Towards Zero-Touch Management in RAN Slicing for Next-Generation Networks

by

Ohood SABR

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, MARCH 30, 2026

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Dr. Kuljeet Kaur, Thesis supervisor
Département de génie électrique, l'École de Technologie Supérieure

Dr. Georges Kaddoum, Thesis Co-Supervisor
Département de génie électrique, l'École de Technologie Supérieure

Dr. Naouel Moha, Chair, Board of Examiners
Département de génie logiciel et TI, l'École de Technologie Supérieure

Dr. Kim Khoa Nguyen, Member of the Jury
Département de génie électrique, l'École de Technologie Supérieure

Dr. Yousef Shayan, External Examiner
Department of Electrical and Computer Engineering, Concordia University

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON 16, MARCH, 2026

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

To my beloved Dad & Mom,

an ocean of love, courage, and kindness in my life...

ACKNOWLEDGEMENTS

Pursuing a Ph.D. degree is a long and frequently lonely journey that can greatly shape your life. No matter how hard you try and how strong you are, reaching this point would not have been possible without the valuable support, help, and inspiration of several people who I wish to convey my heartfelt gratitude for.

I sincerely thank my supervisors, Prof. Kuljeet Kaur and Prof. Georges Kaddoum, and to express my sincere gratitude for their guidance, generous comments, thoughtful suggestions, and support throughout my Ph.D. journey. I am deeply grateful to both of you for your immense patience. Further appreciation goes to Prof. Kuljeet Kaur for her kindness and understanding, particularly during the most difficult times. Your continuous support and understanding gave me the strength to move forward when things felt impossible.

I am equally thankful to Dr. Mathieu Gratuze, for his continuous support, thoughtful discussions, and willingness to share knowledge. Thank you for every moment you dedicated to support me. Additionally, I am grateful to the entire ETS Library team for their kindness, support, and assistance. Their dedication in providing the necessary resources, books, and equipment greatly contributed to my progress.

A Ph.D. journey is incredibly challenging, and having friends who understand those challenges makes all the difference. I am deeply grateful to my best friends and second family— David Sam, Da-Sheena Fulford, Thi Somphos Lam, Wedad Ibrahim Al-Dulaimi, and Manisha Mahagammulle Gamage—for their unwavering help and encouragement throughout. This journey would also not be complete without my friends in Mesopotamia, the UK, and other parts of the world. Thank you all; your words have been the greatest source of strength throughout my entire path.

Finally, my deepest love and gratitude go to my family for their endless support, wisdom, and encouragement. Your patience and understanding have been invaluable, giving me the strength to reach this stage. Words cannot express how truly grateful I am to you.

Vers une gestion sans intervention dans le découpage du RAN pour les réseaux de nouvelle génération

Ohood SABR

RÉSUMÉ

Les réseaux sans intervention (ZTN) représentent un changement de paradigme de pointe vers une gestion de réseau entièrement automatisée et intelligente. Ils fournissent l'automatisation et l'intelligence nécessaires pour gérer la complexité, l'échelle et le comportement dynamique des systèmes mobiles de nouvelle génération, notamment la sixième génération (6G). L'intelligence artificielle (IA) constitue le cerveau et l'épine dorsale des ZTN. En particulier, les algorithmes d'apprentissage par renforcement profond (DRL) ont démontré un fort potentiel pour améliorer l'efficacité opérationnelle et soutenir la prise de décision intelligente. La gestion sans intervention est particulièrement cruciale dans les technologies transformatrices telles que le découpage de réseau (NS), qui permet la création de réseaux logiques isolés sur une infrastructure physique partagée. Chaque réseau logique est conçu pour répondre à des exigences de qualité de service (QoS) distinctes, ce qui rend le NS indispensable pour les générations mobiles actuelles et futures. Cependant, la mise en œuvre des ZTN dans un cadre NS est associée à plusieurs défis, notamment dans le domaine du réseau d'accès radio (RAN). Parmi les nombreuses difficultés rencontrées dans le domaine du RAN, les trois défis suivants se distinguent comme particulièrement critiques : (i) la gestion de l'allocation des ressources inter- et intra-tranches ; (ii) garantir la sécurité du partage des ressources entre tranches ; et (iii) assurer une gestion coordonnée et simultanée des ressources radio aux niveaux inter- et intra-tranches. Ces défis sont encore accentués par la nature dynamique des canaux sans fil, la rareté des ressources radio, la diversité des accords de niveau de service (SLA), la forte densification des dispositifs, l'arrivée aléatoire du trafic et diverses imperfections du réseau telles que des informations d'état du canal (CSI) imparfaites, un accès multiple par répartition orthogonale de la fréquence (OFDMA) imparfait et des défaillances matérielles (HWI).

Dans ce contexte, la présente thèse apporte plusieurs contributions majeures pour relever ces défis. Premièrement, nous étudions l'allocation de ressources intra-tranche pour le découpage du RAN en introduisant un schéma d'auto-optimisation (SO) efficace pour un système multi-utilisateurs à entrées et sorties multiples (MU-MISO), nommé PABSO-DRL. Le schéma PABSO-DRL proposé gère dynamiquement et conjointement l'allocation de puissance et la formation de faisceaux afin de garantir des débits de données élevés pour le haut débit mobile amélioré (eMBB), tout en assurant la haute fiabilité requise par les communications ultra-fiables à faible latence (uRLLC). Ce schéma est conçu pour gérer des exigences de QoS hétérogènes grâce à une approche de réseau Q profond (DQN) multi-agents, tout en tenant compte des informations d'état du canal (CSI) imparfaites, de l'isolation OFDMA incomplète et de la dynamique temporelle de l'environnement RAN.

Nous nous intéressons ensuite à l'allocation des ressources entre les tranches. À cette fin, nous proposons un schéma auto-optimisé et sécurisé, basé sur un cadre coopératif multi-acteurs-

critiques (CoMA2C), appelé SO-CoMA2C, pour gérer les ressources radio multiples (puissance et bande passante) réparties sur un ensemble de tranches hétérogènes. Ce schéma s'adapte aux fluctuations de la charge de trafic et prend en compte la présence d'interfaces matérielles (HWI) dans un réseau d'accès radio ouvert (O-RAN). L'objectif principal du schéma proposé est de maximiser l'efficacité spectrale tout en garantissant le SLA de chaque tranche. De plus, nous assurons la sécurité de l'allocation en intégrant l'algorithme de chiffrement AES (Advanced Encryption Standard) au schéma proposé.

Enfin, nous développons un cadre auto-optimisé hiérarchique visant à maximiser la qualité de service (QoS) à long terme et l'efficacité spectrale des services hétérogènes. Le cadre proposé adopte une stratégie à deux couches, mise en œuvre par deux schémas de gestion de découpage complémentaires : (i) un schéma coopératif multi-acteurs-critiques (CoMA2C) qui alloue la puissance et la bande passante entre les tranches hétérogènes sur une longue période, et (ii) un schéma multi-agents DQN (MADQN) qui gère la puissance et la formation de faisceaux pour les utilisateurs actifs au sein de chaque tranche sur une courte période. Cette conception prend en compte les interruptions matérielles, les fluctuations de trafic et les variations de canal. De plus, un schéma prometteur basé sur l'accès multiple à répartition de débit (RSMA) est étudié afin d'améliorer encore les performances au sein de chaque service hétérogène. Outre l'amélioration des performances, le cadre proposé optimise également l'efficacité de la coordination en minimisant les surcharges. En particulier, le schéma inter-tranches est déclenché uniquement lorsque des variations importantes surviennent dans la charge de trafic des tranches hébergées. Cette conception réduit la surcharge globale du système en termes de consommation de mémoire, de temps d'apprentissage et de coûts opérationnels associés.

Les résultats de simulation, basés sur des hypothèses de modélisation système réalistes, démontrent que le cadre d'approches coopératives multiples proposé, reposant sur l'apprentissage par renforcement profond (DRL), surpasse les méthodes de référence et répond avec succès à diverses exigences de niveau de service (SLA) dans des environnements de déploiement O-RAN dynamiques. Les résultats d'évaluations approfondies montrent également que nos schémas proposés ouvrent la voie à une gestion des ressources plus complète et prédictive, garantissant des performances robustes même dans des conditions de réseau incertaines et des environnements opérationnels imparfaits. De plus, les résultats mettent en évidence la flexibilité, la robustesse et l'évolutivité potentielle de la conception proposée, tout en réduisant la charge de calcul par rapport aux méthodes de référence. Globalement, ces résultats valident le potentiel de l'utilisation de plusieurs agents DRL coopératifs pour permettre un découpage RAN automatisé, évolutif et intelligent dans les réseaux de communication sans fil de nouvelle génération.

Mots-clés: Apprentissage coopératif, apprentissage par renforcement profond, services hétérogènes, découpage inter-RAN, découpage intra-RAN, MISO, découpage de réseau, allocation de ressources, réseaux sans intervention manuelle réseaux

Towards Zero-Touch Management in RAN Slicing for Next-Generation Networks

Ohood SABR

ABSTRACT

Zero-Touch Networks (ZTNs) represent a cutting-edge paradigm shift towards fully automated and intelligent network management, providing the automation and intelligence needed to handle the complexity, scale, and dynamic behavior of next-generation mobile systems, including the sixth generation (6G). Artificial intelligence (AI) functions as the brain and backbone of ZTNs. In particular, deep reinforcement learning (DRL) algorithms have demonstrated strong potential for improving operational efficiency and supporting intelligent decision-making. Zero-touch management is particularly critical in transformative technologies such as network slicing (NS), which enables the creation of isolated logical networks on a shared physical infrastructure. Each logical network is designed to meet distinct quality of service (QoS) requirements, which makes NS indispensable for both current and future mobile generations. However, realizing ZTNs within a NS framework is associated with several challenges, especially in the radio access network (RAN) domain. Among the many difficulties encountered in the RAN domain, the following three challenges stand out as particularly critical: (i) managing inter- and intra-slice resource allocation; (ii) ensuring the security of inter-slice resource sharing; and (iii) achieving coordinated and concurrent management of radio resources at both inter- and intra-slice levels. These challenges are further exacerbated by the dynamic nature of wireless channels, scarcity of radio resources, diverse service-level agreements (SLAs), massive device densification, random traffic arrivals, and various network imperfections such as imperfect channel state information (CSI), imperfect orthogonal frequency division multiple access (OFDMA), and hardware impairments (HWIs).

In this context, the present thesis makes several key contributions to address these challenges. First, we investigate intra-slice resource allocation for RAN slicing by introducing an efficient self-optimizing (SO) scheme for a multi-user multiple-input single-output (MU-MISO) system, named PABSO-DRL. The proposed PABSO-DRL scheme dynamically and jointly manages power allocation and beamforming to ensure high data rates for enhanced mobile broadband (eMBB) while concurrently ensuring high reliability required by ultra-reliable low-latency communications (uRLLC). The scheme is designed to handle heterogeneous QoS requirements using a multi-agent deep Q-network (DQN) approach, while accounting for imperfect CSI, incomplete OFDMA isolation, and time-varying dynamics of the RAN environment.

Next, we focus on investigating inter-slice resource allocation. To this end, we propose a secure self-optimizing scheme based on a cooperative multi-actor–critic (CoMA2C) framework, referred to as SO-CoMA2C, to manage multiple radio resources (power and bandwidth) across a set of heterogeneous slices based on the fluctuating traffic load and in the presence of HWIs in open RAN (O-RAN). The main goal of the proposed scheme is maximizing the spectral efficiency while ensuring the SLA of each slice. Furthermore, we ensure the security of the

allocation by integrating the Advanced Encryption Standard (AES) algorithm into the proposed scheme.

Finally, we develop a hierarchical self-optimizing framework aimed at maximizing the long-term QoS and spectral efficiency of heterogeneous services. The proposed framework adopts a two-layer strategy implemented through the following two complementary slicing management schemes: (i) a cooperative multi-actor–critic (CoMA2C) scheme that allocates power and bandwidth across heterogeneous slices at a large timescale and (ii) a multi-agent DQN (MADQN) scheme that manages power and beamforming for active users within each slice at a small timescale. This design accounts for HWIs, traffic fluctuations and channel variations. Furthermore, a promising rate-splitting multiple-access (RSMA)-based scheme is investigated to further enhance the performance within each heterogeneous service. Beyond performance enhancement, the proposed framework also addresses coordination efficiency by minimizing overheads. In particular, the inter-slice scheme is triggered only when substantial changes occur in the traffic loads of the hosted slices. This design reduces the overall system overhead in terms of memory consumption, training time, and related operational costs.

Simulation results based on realistic system model assumptions demonstrate that the proposed cooperative multiple DRL-based approaches framework outperform baseline methods and successfully meet diverse SLA requirements in dynamic O-RAN deployment environments. The results of extensive evaluations further show that our proposed schemes provide a pathway towards more comprehensive and predictive resource management, ensuring robust performance under uncertain network conditions and imperfect operational environments. Moreover, the results highlight flexibility, robustness, and possible scalability of the proposed design while achieving a lower computational overhead as compared to benchmark baselines. Overall, the findings validate the potential of employing multiple cooperative DRL agents to enable automated, scalable, and intelligent RAN slicing in next-generation wireless communication networks.

Keywords: Cooperative learning, deep reinforcement learning, heterogeneous services, inter-RAN slicing, intra-RAN slicing, MISO, network slicing, resource allocation, zero-touch networks

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

1G	First generation
2G	Second generation
3G	Third generation
3GPP	Third Generation Partnership Project
4G	Fourth generation
5G	Fifth generation
6G	Sixth generation
7G	Seventh generation
A2C	Actor–Critic
AC	Admission control
AI	Artificial intelligence
AP	Application plane
AWGN	Additive white Gaussian noise
B5G	Beyond fifth generation
BS	Base station
BW	Bandwidth
CAPEX	Capital expenditures
CL	Central learning
CN	Core network

CoMA2C	Cooperative multi-actor–critic
CP	Control plane
COTS	Commercial off-the-shelf
C-RAN	Centralized radio access network
CSI	Channel state information
CU	Centralized unit
DL	Deep learning
DNN	Deep neural network
DQN	Deep Q-network
DP	Data plane
DRL	Deep reinforcement learning
DU	Distributed unit
D-RAN	Decentralized radio access network
DoS	Denial-of-Service
E2E	End-to-end
eMBB	Enhanced mobile broadband
ETSI	European Telecommunications Standards Institute
HWIs	Hardware impairments
ICSI	Imperfect channel state information
IL	Independent learning

IoT	Internet of Things
KPI	Key performance indicator
M2M	Machine-to-machine
MADQN	Multi-agent deep Q-network
MADRL	Multi-agent deep reinforcement learning
MANO	Management and orchestration
MARL	Multi-agent reinforcement learning
MDP	Markov decision process
MEC	Multi-access edge computing
MIMO	Multiple-input multiple-output
MISO	Multiple-input single-output
ML	Machine learning
mMTC	Massive machine-type communications
MNO	Mobile network operator
MU-MISO	Multi-user multiple-input single-output
NAT	Network Address Translation
Near-RT	Near-real-time
NGWNs	Next-generation wireless networks
Non-RT	Non-real-time
NFV	Network function virtualization

NS	Network slicing
OFDMA	Orthogonal frequency-division multiple access
OPEX	Operational expenditures
O-RAN	Open radio access network
POMDP	Partially observable Markov decision process
PRB	Physical resource block
PRBs	Physical resource blocks
QoS	Quality of service
RAN	Radio access network
RA	Resource allocation
RRA	Radio resource allocation
RRAM	Radio resource allocation and management
RRM	Radio resource management
RIC	RAN intelligent controller
RL	Reinforcement learning
RSMA	Rate-splitting multiple access
RU	Radio unit
SDN	Software-defined networking
SDNR	Signal-to-distortion-and-noise ratio
SINR	Signal-to-interference-noise ratio

SLA	Service-level agreement
SL	Supervised learning
SNR	Signal-to-noise ratio
TN	Transport network
UE	User equipment
UL	Unsupervised learning
UP	User plane
uRLLC	Ultra-reliable low-latency communications
VoNR	Voice over New Radio
VR	Virtual reality
vRAN	Virtualized radio access network
ZT	Zero touch
ZTNs	Zero-touch networks
ZSM	Zero-touch and service management

INTRODUCTION

0.1 The Motivation Toward Zero-Touch in Network Slicing

In recent decades, the rapid growth of communication applications/ together services has led to revolutionary developments across successive generations of mobile communication, yielding the five generation (5G) of cellular systems (Wang *et al.*, 2023), as illustrated in Fig. 0.1. The first generation (1G), which was launched in the early 1980s, was based on analog communication systems. Following the success of 1G, a new generation, referred to as second generation (2G), was developed, offering new features such as digital calls and texts. The next development was the third generation (3G), which supports a wide range of services, including video calling, roaming, Internet, phone calls, and multimedia services. In addition, 3G technology offered high-speed data transfer at rates of up to 2 Megabits per second (Mbps) and the ability to download data from the internet (Bugade *et al.*, 2021). However, the age of 3G was short due to various technical challenges and several drawbacks with 3G mobile devices, such as their higher power consumption as compared to most 2G versions. Furthermore, 3G network plans cost more than the previous generation (Gupta & Jha, 2015). These issues inspired the creation of a more advanced version of the 3G standards–namely, the fourth generation (4G) (Bugade *et al.*, 2021). 4G constitutes a major improvement over 3G in terms of both the speed of data transfer and the quality of streaming; therefore, 4G is considered superior to all previous generations (Salameh & El Tarhuni, 2022; Bugade *et al.*, 2021). However, 4G could not cope with the exponential growth in the number of connected devices. In particular, 4G failed to adequately address some issues such as the need for higher data rate, lower end-to-end (E2E) latency, higher capacity, massive device connectivity, and cost (Gupta & Jha, 2015).

