10 s
4  /NodeList/5/DeviceList/0/LteEnbRrc/ReceivePreconnectionRequest: T-BS 3 receives Preconnect Request from S-BS 1 for UE 1, at time 10 s
5  /NodeList/3/DeviceList/0/LteEnbRrc/ForwardDLDataToPreconnectedBS: S-BS 1 starts forwarding DL data to T-BS 3 for UE 1, at time 10 s
6  /NodeList/9/DeviceList/0/LteUeRrc/PreconnectionEstablished: UE 1 preconnects to T-BS 3, at time 10.0005 s
7  /NodeList/3/DeviceList/0/LteEnbRrc/3GPPHandoverStart: BS 1 starts 3GPP handover of UE 1 to BS 2, at time 11.4805 s
8  /NodeList/5/DeviceList/0/LteEnbRrc/PreconnectionRelease: Other candidate T-BS 3 releases preconnection to UE 1, at time 11.4805 s
9  /NodeList/9/DeviceList/0/LteUeRrc/3GPPHandoverStart: UE 1 starts 3GPP handover from BS 1 to BS 2, at location 718.917:200:1.6, at time 11.481 s
10 /NodeList/9/DeviceList/0/LteUeRrc/RandomAccessSuccessful: UE 1 notifies random access successful to BS 2, at time 11.4852 s
11 /NodeList/9/DeviceList/0/LteUeRrc/3GPPHandoverEndOk: UE 1 completes successful handover to BS 2 with RNTI 1, at time 11.4852 s
12 /NodeList/9/DeviceList/0/LteUeRrc/ConnectionEstablished: UE 1 connects to BS 2, at time 11.4852s
13 /NodeList/4/DeviceList/0/LteEnbRrc/PathSwitchRequestToMME: BS 2 sends Path Switch Request to MME for preconnected UE 1, at time 11.4857 s
14 /NodeList/4/DeviceList/0/LteEnbRrc/3GPPHandoverEndOk: BS 2 completes handover of UE  1 RNTI 1, at time 11.5058 s
15 /NodeList/4/DeviceList/0/LteEnbRrc/HandoverPathSwitchRequestAck: BS 2 receives Path Switch Request Ack from MME for UE 1, at time 11.5058 s
16 /NodeList/3/DeviceList/0/LteEnbRrc/RemoveUe: BS 1 receives UE Context Release from BS 2. Remove UE 1, at time 11.5058 s
```

Figure 6.5 NS-3 Tracing Logs of Simulated PHO Process: Pre-connection Cancellation

Thirdly, T-BS mis-selection may occur when an unoptimized T-BS is selected during the agent training process of the DQN-assisted PHO solution. This specific example is demonstrated in Figure 6.5. In that case, a handover is triggered and processed by the standard 3GPP baseline handover procedure; the established pre-connection and the

reserved resources at the preconnected BS(s) should be released, which is shown at line 8. Since the standard 3GPP baseline handover takes over the execution and completion phases, the functionalities of early data forwarding and packet statistics are not supported in the simulation system.

6.4 Evaluation of DQN-Assisted PHO Mechanism

In RL, an agent is trained in episodes. It is a common practice to let a RL agent interact for a fixed amount of time with the environment before resetting it and repeating the process in a series of episodes [97]. This method was applied to all the experiments in this section. The experiments are trying to answer the following questions:

1. What is the performance of the proposed DQN-assisted and MADRL-based PHO management?
2. How do the hyperparameters of discount factor and reply memory capacity affect DQN learning performance?
3. How do ReLU and Leaky ReLU activation functions affect DQN learning performance?

The following two metrics are used for performance evaluations:

- Episode reward: The proposed DQN-assisted solution trains the agent through the repeated episodes. The episode reward is the cumulative reward value in one episode.
- PHO-SR: It is defined by Equation 4-1. In the experiments, PHO-SR is calculated at the end of each episode.

6.4.1 Evaluation of Training Execution Time

❖ **Experiment Objective**:

One of the drawbacks of RL is the slowness in convergence. Additionally, in the proposed MADRL-assisted PHO management scheme, multiple agents are trained in a distributed manner, which increases the computation complexity. This experiment aims to evaluate the training time factors in DQN-based and MADRL-based PHO management solutions.

❖ **Results:**

The execution time against different simulation duration settings in a single UE scenario is demonstrated in Figure 6.6. The training execution time with 2,000 episodes against the number of UEs in the MADRL-based PHO management is shown in Figure 6.7.

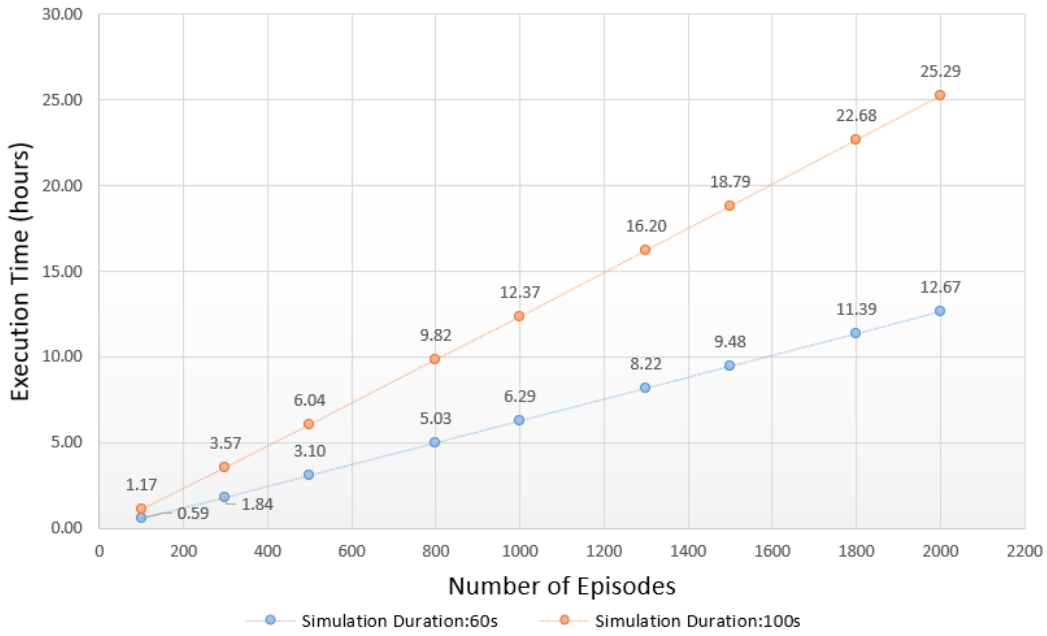


Figure 6.6 Training Execution Time vs. Simulation Time Duration

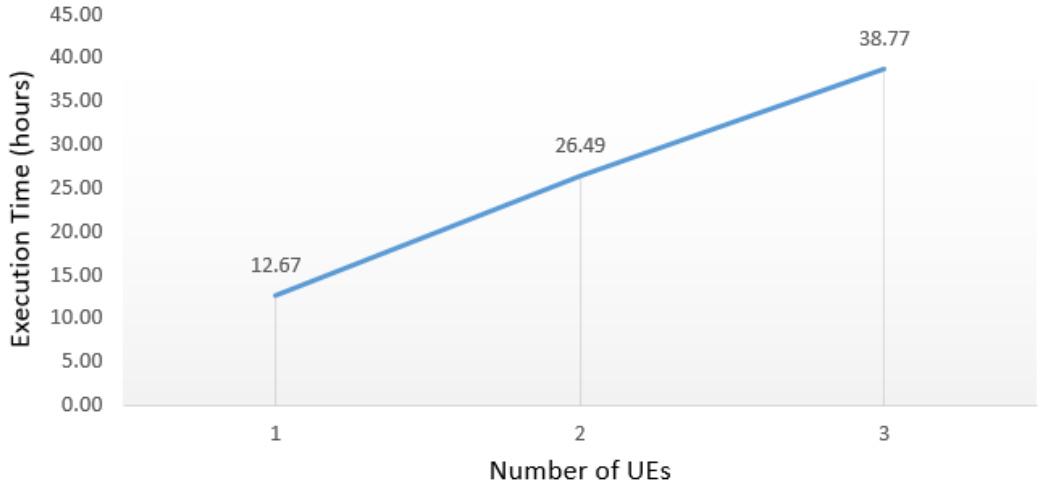


Figure 6.7 Training Execution Time vs. Number of UEs in MADRL-based Solution

The comparison results show that the simulation duration and the number of UEs in MADRL linearly impact the training execution time. Training agents is a time-consuming process.

As mentioned in Section 4.3, the high-performance computing servers with the parallel training for the multi-agent scenario can be considered to accelerate the learning process.

6.4.2 Single-Agent DQN-Assisted PHO Mechanism

❖ **Experiment Objective**:

In the proposed DQN-based solution, the agent is tied to the UE. This experiment aims to evaluate the single UE scenario with the proposed DQN-based PHO management scheme, which is presented as Algorithm 1 in Section 4.2.2.2.

❖ **Topology and Parameters:**

The adopted topology is depicted in Figure 6.1, which is a system of a single UE and 6 BSs. A UE departs from the coordinates (100, 200, 1.6) with a constant velocity of 22 m/s

(80 km/h) and time-based uniformly random direction. The moving direction is changed every 4 seconds in the range of 0 to 1.8 radians. Although the direction change frequency is uncommon in the real-world, the model considers the system randomness, which is common in reality. According to the specific parameters listed in Table 6.5, there are five handovers during the UE's trajectory. The sequential occurrences are BS1-> BS6, BS6-> BS5, BS5-> BS4, BS4-> BS3, and BS3->BS2.