At present, 5G is considered a promising technology for future networks. 5G is expected to provide enhanced capabilities, such as high reliability, ultra-low latency, mobility support, and seamless connections (Liyanage *et al.*, 2022). 5G provides significant improvements over

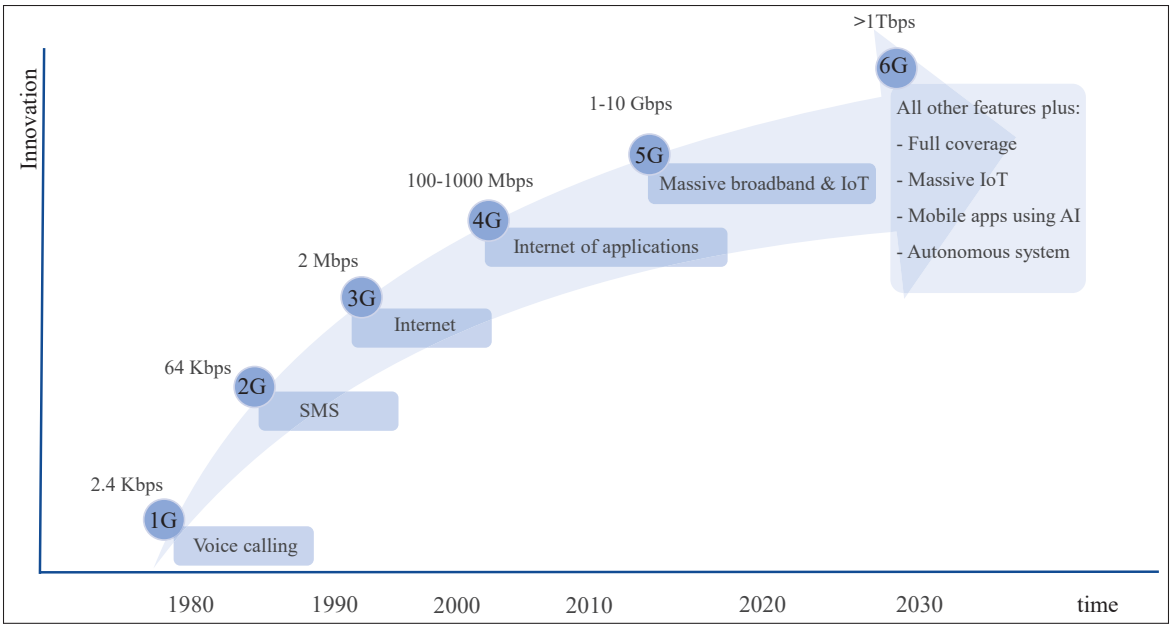

Figure 0.1 The evolution of wireless communications toward future 6G networks
Adapted from Giordani *et al.* (2020)

4G systems. Particularly noteworthy is its architecture (Wang *et al.*, 2023) that incorporates network slicing (NS) technology. NS emerged as a transformative paradigm to support the creation of logical networks with diverse quality of service (QoS) requirements on a common physical infrastructure (Qiao *et al.*, 2025). The architecture of 5G supports the following three core services or slices: (i) enhanced mobile broadband (eMBB); (ii) ultra-reliable low-latency communication (uRLLC); and (iii) massive machine-type communication (mMTC) (Indoonundon & Pawan Fowdur, 2021). eMBB refers to traffic with high data rate demands, with up to 20 Gb/s peak data rate and 100 Mb/s everywhere (e.g., automated driving). Furthermore, uRLLC services were introduced to serve use cases with strict criteria for extremely low latency around 5 ms and extremely high dependability of 99.9999 % (e.g., automated driving). Finally, mMTC enables connection densities of up to 1 million devices per square kilometer of affordable, low-energy devices (10 years of battery life), while satisfying relaxed data rate and latency requirements, such as those in smart city applications (Ojaghi, Adelantado, Antonopoulos & Verikoukis, 2022).

Despite the advantages of 5G, it will not be capable of meeting future needs because 5G capabilities are expected to reach their limits by 2030 (Ojaghi *et al.*, 2022). With the rapid growth of machine-to-machine (M2M) communications, the number of internet-connected devices is expected to reach hundreds of billions, while 5G networks can support only up to about a billion devices (Salameh & El Tarhuni, 2022). Moreover, some future applications and services referred to as fully dependable machine operations (e.g., remote surgery and remote management of a smart factory) require extremely high availability, ultra-high data rate transmission, and ultra-low latency as compared to what 5G offers (Gangakhedkar *et al.*, 2018; Wang *et al.*, 2023).

Therefore, extensive effort is currently invested into achieving these stringent requirements in order to satisfy the anticipated demand for future services by shifting towards the sixth generation (6G) system (Jiang, Han, Habibi & Schotten, 2021). By 2030, the number of connected devices supported by 6G wireless networks is expected to exceed 125 billion (Setayesh, 2024). Contemplating the future of 6G networks and considering their varied components, high density, complexity, expected performance and intelligence levels, many questions arise about the suitable tools for managing such networks. Specifically, 6G systems are anticipated to support a massive number of extremely diverse network slices spanning multiple technological domains (*i.e.*, radio access network (RAN), edge, cloud, and core), which will pose significant challenges to traditional centralized management and orchestration strategies in terms of scalability and sustainability (Chergui, Ksentini, Blanco & Verikoukis, 2022). Moreover, as argued by Pennanen, Hänninen, Tervo, Tölli & Latva-Aho (2025) 6G will also approach its limits, after which the next generation, seventh-generation (7G) will begin to emerge. This transition is expected to follow the same ten-year cycle observed in the past. While 6G is anticipated to be the dominant technology throughout the 2030s, the 2040s are anticipated to bring the emergence of 7G, which will be much more complicated than that of 6G.

What the above discussion suggests is that, with the advances of wireless technologies, the range of capabilities and the number of connected devices will increase substantially. From an architectural perspective, mobile communication systems have evolved towards incorporating more antennas, more advanced multiple access techniques, and a broader set of services (Wang *et al.*, 2023). Consequently, it becomes essential to ask how network management and orchestration (MANO) frameworks can be designed to handle the scalability, automation, and complexity of such networks.

This has motivated scholars to explore different approaches to address the shortcomings of current network MANO systems (Liyanagea *et al.*, 2022). The most recent approach is zero-touch networks (ZTNs), which largely rely on artificial intelligence (AI) and machine learning (ML) techniques which serve as those networks' backbone algorithms (Yang *et al.*, 2025). In particular, deep reinforcement learning (DRL) algorithms have recently attracted attention for their ability to perform complex decision-making and offer automated optimal control in dynamic telecommunications environments (Rezazadeh, Chergui, Christofi & Verikoukis, 2021a; Iacoboaiea, Krolikowski, Houidi & Rossi, 2023). The ZTNs paradigm aims at full automation of network operations and services while minimizing the need for human intervention, which is essential for managing complexity and dynamic requirements of next-generation networks such as 6G (Yang *et al.*, 2025).

There are several key enabling technologies for realizing ZTNs functionality, such as network function virtualization (NFV), software-defined networking (SDN), multi-access edge computing (MEC), and NS, whose integration with AI is essential (Sabr, Kaddoum & Kaur, 2025a). While wireless communications witnessed the first birth of NS in 5G, it is expected to be inherited and further evolved as a key enabler of intelligent, automated, and flexible network management (Mei *et al.*, 2021). NS is also expected to become a foundational element in future systems (Alam *et al.*, 2025). To achieve an E2E NS system, slicing should span across the following three

domains: the core network (CN), transport network (TN), and RAN domains (Li *et al.*, 2020b; Alam *et al.*, 2025; Awada, Berri & Chorti, 2024). In contrast to the CN and TN domains, RAN slicing faces several open issues, including potential radio resource sharing, efficient utilization of radio resources, dynamic variations in service traffic flows, and maintaining performance isolation among slices. (Mei *et al.*, 2021; Li *et al.*, 2020b). The main challenges include the need to balance resource efficiency with isolation, harmonize inter- and intra-RAN allocation schemes, and manage slice prioritization across multiple RAN layers. Moreover, the scarcity of radio resources necessitates adopting highly efficient management strategies to sustain optimal network performance (Alam *et al.*, 2025). Another critical challenge is the challenge of concurrently satisfying heterogeneous service requirements in time-varying wireless channel environments (Zhou *et al.*, 2023).

Considering the essential role of NS as a cornerstone of 6G and beyond mobile network deployment, also as a key enabler for realizing zero-touch (ZT) management, in the present thesis, we investigate the implementation of ZT capabilities within RAN slicing for next-generation wireless systems, considering the remaining open challenges in the RAN domain.

0.2 Objectives

The major aim of this thesis is to design self-optimizing (SO) radio resource management (RRM) schemes that will support the coexistence of heterogeneous services with stringent and diverse QoS requirements in a multi-user multiple-input single-output (MU-MISO) system. The design of the proposed schemes uses a multi-agent DRL (MADRL) approach, paving the way toward full ZT management in RAN slicing for future networks.

Accordingly, the first objective of the thesis is to design a self-optimizing scheme for the intra-RAN slicing level based on MADRL, thus ensuring effective resource management for eMBB and uRLLC services with different QoS requirements, while accounting for practical constraints

such as imperfect channel state information (ICSI) and imperfect orthogonal frequency division multiple access (OFDMA).

The second objective is to investigate inter-slice resource management by designing a secure and scalable RRM scheme to manage resources among a set of heterogeneous services with diverse QoS requirements based on the cooperative MADRL framework, adapting to fluctuating traffic loads and considering hardware impairments (HWIs).

Finally, the third objective is to design a hierarchical radio resource allocation (RRA) framework that will jointly manage radio resources on both the inter-slice and intra-slice levels, while evaluating the performance of rate-splitting multiple access (RSMA) in scenarios where heterogeneous services coexist. The proposed framework is developed using a cooperative DRL approach and accounts for the effects of fluctuating traffic loads and HWIs.

The overview of the objectives, challenges, and solutions covered in the present thesis is provided in Fig. 0.2.

0.3 Thesis Contributions and Related Publications

This section provides an overview of the core contributions of this thesis and related publications derived from the conducted research.

0.3.1 Thesis Contributions

Chapter 2 addresses the coexistence of eMBB and uRLLC services at the intra-slice of the RAN domain. The simultaneous support of these services with heterogeneous QoS requirements in a MU-MISO system, particularly in the presence of ICSI and imperfect orthogonality in OFDMA, introduces complex resource management challenges. More specifically, we focus on the joint power allocation and beamforming problem, aiming to maximize the data rate of

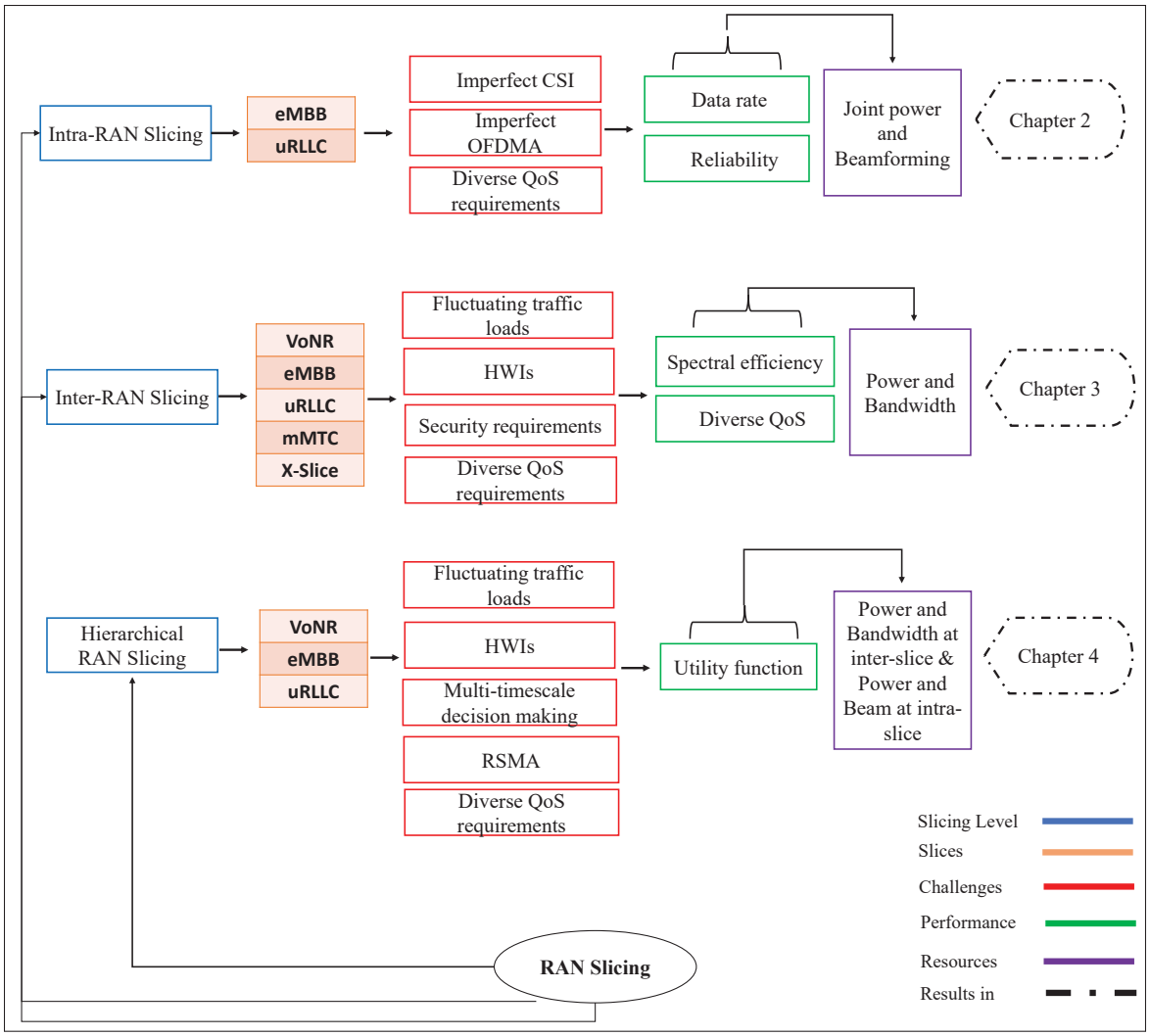

Figure 0.2 Overview of the present thesis' objectives, challenges, and solutions

eMBB users while minimizing the outage probability of uRLLC users. To this end, we develop a self-optimizing scheme based on multiple deep Q-network (DQN) agents to optimally allocate and manage resources for both eMBB and uRLLC users. Simulation results demonstrate that the proposed scheme outperforms baseline methods in satisfying the QoS requirements. The contributions of Chapter 2 were published in the *IEEE Transactions on Consumer Electronics*.

Chapter 3 shifts the focus to inter-slice resource management within the open RAN (O-RAN) domain. In this chapter, we examine the challenges of allocating multiple radio resources

to a set of heterogeneous services, each with its distinct traffic characteristics and diverse service level agreement (SLA), all under fluctuating traffic loads and in the presence of HWIs. Building on this evidence, we propose a secure, self-optimizing scheme to manage power and bandwidth allocation among voice over new radio (VoNR), eMBB, uRLLC, mMTC, and X-slice. The problem is formulated to maximize the system spectral efficiency while satisfying heterogeneous QoS constraints under HWIs. To address this NP-hard problem, it is decomposed into subproblems that are then solved using cooperative multiple actor-critic (CoMA2C) agents. To ensure secure allocation, Advanced Encryption Standard (AES) algorithm is integrated into the Markov decision process (MDP) framework. The proposed SO-CoMA2C scheme is then evaluated against the state-of-the-art schemes. The results reveal that the proposed scheme outperforms the baseline methods, and the complexity analysis demonstrates that the scheme achieves a high level of security while significantly reducing computational overhead. The results reported in Chapter 3 were published in the *IEEE Internet of Things Journal*.

Building upon the work presented in Chapter 3, we then extend our analysis to multiple slicing levels in Chapter 4, where the coexistence of VoNR, eMBB, and uRLLC services introduces a more complex resource management challenge across both inter- and intra-slice levels simultaneously. To address this challenge, we propose a hierarchical self-optimizing framework designed to jointly manage power and bandwidth among heterogeneous services on the inter-slice level, and jointly optimize power and beamforming for the active users within each slice on the intra-slice level. The problem is formulated to maximize the utility function while satisfying the QoS requirements of each slice. We then decompose it into two subproblems: the inter-slice level is reformulated as a partially observable Markov decision process (POMDP), while the intra-slice level is reformulated as a MDP. To solve the POMDP, we propose a cooperative multi-agent actor-critic (A2C) based scheme; to address the intra-slice level, we design a cooperative multi-agent DQN-based solution. Simulation results demonstrate superiority of the proposed framework over several baseline frameworks, while also maintaining

lower overhead, making the proposed framework well-suited for joint inter- and intra-RAN slicing deployments. The contributions of Chapter 4 were published in *IEEE Open Journal of the Communications Society*.

0.3.2 Related Publications

The research outcomes of the present thesis resulted in several published research articles listed below. In what follows, journal publications are labeled as J, whereas conference contributions are labeled as C.

J1: **O. Sabr**, G. Kaddoum and K. Kaur, "PABSO-DRL: Power and Beam Self-Optimization Scheme for Multiple Slices in MU-MISO Systems," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 2, pp. 4343-4358, May 2025, doi: 10.1109/TCE.2024.3504958.

J2: **O. Sabr**, K. Kaur and G. Kaddoum, "A Secure Multi-Radio Resource Scheme Using Cooperative DRL Agents for Heterogeneous Inter-RAN Slicing Under Hardware Impairments," *IEEE Internet of Things Journal*, July 2025, doi: 10.1109/JIOT.2025.3593409.

J3: **O. Sabr**, G. Kaddoum and K. Kaur, "HiSO-CoMA: Hierarchical Self-Optimising Framework for O-RAN Slicing Using Cooperative Multiple Agent Deep Reinforcement Learning," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 9632-9653, 2025, doi: 10.1109/OJCOMS.2025.3631799.

C1: **O. Sabr**, K. Kaur and G. Kaddoum, "SOIS-A2C Scheme: Facilitating Management of Multi-Radio Resources in Heterogeneous Inter-RAN Slicing in the Presence of Hardware Impairments,"*ICC 2025 - IEEE International Conference on Communications, Montreal, QC, Canada, 2025, pp. 1414-1419*, doi: 10.1109/ICC52391.2025.11161109.

C2: **O. Sabr**, G. Kaddoum and K. Kaur, "PoB-MDRL: Multi-Agent Deep Reinforcement Learning-Based Joint Power Allocation and Beamforming for Heterogeneous Services under

Non-Ideal Network Conditions,"*The 39th Annual Canadian Conference on Electrical and Computer Engineering (CCECE 2026), Montreal, QC, Canada* (Accepted).

0.4 Thesis Organization

In Chapter 1, we provide a comprehensive overview of ZTNs, enabling technologies, overview of O-RAN slicing, and RRM in RAN slicing. The chapter concludes with a discussion of resource allocation optimization approaches. Chapter 2 introduces our proposed PABSO-DRL scheme, a multi-agent DRL–based scheme designed to provide self-optimizing management for intra-RAN slicing. In Chapter 3, we present the secure resource allocation scheme based on cooperative learning approach, which serves as a self-optimizing scheme for managing heterogeneous inter-RAN slicing. Chapter 4 presents and evaluates a hierarchical HiSO-CoMA framework, which coordinates both inter- and intra RAN slicing management based on cooperative MADRL. Finally, the thesis concludes with a summary of the key contributions and an outline of potential future research.

CHAPTER 1

BACKGROUND

In this chapter, we first introduce the concept of ZTNs in Section 1.1. Section 1.2 then provides an overview of key enabling technologies supporting ZTNs. Next, Section 1.3 outlines the evolution of the RAN domain, followed by a discussion of the O-RAN architecture in Section 1.3.1. Section 1.3.2 describes resource allocation in O-RAN slicing. Finally, Section 1.4 provides an overview of traditional and ML–based optimization techniques.

1.1 Overview of Zero Touch Networks

As discussed earlier in the Introduction, the shift towards network automation has been driven by the envisaged complexity of operating and managing next-generation networks. ZTNs have emerged as a ground-breaking network management paradigm for next-generation networks such as 6G (Yang *et al.*, 2025). The aim of ZTNs is to provide full automation, whereby all operational tasks and processes are completed automatically without human intervention, including configuration, optimization, healing, and protection (Yang *et al.*, 2025). AI is expected to be a key facilitator of self-managing capabilities, lowering operational costs, and eliminating the risk of human errors (Benzaid & Taleb, 2020; Yang *et al.*, 2025). ZTNs are integral to the zero-touch and service management (ZSM) framework (Yang *et al.*, 2025), developed by the European Telecommunications Standards Institute (ETSI) group, which was set up in 2017 with the goal of automating network management and E2E provision of services in multi-domain contexts. The ETSI proposed a reference architecture for delivering ZSM in B5G networks (Gallego-Madrid, Sanchez-Iborra, Ruiz & Skarmeta, 2022).

Zero-touch capabilities in ZTNs build upon the following four core autonomous operational categories, as illustrated in Fig. 1.1 (Arzo *et al.*, 2021; Yang, El Rajab, Shami & Muhaidat, 2024a):

1. **Self-configuration**, refers to a network's ability to set up and change its configuration according to specified policies in order to accomplish a specific performance. This ought to take place automatically, without any human interference.
2. **Self-optimization**, which refers to a network's ability to make the optimum use of its resources at all times, even in a very diverse environment. The network should continuously and automatically evaluate its current performance and develop plans to efficiently work in the event where it deviates from expectations or goals.
3. **Self-healing**, which is required in the network when one of the elements fail. This feature enables networks to achieve the fastest and the most efficient possible recovery from such breakdowns. Also, self-healing enables the network to identify failed components and automatically repair them to maintain service.
4. **Self-protection**, which ensures the network can protect itself from any foreseeable threats like Denial-of-Service (DoS) and Man-in-the-middle attacks.

According to Aleyadeh (2023), ZT management is becoming increasingly important and promising for next-generation networks due to several key factors. First, it offers significant gains in efficiency and speed, which are essential as networks grow in complexity and scale, particularly with the adoption of NS technology. ZT automation also reduces operational expenses (OPEX), as automated systems require minimal human intervention. This substantially decreases the costs associated with network management and increases the overall cost-effectiveness of the services. Moreover, ZT improves reliability and consistency by reducing the risk of human errors, which remain a major cause of network outages, as well as enables real-time network management and optimization, which is a critical requirement for future networks, where conditions can change rapidly owing to user mobility or sudden fluctuations in traffic demands.

1.2 Enabling Technologies for ZTNs

This section outlines the fundamental technologies enabling ZTNs, including SDN, NFV, and NS, which provide programmability, flexibility, and automation for intelligent network management.

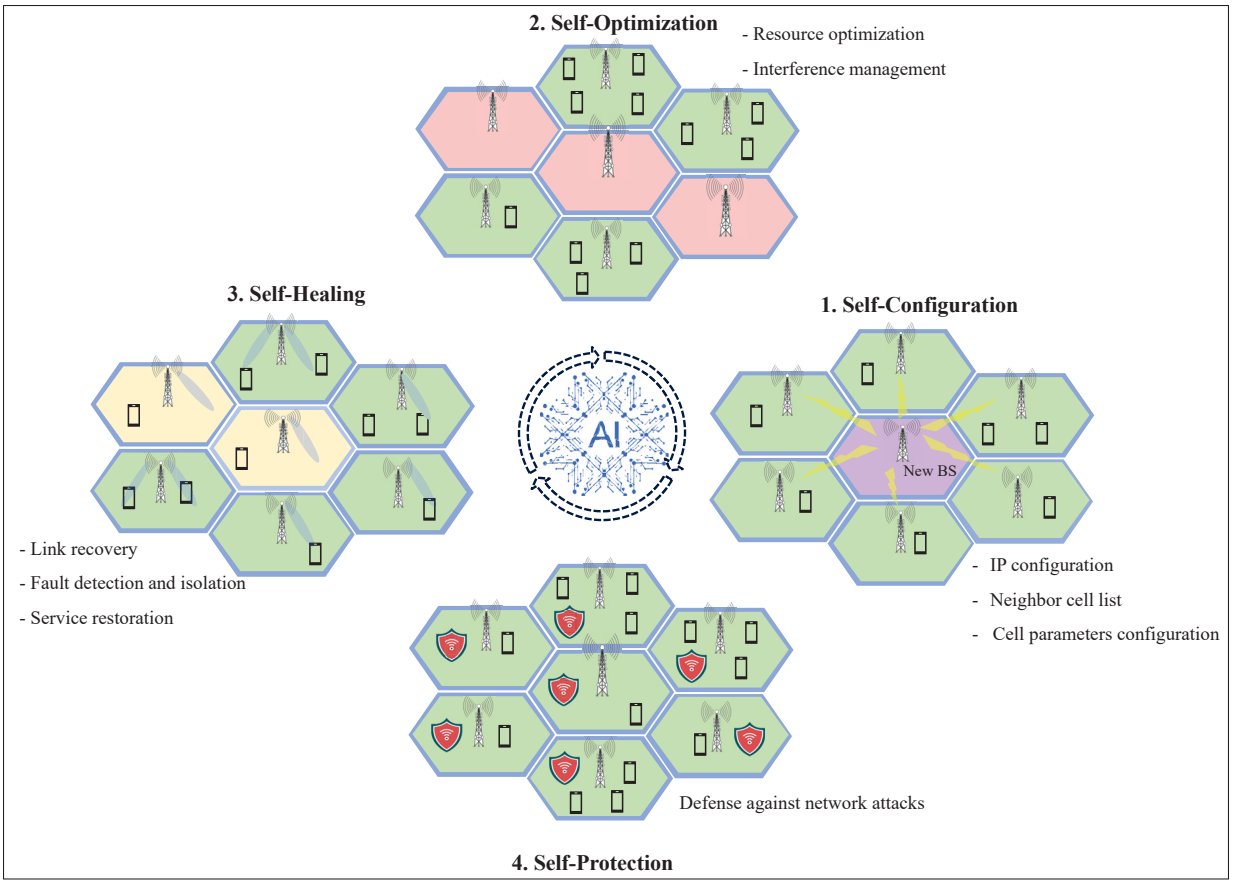

Figure 1.1 ZTN Classes

1.2.1 Software-Defined Networking

SDN enables network programmability by decoupling the control or management plane from the data plane (Zhang, 2019). This architectural separation offers multiple benefits, such as simplifying network control and management (Amin, Reisslein & Shah, 2018). SDN represents a promising approach to enable network automation (Arzo *et al.*, 2021). Figure 1.2 illustrates the general architecture of SDN, which includes the following three planes: (i) the data plane (DP), which includes all network hardware, such as switches, routers, and firewalls; (ii) control plane (CP); and (iii) the application plane (AP). It also includes two interfaces: a southbound interface responsible for controlling resources between the DP and the CP, as well as a northbound interface to connect the CP to the AP (Rahmanian, Shahhoseini & Pozveh, 2021). The CP is executed in a centralized controller. The controller is a single, sophisticated software managing

all aspects of the network. The controller uses OpenFlow, a standard protocol that allows it to forward rules for all network interfaces (Arzo *et al.*, 2021).

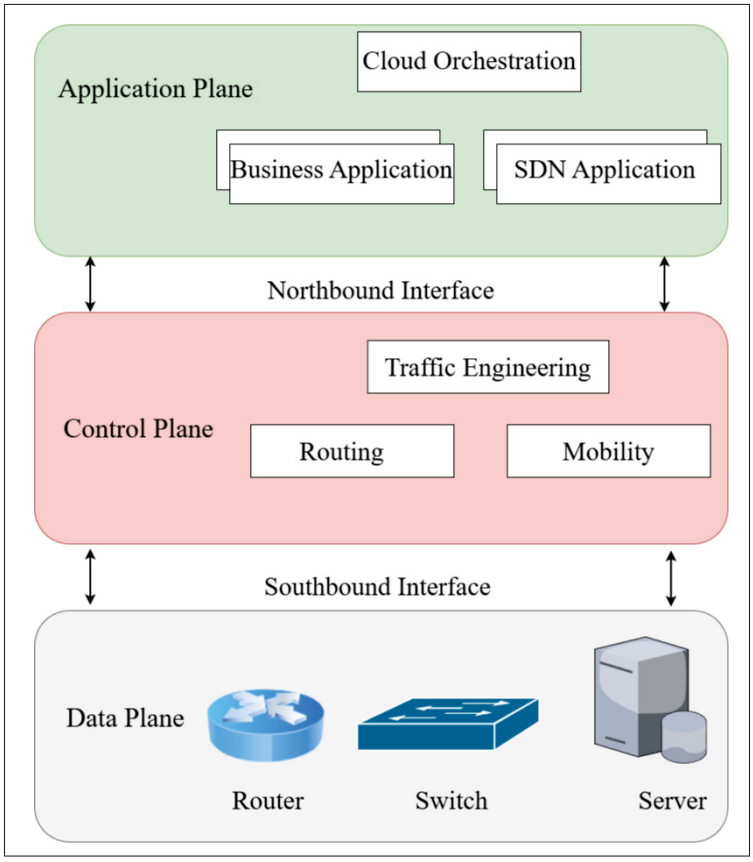

Figure 1.2 Overview of SDN architecture
Adapted from Arzo *et al.* (2021)

1.2.2 Network Functions Virtualization

To deliver meaningful services to end users, service providers require a wide range of hardware components. In response to users' growing demand for new network services, service providers must invest time and effort in deploying physical hardware and equipment for each network function. In addition, there is a need for highly qualified network designers and operators to handle the difficulty of establishing and maintaining huge networks. Therefore, ETSI was the first group studying the use of the NFV idea in operator infrastructures, primarily to address issues with adaptive and flexible services and to establish a foundation for next-generation

networks (Barakabitze, Ahmad, Mijumbi & Hines, 2020; Alwakeel, Alnaim & Fernandez, 2019). NFV is a technology that aims to separate network functions like firewalls, gateways, Network Address Translation (NAT), and routers from specialised hardware and runs them as software on a cloud server. Figure 1.3 shows the differences between the traditional network strategy and the NFV strategy (Alwakeel *et al.*, 2019).

Consequently, operators can offer virtualized services like virtual firewalls, virtual gateways, and virtualized network components. If all functions are given by software rather than hardware, providing virtual services to the end users results in enabling dynamic and fast deployment of new services with more agility, eliminating the time and effort required to expand the network, and significant savings in both capital expenditures (CAPEX) and OPEX (Alwakeel *et al.*, 2019). NFV promises significant benefits, including flexibility, scalability, and greater independence (Alwakeel *et al.*, 2019).

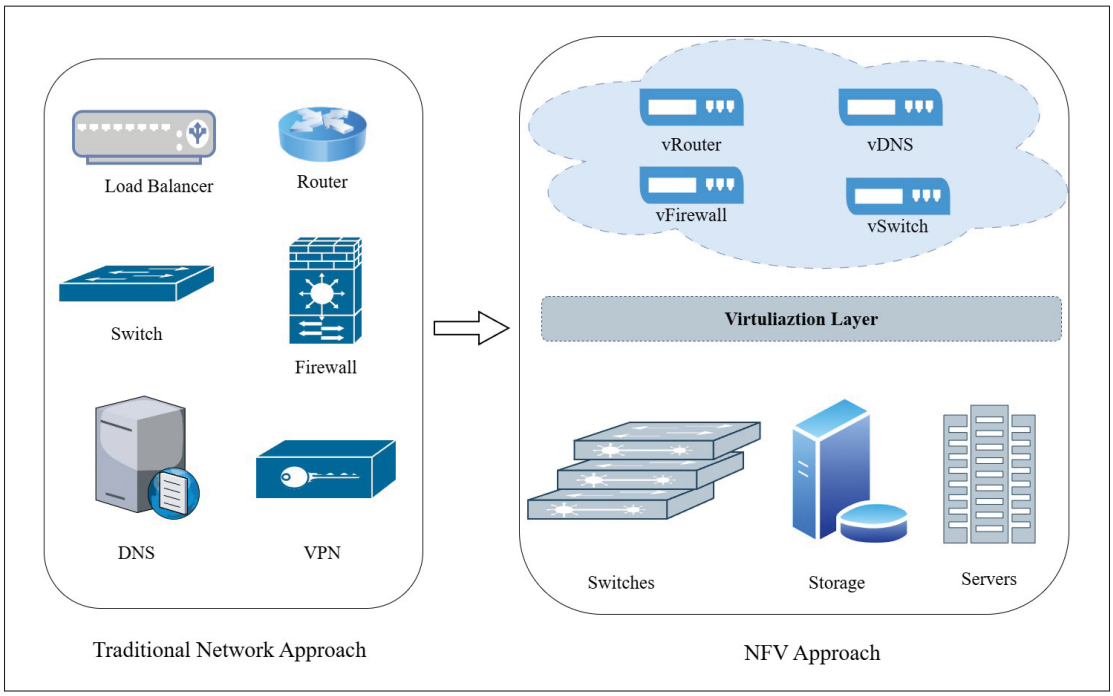

Figure 1.3 Traditional network approach vs. NFV
Adapted from Alwakeel *et al.* (2019)

1.2.3 Network Slicing

As mentioned earlier, 5G systems are designed to support a wide range of services and emerging applications, each characterized by distinct QoS requirements. By contrast, traditional mobile communication networks rely on the "one-size-fits-all" architecture, lack the adaptability, reconfigurability, and scalability needed to simultaneously accommodate the extremely diverse requirements of the different services in terms of performance, availability, cost, and security (Rahmanian *et al.*, 2021; Zhang, 2019; Ojaghi *et al.*, 2022). To efficiently manage numerous services and deliver customized network capabilities under limited resource availability while concurrently reducing capital and operational expenditures (Zhang, 2019), NS was proposed by academia and industry as a key enabler for providing on-demand, tailored 5G services (Zhang, 2019). The main idea behind NS is dividing a physical network into a set of virtual networks, each of which can be designed and optimized for a certain kind of application (Qiao *et al.*, 2025); (see Fig. 1.4).