Table 6.5 Parameters for Free Space Topology - Single UE Scenario

	Parameter Name	Values
NS3 Parameters	**Simulation Duration (Per Episode)**	100 seconds
	Number of BSs	6
	Number of UEs	1
	UE's Mobility Model	RandomWalk2dMobilityModel
	Time Mode	4s
	UE's Velocity	22 m/s (80 km/h)
	UE's Direction	Uniform Random [0, 1.8] radians
	UE's Coordinates	(100, 200, 1.6)
DQN Hyperparameters	**Discount Factor**	0.99
	Replay Buffer Capacity	50,000

❖ Results:

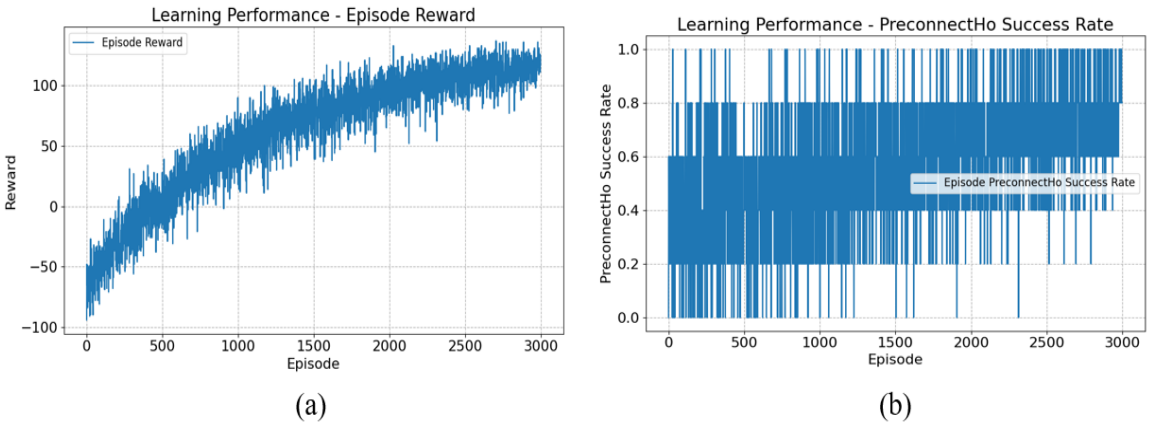


Figure 6.8 Learning Performance: (a) Episode Reward; (b) PHO-SR

The learning performance is shown in Figure 6.8. It can be seen that:

- The UE-associated agents are able to converge successfully within the period of each simulation.
- The number of episodes is not a part of the environment and is used to facilitate learning. However, the number of episodes may affect the learning performance. The learning performance of episode reward and PHO-SR are improved when increasing the number of episodes. When increasing the number of episodes from 2,000 to 3,000, the PHO-SR increases from 80% to 100% in the online prediction.

The simulation delivers the ideal outcome of all triggered handovers that can be completed successfully. In the system, the handover mechanism is configured as the standard 3GPP baseline handover by default. Even though the DQN-assisted PHO-SR is 80%, the standard 3GPP baseline handover takes over the 20% partition. The total handover success rate can reach 100%.

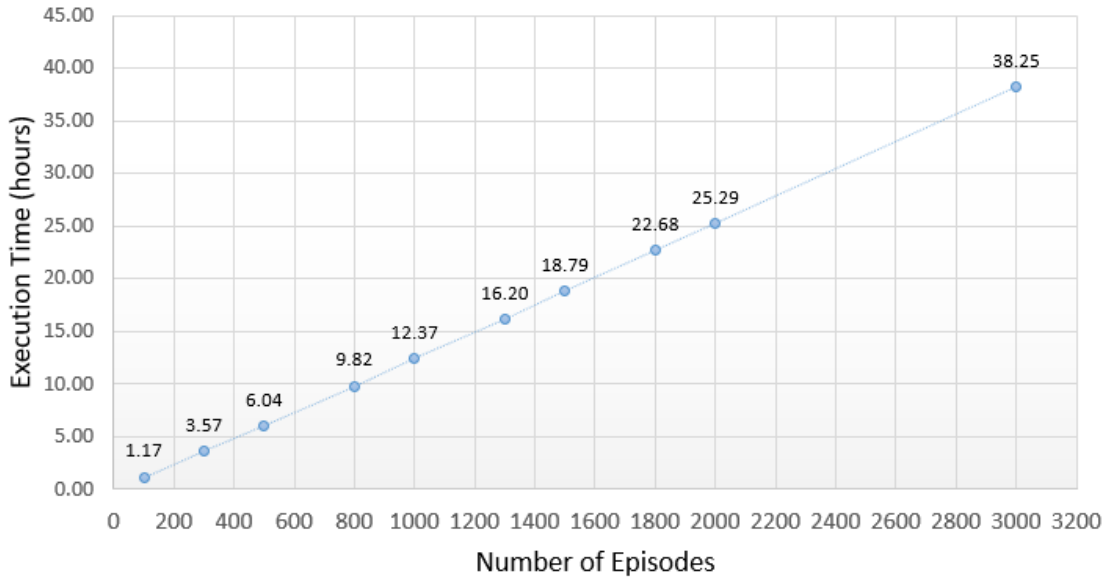


Figure 6.9 Execution Time vs. Number of Episodes

To reach the PHO-SR of 100%, the trade-off is the extra training time needed, from 25.29 hours for 2,000 episodes to 38.25 hours for 3000 episodes, as depicted in Figure 6.9. As a result, the number of episodes of 2,000 is used in all the following experiments in the interest of time.

6.4.3 Multi-Agent DQN-Assisted PHO Mechanism

6.4.3.1 Two UEs with Different Velocities

❖ **Experiment Objective**:

This experiment aims to evaluate the performance of the proposed multi-agent DQN-assisted PHO management scheme, which is presented as Algorithm 2 in Section 4.3.

❖ **Topology and Parameters:**

The topology adopted is depicted as Figure 6.2. However, only two UEs are conducted. The two UEs move with different velocities and depart from the different locations. All the parameters are listed in Table 6.6.

Table 6.6 Parameters for Evaluation of 2 UEs with Different Velocities in MADRL Solution

	Parameter Name	Values
NS3 Parameters	**Number of Episodes**	2,000
	Simulation Duration (Per Episode)	60 seconds
	Number of BSs	6
	Number of UEs	2
	UEs' Mobility Model	ConstantVelocityMobilityModel
	UEs' Velocities	UE1: 27.78 m/s (100 km/h) UE2: 29.17 m/s (105 km/h)
	UEs' Coordinates	UE1: (500, 0, 1.6) UE2: (550, 0, 1.6)
DQN Hyperparameters	**Discount Factor**	0.99
	Replay Buffer Capacity	50,000

❖ **Results**:

The learning performance is shown in Figure 6.10, which includes performances of episode reward shown in (a) (b), and PHO-SR shown in (c) (d).

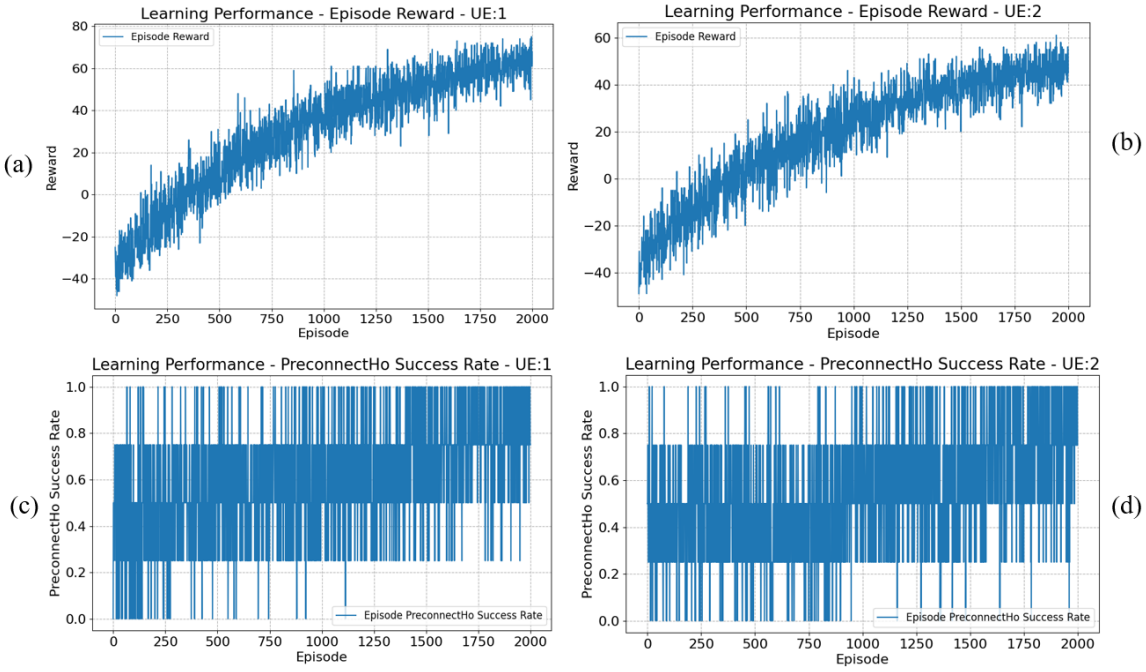


Figure 6.10 Learning Performance Comparison - 2 UEs with Different Velocities

It can be seen that:

- Both UE-associated agents are able to converge successfully within the period of each simulation.
- The performances of PHO-SR for both agents are improved when the number of episodes increases. PHO-SR can converge to 100% at the end of episode 2,000.
- The agent for UE1 outperforms on reward during the learning process.

Table 6.7 Online Prediction Results of 2 UEs with Different Velocities in MADRL Solution

Online Prediction			
UE1 (100 km/h)		**UE2 (105 km/h)**	
PHO-SR	**Reward**	**PHO-SR**	**Reward**
100%	75	100%	61

The online prediction results are shown in Table 6.7. Both UE-associated agents can reach the PHO-SR of 100%. However, the agent for UE1 outperforms on reward, which is the same as offline learning performance.

Additionally, comparing the experiment in Section 6.4.2, the UE is configured with RandomWalk2dMobilityModel, and PHO-SR can only reach 80% when the number of training episodes is 2,000. The randomness in the mobility model increases the learning complexity, which needs longer learning time for convergence.