The use of SDN and NFV enables slice creation, allowing mobile networks to gradually transition to a flexible and programmable network architecture (Yan, Feng, Zhou, Sun & Liang, 2019). In order to meet the constantly changing requirements of slices, NS needs to be dynamic and adaptable (Rahmanian *et al.*, 2021). Each slice supports a specific type of service, each with distinct performance requirements. For instance, eMBB supports applications demanding ultra-high bandwidth, such as virtual reality (VR), video streaming; uRLLC provides services for applications requiring highly reliable and low-latency communication, such as the remote control of critical operations; and mMTC supports massive IoT connectivity, as seen in smart city deployments interconnecting numerous embedded sensors (Alcaraz, Losilla, Zanella & Zorzi, 2023; Rahmanian *et al.*, 2021).

The infrastructure provider is responsible for allocating the necessary radio and computing resources to each NS to satisfy the SLA made with its tenant (Qiao *et al.*, 2025). The SLA defines a set of criteria for performance indicators, such as throughput or latency, that depend on the tenant requirements and service type. In NS, the resource allocation must ensure

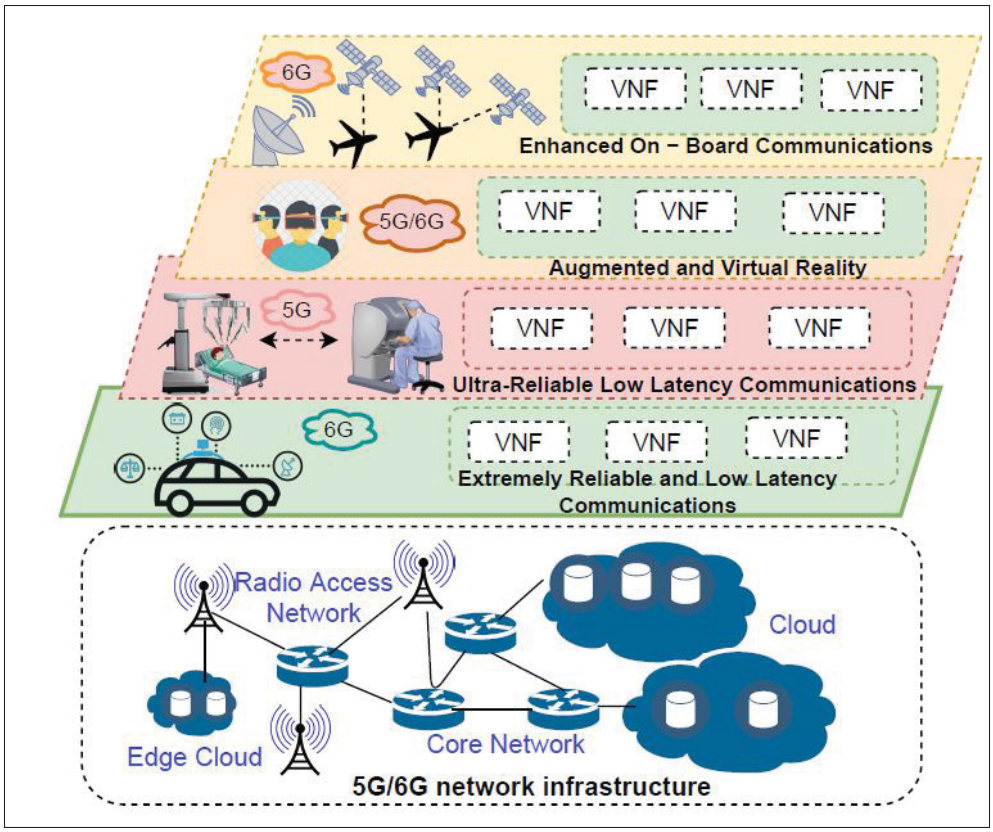

Figure 1.4 Network slicing architecture
Taken from Hurtado Sanchez *et al.* (2022)

isolation between network slices while concurrently remaining resource-efficient and adaptable to fluctuations in network traffic conditions (Alcaraz *et al.*, 2023).

Slice isolation maintains the integrity of individual network slices by prohibiting errors or malfunctions in one slice from impacting others, thus supporting independence and reliability across virtualized networks (Alam *et al.*, 2025). As stressed by Sallent, Perez-Romero, Ferrus & Agusti (2017) maintaining isolation between slices is crucial to prevent the traffic of one slice from adversely affecting others. Furthermore, such isolation must be designed to support effective resource efficiency. Therefore, robust isolation between slice instances is one of the most essential requirements for guaranteeing NS security (De Alwis, Porambage, Dev, Gadekallu & Liyanage, 2024). In a single-cell RAN slicing setup, isolation can be ensured by allocating an orthogonal or dedicated set of physical radio resources to each slice for a

specified duration based on its requirements. During this interval, the slice retains the flexibility to distribute these resources among its own users. However, extending slicing to a multi-cell RAN makes isolation significantly more complex, as it requires careful consideration of the interference that may arise between transmitters belonging to different tenants across multiple cells (Sallent *et al.*, 2017).

To achieve E2E NS and fully realize its benefits, the slice must span all network domains– CN, TN, and RAN–all of which need to be considered collectively within an integrated framework (Ebrahimi, Bouali & Haas, 2024). In what follows, we briefly define the function of each domain.

- **CN domain** supports end-user services by delivering essential functions such as mobility management, session management, and network slice selection (Ebrahimi *et al.*, 2024).
- **TN domain** includes the links between the RAN and the core network, as well as the internal nodes along these paths. When slicing the TN, the main factors to consider are link capacity, latency limits, and efficiency of routing traffic through SDN-based switches (Ebrahimi *et al.*, 2024).
- **RAN domain** is composed of base stations (BSs), where each BS generally includes a radio unit (RU) containing the RF circuitry for signal transmission and reception, and a baseband unit (BBU) that performs computational functions such as radio management and resource allocation (Wani, Kretschmer, Schröder, Grebe & Rademacher, 2025).

As argued by Azimi, Yousefi, Kalbkhani & Kunz (2022a) the RAN domain presents the most significant slicing challenges compared to the TN and CN domains. One of the most critical challenges in RAN slicing is resource scheduling, where limited resources must be allocated to heterogeneous users with diverse QoS requirements while accounting for variations in traffic and network-state dynamics.

1.2.3.1 Advantages of NS

Compared with traditional networks, NS offers several significant benefits (Li *et al.*, 2017) that are briefly outlined below.

- **Enhanced performance:** Instead of relying on a single uniform network, slicing allows the creation of dedicated logical networks that can deliver superior performance for specific services;
- **Adaptive capacity:** Each slice can automatically adjust its resources—either increasing or decreasing them—according to real-time service needs and user load;
- **Reliability and Security:** Network slices are separated from one another, resulting in that modifications or issues in one slice do not impact others. This separation improves both the reliability and security of each individual slice;
- **Efficient resource utilization:** Since slices are built to meet precise requirements of a service, the underlying physical resources can be more effectively allocated, leading to optimized overall network usage.

1.3 Evolution of RAN

Generally, a mobile network architecture consists of three main elements, shown in Fig. 1.5: user equipment (UE), RAN, and the core network. The UE refers to the device used by the end user to access the mobile network, such as a smartphone. The RAN enables radio communication between the UE and the core network (Wani *et al.*, 2025).

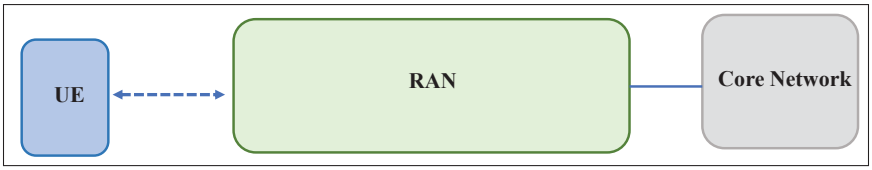

Figure 1.5 Overview of the basic components of a mobile network
Adapted from Wani *et al.* (2025)

Over the last three decades, RAN has undergone several evolutionary stages, transitioning from tightly integrated, hardware-dependent systems to more open and flexible architectures. This evolution began with the decentralized RAN (D-RAN), where both the RU and BBU were co-located at each cell site, leading to limited scalability and high operational costs. To overcome these limitations, a centralized RAN (C-RAN) architecture was introduced. In C-RAN, BBUs from multiple sites are centralized, whereas RUs remain distributed at their respective cell sites, thus reducing site expenditures and improving management efficiency (Wani *et al.*, 2025; Agarwal, Irmer, Lister & Muntean, 2025). However, C-RAN still relies on proprietary hardware and interfaces, which limit flexibility and vendor interoperability (Wani *et al.*, 2025). In addition, considering that all BBUs are hosted in a central cloud location, C-RAN is prone to single-point failures which may cause complete network outages (Gavrilovska, Rakovic & Denkovski, 2020).

Building upon C-RAN, virtualized RAN (vRAN) replaces traditional proprietary BBU hardware with commercial off-the-shelf (COTS) servers and separates the software from the hardware using NFV principles. This allows network functions to run in virtual machines or containers on COTS platforms. However, the interface between COTS-based BBUs and RRUs remains proprietary. Although vRAN improves resource utilization and scalability, it still inherits these proprietary interfaces (Agarwal *et al.*, 2025; Nagib, 2024; Wani *et al.*, 2025). vRAN also provides benefits such as increased flexibility, enhanced scalability, and reduced costs through virtualization and cloud-based design. However, despite these advantages, several challenges persist. Specifically, the virtualization layer may introduce additional latency, making it difficult to satisfy the stringent real-time performance requirements of 5G. Furthermore, coordination and management of virtualized resources across distributed infrastructures are also complex and can affect operational efficiency. In addition, current vRAN implementations may struggle to deliver the high processing power and reliable low-latency connectivity demanded by 5G services (Agarwal *et al.*, 2025).

Overall, conventional RANs still lack the adaptability and transparency needed to meet simultaneous and diverse service requirements (Nagib, 2024). These architectures frequently do not incorporate advanced data-driven optimization or closed-loop control mechanisms, which

are essential to efficiently manage the complexity and heterogeneity of modern networks (Wani *et al.*, 2025).

To overcome the aforementioned limitations, O-RAN, representing the latest evolution of the RAN architecture, was proposed as the future direction of RAN (Wani *et al.*, 2025). The O-RAN Alliance first introduced the term O-RAN as part of its efforts to standardize and advance the RAN domain. The Alliance aims to drive the evolution of mobile networks toward more intelligent, open, virtualized, and interoperable architectures (Agarwal *et al.*, 2025).

O-RAN systems are structured around virtualized, software-based components, and disaggregated components interconnecting through open, standardized interfaces and can be coordinated through advanced intelligent controllers. This disaggregation and virtualization enable network functions to be deployed in a more adaptable, cloud-native manner, improving the system's robustness and ability to reconfigure. The reliance on open interfaces enables interoperability across multiple vendors and, encouraging a more diverse RAN environment. In addition, intelligent controllers exploiting these open interfaces can elevate the degree of automation within the RAN, supporting more autonomous, cost-efficient, and streamlined optimization processes for network operators (Wani *et al.*, 2025).

1.3.1 Architecture of O-RAN

As RAN evolves towards O-RAN, BS functions are split into the following three components: (i) RU, (ii) the distributed unit (DU), and (iii) the centralized unit (CU), as illustrated in Fig. 1.6. The CU is further partitioned into two logical units: one responsible for the CP and the other for the user plane (UP) (Polese, Bonati, D'Oro, Basagni & Melodia, 2023; Agarwal *et al.*, 2025). These components are connected through open interfaces such as the Open Fronthaul. Using these standardized open interfaces, equipment from different vendors can work together, enabling a more flexible and interoperable RAN architecture (Agarwal *et al.*, 2025). Furthermore, a fundamental element is the RAN intelligent controller (RIC) that acts as the brain of the RAN (Ngo *et al.*, 2024b), providing a programmable environment that enables optimization functions

to monitor, analyze, and control RAN. The RIC hosts a range of control functionalities that were traditionally implemented on the BS level, such as mobility management, admission control (AC), and interference coordination. These functions are deployed as modular applications operating on the RIC platform, and their control decisions are executed through standardized open interfaces, such as E2 interface. The behavior of these applications is driven by network conditions, traffic dynamics, and the operational objectives of the mobile network operator (MNO). These applications can leverage AI and ML techniques to enable adaptive and intelligent network management (Wani *et al.*, 2025).

The O-RAN architecture consists of two types of RICs: non-real-time (non-RT) and near-real-time (near-RT), which perform the control and management of RAN nodes at different time scales (Almeida *et al.*, 2024).

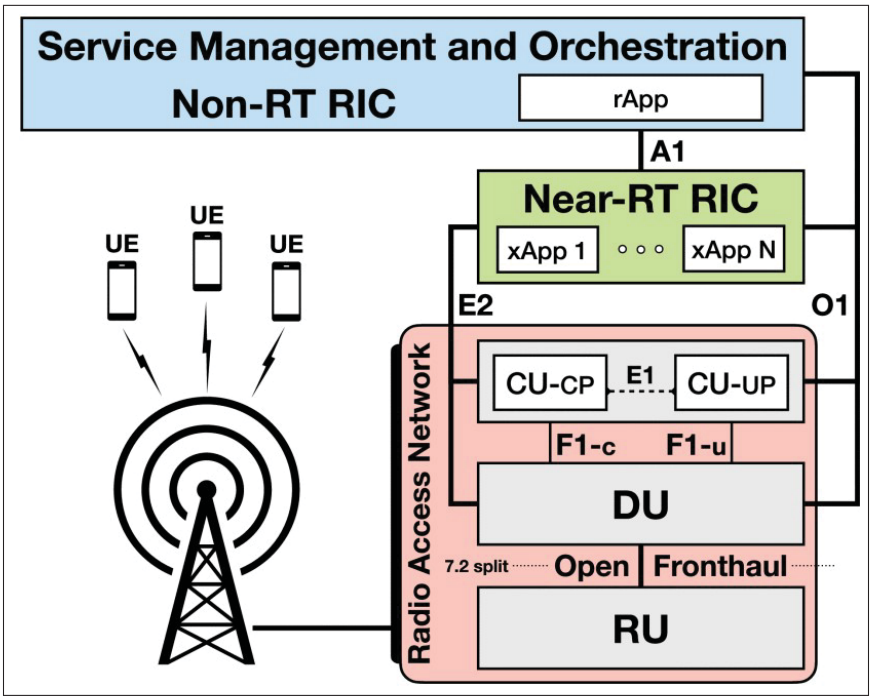

Figure 1.6 Overview of the O-RAN architectural components
Taken from Groen *et al.* (2024a)

- **Near-RT RIC** is a logical component whose main goal is hosting and running external applications, known as xApps. These xApps are microservice-based applications designed to run on the near-RT RIC (for tasks requiring latency between 10 ms and 1 s) and can be

developed by third-party software companies (Wani *et al.*, 2025). By operating through near-real-time closed control loops, xApps continuously monitor, analyze, and optimize network parameters, thereby enabling the network to achieve its intended performance and behavior (Santos *et al.*, 2025). Through the E2 interface, the near-RT RIC and its xApps can monitor, control, or override specific network functions, exchange control messages, and collect real-time data and feedback from the BS. This capability allows for the efficient management and optimization of operations within the BS (Agarwal *et al.*, 2025). The near-RT RIC is primarily designed to enable advanced and intelligent functionalities, including radio resource management for optimizing network performance. It also plays a key role in balancing traffic loads, ensuring efficient data transmission, and mitigating interference by identifying and addressing interference-related problems. In addition, the near-RT RIC manages QoS to support various service types and handles handover procedures as users transition among cells. Furthermore, it provides an open platform that allows third-party applications and services to enhance RAN management and overall network flexibility (Choudhary, Srivastava & Jha, 2024).

- **Non-RT RIC** is a logical component responsible for non-real-time activities requiring durations exceeding one second, such as ML model training, policy management, and optimization using data obtained over time (Wani *et al.*, 2025).

1.3.2 Resource Allocation in RAN Slicing

NS resource management covers computational, communication, and radio resources in both the CN and RAN domains, where each resource domain is subject to its own set of optimization objectives. Due to its potential to enable a wide range of emerging services in future networks, RAN slicing has recently garnered substantial attention from both academia and industry (Nagib, 2024). However, RRM in RAN slicing faces many challenges. Among these, MNOs need to satisfy the SLAs of slices while simultaneously minimizing radio-resource consumption (Zangooei, Golkarifard, Rouili, Saha & Boutaba, 2024). The complexity of RRM in the RAN domain arises from factors such as interferences, the dynamic nature of wireless channels,

scarcity of inherent radio resources, diverse SLA, densification of devices, and random traffic arrivals (Zangooei *et al.*, 2024; Nagib, 2024; Feng *et al.*, 2020; Anıl Akyıldız, Faruk Gemici, Hökelek & Ali Çırpan, 2024). The available radio resources are strongly affected by the stochastic nature of the channel conditions and time-varying user demand for different services. Estimating the short-term traffic demand for each service type is particularly challenging, especially with the emergence of new 6G services (Nagib, 2024). Radio resource allocation and management (RRAM) can plays a crucial role in providing enhanced communications within complex and expansive networks. Intelligent RRAM solutions ensure improved network connectivity, higher system efficiency, and reduced energy consumption. Overall, there are two significant factors impact wireless network performance: the efficient utilization and coordination of radio resources, and the ability of the system to adapt to unpredictable changes in network dynamics, such as wireless channel variations, user mobility patterns, real-time resource availability, and traffic load fluctuations (Alwarafy, Abdallah, Ciftler, Al-Fuqaha & Hamdi, 2021a).

Among the different radio resources that must be carefully managed in RAN slicing, power is one of the most critical. As 5G networks become increasingly dense and diverse, handling interference grows significantly more challenging, and traditional intercell interference-coordination algorithms lose their effectiveness (Xiong *et al.*, 2019). This is due to the fact that these kinds of algorithms need to solve a non-convex problem for each mini-slot, resulting in substantial computational complexity that renders real-time processing impractical (Xue *et al.*, 2024a). Power control challenge is particularly pronounced for uRLLC services, where links must maintain a high signal-to-noise ratio (SNR) to satisfy stringent QoS requirements (Xue *et al.*, 2024a). Nevertheless, increasing the transmit power is not a viable solution, as it may cause severe inter-cell interference and excessive energy consumption, especially in multi-connectivity scenarios (Xue *et al.*, 2024a). Although reducing the transmit power can help limit interference, it frequently comes at the expense of the user's data rate (Xiong *et al.*, 2019). Consequently, interference management in modern mobile systems is commonly formulated as a power-control optimization problem (Xiong *et al.*, 2019). Therefore, the power allocation problem in RAN slicing requires an

effective approach capable of operating in real time and adapting to dynamically changing active users.

An overview of RAN slicing resource allocation is shown in Fig. 1.7 that illustrates a scenario comprising a set of heterogeneous slices corresponding to different service types, including VoNR, eMBB, uRLLC, mMTC, and X-slice. Each service category is designed for distinct functions and application scenarios, which consequently leads to different latency, reliability, and data rate requirements for each category. In forthcoming 6G networks, these categories are expected to be subject to even stricter performance requirements (Nagib, 2024). When developing resource allocation methods for diverse services, a variety of factors must be considered, as stressed in (Qiao *et al.*, 2025). Among these factors is the recognition that a single-timescale radio resource allocation mechanism is inadequate for serving heterogeneous services. Therefore, to meet real-time demands of different services, resource allocation for slices must be performed across multiple timescales (Qiao *et al.*, 2025).

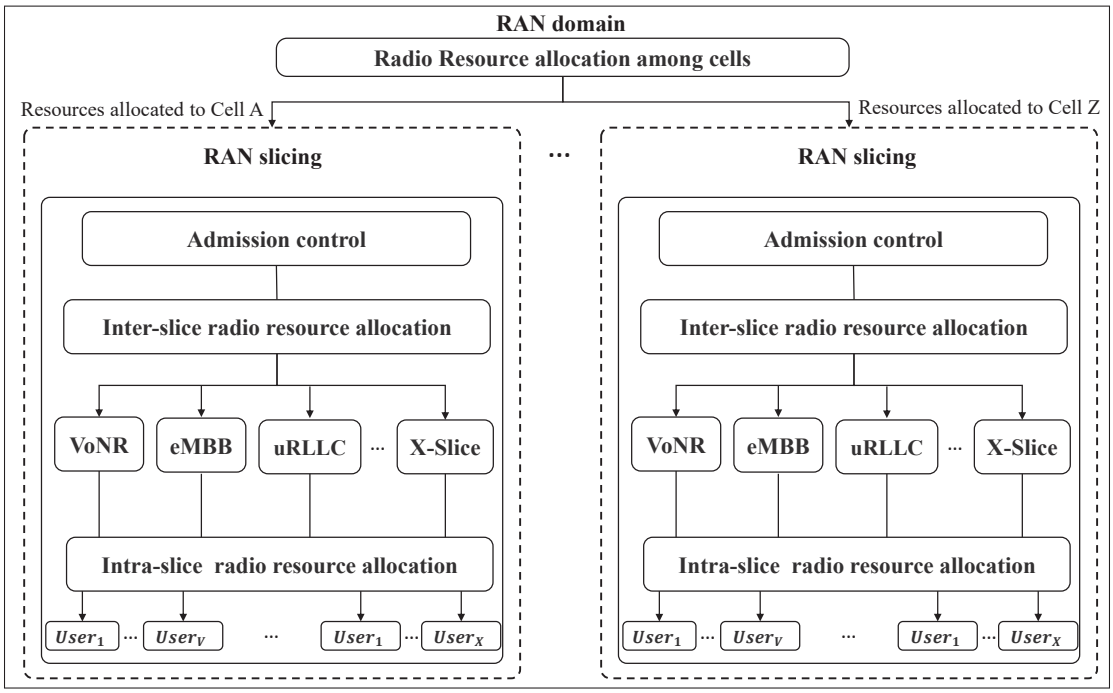

Figure 1.7 Overview of RAN slicing resource allocation
Adapted from Nagib (2024)

As shown in Fig. 1.7, the initial step in the resource-allocation process in NS is AC function. The main aim of the AC is to ensure that the required resources will be available when the network slice becomes operational. The AC is consistently part of the preparation phase in the NS (Ebrahimi *et al.*, 2024). Specifically, the AC decides whether the network has sufficient resources to admit slice requests and meet their specified requirements or reject them in a given cell. This decision enables the infrastructure provider to more efficiently manage the underlying resources, ensure fairness in resource distribution across the slice types, and increase revenue while minimizing SLA violations among admitted slices (Chakraborty & Sivalingam, 2023).

Once the required number of slice requests is approved by the AC, the system must allocate adequate resources to each slice. Adaptive resource assignment to admitted slices is a core function of NS management. At this stage, operators need to decide how much capacity each slice should receive so that network resources are efficiently used while maintaining low operational costs. The main difficulty lies in maintaining a balanced allocation that avoids the following two undesirable situations (Rahmanian *et al.*, 2021):

- **Under-provisioning:** Allocating fewer resources than a slice requires, which may result in violating the SLA established with the tenant (Rahmanian *et al.*, 2021);
- **Over-provisioning:** Allocating more resources than necessary, causing part of the available capacity to remain unused and reducing overall efficiency (Rahmanian *et al.*, 2021).

To realize full automation and ZT functionality within the RAN domain, radio resource management must be performed at two complementary levels: inter-slice and intra-slice. Furthermore, RAN resource management schemes should be capable of self-adapting to rapidly changing network conditions, efficient scaling with the growing number of connected devices and hosted slices, and autonomous operation in order to handle the challenges of future networks.

1.3.2.1 Inter-Slice Resource Allocation

The inter-slice resource allocation layer was previously introduced in next-generation wireless networks (NGWNs) as part of the NS RRM framework to ensure that each network slice adheres

to its specific SLA requirements (Debbabi, Jmal, Fourati & Aguiar, 2022). The aim of inter-slice allocation is to distribute network resources among admitted slices at a large timescale (Qiao *et al.*, 2025); (as shown in Fig. 1.8).

1.3.2.2 Intra-Slice Resource Allocation

This type of allocation occurs after resources are partitioned among slices. It operates over short timescale, and during the RRM process, the scheduler allocates available resources among users within the same slice, considering factors such as interference control, spectral efficiency, load balancing, and user fairness (Debbabi *et al.*, 2022); (as presented in Fig. 1.8).

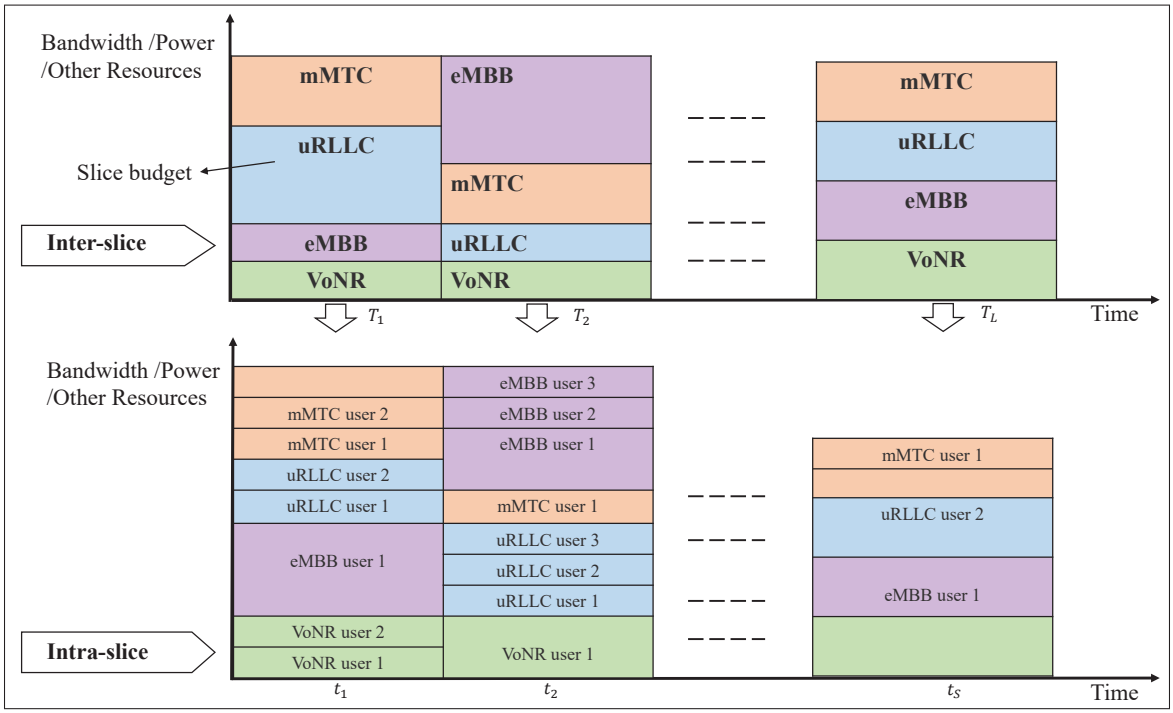

Figure 1.8 Illustration of inter- and intra-slice resource allocation
Adapted from Nagib (2024)

1.4 Optimization Approaches for RAN-Slicing RA

In this section, we provide an overview of the optimization approaches used for RAN-slicing resource allocation. A review of traditional optimization methods is provided in Section 1.4.1, which is followed by a presentation of ML–based approaches in Section 1.4.2.

1.4.1 Traditional Optimization Approaches

To date, a variety of techniques have been used to manage and solve radio resource problems in RAN slicing. These techniques include heuristic methods (Chang & Nikaein, 2018), and lagrange methods for optimization (Zhang *et al.*, 2017). However, these approaches are not well-suited to B5G networks and their heterogeneous service demands (Zangooei, Saha, Golkarifard & Boutaba, 2023). This is due to various reasons. First, the approximate mathematical models used in traditional optimization approaches lack the ability to accurately and fully describe complex dynamic environments in real-world networks (Liu, Ding & Liu, 2020; Zangooei *et al.*, 2023). In addition, the massive number of connected devices increases the complexity of mobile networks, evolving communication models, and diverse QoS requirements (Zangooei *et al.*, 2023). This leads to an increase in the network optimization parameters. As pointed out by Abiko *et al.* (2020), the number of network parameters required for mobile network optimization has grown from 1500 parameters for 4G to over 2000 parameters for 5G. This raises an important question: How much more complex will these parameters become in 6G and beyond?

Furthermore, the significant computational complexity of searching in large, complicated, and dynamic environments frequently results in suboptimal performance of the aforementioned approaches (Zangooei *et al.*, 2023; Nagib, 2024). Finally, the dynamics of service traffic and network circumstances frequently exhibit long-term and short-term trends. However, traditional methods, do not adapt to such hidden network dynamics because they lack the ability to learn data patterns (Zangooei *et al.*, 2023). Different programming approaches, such as mixed integer programming, dynamic programming, and stochastic programming, have been used to solve various optimization problems. These approaches work well, especially when dealing

with sequence optimization problems. However, a notable drawback of these programming approaches is that they require performing calculations from scratch for each iteration, which can be time-consuming and inefficient (Zhang, Zhang & Qiu, 2020). Furthermore, conventional techniques, such as exhaustive search, genetic algorithms, combinatorial approaches, and branch and bound methods, have historically been used to address the non-convex nature of optimization problems. However, in the case of expansive cellular networks, computational demands imposed by these techniques make them unviable and infeasible (Ishfaq Ahmed & Hossain, 2019).

According to Alwarafy, Albaseer, Ciftler, Abdallah & Al-Fuqaha (2021b) traditional RRA techniques struggle to cope with the rising heterogeneity and scalability demands of future 6G heterogeneous networks. These methods depend on perfect CSI and become computationally infeasible even when such complete CSI is available. Consequently, there is a continued need for scalable and robust RRA approaches capable of managing substantial complexity and diversity of 6G systems.

1.4.2 Machine Learning-Based Approaches

ML is an area of AI that develops algorithms capable of learning from data, making predictions, and enabling data-driven decision making. ML techniques are widely applied in NS frameworks to anticipate network performance, optimize resource allocation, and support automated decision making (Ebrahimi *et al.*, 2024). Their use is growing in NGWNs to handle increasing network complexity while preserving strong performance. ML methods can effectively learn the unknown and nonlinear behaviors of dynamic wireless environments, offering advantages over traditional techniques. They also help to achieve a good balance between the system performance and computational cost. Furthermore, NGWNs are expected to include built-in intelligence in the O-RAN architecture, enabling more flexible and AI-driven RAN slicing. Consequently, numerous recent studies have proposed ML-based approaches to address the challenges related to NS (Nagib, 2024). These approaches generally fall into the following four major ML categories: supervised learning (SL), unsupervised learning (UL), deep learning (DL), and reinforcement learning (RL). In what follows, we discuss these three categories in further detail.

1.4.2.1 SL-Based Approach

SL uses labeled data to learn input–output relationships and make predictions for unseen inputs (Friesen, Wisniewski & Jasperneite, 2022). It enables an accurate prediction of network performance metrics using labeled data and supports resource management decisions within the NS. For example, regression tree methods can be applied to estimate physical resource blocks (PRBs) utilization in the RAN based on users' traffic patterns (Ebrahimi *et al.*, 2024). SL techniques are particularly effective for classification and regression tasks, where the goal is to predict a specific output value or class based on given inputs. In ZTNs, SL is applied to tasks such as traffic classification, predicting service requirements, and anticipating user behavior (Yang, Rajab, Shami & Muhaidat, 2024b). However, such approaches still largely depend on the availability of high-quality training data that reliably capture the characteristics of network slices (Ebrahimi *et al.*, 2024).

1.4.2.2 UL-Based Approach

The goal of UL approaches is to analyzes unlabeled data to identify patterns (Dulaj, Alhammadi, Shayea, El-Saleh & Alnakhli, 2025), guiding decision-making by identifying hidden relationships in NS data (Ebrahimi *et al.*, 2024), without receiving any feedback from the environment (Friesen *et al.*, 2022). Among the most commonly used UL methods are clustering algorithms. These methods are used to divide input data into separate clusters (*i.e.,* groups) based on their similarity. Each observation within the same cluster has a higher degree of similarity than observations in other clusters (Aouedi, Piamrat, Hamma & Perera, 2022). For instance, clustering is applied for slice AC, allowing new slice requests to be matched with the most appropriate existing slices for efficient resource utilization. However, interpreting the results of such methods frequently requires expert knowledge and careful analysis (Ebrahimi *et al.*, 2024).