6.4.3.2 Three UEs with Different Velocities

❖ **Experiment Objective**:

The purpose of this experiment is to evaluate the more complex MADRL-based PHO solution by extending the number of UEs. Compared with the experiment in Section 6.4.3.1, one more UE (UE3) is added to the system.

❖ **Topology and Parameters:**

The adopted topology is depicted in Figure 6.2. All the parameters are listed in Table 6.8.

Table 6.8 Parameters for Evaluation of 3 UEs with Different Velocities in MADRL Solution

	Parameter Name	Values
NS3 Parameters	**Number of Episodes**	2,000
	Simulation Duration (Per Episode)	60 seconds
	Number of BSs	6
	Number of UEs	3
	UEs' Mobility Model	ConstantVelocityMobilityModel
	UEs' Velocities	UE1: 27.78 m/s (100 km/h) UE2: 29.17 m/s (105 km/h) UE3: 30.56 m/s (110 km/h)
	UEs' Coordinates	UE1: (500, 0, 1.6) UE2: (550, 0, 1.6) UE3: (600, 0, 1.6)
DQN Hyperparameters	**Discount Factor**	0.99
	Replay Memory Capacity	50,000

❖ **Results:**

The agents' learning performance on episode reward and PHO-SR is shown in Figure 6.11. The online prediction results are summarized in Table 6.9.

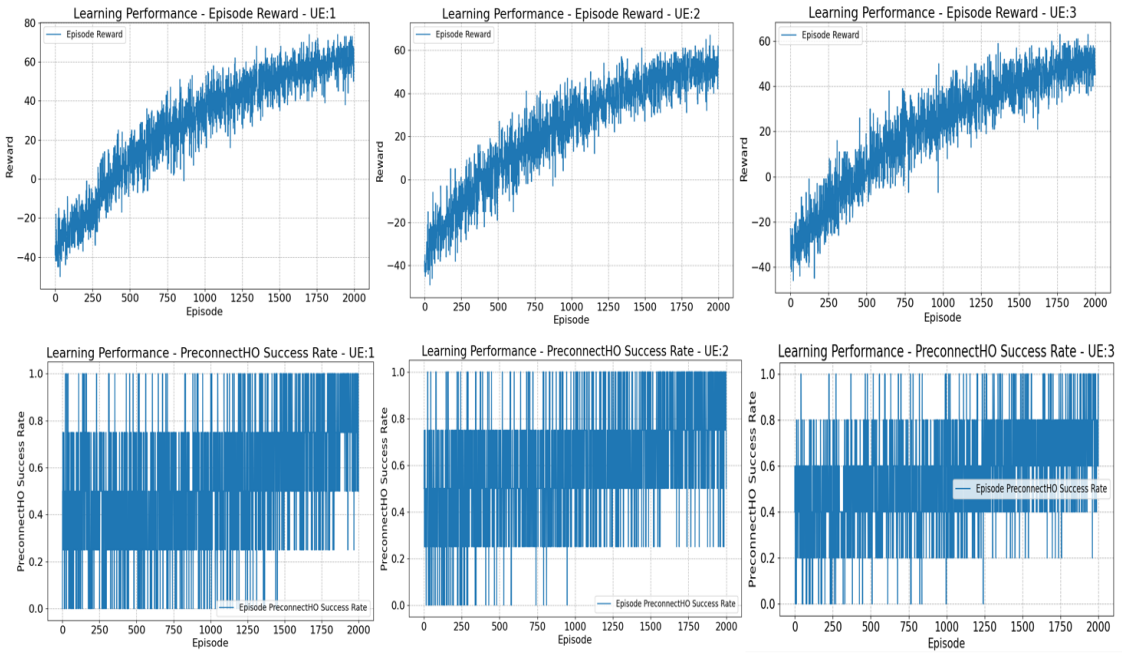


Figure 6.11 Learning Performance - 3 UEs with Different Velocities

Table 6.9 Online Prediction Results of 3 UEs with Different Velocities in MADRL Solution

Online Prediction Results					
UE1 (100 km/h)		**UE2 (105 km/h)**		**UE3 (110 km/h)**	
PHO-SR	**Reward**	**PHO-SR**	**Reward**	**PHO-SR**	**Reward**
100%	77	100%	67	80%	63

It can be seen that:

- Both UE1 and UE2 consistently perform well in the learning process, and can reach the online prediction of PHO-SR at 100%, however, UE1 outperforms on reward. The results are similar to Section 6.4.3.1.
- UE3 underperforms on both learning and prediction.

6.4.3.3 Three UEs with Same Velocity

❖ **Experiment Objective**:

The experiment results in Section 6.4.3.2 show that learning and prediction performance of UE3 (110 km/h) is worse than the other two UEs with lower speed values (100 km/h and 105 km/h). This experiment explores the scenario of multiple UEs with the same velocity of 110 km/h.

❖ **Topology and Parameters:**

The network topology is depicted in Figure 6.2. All the parameters are listed in Table 6.10.

Table 6.10 Parameters for Evaluation of 3 UEs with Same Velocity in MADRL Solution

	Parameter Name	Values
NS3 Parameters	**Number of Episodes**	2,000
	Simulation Duration (Per Episode)	60 seconds
	Number of BSs	6
	Number of UEs	3
	UEs' Mobility Model	ConstantVelocityMobilityModel
	UEs' Velocities	UE1: 30.56m/s (110 km/h)
	UEs' Coordinates	UE1: (400, 0, 1.6) UE2: (450, 0, 1.6) UE3: (600, 0, 1.6)
DQN Hyperparameters	**Discount Factor**	0.99
	Replay Memory Capacity	50,000

❖ **Results:**

The offline learning performances are illustrated in Figure 6.12 and the online prediction results are summarized in Table 6.11. Although all three UEs are configured with the same velocity, their learning performance is different. UE1 and UE2 can reach the PHO-SR of 100%. Additionally, UE1 outperforms on reward. UE3 still underperforms on both reward and PHO-SR compared to that of UE1 and UE2, which is the same as that of in Table 6.9.

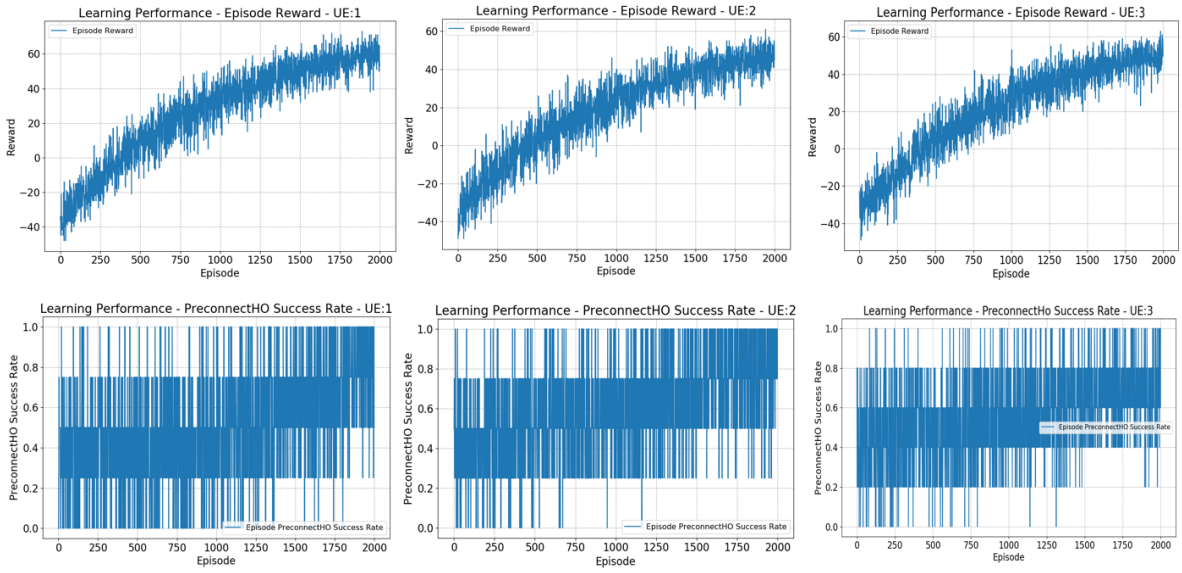


Figure 6.12 Learning Performance - 3 UEs with Same Velocity

Table 6.11 Online Prediction Results of 3 UEs with Same Velocity in MADRL Solution

UE1 (110 km/h)		UE2 (110 km/h)		UE3 (110 km/h)	
PHO-SR	**Reward**	**PHO-SR**	**Reward**	**PHO-SR**	**Reward**
100%	74	100%	60	80%	63

The learning instability of UE3 is caused by the time limits in RL. In the simulated environment, the simulation time per episode is 60 seconds, with the time step of 1 second. The coordinates of all possible handover occurrences are marked as the red stars in Figure 6.13.

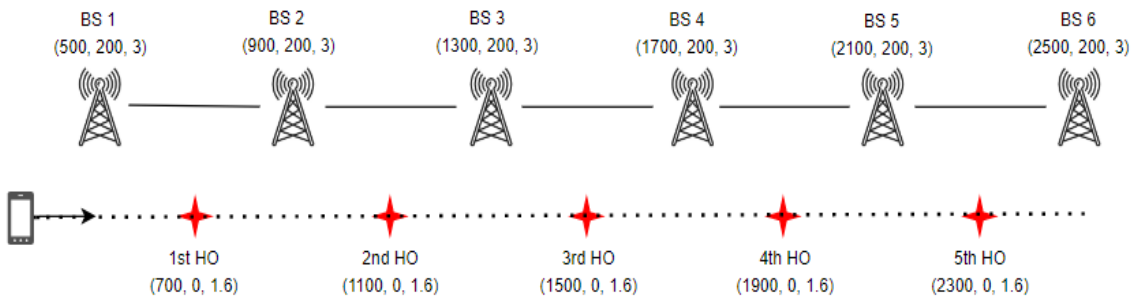


Figure 6.13 Coordinates of Handover Occurrences

During UEs' trajectories, the total number of handover occurrences in a UE's trajectory depends on UE's velocity and departure location. According to the UE's settings in Table 6.10. There are 4 handovers that occurred for UE1 and UE 2, which are from 1st to 4th. By contrast, there are 5 handover occurrences for UE3. The PHO-SR at 80% means that there is one PHO failure. By observing the online prediction process, it can be seen that the PHO failure occurred at the 5th handover. The 5th handover occurs at a simulation time of 55.63 seconds. The UE3-associated agent cannot be trained adequately for the 5th handover with a simulation time limit of 60 seconds.