1.4.2.3 DL-Based Approach

DL uses artificial neural networks (ANNs), also known as neural networks (NNs), which are sets of mathematical functions designed to produce the desired output from an input dataset by imitating the behavior of biological neurons (Djigal, Xu, Liu & Zhang, 2022). A neural network consists of interconnected nodes, including an input layer, one or more hidden layers, and an output layer, as illustrated in Fig. 1.9. Each node (artificial neuron) has an associated weight and bias (Djigal *et al.*, 2022). DL models acquire knowledge from data by propagating inputs through multiple layers, each of which transforms the information to capture essential features. Activation functions are central to this process, as they introduce non-linearity, enabling the network to learn complex patterns and relationships. By applying functions such as ReLU, sigmoid, or softmax, the model can effectively control neuron outputs and optimize predictions for various tasks (Mienye & Swart, 2024). DL algorithms rely on NNs that operate in two main phases: training and inference (Djigal *et al.*, 2022). In the context of NS, one widely used DL approach is the Long Short-Term Memory (LSTM) model, which is particularly effective for optimizing radio resource allocation (Abood *et al.*, 2023).

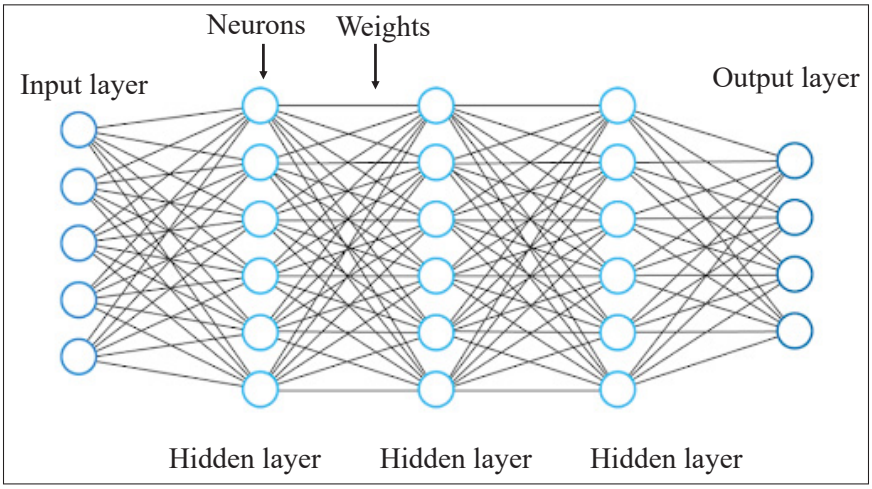

Figure 1.9 Overview of neural network

1.4.2.4 RL and DRL-Based Approaches

Originally inspired by behaviorist psychology (Arulkumaran, Deisenroth, Brundage & Bharath, 2017), RL is a type of ML that fundamentally depends on the trial-and-error learning approach. RL supports automated decision making in dynamic environments (Ebrahimi *et al.*, 2024). It follows a MDP, which is a mathematical framework for modeling decision making. MDP is made up of four components $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T})$, where: $\mathcal{S}$ refers to a set of states, $\mathcal{A}$ is a set of possible actions, $\mathcal{R}$ is the reward function, and $\mathcal{T}(s'|s_t, a_t)$ is the state transition probability representing the probability of action $a \in \mathcal{A}$ under current state $s \in \mathcal{S}$ at slot t, leading to state $s' \in \mathcal{S}$ at slot $t + 1$ (Li *et al.*, 2018). In RL, at each time step t, the agent receives a state $s_t \in \mathcal{S}$; accordingly, the agent selects an action $a_t \in \mathcal{A}$. The action is chosen based on policy π, which represents rules that the agent follows to select an action. The policy maps each state s_t to the corresponding action a_t. After taking an action, the agent receives a scalar reward r_t from the environment. The reward can be positive, negative, or zero. Following the agent's action, the environment transitions to a new state (s') based on its dynamics (Li, 2017). The interaction between the agent and its environment is depicted in Fig. 1.10, where the main goal of the agent-learning process is finding the best policy that maximizes future rewards, as shown in Eq. (1.1) (Arulkumaran *et al.*, 2017):

$$R_t = \sum_{t=1}^{T} \gamma^{t-1} r_t = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{T-1} r_T, \tag{1.1}$$

where $\gamma \in [0, 1]$ discount factor balances the importance between immediate and future rewards; here, smaller values prioritize immediate rewards (Arulkumaran *et al.*, 2017). Furthermore, t is the current time and T denotes the last time step in the episode.

In general, RL algorithms are classified into the following two main types: value-based and policy-based (Hao *et al.*, 2024).

- **Policy-based RL** aims to directly optimize policy π by maximizing the expected return, where π associates states with the probability of performing an action. A well-known example of this type is the REINFORCE algorithm (Meng, Chen, Wu & Cheng, 2020).

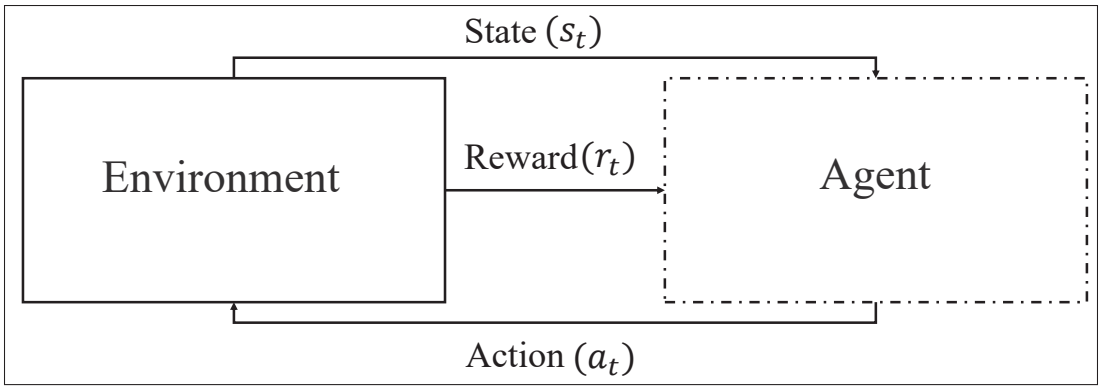

Figure 1.10 Overview of the interaction between the RL agent and its environment

- **Value-based RL** focuses on learning the optimal value function, which evaluates how good it is for an agent to be in a particular state; and this is known by state-value function ($V(s)$); or how good it is for an agent to be in a given state and taking a specific action; this is known by action-value function ($Q(s,a)$). Specifically, value function $V(s)$ of the state s under policy π is calculated by getting the expected return value of s and mathematically defined by $V(s) = \mathbb{E}[R_t \mid s_t = s, \pi]$, where $\mathbb{E}[.]$ denotes the expected value of a random variable. Whereas the action-value function $Q(s,a)$ measures the expected cumulative reward when taking an action in a given state which mathematically defined by $Q(s,a) = \mathbb{E}[R_t|s_t = s, a_t = a, \pi]$ (Nguyen, Nguyen & Nahavandi, 2020). The Q-value function can be iteratively calculated using the Bellman equation (Ge, Liang, Joung & Sun, 2020).

A well known example of value-based RL is Q-learning algorithm, which is a model-free RL algorithm. It employs a Q-function that maps each state–action pair to an estimated cumulative reward, referred to as the Q-value. By repeatedly updating these Q-values and choosing the actions that yield the highest Q-values, the agent gradually learns the optimal set of state–action choices, which represents the optimal policy (Xiong *et al.*, 2019). Q-learning algorithm uses a Q-table to save and update the Q-function values (Youssef, Nour, Lagrange & Douillard, 2022), where the agent must keep a table of Q-values and continually update them for every combination

of state and action. The Q-table is structured as $|\mathcal{S}| \times |\mathcal{A}|$ matrix, with $\mathcal{S}$ representing the set of states within the environment and $\mathcal{A}$ corresponding to the set of available actions (Kantasewi, Marukatat, Thainimit & Manabu, 2019; Ge *et al.*, 2020). In the future wireless networks characterized by massive scale, heterogeneity, and decentralized architectures, the range of possible system states can grow extremely large. Under these conditions, tracking and updating all Q-values becomes impractical. Hence, Q-learning algorithm becomes inefficient because it needs a huge Q-table and requires a long time to converge (Youssef *et al.*, 2022).

To tackle the challenges of RL, in 2015, a significant achievement was made by Mnih *et al.* who introduced the integration of DL into RL (Nguyen *et al.*, 2020), which led to the development of DRL. Following this advancement, DRL techniques have gained significant research attention for future mobile networks, as they offer strong capabilities for handling large-scale and rapidly changing network conditions (Xiong *et al.*, 2019). Moreover, the ability of DRL to autonomously optimize system performance through interaction with unknown environments (Mei *et al.*, 2021) has also captured considerable research interest.

For instance, one of the well-known DRL algorithms is DQN, a value-based algorithm that uses a deep neural network (DNN) to approximate the Q-function. The DQN algorithm receives system states as input; then, each input state is sent to a different layer of the neural network using certain weights. The DNN then produces Q-values corresponding to every possible action the agent can take (Xiong *et al.*, 2019). Figure 1.11 illustrates the difference between Q-learning and DQN in evaluating the Q-value. Another example of DRL algorithms is A2C, which combines the strengths of both policy-based and value-based methods (Kouchaki & Marojevic, 2022). A2C is known for its effectiveness in addressing RAN resource management problems.

For allocating resources for multiple slices, one possible approach is to employ a single centralized DRL agent that has full visibility of the system state and controls the behavior of all slices. However, although this approach may achieve the best overall performance, it is generally impractical due to the significant signaling overhead it introduces (Kim & Lim, 2021). Moreover, in real-world implementations, finding an optimal solution becomes challenging because the

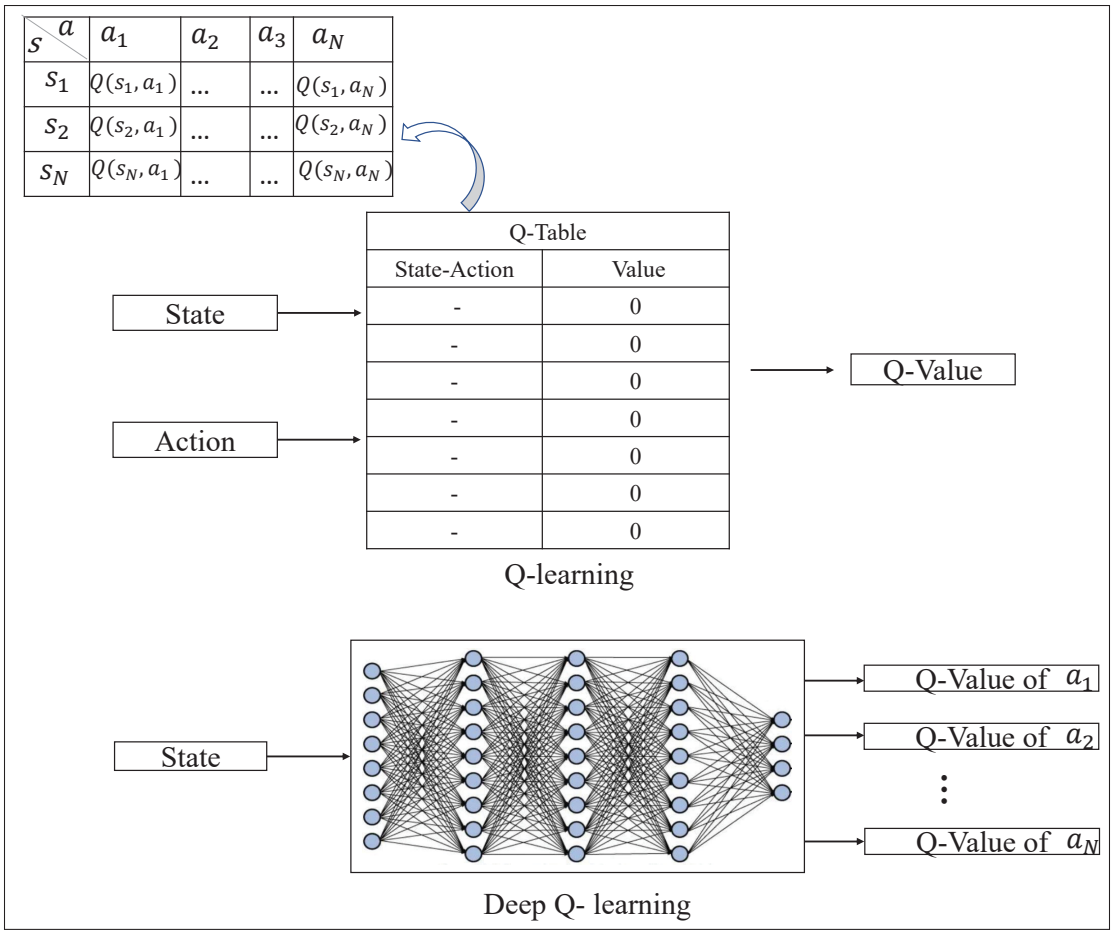

Figure 1.11 Q-Learning vs Deep-Q-Learning
Adapted from Badini *et al.* (2024)

action space grows exponentially with the number of slices (Kim & Lim, 2021), the number of resources that can be managed by the agent for each slice, as well as by the number of users per slice. Therefore, in this thesis, we adopt instead decentralized MADRL approaches that are briefly introduced in the next section.

1.4.2.5 Multi-Agent DRL

Multi-agent reinforcement learning (MARL) is a significant branch of RL (Shi *et al.*, 2022) that addresses sequential decision-making scenarios involving a set of agents, where the overall system dynamics depend on their combined/ joint actions. Consequently, the reward obtained by

any individual agent is determined not solely by its own actions, but also by the joint actions of all agents in the shared environment (Feriani & Hossain, 2021). Based on the relationship between different agents, multi-agent tasks are classified into the following three categories: fully cooperative, fully competitive, and mixed (cooperative–competitive) settings (Shi *et al.*, 2022). In cooperative situations, all agents operate under a shared reward function. By contrast, competitive situations involve agents whose objectives oppose one another (Feriani & Hossain, 2021). A simple schematic representation of the MARL interactions is depicted in Fig. 1.12, where a set of N agents observes their individual states of the environment and take a joint decisions to improve it. Each agent then receives a reward and a new state, and the process continues until a terminal condition is satisfied (Albrecht, Christianos & Schäfer, 2024). The learning and training in cooperative MADRL can be classified into two main approaches outlined below:

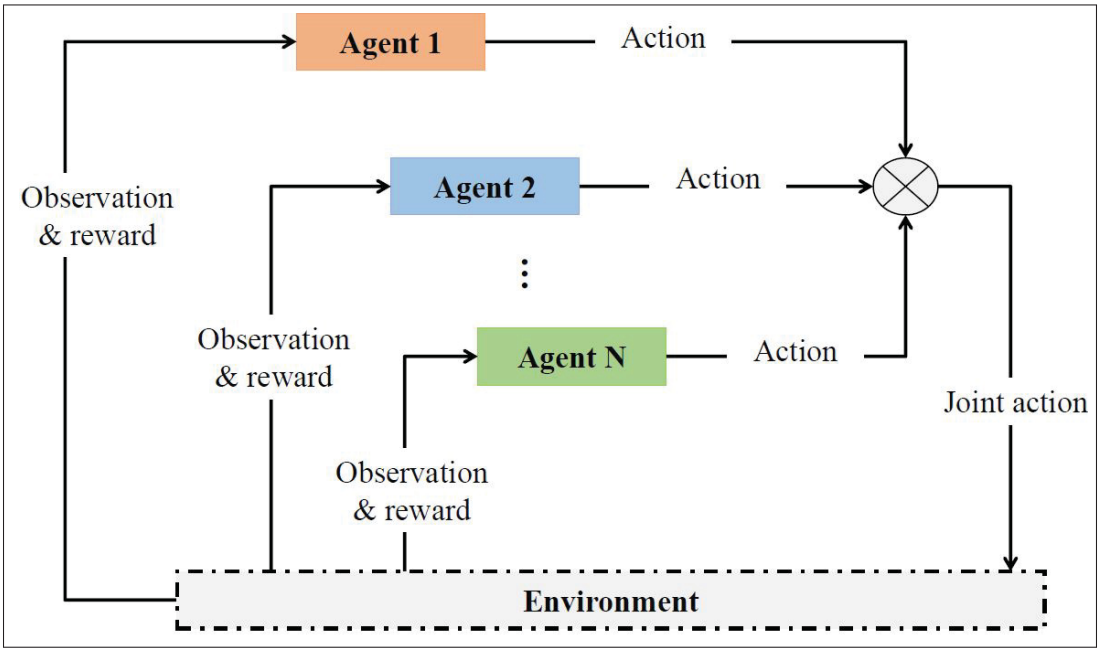

Figure 1.12 Schematic of MADRL
Adapted from Albrecht *et al.* (2024)

- **Central Learning (CL) Approach:** In this approach, a single central policy π_c is trained and acts as a central controller that gathers all agents' observations and uses the aggregated information to determine joint actions from the set $A = A_1 \times \cdots \times A_N$ (Albrecht *et al.*,

2024; Jin *et al.*, 2025). One of the main advantages of this strategy is that it avoids common multi-agent challenges such as non-stationarity and credit assignment (Albrecht *et al.*, 2024). However, its key disadvantage is that this approach requires centralization during both training and execution, causing the state and action spaces to grow exponentially with the number of agents (Gupta, Egorov & Kochenderfer, 2017), which makes learning computationally expensive. CL also faces a fundamental limitation imposed by the distributed nature of multi-agent systems, where direct communication between a central policy π_c and all agents may be infeasible or undesirable in many practical scenarios (Albrecht *et al.*, 2024). Furthermore, this approach is not feasible for large-scale or real-time systems (Feriani & Hossain, 2021; Jin *et al.*, 2025).

- **Independent learning (IL) Approach:** In the IL approach, each agent learns and optimizes its own policy π_n independently, based solely on its local information (states, actions, and rewards), while neglecting the presence of other agents. Agents neither observe nor use information about others; instead, other agents' behavior is treated as part of the environment's dynamics (Albrecht *et al.*, 2024). In this framework, each agent trains and operates fully independently, updating its strategy only from its own observations and rewards (Jin *et al.*, 2025).

  An important benefit of the IL approach is that it simplifies the development of heterogeneous policies (Gupta *et al.*, 2017). Additionally, IL the approach naturally avoids the exponential growth in action spaces that arises in CL and is well suited to systems where agents operate based on local policies. However, it faces a key challenge as the environment becomes non-stationary, because all agents learn and adjust their behaviors simultaneously (Albrecht *et al.*, 2024). Despite this limitation, the IL approach can perform well in real-world settings, as recent studies demonstrated its successful application to various resource allocation and control problems in wireless communication networks (Feriani & Hossain, 2021). Furthermore, the IL approach demonstrated strong scalability and robustness, particularly in environments with limited communication (Jin *et al.*, 2025).

1.5 Conclusion

This chapter provided a comprehensive background on ZTNs, beginning with their definition, objectives, and significance in next-generation wireless networks. It then discussed the key enabling technologies, including SDN, NFV, and NS, highlighting how they together support flexible, programmable, and efficient network management. The evolution of RAN and the O-RAN architecture was examined, emphasizing the role of intelligent RAN design in enabling advanced resource allocation mechanisms. Following this, the chapter addressed resource allocation in NS, distinguishing between inter- and intra-slice strategies and highlighting their importance for achieving high performance and QoS. The chapter then turns to the traditional optimization approaches used in NS, showing their limitations in handling dynamic, large-scale networks. This led to a discussion of ML–based methods, including SL, UL, DL, RL and DRL for autonomous, adaptive decision-making. The chapter concluded with MADRL, demonstrating how decentralized, scalable approaches can effectively manage resources in complex, distributed, and heterogeneous network environments. By following this sequence, it provides a clear context and motivation for adopting advanced ML-based and multi-agent solutions, which are further explored and evaluated in the subsequent chapters.

CHAPTER 2

PABSO-DRL: POWER AND BEAM SELF-OPTIMIZATION SCHEME FOR MULTIPLE SLICES IN MU-MISO SYSTEMS

Ohood Sabr[1], Georges Kaddoum[1,2], and Kuljeet Kaur[1,3]

[1] Department of Electrical Engineering, École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, QC H3C 1K3, Canada
[2] Cyber Security Systems and Applied AI Research Center, Lebanese American University, Beirut, Lebanon
[3] Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, 140401, Punjab, India

Paper published in *IEEE Transactions on Consumer Electronics*, November 2024

2.1 Abstract

Recently, the concept of zero-touch networks (ZTNs) relying heavily on pervasive artificial intelligence (AI) algorithms for the full automation of future networks have emerged. ZTNs, empowered by AI, will play a significant role in reducing the management complexity of beyond fifth-generation (B5G) networks. One enabling technology of ZTNs is network slicing (NS), which is the cornerstone of B5G networks. However, NS faces challenges, particularly in terms of radio resource allocation management. Therefore, this paper presents an efficient self-optimization scheme for a multi-user multiple-input single-output (MU-MISO) system across different slices; named PABSO-DRL. This scheme dynamically and jointly manages power allocation and beam direction based on a deep reinforcement learning (DRL) framework, considering imperfect channel state information (CSI) and the time-varying dynamics of the NS environment. PABSO-DRL ensures high data rates for enhanced mobile broadband (eMBB) while guaranteeing high reliability for ultra-reliable low latency communications (uRLLC). The problem is formulated as a multi-objective optimization problem and solved by designing multiple deep Q-learning (DQL) agents. The proposed scheme is extensively evaluated under two scenarios: perfect and imperfect CSI, comparing its performance with four traditional benchmark algorithms, and state-of-the-art schemes. Results demonstrate the superiority of

the proposed DRL scheme, even under imperfect CSI, highlighting its adaptability to various network slicing conditions.

2.2 Introduction

Fifth generation (5G) of mobile communications has emerged in response to anticipated future network needs, aiming to address challenges associated with the handling of high traffic volumes, enhancing the quality of experience (QoE), and achieving greater affordability through cost reduction strategies (Ojaghi *et al.*, 2022). Expected to bring significant improvements in mobile connectivity, 5G incorporates three main slices: enhanced mobile broadband (eMBB), ultra-reliable low latency communications (uRLLC), and massive machine-type communications (mMTC) (Abbas *et al.*, 2024). eMBB refers to traffic with high data rate demands, with up to 20 Gb/s peak data rate and 100 Mb/s everywhere; it is ideal for multimedia applications, such as augmented reality. The uRLLC slice, on the other hand, is designed to meet the needs of applications with stringent requirements for extremely low latency (5ms) and high reliability (99.9999%), such as autonomous driving and smart factories. Finally, mMTC enables connection densities of up to 1 million devices per square kilometer of affordable and low-energy devices (10 years of battery life) (Ojaghi *et al.*, 2022). There are essential technologies driving 5G networks, including software-defined networking (SDN), network function virtualization (NFV), and network slicing (NS) (Yang, El Rajab, Shami & Muhaidat, 2023). NS enables the creation of multiple service-specific logical networks, such as eMBB, uRLLC, and the consumer internet of things (CIoT), all functioning on a shared common physical infrastructure (Abbas *et al.*, 2024; Marzouk, Radwan, Chi & Barraca, 2023). Each logical network is designed and optimized to serve a specific group of consumers (Abbas *et al.*, 2024; Marzouk *et al.*, 2023). Despite these advancements, the capabilities of 5G are expected to reach their limits by 2030 (Tariq *et al.*, 2020). This is attributed to its shortcomings, such as scalability, communication speed, link reliability, and latency, which will not meet the requirements of future services and applications (Tariq *et al.*, 2020). Therefore, intensive efforts are underway to enhance network infrastructure by shifting towards the sixth generation (6G) to accommodate the increasing

number of connected internet of things (IoT) devices, and meet the stringent requirements of future services (Jiang *et al.*, 2021). 6G is expected to facilitate full dimensional wireless coverage by integrating existing heterogeneous networks, including ground, space, air, and underwater. This integration, along with the emergence of new services and applications, will result in extremely complicated and heterogeneous mobile networks (Liyanage *et al.*, 2022; Alwarafy *et al.*, 2021b). Conventional network management and orchestration strategies will struggle to efficiently manage network operations such as monitoring, optimization, and control (Liyanage *et al.*, 2022). This requires a network management system that ensures low operational costs, better scalability, and is free from human error (Liyanage *et al.*, 2022).

To this end, zero-touch networks (ZTNs) have emerged as a modern solution for automating all network operations and reducing the need for manual interventions. ZTNs strive to achieve 100% automation using artificial intelligence (AI) and machine learning (ML) to ensure end-to-end automation in cellular networks (Yang *et al.*, 2023). This is achieved by classifying network operations into four classes: self-configuration, self-optimization, self-protection, and self-healing (Yang *et al.*, 2023). The integration of NS, SDN, and NFV with AI forms enabling technologies that lay the groundwork for achieving ZTNs functionality (Yang *et al.*, 2023). One of the essential enabling technologies for 6G systems is expected to be NS (Mei *et al.*, 2021). However, the heterogeneous and dynamic quality of service (QoS) requirements of future applications cannot be fully supported by the NS mechanism currently utilized in 5G (Mei *et al.*, 2021). Therefore, NS must continue to advance in its flexibility and adaptability to meet the requirements of future networks. Nevertheless, NS will significantly increase the complexity of wireless network operations, rendering conventional mathematical model-based approaches to network operation insufficient (Mei *et al.*, 2021). This motivates us to incorporate AI capabilities into NS for future 6G-based applications. AI algorithms, particularly deep reinforcement learning (DRL) algorithms, have shown potential for achieving automation in RAN slicing (Xiong *et al.*, 2019; Mei *et al.*, 2021). DRL algorithms, such as deep Q-learning (DQL), are model-free, which enables them to operate without the need for prior knowledge of

the NS environment. Instead, they rely on direct interactions with the environment through a trial-and-error strategy (Ge *et al.*, 2020).

NS is implemented across three levels: core, transport, and radio access network (RAN) (Azimi, Yousefi, Kalbkhani & Kunz, 2022b). However, it is noted in the literature that RAN slicing is much more challenging, particularly due to resource allocation (Azimi *et al.*, 2022b). Despite existing efforts, there remain unresolved issues in RAN slicing, namely, dynamic radio resource sharing, isolation, and the potential conflicts in sharing radio resources among slices (Mei *et al.*, 2021; Azimi *et al.*, 2022b).

To address these challenges and ensure future network designs incorporate automation, integrating zero touch (ZT) operations into the resource allocation framework becomes essential. Given the anticipated integral role of NS in shaping 6G networks and its consideration as an enabling technology for ZTNs, this paper aims to explore radio resource allocation and management for NS. The objective is to design a power and beam self-optimization scheme for multiple slices in a multi-user multiple-input single-output (MU-MISO), named PABSO-DRL. The technical contributions of this paper can be summarized as follows:

- To propose an efficient self-optimization scheme for uRLLC and eMBB that dynamically and jointly manages power allocation and beam direction in a MU-MISO system under perfect and imperfect channel state information (CSI).
- To adopt a unicast transmission mode within slices, with the capability to simultaneously send multiple beams to serve different users. It enables multi-beam transmission with different QoS requirements in the NS domain.
- To maximize the long-term data rates for eMBB while minimizing the outage probability to ensure connection reliability for uRLLC. This is to be accomplished by formulating the problem as a Markov decision process (MDP) framework and solved by designing multiple DQL agents.
- To enhance the scalability, the proposed scheme is designed for flexibility in accommodating various sizes of uniform linear array antennas.

- To extensively evaluate the proposed scheme in the NS environment, while taking into consideration non-line-of-sight, imperfections, and intra-interference. Also, to evaluate its performance by comparing with four traditional benchmark algorithms, and current state-of-the-art schemes.

2.2.1 Organization

The rest of this paper is structured as follows. In Section 2.3, the literature review is presented. Section 2.4 elaborates on the system model. Section 2.5 presents the problem formulation. Thereafter, Section 2.6 provides the proposed solution. Section 2.7 highlights the performance of the proposed scheme and discusses the obtained results. The discussion is presented in Section 2.8. Finally, conclusions and future works are summarized in Section 2.9.

2.3 Literature Review

This section discusses related works and the importance of DRL along with its applications in RAN.

Table 2.1 Summary of related works

Ref.	Considered Slices	Optimized Parameters	Study Goals	Type of CSI	Adopted Beamforming Technique	Methodology
(Zhao, Chi, Qian, Zhu & Hou, 2022)	eMBB & uRLLC	Power & bandwidth	Minimize power & bandwidth	ICSI only for uRLLC	—	Lyapunov drift-plus-penalty method
(Setayesh, Bahrami & Wong, 2022)	eMBB & uRLLC	Power & bandwidth	Maximize throughput for eMBB & minimize the power consumption for uRLLC	PCSI	—	Attention-based DNN algorithm
(Filali, Mlika, Cherkaoui & Kobbane, 2022)	eMBB & uRLLC	RB	Maximize the total sum rates	PCSI	—	DRL
(Zhang, She, Ying, Li & Vucetic, 2023)	uRLLC	RB	Maximize resource utilization efficiency	ICSI	—	DRL
(Yan, NG, Ke & Lam, 2024)	eMBB, uRLLC & mMTC	RB & beamforming	Maximize the utility function	PCSI	Codebook	DRL & K-means algorithm
(Ghanem, Jamali, Sun & Schober, 2020)	uRLLC	Power	Maximize the weighted sum throughput	PCSI & ICSI	—	Monotonic optimization
(Elsayed & Erol-Kantarci, 2020)	eMBB & uRLLC	RB & beamforming	Maximize sum rate for eMBB & minimize latency for uRLLC	PCSI	Not declared	Online clustering
(Tang, Shim, Chang & Quek, 2019a)	eMBB & uRLLC	Power & bandwidth	Minimize total power consumption	PCSI	Not declared	SCA & SDR techniques
(Tang, Shim & Quek, 2019b)	eMBB & uRLLC	Bandwidth & beamforming	Maximize operator's revenue	PCSI	Not declared	SCA & SDR techniques
(Ginige, Shashika Manosha, Rajatheva & Latva-aho, 2020b)	eMBB & uRLLC	Power, bandwidth & beamforming	Maximize eMBB user admission	PCSI	Not declared	Approximation methods & SCP
(Slalmi, Chaibi, Saadane & Chehri, 2021b)	eMBB & uRLLC	Power, bandwidth & beamforming	Maximize eMBB user admission	PCSI	Not declared	SCP

* PCSI: perfect CSI, ICSI: imperfect CSI.

Existing literature on resource allocation for eMBB and uRLLC slices can be classified into two main lines of research. The first line of research focuses on designing resource-allocation algorithms based on different methods for jointly managing power and bandwidth allocation for eMBB and uRLLC slices in single-input and single-output (SISO) systems. Examples of this include the algorithm designed by Zhao *et al.* (Zhao *et al.*, 2022) based on the Lyapunov drift-plus-penalty method and the algorithm proposed by Setayesh *et al.* (Setayesh *et al.*, 2022) which utilizes attention mechanism with deep neural networks (DNNs). Another example of

resource allocation schemes for SISO system was introduced by Filali *et al.* (Filali *et al.*, 2022) to manage resource block (RB) for eMBB and uRLLC slices, where the authors designed a scheme-based on DRL. However, the proposed approach in (Zhao *et al.*, 2022) assumes perfect CSI for the eMBB slice, while (Setayesh *et al.*, 2022; Filali *et al.*, 2022) both assume perfect CSI for eMBB and uRLLC slices. Additionally, (Zhao *et al.*, 2022; Setayesh *et al.*, 2022; Filali *et al.*, 2022) neglect intra-interference, which may significantly impact the system's performance in real-world deployment scenarios. Moreover, the dynamic allocation of power, which is a scarce resource, has not been addressed in (Filali *et al.*, 2022).

The second line of research focuses on designing resource allocation algorithms for eMBB and uRLLC in MISO systems. For instance, Zhang *et al.* (Zhang *et al.*, 2023) proposed a DRL algorithm for managing the RB allocation for uRLLC slice. Ghanem *et al.* (Ghanem *et al.*, 2020) developed a power allocation algorithm under both perfect and imperfect CSI based on monotonic optimization techniques for the uRLLC slice. Elsayed *et al.* (Elsayed & Erol-Kantarci, 2020) pointed out that beyond 5G (B5G), wireless networks are expected to face increased complexity and high dynamicity due to user mobility and diverse applications with various requirements. This will result in challenges not only in resource allocation but also in beam management. To address these challenges, Elsayed *et al.* introduced a scheme for RB allocation and beam management for eMBB and uRLLC slices. The authors applied a DRL algorithm for RB allocation and utilized an online clustering algorithm, known as density-based spatial clustering of applications with noise, for beam management. The proposed solution has a limitation in terms of beam management if applied in ZT environments, due to its static nature and lack of adaptability. Yan *et al.* (Yan *et al.*, 2024) proposed a scheme for inter-slicing that utilizes DRL and K-means algorithm to manage RB and analog beamforming. However, perfect CSI is assumed, and intra-cluster interference is overlooked, which may significantly impact the system's performance in real deployment environments. Additionally, the proposed approach utilizes the K-means algorithm to classify users based on their channel characteristics and then selects a beam to serve each cluster. As the number of users grows, the clustering algorithm may encounter performance issues, particularly if users exhibit diverse and unique channel conditions,

potentially leading to ineffective beam selection. Moreover, there is another challenge: how to ensure that the users of each cluster can be effectively served by a single beam, especially when they are not located close to each other. Tang *et al.* (Tang *et al.*, 2019a) proposed an algorithm based on successive convex approximation (SCA) and semidefinite relaxation (SDR) techniques to manage power allocation and beamforming for eMBB and uRLLC slices. The authors implemented multicast transmission for the eMBB slice and unicast transmission for the uRLLC slice. Flexible frequency division duplex was employed to prevent interference between slices and within each slice.