The importance of time-awareness for optimizing a time-limited objective has been well-studied in the literature. The authors in [97] investigated and thorough analyzed the impact of time limits in RL, and their experimental results show that time limits can cause invalidation of experience replay and result in suboptimal policies and training instability, they suggested adding time limits parameter for time-limited tasks to facility the agent to maximize its performance over a limited time.

6.4.4 DQN Hyperparameters Optimization

The hyperparameters in DQN algorithms are used to control the learning process and can have a significant impact on the learning and prediction performance. A model with well-tuned hyperparameters can achieve expected outcomes. The optimized hyperparameters can minimize the loss function and maximize the learning performance. On the contrary, it can lead to an endless training process or raise many stability problems and can be ended without achieving the desired goal.

In this section, several hyperparameters are evaluated in the DQN-assisted PHO management solution, which are discount factor γ, the capacity of experience replay memory, ReLU and Leaky ReLU activation functions.

6.4.4.1 Discount Factor γ in DQN

❖ **Experiment Objective**:

Discount factor γ is a significant hyperparameter in RL, which is introduced in Section 2.4.2.1. It helps with proving the convergence of a RL algorithm. If $\gamma = 0$, the agent only concerns the immediate rewards. If $\gamma = 1$, the agent cares for all future rewards. Tuning the discount factor implies a trade-off. The higher γ the higher possibility of ensuring average optimality for discounted-optimal policies, but the bigger the computational costs.

❖ **Topology and Parameters:**

The selected topology is depicted in Figure 6.2. All the parameters are listed in Table 6.12, except for the parameter of interest: capacity of experience replay memory. Two different discount factor values are evaluated in these experiments, which are 0.95 and 0.99.

Table 6.12 Parameters for Evaluation of Discount Factor

	Parameter Name	Values
NS3 Parameters	**Number of Episodes**	2,000
	Simulation Duration (Per Episode)	60 seconds
	Number of BSs	6
	Number of UEs	1
	UE's Mobility Model	ConstantVelocityMobilityModel
	UE's Velocity	UE: 27.78 m/s (100 km/h)
	UE's Coordinate	UE: (500, 0, 1.6)
DQN Hyperparameters	**Replay Memory Capacity**	50,000

❖ **Results:**

The offline learning performance results are demonstrated in Figure 6.14 for γ = 0.95, and Figure 6.15 for γ = 0.99. The online prediction results are summarized in Table 6.13. It can be seen that:

- The discount factor γ does not affect the execution time.
- The discount factor γ has significant effects on an agent's learning performance. γ = 0.99 outperforms in the proposed solution.
- In the online prediction, the agent trained with γ = 0.95 can only achieve PHO-SR at 50% with a reward value of 10. In contrast, the agent trained with γ = 0.99 can maximize the PHO-SR at 100% with the reward value of 65.

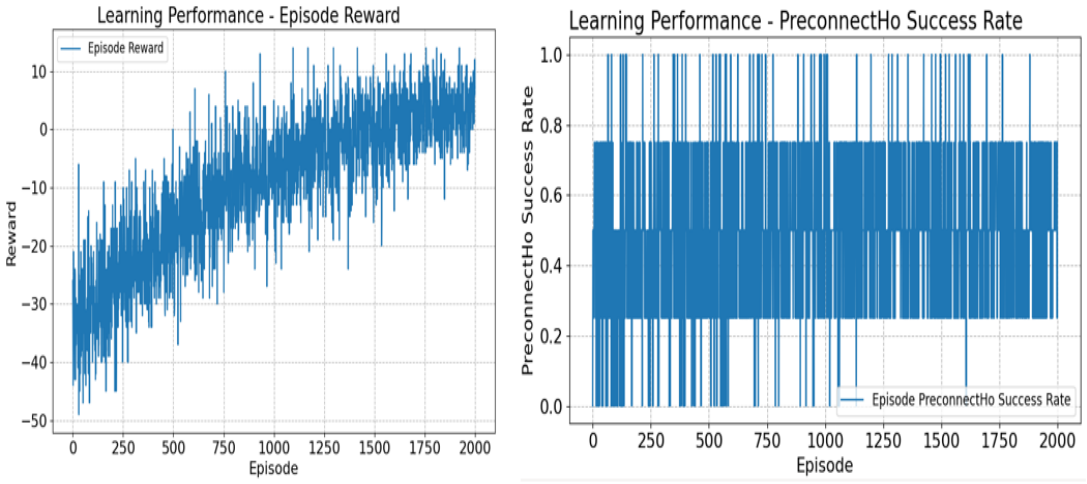


Figure 6.14 Learning Performance - Discount Factor γ = 0.95

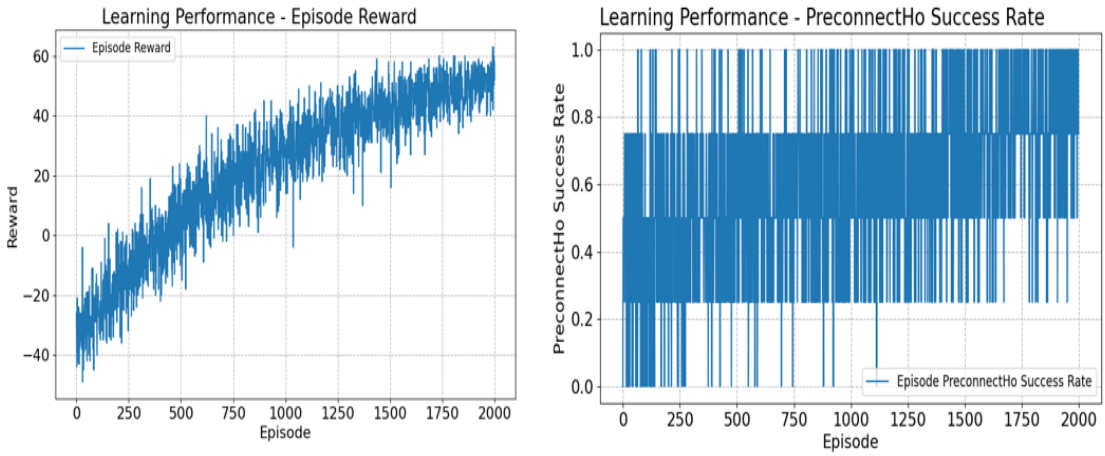


Figure 6.15 Learning Performance - Discount Factor γ = 0.99

Table 6.13 Impacts of Discount Factor

Experiment #	Hyperparameter	Offline Training		Online Prediction Results	
	Discount Factor	Execution Time (hours)		PHO-SR	Reward
#1	0.95	13.53		50%	10
#2	0.99	13.95		100%	65

6.4.4.2 Capacity of Experience Replay Memory in DQN

❖ **Experiment Objective**:

As highlighted in Section 2.4.3.2, the experience replay memory is one of the core techniques in the proposed DQN-based solution. The capacity of experience replay memory determines the total number of transition samples to be stored and further used for model training. The empirical evidence in [57] shows that the replay memory size is a task-dependent hyperparameter. The agent is sensitive to the replay memory size in a complex learning system. The optimal memory size can improve data efficiency and stabilize the training of a neural network. This experiment aims to evaluate the impact of the replay memory size in the proposed DQN-associated PHO management.

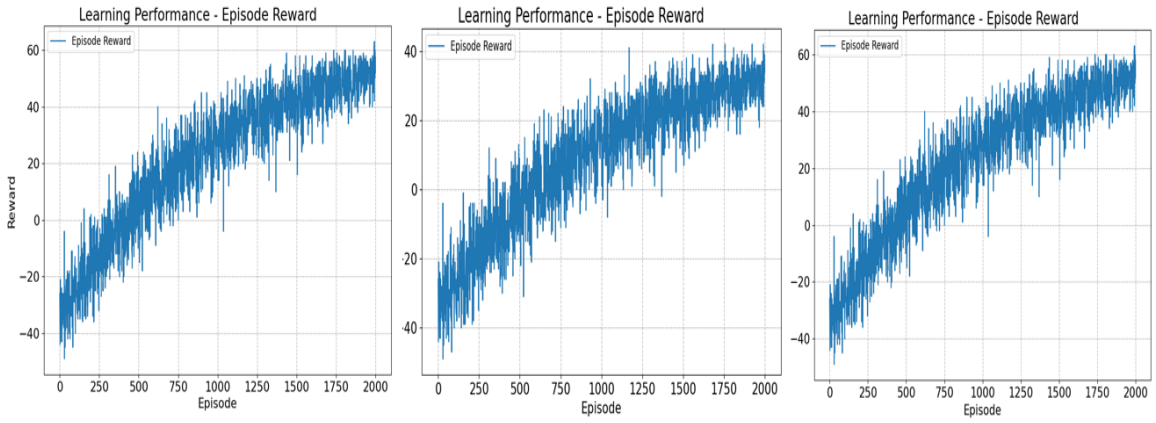
❖ **Topology and Parameters:**

The deployed topology is depicted in Figure 6.2. All the parameters are the same as that of Table 6.12, except for the parameter of interest: capacity of experience replay memory. The discount factor used in this experiment is 0.99, which is selected based on the empirical results in Table 6.13. Three capacity values are evaluated in these experiments, which are 10, 000, 30,000 and 50,000.

❖ **Results:**

The comparison of offline learning performances evaluated by reward are demonstrated in Figure 6.16. Figure 6.17 is the comparison of the PHO-SRs. The online prediction results are illustrated in Table 6.14. It shows that:

- The capacity values of 10,000 and 50,000 lead to the same prediction results: PHO-SR is 100%, and reward value is 65. The performances on both metrics are better than that of 30,000.
- When increasing the capacity of experience replay memory, extra execution times are needed in model training, but the impact is negligible.



(a) capacity = 10,000 (b) capacity = 30,000 (c) capacity = 50,000

Figure 6.16 Learning Performance Comparison of Capacity: Reward

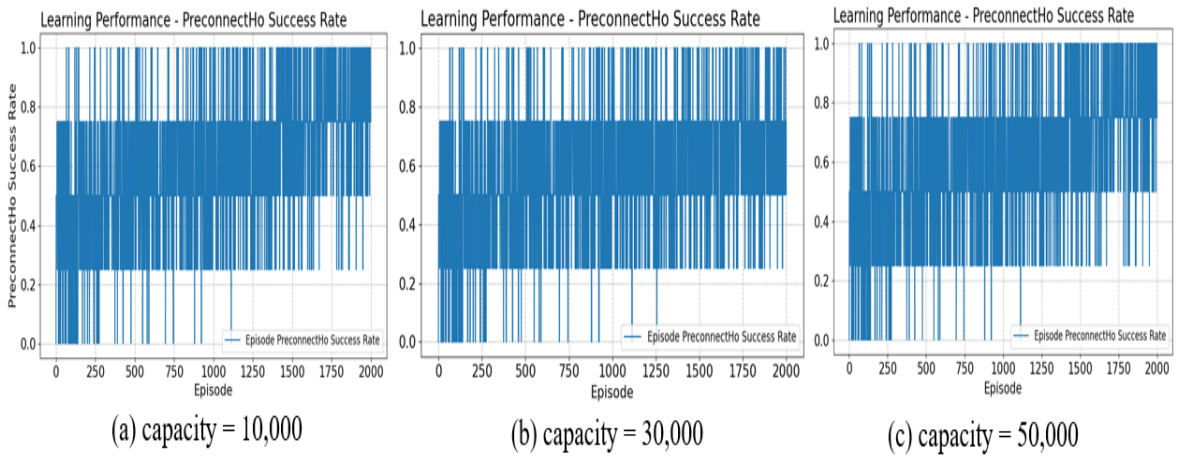


Figure 6.17 Learning Performance Comparison of Capacity: PHO-SR

Table 6.14 Impacts of Replay Memory Capacity

Experiment #	Hyperparameters		Offline Training		Online Prediction Results	
	Discount Factor	Capacity of Replay Memory	Execution Time (hours)		PHO-SR	Reward
#1	0.99	10,000	13.51		100%	65
#2	0.99	30,000	13.79		75%	42
#3	0.99	50,000	13.95		100%	65

6.4.4.3 Activation Function in DNN: ReLU vs. Leaky ReLU

In the proposed DQN-based PHO management, a two-feature state vector {RSRQs and RSRQ change rates} is considered as the observations of the environment, and used for training the agent. The state vector is defined as Equation 4-2. The RSRQs are negative values, typically in the range of -19 to -3dB, but the RSRQ change rates can be either positive or negative. The advantage of the Leaky ReLU activation function compared with ReLU is highlighted in Section 4.2.2.1.

❖ **Experiment Objective**:

This experiment aims to compare the learning performance of ReLU and Leaky ReLU activation factions in the proposed multi-agent DQN-assisted PHO solution.

❖ **Topology and Parameters:**

The topology is depicted in Figure 6.2. All the parameters are listed in Table 6.15, except for the parameter of interest: ReLU and Leaky ReLU activation function; the parameter α in Leaky ReLU.

Table 6.15 Parameters for Activation Function Evaluation

	Parameter Name	Values
NS3 Parameters	**Number of Episodes**	2,000
	Simulation Duration (Per Episode)	60 seconds
	Number of BSs	6
	Number of UEs	2
	UEs' Mobility Model	ConstantVelocityMobilityModel
	UEs' Velocities	UE1: 27.78 m/s (100 km/h) UE2: 29.17 m/s (105 km/h)
	UEs' Coordinates	UE1: (500, 0, 1.6) UE2: (550, 0, 1.6)
DQN Hyperparameters	**Discount Factor**	0.99
	Replay Memory Capacity	50,000

❖ **Results:**

The results are summarized in Table 6.16. It shows that:

Table 6.16 Performance Comparison: ReLU vs. Leaky ReLU

Experiment #	Hyperparameters		Online Prediction Results			
			UE1 (100km/h)		UE2 (105 km/h)	
	Activation Function		PHO - SR	Reward	PHO - SR	Reward
# 1	Leaky ReLU (α = 0.01)		75%	27	100%	61
# 2	Leaky ReLU (α = 0.1)		75%	42	100%	67
# 3	Leaky ReLU (α = 0.2)		100%	77	100%	67
# 4	ReLU		100%	65	100%	65

- The parameter α in Leaky ReLU activation function impacts the prediction accuracy, and it is task-dependent.
- For UE1, Leaky ReLU activation function with parameter α = 0.2 outperforms others on both reward and PHO-SR.

- For UE2, all the four experiments can achieve PHO-SR at 100%; Leaky ReLU activation function with parameter $\alpha = 0.1$ and 0.2 perform better on reward.
- ReLU activation function can also be considered as a good choice, because both UEs can achieve PHO-SR at 100%.

6.5 Summary

The evaluation results obtained from various experiments described in this chapter demonstrated that the proposed PHO handover technique is achievable, and the MADRL-assisted solution delivered a flexible and intelligent PHO management solution. The results show that the devised solution is suitable for the problem and the performance remains consistent.

Some areas for improvement were identified during the experiments which include the reward function refactoring to explicitly encourage a better policy on best T-BS selection, hyperparameters optimization to accelerate the learning process and improve the performance, and new techniques for improving the learning efficiency.

Chapter 7: Conclusions and Future Work

7.1 Summary and Conclusions

The heterogeneous services with consistent QoS requirements in 5G networks bring more challenges to cellular networks. Handover management is an essential and significant functionality in cellular networks, and it is one of the challenges for URLLC. Enhanced handover mechanisms have been attracting widespread interests from both the research community and the networking industry. The design and development of an approach that deals with efficient and effective handovers and the evaluation of system performance through scalable and realistic simulations are indeed recognized as significant topics to be properly investigated.

In this thesis, both objectives of the design and evaluation of a handover methodology using simulation were achieved. The proposed PHO aimed to increase the handover success rate by integrating three main techniques, which are MBB scheme, candidate T-BS(s) pre-connection and early data duplicating-forwarding-buffering during handover execution. In addition, solutions based on DRL and MADRL were designed and evaluated. The DRL-based approach satisfied the objective of the selection of candidate T-BS(s) to maximize the individual UE's PHO success rate. The MADRL-based solution further extended the DRL-based UE-associated solution to a multiple UEs scenario in which the agents learned simultaneously and independently in a distributed manner.

In order to provide the proposed algorithms with an environment as realistic as possible, the implementation and evaluation of the proposed solutions were performed with the NS3 simulator and its extension tool of NS3-Gym. The integration of the RL framework

with realistic network simulations provides a powerful tool for investigating complex network scenarios.

A detailed and long trial-and-error session has been performed in order to identify the more optimal parameters and reward function for the DQN algorithm. The performance evaluation and experimental results seemed promising based on the facts that results of the PHO success rate were consistent and all agents were able to converge successfully within the interval of each simulation. In spite of some limitations in the proposed approach, it has potential to meet performance challenges and requirements of 5G networks.

7.2 Future Research Directions

Although the maximum reward achieved is consistent, a threat to validity can be observed from the research, which should be considered and addressed in future works.

First of all, the DQN-based and MADRL-based PHO management did not consider resource constraints at T-BSs. Since the admission control can be configured in T-BS, shown in Figure 4.1, so if no resource is available, the pre-connect request should be rejected and the PHO success rate may be reduced due to the fact that resources at T-BSs may be exhausted or UEs may compete for the limited available resources.

Secondly, the experiments were conducted based on a small number of UEs due to the amount of training time needed, as demonstrated in Section 6.4.1. A large number of UEs and a wider range of velocities could result in higher variations on the results. Thirdly, the two factors of RSRQs and RSRQ change rates in the state vector with equal weights were considered for the decision making in the proposed DQN-assisted PHO solution. If different weights or other factors need to be incorporated, the results need to be re-evaluated.

Thirdly, the choice of task-dependent hyperparameters of the DQN algorithm could affect the learning performance. The exploration experiments, in terms of discount factor, the experience replay memory capacity, and activation function were conducted in Section 6.4.4. However, the optimization of some other hyperparameters should also be considered.

There are potential areas that warrant further research for PHO, especially those regarding reward function refactoring and hyperparameter optimization to accelerate the training process, in order to maximize reward with the minimum number of episodes.

In addition, further research is suggested to consider improving the learning efficiency by exploring new methodology or other DRL algorithms. For example, the prioritized experience replay [98] can improve the learning efficiency of the experience replay in DQN applications; Double Deep Q-Networks can improve over DQN in convergence, value accuracy, policy quality, and learning stability [99] .

Furthermore, the proposed PHO focused on mobility management and applied the simplest MADRL approach to the optimization problem, which involves independence of UEs learning in a distributed manner in the system. Further research is suggested to integrate resource management and QoS into PHO to enable competition among UEs for decision making.

Appendices

Appendix A Standardized QCI Characteristics [29]

QCI	Resource Type	Priority Level	Packet Delay Budget (NOTE 13)	Packet Error Loss Rate (NOTE 2)	Example Services
1 (NOTE 3)	GBR	2	100 ms (NOTE 1, NOTE 11)	10^{-2}	Conversational Voice
2 (NOTE 3)		4	150 ms (NOTE 1, NOTE 11)	10^{-3}	Conversational Video (Live Streaming)
3 (NOTE 3, NOTE 14)		3	50 ms (NOTE 1, NOTE 11)	10^{-3}	Real Time Gaming, V2X messages Electricity distribution - medium voltage (e.g. clause 7.2.2 of TS 22.261 [51]) Process automation - monitoring (e.g. clause 7.2.2 of TS 22.261 [51])
4 (NOTE 3)		5	300 ms (NOTE 1, NOTE 11)	10^{-6}	Non-Conversational Video (Buffered Streaming)
65 (NOTE 3, NOTE 9, NOTE 12)		0.7	75 ms (NOTE 7, NOTE 8)	10^{-2}	Mission Critical user plane Push To Talk voice (e.g., MCPTT)
66 (NOTE 3, NOTE 12)		2	100 ms (NOTE 1, NOTE 10)	10^{-2}	Non-Mission-Critical user plane Push To Talk voice
67 (NOTE 3, NOTE 12)		1.5	100 ms (NOTE 1, NOTE 10)	10^{-3}	Mission Critical Video user plane
75 (NOTE 14)		2.5	50 ms (NOTE 1)	10^{-2}	V2X messages
71		5.6	150ms (NOTE 1, NOTE 16)	10^{-6}	"Live" Uplink Streaming (e.g. TS 26.238 [53])
72		5.6	300ms (NOTE 1, NOTE 16)	10^{-4}	"Live" Uplink Streaming (e.g. TS 26.238 [53])
73		5.6	300ms (NOTE 1, NOTE 16)	10^{-8}	"Live" Uplink Streaming (e.g. TS 26.238 [53])
74		5.6	500ms (NOTE 1, NOTE 16)	10^{-8}	"Live" Uplink Streaming (e.g. TS 26.238 [53])
76		5.6	500ms (NOTE 1, NOTE 16)	10^{-4}	"Live" Uplink Streaming (e.g. TS 26.238 [53])
5 (NOTE 3)	Non-GBR	1	100 ms (NOTE 1, NOTE 10)	10^{-6}	IMS Signalling
6 (NOTE 4)		6	300 ms (NOTE 1, NOTE 10)	10^{-6}	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
7 (NOTE 3)		7	100 ms (NOTE 1, NOTE 10)	10^{-3}	Voice, Video (Live Streaming) Interactive Gaming
8 (NOTE 5)		8	300 ms (NOTE 1)	10^{-6}	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
9 (NOTE 6)		9			

QCI	Resource Type	Priority Level	Packet Delay Budget (NOTE B1)	Packet Error Loss Rate (NOTE B2)	Maximum Data Burst Volume (NOTE B1)	Data Rate Averaging Window	Example Services
82 (NOTE B6)	GBR	1.9	10 ms (NOTE B4)	10^{-4} (NOTE B3)	255 bytes	2000 ms	Discrete Automation (TS 22.278 [38], clause 8 bullet g, and TS 22.261 [51], table 7.2.2-1, "small packets")
83 (NOTE B6)		2.2	10 ms (NOTE B4)	10^{-4} (NOTE B3)	1354 bytes (NOTE B5)	2000 ms	Discrete Automation (TS 22.278 [38], clause 8 bullet g, and TS 22.261 [51], table 7.2.2-1, "big packets")
84 (NOTE B6)		2.4	30 ms (NOTE B7)	10^{-5} (NOTE B3)	1354 bytes (NOTE B5)	2000 ms	Intelligent Transport Systems (TS 22.278 [38], clause 8, bullet h, and TS 22.261 [51], table 7.2.2).
85 (NOTE B6)		2.1	5 ms (NOTE B8)	10^{-5} (NOTE B3)	255 bytes	2000 ms	Electricity Distribution- high voltage (TS 22.278 [38], clause 8, bullet i, and TS 22.261 [51], table 7.2.2 and Annex D, clause D.4.2).

10	9	1100 ms (NOTE 1, NOTE 17)	10^{-6}	Video (Buffered Streaming) TCP-based (e.g. www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.) and any service that can be used over satellite access with these characteristics
69 (NOTE 3, NOTE 9, NOTE 12)	0.5	60 ms (NOTE 7, NOTE 8)	10^{-6}	Mission Critical delay sensitive signalling (e.g., MC-PTT signalling, MC Video signalling)
70 (NOTE 4, NOTE 12)	5.5	200 ms (NOTE 7, NOTE 10)	10^{-6}	Mission Critical Data (e.g. example services are the same as QCI 6/8/9)
79 (NOTE 14)	6.5	50 ms (NOTE 1, NOTE 10)	10^{-2}	V2X messages
80 (NOTE 3)	6.8	10 ms (NOTE 10, NOTE 15)	10^{-6}	Low latency eMBB applications (TCP/UDP-based); Augmented Reality

Appendix B E-UTRA Channel Numbers [94]

E-UTRA Operating Band	Downlink			Uplink		
	F_{DL_low} (MHz)	$N_{offs-DL}$	Range of N_{DL}	F_{UL_low} (MHz)	$N_{offs-UL}$	Range of N_{UL}
1	2110	0	0 – 599	1920	18000	18000 – 18599
2	1930	600	600 – 1199	1850	18600	18600 – 19199
3	1805	1200	1200 – 1949	1710	19200	19200 – 19949
4	2110	1950	1950 – 2399	1710	19950	19950 – 20399
5	869	2400	2400 – 2649	824	20400	20400 – 20649
6	875	2650	2650 – 2749	830	20650	20650 – 20749
7	2620	2750	2750 – 3449	2500	20750	20750 – 21449
8	925	3450	3450 – 3799	880	21450	21450 – 21799
9	1844.9	3800	3800 – 4149	1749.9	21800	21800 – 22149
10	2110	4150	4150 – 4749	1710	22150	22150 – 22749
11	1475.9	4750	4750 – 4949	1427.9	22750	22750 – 22949
12	729	5010	5010 – 5179	699	23010	23010 – 23179
13	746	5180	5180 – 5279	777	23180	23180 – 23279
14	758	5280	5280 – 5379	788	23280	23280 – 23379
…						
17	734	5730	5730 – 5849	704	23730	23730 – 23849
18	860	5850	5850 – 5999	815	23850	23850 – 23999
19	875	6000	6000 – 6149	830	24000	24000 – 24149
20	791	6150	6150 – 6449	832	24150	24150 – 24449
21	1495.9	6450	6450 – 6599	1447.9	24450	24450 – 24599
22	3510	6600	6600 – 7399	3410	24600	24600 – 25399
23	2180	7500	7500 – 7699	2000	25500	25500 – 25699
24	1525	7700	7700 – 8039	1626.5	25700	25700 – 26039
25	1930	8040	8040 – 8689	1850	26040	26040 – 26689
26	859	8690	8690 – 9039	814	26690	26690 – 27039
27	852	9040	9040 – 9209	807	27040	27040 – 27209
28	758	9210	9210 – 9659	703	27210	27210 – 27659
29[2]	717	9660	9660 – 9769	N/A		
30	2350	9770	9770 – 9869	2305	27660	27660 – 27759
31	462.5	9870	9870 – 9919	452.5	27760	27760 – 27809
32[2]	1452	9920	9920 – 10359	N/A		
33	1900	36000	36000 – 36199	1900	36000	36000 – 36199
34	2010	36200	36200 – 36349	2010	36200	36200 – 36349
35	1850	36350	36350 – 36949	1850	36350	36350 – 36949
36	1930	36950	36950 – 37549	1930	36950	36950 – 37549
37	1910	37550	37550 – 37749	1910	37550	37550 – 37749
38	2570	37750	37750 – 38249	2570	37750	37750 – 38249
39	1880	38250	38250 – 38649	1880	38250	38250 – 38649
40	2300	38650	38650 – 39649	2300	38650	38650 – 39649
41	2496	39650	39650 –41589	2496	39650	39650 –41589
42	3400	41590	41590 – 43589	3400	41590	41590 – 43589
43	3600	43590	43590 – 45589	3600	43590	43590 – 45589
44	703	45590	45590 – 46589	703	45590	45590 – 46589
45	1447	46590	46590 – 46789	1447	46590	46590 – 46789
46	5150	46790	46790 – 54539	5150	46790	46790 – 54539
47	5855	54540	54540 - 55239	5855	54540	54540 – 55239
48	3550	55240	55240 – 56739	3550	55240	55240 – 56739
49	3550	56740	56740 – 58239	3550	56740	56740 – 58239
50	1432	58240	58240 - 59089	1432	58240	58240 - 59089
51	1427	59090	59090 - 59139	1427	59090	59090 - 59139
52	3300	59140	59140 - 60139	3300	59140	59140 - 60139
53	2483.5	60140	60140 - 60254	2483.5	60140	60140 - 60254
…						
64	Reserved					
65	2110	65536	65536 – 66435	1920	131072	131072 – 131971
66[5]	2110	66436	66436 – 67335	1710	131972	131972 – 132671
67[2]	738	67336	67336 – 67535	N/A		
68	753	67536	67536 - 67835	698	132672	132672 - 132971
69[2]	2570	67836	67836 - 68335	N/A		
70[6]	1995	68336	68336 - 68585	1695	132972	132972 - 133121
71	617	68586	68586 - 68935	663	133122	133122 - 133471
72	461	68936	68936 - 68985	451	133472	133472 - 133521

73	460	68986	68986 - 69035	450	133522	133522 - 133571
74	1475	69036	69036 - 69465	1427	133572	133572 - 134001
75[2]	1432	69466	69466 - 70315	N/A		
76[2]	1427	70316	70316 - 70365	N/A		
85	728	70366	70366 - 70545	698	134002	134002 - 134181
87	420	70546	70546 - 70595	410	134182	134182 - 134231
88	422	70596	70596 - 70645	412	134232	134232 - 134281
103	757	70646	70646 – 70655	787	134282	134282 – 134291

NOTE 1: The channel numbers that designate carrier frequencies so close to the operating band edges that the carrier extends beyond the operating band edge shall not be used. This implies that the first 7, 15, 25, 50, 75 and 100 channel numbers at the lower operating band edge and the last 6, 14, 24, 49, 74 and 99 channel numbers at the upper operating band edge shall not be used for channel bandwidths of 1.4, 3, 5, 10, 15 and 20 MHz respectively.

NOTE 2: Restricted to E-UTRA operation when carrier aggregation is configured.

NOTE 3: For ProSe and V2X the corresponding UL channel number are also specified for the DL for the associated ProSe/V2X operating bands i.e. ProSe_F_{UL} - F_{UL} and ProSe_F_{DL} - F_{UL}; V2X_F_{UL} = F_{DL} and V2X_F_{DL} - F_{UL}.

NOTE 4: Requirements for uplink operations are not specified in this version of the specification.

NOTE 5: The range 2180-2200 MHz of the DL operating band is restricted to E-UTRA operation when carrier aggregation is configured.

NOTE 6: The range 2010-2020 MHz of the DL operating band is restricted to E-UTRA operation when carrier aggregation is configured and TX-RX separation is 300 MHz The range 2005-2020 MHz of the DL operating band is restricted to E-UTRA operation when carrier aggregation is configured and TX-RX separation is 295 MHz.

References

[1] Ericsson, "Ericsson Mobility Report, Nov. 2021" Stockholm, Sweden, *https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/november-2021*, accessed in July, 2022.

[2] M. S. Mollel *et al.*, "A Survey of Machine Learning Applications to Handover Management in 5G and Beyond," *IEEE Access*, vol. 9, pp. 45770–45802, 2021.

[3] M. Tayyab, X. Gelabert, and R. Jantti, "A Survey on Handover Management: From LTE to NR," *IEEE Access*, vol. 7, pp. 118907–118930, 2019.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Second. MIT press, 2018.

[5] A. Feriani and E. Hossain, "Single and Multi-Agent Deep Reinforcement Learning for AI-Enabled Wireless Networks: A Tutorial," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 2, pp. 1226–1252, Apr. 2021.

[6] N. C. Luong *et al.*, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, Oct. 2019.

[7] A. Lazaridis, A. Fachantidis, and I. Vlahavas, "Deep Reinforcement Learning: A State-of-the-Art Walkthrough," *Journal of Artificial Intelligence Research*, vol. 69, pp. 1421–1471, Dec. 2020.

[8] K. Zhang, Z. Yang, and T. Başar, "Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms," *arXiv preprint arXiv:1911.10635*, Nov. 2019.

[9] L. Busoniu, R. Babuska, and B. de Schutter, "A Comprehensive Survey of Multiagent Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[10] A. M. Ibrahim, K.-L. A. Yau, Y.-W. Chong, and C. Wu, "Applications of Multi-Agent Deep Reinforcement Learning: Models and Algorithms," *Applied Sciences*, vol. 11, no. 22, p. 10870, Nov. 2021.

[11] 3rd Generation Partnership Project (3GPP), "TR 138 913 - V17.0.0 - 5G; Study on scenarios and requirements for next generation access technologies (3GPP TR 38.913 version 17.0.0 Release 17)," 2022.

[12] M. Lauridsen, L. C. Gimenez, I. Rodriguez, T. B. Sorensen, and P. Mogensen, "From LTE to 5G for Connected Mobility," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 156–162, Mar. 2017.

[13] 3rd Generation Partnership Project (3GPP), "TS 136 881 - V14.0.0 - LTE; Study on Latency Reduction Techniques for LTE." Jun. 2016.

[14] I. Shayea, M. Ergen, M. Hadri Azmi, S. Aldirmaz Colak, R. Nordin, and Y. I. Daradkeh, "Key Challenges, Drivers and Solutions for Mobility Management in 5G Networks: A Survey," *IEEE Access*, vol. 8, pp. 172534–172552, 2020.

[15] Xinjie Yang, S. Ghaheri-Niri, and R. Tafazolli, "Evaluation of soft handover algorithms for UMTS," in *Proceedings of 11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), 2020,* pp. 772–776.

[16] L. Buşoniu, R. Babuška, and B. de Schutter, "Multi-agent Reinforcement Learning: An Overview," *Innovations in multi-agent systems and applications-1*, Springer, pp. 183–221, 2010.

[17] 3rd Generation Partnership Project (3GPP), "TS 138 300 - V17.0.0 - 5G; NR; NR and NG-RAN Overall description; Stage-2 (3GPP TS 38.300 version 17.0.0 Release 17)," 2022.

[18] S. Ahmadi, "IEEE 802.16m System Operation and State Diagrams," *Mobile WiMAX*, pp. 97–147, Jan. 2011.

[19] Mihret, Estifanos & Haile, Getamesay, "4G, 5G, 6G, 7G and Future Mobile Technologies," *American Journal of Computer Science and Technology*, vol. 9, no. 2, pp. 75, 2021.

[20] 3rd Generation Partnership Project (3GPP), "TS 136 300 - V17.0.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (3GPP TS 36.300 version 17.0.0 Release 17)," 2022.

[21] P. Lin, J. Hou, and X. Xu, "Research on 5G SA Mobility Management," in *Proceedings of International Wireless Communications and Mobile Computing (IWCMC)*, 2021, pp. 503–507.

[22] 3rd Generation Partnership Project (3GPP), "TS 136 133 - V17.5.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management (3GPP TS 36.133 version 17.5.0 Release 17)," 2022.

[23] 3rd Generation Partnership Project (3GPP), "TS 138 133 - V17.5.0 - 5G; NR; Requirements for support of radio resource management (3GPP TS 38.133 version 17.5.0 Release 17)," 2022.

[24] 3rd Generation Partnership Project (3GPP), "TS 136 420 - V17.0.0 - LTE; Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 general aspects and principles (3GPP TS 36.420 version 17.0.0 Release 17)," 2022.

[25] 3rd Generation Partnership Project (3GPP), "TS 136 423 - V17.0.0 - LTE; Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 Application Protocol (X2AP) (3GPP TS 36.423 version 17.0.0 Release 17)," 2022.

[26] 3rd Generation Partnership Project (3GPP), "TS 138 420 - V17.0.0 - 5G; NG-RAN; Xn general aspects and principles (3GPP TS 38.420 version 17.0.0 Release 17)," 2022.

[27] F. Afroz, R. Subramanian, R. Heidary, K. Sandrasegaran, and S. Ahmed, "SINR, RSRP, RSSI and RSRQ Measurements in Long Term Evolution Networks," *International Journal of Wireless & Mobile Networks*, vol. 7, no. 4, pp. 113–123, Aug. 2015.

[28] T. Akhtar, C. Tselios, and I. Politis, "Radio resource management: approaches and implementations from 4G to 5G and beyond," *Wireless Networks*, vol. 27, no. 1, pp. 693–734, Jan. 2021.

[29] TSGS, "TS 123 203 - V17.2.0 - Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Policy and charging control architecture (3GPP TS 23.203 version 17.2.0 Release 17)," 2022.

[30] C.-C. Lin, K. Sandrasegaran, H. A. M. Ramli, and R. Basukala, "Optimized Performance Evaluation of LTE Hard Handover Algorithm with Average RSRP Constraint," *arXiv preprint arXiv: 1105.0234*, May 2011.

[31] X. Chen, K. T. Kim, B. Lee, and H. Y. Youn, "DIHAT: Differential Integrator Handover Algorithm with TTT window for LTE-based systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 162, Dec. 2014.

[32] L. C. Gimenez, P. H. Michaelsen, K. I. Pedersen, T. E. Kolding, and H. C. Nguyen, "Towards Zero Data Interruption Time with Enhanced Synchronous Handover," in *Proceedings of 2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, 2017, pp. 1–6.

[33] N. Aljeri and A. Boukerche, "Smart and Green Mobility Management for 5G-enabled Vehicular Networks," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, Mar. 2022.

[34] 3rd Generation Partnership Project (3GPP), "TS 136 331 - V17.0.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (3GPP TS 36.331 version 17.0.0 Release 17)," 2022.

[35] 3rd Generation Partnership Project (3GPP), "TS 138 331 - V17.0.0 - 5G; NR; Radio Resource Control (RRC); Protocol specification (3GPP TS 38.331 version 17.0.0 Release 17)," 2022.

[36] H. Hendrawan, A. R. Zain, and S. Lestari, "Performance Evaluation of A2-A4-RSRQ and A3-RSRP Handover Algorithms in LTE Network," *Jurnal Elektronika dan Telekomunikasi*, vol. 19, no. 2, Dec. 2019.

[37] 3rd Generation Partnership Project (3GPP), "TS 136 214 - V17.0.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer; Measurements (3GPP TS 36.214 version 17.0.0 Release 17)," 2022.

[38] M. Tayyab, G. P. Koudouridis, and X. Gelabert, "A simulation study on LTE handover and the impact of cell size," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 263, pp. 398–408, 2019.

[39] H. Martikainen, I. Viering, A. Lobinger, and T. Jokela, "On the Basics of Conditional Handover for 5G Mobility," in *Proceedings of 2018 IEEE 29th Annual International*

Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2018, pp. 1–7.

[40] H.-S. Park, Y. Lee, T.-J. Kim, B.-C. Kim, and J.-Y. Lee, "Handover Mechanism in NR for Ultra-Reliable Low-Latency Communications," *IEEE Network*, vol. 32, no. 2, pp. 41–47, Mar. 2018.

[41] U. Challita, H. A. Ryden, and H. Tullberg, "When Machine Learning Meets Wireless Cellular Networks: Deployment, Challenges, and Applications," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 12-18, June 2020.

[42] B. Mahesh, "Machine Learning Algorithms-A Review," *International Journal of Science and Research*, 2018.

[43] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4. pp. 2392–2431, Oct. 01, 2017.

[44] Z. Wang, L. Li, Y. Xu, H. Tian, and S. Cui, "Handover Control in Wireless Systems via Asynchronous Multiuser Deep Reinforcement Learning," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4296–4307, Dec. 2018.

[45] G. Thomas, "Markov Decision Processes," *https://ai.stanford.edu/*, accessed in July, 2022.

[46] M. van Otterlo and M. Wiering, "Reinforcement Learning and Markov Decision Processes", *Wiering, M., van Otterlo, M. (eds) Reinforcement Learning. Adaptation, Learning, and Optimization,* vol 12. Springer, 2012.

[47] F. Studzinski Perotto and L. Vercouter, "Tuning the Discount Factor in Order to Reach Average Optimality on Deterministic MDPs," *Bramer, M., Petridis, M. (eds) Artificial Intelligence XXXV. SGAI 2018. Lecture Notes in Computer Science,* vol 11311. Springer, 2018.

[48] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, May 1996.

[49] H. Zhang and T. Yu, "Taxonomy of reinforcement learning algorithms," *Deep Reinforcement Learning: Fundamentals, Research and Applications*, pp. 125–133, Jan. 2020.

[50] J. Gläscher, N. Daw, P. Dayan, and J. P. O'Doherty, "States versus rewards: Dissociable neural prediction error signals underlying model-based and model-free reinforcement learning," *Neuron*, vol. 66, no. 4, pp. 585–595, May 2010.

[51] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[52] M. Wunder, M. Littman, and M. Babes, "Classes of Multiagent Q-learning Dynamics with-greedy Exploration.", in *Proceedings of the 27th International Conference on Machine Learning (ICML),*2010, pp.1167-1174.

[53] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A Brief Survey of Deep Reinforcement Learning," *arXiv preprint arXiv: 1708.05866,* Aug. 2017.

[54] X. You, X. Li, Y. Xu, H. Feng, and J. Zhao, "Toward Packet Routing with Fully-distributed Multi-agent Deep Reinforcement Learning," in *Proceedings of 2019 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, 2019, pp. 1–8.

[55] L.-J. Lin, "Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching," *Machine learning*, vol. 8, no. 3-4, pp. 293–321, 1992.

[56] W. Fedus, P. Ramachandran, R. Agarwal, Y. Bengio, H. Larochelle, M. Rowland, and W. Dabney, "Revisiting Fundamentals of Experience Replay," *arXiv preprint arXiv:* 2007.06700, 2020.

[57] S. Zhang and R. S. Sutton, "A Deeper Look at Experience Replay," *arXiv preprint arXiv: 1712.01275*, 2017.

[58] P. Stone and M. Veloso, "Multiagent Systems: A Survey from a Machine Learning Perspective," *Autonomous Robots 2000*, vol. 8, no. 3, pp. 345–383, Jun. 2000.

[59] M. Tan, "Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents," in *Proceedings of Machine Learning*, 1993, pp. 330–337.

[60] L. Matignon, G. J. Laurent, N. Le Fort-Piat. "Independent reinforcement learners in cooperative Markov games: a survey regarding coordination problems," *Knowledge Engineering Review, Cambridge University Press (CUP)*, vol. 27, no. 1, pp. 1–31, 2012.

[61] T. R. Henderson, M. Lacage, and G. F. Riley, "Network Simulations with the ns-3 Simulator," in *Proceedings of SIGCOMM Demonstration,* 2008, vol. 14, no. 14, p.527.

[62] G. F. Riley and T. R. Henderson, "The ns-3 Network Simulator," in *Modeling and Tools for Network Simulation*, Springer, pp. 15–34, 2010.

[63] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," *arXiv preprint arXiv:1606.01540*, Jun. 2016.

[64] P. Gawłowicz and A. Zubow, "ns3-gym: Extending OpenAI Gym for Networking Research," *arXiv preprint arXiv:1810.03943*, Oct. 2018.

[65] P. Gawłowicz and A. Zubow, "ns-3 meets OpenAI Gym: The Playground for Machine Learning in Networking Research," in *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2019, pp.113–120.

[66] P. Hintjens, "ZeroMQ: Messaging for Many Applications." *O'Reilly Media, Inc.*, 2013.

[67] H. Yin *et al.*, "ns3-ai: Fostering Artificial Intelligence Algorithms for Networking Research" in *Proceedings of Workshop on ns-3 – WNS3 2020*, June 2020.

[68] C. Wang, Z. Zhao, Q. Sun, and H. Zhang, “Deep Learning-Based Intelligent Dual Connectivity for Mobility Management in Dense Network,” in *Proceedings of 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall),* 2018, pp. 1-5.

[69] L. Yan *et al.*, “Machine Learning-Based Handovers for Sub-6 GHz and mmWave Integrated Vehicular Networks,” in *Proceedings of IEEE Transactions on Wireless Communications*, 2019, vol. 18, no. 10, pp. 4873–4885.

[70] N. Aljeri and A. Boukerche, “An Efficient Handover Trigger Scheme for Vehicular Networks Using Recurrent Neural Networks,” in *Proceedings of the 15th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, 2019, pp.85–91.

[71] S. Khunteta and A. K. R. Chavva, “Deep Learning Based Link Failure Mitigation,” in *Proceedings of the 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 806–811.

[72] Y. Koda, K. Yamamoto, T. Nishio, and M. Morikura, “Reinforcement learning based predictive handover for pedestrian-aware mmWave networks,” in *Proceedings of INFOCOM 2018 - IEEE Conference on Computer Communications Workshops*, 2018, pp. 692–697.

[73] Y. Chen, X. Lin, T. Khan, and M. Mozaffari, “Efficient Drone Mobility Support Using Reinforcement Learning,” in *Proceedings of 2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1-6.

[74] Y. Koda, K. Nakashima, K. Yamamoto, T. Nishio, and M. Morikura, “Handover Management for mmWave Networks with Proactive Performance Prediction Using Camera Images and Deep Reinforcement Learning,” in *Proceedings of IEEE Transactions on Cognitive Communications and Networking*, 2020, vol. 6, no. 2, pp. 802-816.

[75] M. S. Mollel *et al.*, “Intelligent handover decision scheme using double deep reinforcement learning,” *Physical Communication*, vol. 42, Oct. 2020.

[76] M. Sana, A. de Domenico, E. C. Strinati, and A. Clemente, “Multi-Agent Deep Reinforcement Learning for Distributed Handover Management in Dense MmWave Networks,” in *Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8976–8980.

[77] 3rd Generation Partnership Project (3GPP), “TS 123 003 - V17.5.0 - Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Numbering, addressing and identification (3GPP TS 23.003 version 17.5.0 Release 17),” 2022.

[78] 3rd Generation Partnership Project (3GPP), “TR 121 905 - V17.1.0 - Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Vocabulary for 3GPP Specifications (3GPP TR 21.905 version 17.1.0 Release 17),” 2022.

[79] 3rd Generation Partnership Project (3GPP), “TS 136 321 - V17.0.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification (3GPP TS 36.321 version 17.0.0 Release 17),” 2022.

[80] C. Pai and K. Potdar, “A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers,” *Article in International Journal of Computer Applications*, vol. 175, no. 4, pp. 975–8887, 2017.

[81] K. v. Katsikopoulos and S. E. Engelbrecht, “Markov decision processes with delays and asynchronous cost collection,” in *Proceedings of IEEE Transactions on Automatic Control*, 2003, vol. 48, no. 4, pp. 568–574.

[82] S. Sharma, S. Sharma, and A. Athaiya, “Activation Functions in Neural Networks,” *International Journal of Engineering Applied Sciences and Technology*, vol. 4, pp. 310–316, 2020.

[83] L. Lu, Y. Shin, Y. Su, and G. E. Karniadakis, “Dying ReLU and Initialization: Theory and Numerical Examples”, *arXiv preprint arXiv:1903.06733*, 2019

[84] A. K. Dubey and V. Jain, “Comparative Study of Convolution Neural Network’s Relu and Leaky-Relu Activation Functions,” *Lecture Notes in Electrical Engineering*, vol. 553, pp. 873–880, 2019.

[85] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier Nonlinearities Improve Neural Network Acoustic Models,” in *Proceedings of ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[86] C. Enyinna Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of Trends in Practice and Research for Deep Learning”, *arXiv preprint arXiv: 1811.03378*, 2018.

[87] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” in *Proceedings of 3rd International Conference on Learning Representations (ICLR)*, 2015.

[88] H. Mao, Z. Gong, and Z. Xiao, “Reward Design in Cooperative Multi-agent Reinforcement Learning for Packet Routing,” *arXiv preprint arXiv: 2003.03433*, 2020.

[89] B. James Lansdell, P. Ravi Prakash, and K. Paul Kording, “Learning to Solve the Credit Assignment Problem,” *arXiv preprint arXiv: 1906.00889*, 2019.

[90] NS-3 project, “NS-3 Design Documentation - Model Library,” *https://www.nsnam.org/docs/models/ns-3-model-library.pdf*, accessed in July, 2022.

[91] NS-3 project, “NS3 Tracing - Tutorial.” *https://www.nsnam.org/docs/tutorial/html/tracing.html*, accessed July, 2022.

[92] M. Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” *arXiv preprint arXiv: 1603.04467*, 2019.

[93] “Keras Documentation,” *https://faroit.com/keras-docs/2.0.2/*, accessed in July, 2022.

[94] 3rd Generation Partnership Project (3GPP), “TS 136 101 - V17.5.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception (3GPP TS 36.101 version 17.5.0 Release 17),” 2022.

[95] H. T. Friis, “A Note on a Simple Transmission Formula,” in *Proceedings of the IRE*, 1946, vol. 34, no. 5, pp. 254–256.

[96] M. Assyadzily, A. Suhartomo, and A. Silitonga, “Evaluation of X2-handover performance based on RSRP measurement with Friis path loss using network simulator version 3 (NS-3),” in *Proceedings of 2014 2nd International Conference on Information and Communication Technology (ICoICT)*, 2014, pp. 436-441.

[97] F. Pardo, A. Tavakoli, V. Levdik, and P. Kormushev, “Time Limits in Reinforcement Learning,” in *Proceedings of the 35th International Conference on Machine Learning (ICML)* 2018, vol. 9, pp. 6443–6452.

[98] T. Schaul, J. Quan, I. Antonoglou, D. Silver, and G. Deepmind, “Prioritized Experience Replay,” *arXiv preprint arXiv: 1511.05952*, 2015.

[99] H. van Hasselt, A. Guez, and D. Silver, “Deep Reinforcement Learning with Double Q-learning,” *arXiv preprint arXiv: 1509.06461*, 2015.