Furthermore, Tang *et al.* (Tang *et al.*, 2019b) proposed an algorithm to manage bandwidth and beamforming for eMBB and uRLLC slices, following a methodology similar to that in (Tang *et al.*, 2019a). Ginige *et al.* (Ginige *et al.*, 2020b) introduced an algorithm based on approximation methods to manage the power, bandwidth, and beamforming for eMBB and uRLLC slices. The authors considered orthogonal spectrum sharing strategies to prevent interference between the slices and employed orthogonal frequency division multiple access (OFDMA) to avoid interference among uRLLC users. Slalmi *et al.* (Slalmi *et al.*, 2021b) proposed an algorithm based on the sequential convex programming (SCP) method to manage allocating power, bandwidth, and beamforming for eMBB and uRLLC. The authors followed the same strategies as in (Ginige *et al.*, 2020b) to prevent interference between slices and within the uRLLC slice.

To the best of the authors' knowledge, only a few studies have focused on power allocation or beamforming in multi-slice multi-antenna systems (Ghanem *et al.*, 2020; Elsayed & Erol-Kantarci, 2020; Tang *et al.*, 2019a,b; Ginige *et al.*, 2020b; Slalmi *et al.*, 2021b), all of which are based on iterative optimization algorithms. Additionally, only one of these studies (Ghanem *et al.*, 2020) considered imperfect CSI for a single slice. However, these traditional approaches are not well-suited for B5G networks and their heterogeneous service demands. This is because the approximate mathematical models utilized in traditional optimization approaches lack the ability to accurately and fully describe complex dynamic environments in real-world networks (Zangooei *et al.*, 2023). Additionally, a significant drawback of these approaches is their need to

perform calculations from scratch for each iteration, resulting in high computational complexity and inefficiency (Ge, Liang, Zhang, Long & Sun, 2023; Zangooei *et al.*, 2023). Consequently, they lack dynamic adaptability, scalability, and self-learning capabilities, which are essential for ZT operations.

In contrast to traditional approaches, DRL emerges from the integration of reinforcement learning (RL) principles with DNNs, proving highly effective in addressing complex, high-dimensional, and dynamic problems such as resource allocation in NS (Xiong *et al.*, 2019). DRL is a promising candidate for managing and controlling networks in complex, dense, and diverse environments (Xiong *et al.*, 2019). Moreover, the ability of DRL to autonomously optimize system performance through interactions with unknown environments (Mei *et al.*, 2021) has captured considerable research interest. Accordingly, DRL was chosen as the main methodology to design the proposed solution.

Hence, this research is motivated by the absence of a dedicated joint power allocation and beam management scheme for eMBB and uRLLC in MU-MISO systems based on DRL. This study aims to address this gap and propose a solution that seamlessly integrates into the self-optimization class of ZTNs. The aim is achieved by introducing the PABSO-DRL scheme, which considers the influence of imperfect CSI and intra-interference while also maintaining slice isolation.

2.4 System Model

The system model of the proposed PABSO-DRL scheme is presented as follows.

We consider a network-slicing scenario in which a slice orchestrator (SO) communicates with a single base station (BS) using a southbound protocol to manage the BS remotely. In this context, the following assumptions are made:

- The BS hosts two slices: eMBB and uRLLC, denoted by S^e and S^u, respectively.
- These slices have already been admitted by the SO. The main role of the SO is to set up and manage the lifecycle of the network slices in the mobile network.

- Inter-slice resource allocation is performed by the SO by assigning a budget of resources (*e.g.,* power) to the BS based on the number of admitted slices, following a hard slicing strategy. Consequently, the total power allocated to the BS is denoted by P_{total}, whereas the total power allocated to the eMBB and uRLLC slices is denoted by P^{e}_{max} and P^{u}_{max}, respectively.

The set of all S^e users is denoted by $m = \{1, 2, \ldots, M\}$ and all S^u users by $k = \{1, 2, \ldots, K\}$, where M and K indicate the total number of eMBB and uRLLC users, respectively. Therefore, the set of all users in the system can be represented as $v = \{m, k\} = \{1, 2, \ldots, V\}$. The total number of users is $V = M + K$. We consider a BS equipped with a uniform linear array (ULA) of N_T ($N_T > 1$) antennas, where all users in the system are equipped with a single antenna to ensure low hardware complexity. We focus on the downlink of the BS, assuming that each user belongs to only one network slice based on the required services, as shown in Fig. 2.1. To avoid inter-slice interference between admitted slices, we considered an orthogonal spectrum-sharing strategy between the eMBB and uRLLC users, allowing them to coexist in the same system, as discussed in (Slalmi *et al.*, 2021b; Ginige *et al.*, 2020b; Tang *et al.*, 2019a,b). This approach ensures perfect isolation between the eMBB and uRLLC slices, and treats each slice as an independent network. The advantage of this isolation method is that it enhances the reliability and QoS for each slice without sacrificing the requirements of one slice for another. Moreover, this approach supports customization (Shirzad, 2023), *i.e.,* each slice can control the allocation of resources to its users. Furthermore, this approach will play a vital role in enhancing the reliability of the uRLLC slice in this study by effectively eliminating interference between the admitted slices, thereby, directly improving the connection reliability.

Assuming that the total bandwidth allocated to the BS is B_{total}, whereas the total bandwidth allocated to eMBB users and uRLLC users is denoted by b^e and b^u, respectively. Thus, the expression for bandwidth sharing can be written as.

$$B_{\text{total}} = b^e + b^u \tag{2.1}$$

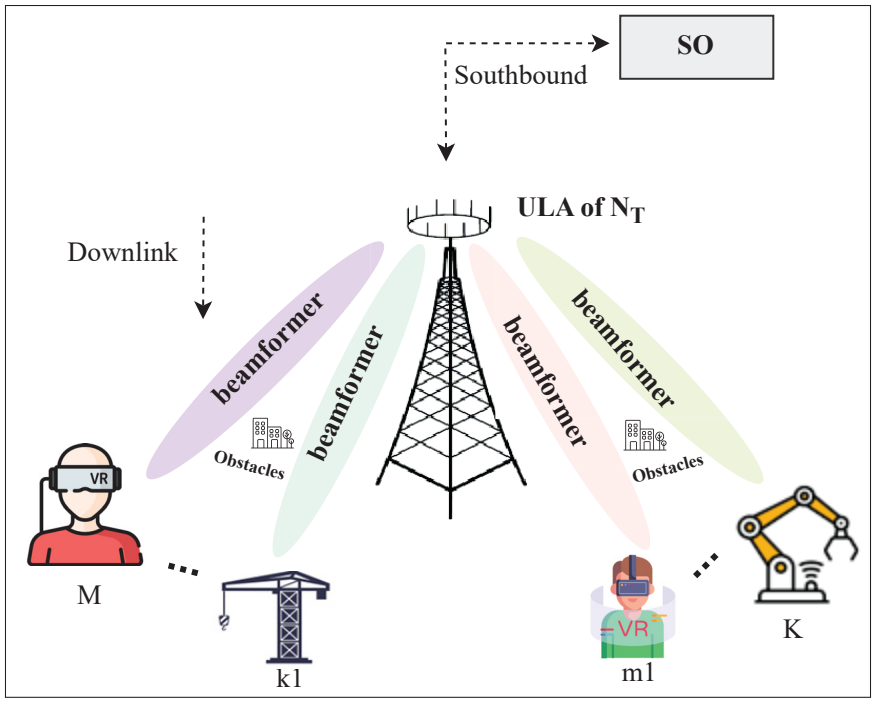

Figure 2.1 System model for PABSO-DRL scheme

Moreover, we consider that OFDMA technology is employed inside both slices, and each eMBB user is allocated $b_m^e = \frac{b^e}{M}$ in bandwidth, as we assume a fair distribution of frequency resources among them. Despite the significant advantages of OFDMA in mitigating interference compared to other multiple access methods, achieving complete interference elimination or isolation can be difficult because some imperfections still exist in real-world scenarios. For instance, imperfect synchronization can result in a carrier frequency offset, leading to subcarrier interference (Zhang & Tellambura, 2009; García & Oberli, 2009). This led us to consider incomplete interference isolation within the OFDMA for intra-slicing in our proposed system.

As mentioned earlier, we consider eMBB and uRLLC slices, to support different types of users. The former requires a high data rate, whereas the latter requires high reliability.

According to (Adamu, López-Benítez & Zhang, 2023), uRLLC reliability is often defined as the probability of successfully transmitting a certain amount of data within a specific interval. This ensures that the data is transmitted with a high probability of success, thereby meeting stringent

latency requirements. However, when the focus of the work is only on reliability, and latency is not explicitly addressed, various definitions of reliability can be considered for the uRLLC slices. Of those definitions, the focus of this work is on the reliability of link connectivity, which is typically expressed in terms of the outage probability.

2.5 Problem Formulation

The proposed scheme focuses specifically on intra-slicing; wherein the major attention is to diverse QoS needs of eMBB and uRLLC users. Specifically, the power allocation and beam are optimized to maximize the data rate of users in the eMBB slice while minimizing the outage probability for the uRLLC slice. The following subsections present mathematical models for the channel, eMBB slice, and uRLLC slice.

2.5.1 Channel Model

The downlink multi-slice MISO channel is modeled as a flat-and-block fading channel (Ge *et al.*, 2020; Zhou, Wang, Umehira & Ji, 2022), which is a realistic and commonly used channel model. Thus, the downlink channel vector between the BS and user v at time slot t, where $v \in \{m, k\}$, is expressed as

$$\mathbf{h}_v(t) = \sqrt{\frac{\beta_v}{L}}\alpha(N_T, \theta_v, \Delta)\mathbf{g}_v;\ v \in \{m, k\} \tag{2.2}$$

where β_v represents large-scale fading that includes path loss and shadowing. We assume that β_v remains the same over several time slots. Moreover, it is considered that there are L paths between BS and the v^{th} user. The matrix $\alpha(N_T, \theta_v, \Delta) \in \mathbb{C}^{N_T \times L}$ represents the N_T antenna array response matrix for the L paths, given by

$$\alpha(N_T, \theta_v, \Delta) = \begin{bmatrix} \alpha_1(N_T, \theta_1) & \dots & \alpha_L(N_T, \theta_L) \end{bmatrix} \in \mathbb{C}^{N_T \times L}, \tag{2.3}$$

where θ_v represents the direction of departure (DoD) of the channel between the BS and user v, and Δ denotes a small angle. In Eq. (2.3), $\alpha_l(N_T, \theta_l) \in \mathbb{C}^{N_T \times 1}$ denotes the array response vector of the l^{th} path, given by

$$\alpha_l(N_T, \theta_l) = \left[1, e^{j2\pi \frac{d}{\lambda} \cos\theta_2} \quad \dots \quad e^{j2\pi \frac{d}{\lambda}(N_T-1)\cos\theta_l}\right]^T, \tag{2.4}$$

where λ denotes the wavelength of the downlink carrier wave, d denotes the distance between antennas, (which is set to $\lambda/2$ in our setup), and θ_l is the DoD of the l^{th} multipath. Here, we assume that the DoDs of all the paths are uniformly distributed. Finally, $\mathbf{g}_v(t) \in \mathbb{C}^{L \times 1}$ denotes a small-scale fading vector from the BS to v. It is assumed that $\mathbf{g}_v$ remains the same within each time slot but varies over time.

In reality, CSI is often subject to imperfections, making it challenging to assume otherwise, particularly at the transmitter's side, where CSI is typically gathered through feedback from the receiver. This imperfection is caused by a variety of sources, including channel estimate error, feedback error/delay, and quantization error (Gharehgoli *et al.*, 2023). To address this issue, imperfect CSI is considered at the BS, modeled as (Cui, Ding & Fan, 2018).

$$\mathbf{h}_v = \underbrace{\widehat{\mathbf{h}}_v}_{\text{estimated channel vector}} + \underbrace{\boldsymbol{\varepsilon}_h}_{\text{imperfection}} \quad ; \quad v \in \{m, k\} \tag{2.5}$$

where, $\boldsymbol{\varepsilon}_h$ follows a Gaussian distribution $\mathcal{CN}(0, \sigma^2_{\varepsilon_h})$, with zero mean and $\sigma^2_{\varepsilon_h}$ is the variance of the estimation error.

2.5.2 eMBB Slice

The received signal of the m^{th} eMBB user in S^e at time t is,

$$y_m(t) = \underbrace{\mathbf{h}_m^H(t)\mathbf{w}_m(t)x_m(t)}_{\text{desired signal}} + \sum_{i=1,i\neq m}^{M} \underbrace{\mathbf{h}_m^H(t)\mathbf{w}_i(t)x_i(t)}_{\text{intra interference}} + \underbrace{\mathbb{Z}_m(t)}_{\text{noise}}, \tag{2.6}$$

where $\mathbf{h}_m \in \mathbb{C}^{N_T\times 1}$ represents the complex channel vector from the BS to the m^{th} eMBB user. The superscript $(\cdot)^H$ denotes Hermitian operator, $\mathbf{w}_m \in \mathbb{C}^{N_T\times 1}$ denotes the beamformer of the BS, x_m is the data symbol for the m^{th} user, with $\mathbb{E}[|x_m|^2] = 1$. The second term of the received signal represents the intra-interference experienced by the m^{th} user from other users within the same slice, in addition to additive white Gaussian noise (AWGN) $\mathbb{Z}_m \in \mathcal{CN}(0, \sigma^2)$ at the m^{th} user.

The instantaneous signal-to-interference-noise-ratio (SINR) of the m^{th} eMBB user in time slot t is given as follows:

$$\gamma_m(t) = \frac{|\mathbf{h}_m^H(t)\mathbf{w}_m(t)|^2}{\underbrace{\sum_{i=1,i\neq m}^{M} |\mathbf{h}_m^H(t)\mathbf{w}_i(t)|^2}_{\text{intra-slice interference}} + \underbrace{\sigma_m^2}_{\text{noise power}}}. \tag{2.7}$$

The achievable downlink data rate for the m^{th} eMBB user in S^e can be computed as:

$$\xi_m(t) = b_m^e \log(1 + \gamma_m(t)). \tag{2.8}$$

To achieve the goal of the proposed scheme in terms of ensuring high data rates for eMBB users, the objective function is formulated to maximize the downlink data rate, represented as follows:

$$\mathcal{F}_1: \quad \underset{\mathbf{W}^e(t)}{\text{maximize}} \quad \sum_{m=1}^{M} \xi_m(t) \tag{2.9}$$

subject to

$$C1: \|\mathbf{w}_m\|^2 \geq 0; \quad \forall m \in \mathcal{S}^e,$$
$$C2: \sum_{m=1}^{M} \|\mathbf{w}_m\|^2 \leq P^e_{\max},$$
$$C3: \xi_m \geq \xi_{min}; \quad \forall m \in \mathcal{S}^e,$$
$$C4: \sum_{m=1}^{M} \|\mathbf{w}_m\|^2 + P^u_{\max} \leq P_{\text{total}},$$

where $\mathbf{W}^e(t) = [\mathbf{w}_{m1}(t), \mathbf{w}_{m2}(t), \ldots, \mathbf{w}_M(t)]$. In Eq. (2.9), constraint C1 ensures that the power allocated to each user is non negative. Constraint C2 guarantees that the power allocated to all users in the given eMBB slice does not exceed the slice budget. Constraint C3 ensures fairness among eMBB users by ensuring that each eMBB user recieves a data rate greater than the minimum data rate (ξ_{min}). C4 ensures that the total power allocated to the eMBB and uRLLC slices does not exceed the total power budget of the BS.

2.5.3 uRLLC Slice

The received signal of the k^{th} uRLLC user in S^u at time t is written as follows:

$$y_k(t) = \underbrace{h_k^H(t)\mathbf{w}_k(t)x_k(t)}_{\text{desired signal}} + \underbrace{\sum_{j=1, j\neq k}^{K} h_k^H(t)\mathbf{w}_j(t)x_j(t)}_{\text{intra interference signal}} + \underbrace{\mathbb{Z}_k(t)}_{\text{noise}}, \tag{2.10}$$

where $\mathbf{h}_k \in \mathbb{C}^{N_T \times 1}$ represents the complex channel vector from the BS to the k^{th} uRLLC user. Here, $\mathbf{w}_k \in \mathbb{C}^{N_T \times 1}$ denotes the beamformer of the BS, x_k is the data symbol for the k^{th} uRLLC user with $\mathbb{E}[|x_k|^2] = 1$. The second term of the received signal represents the intra-interference experienced by the k^{th} user from other users within the same slice, in addition to AWGN $\mathbb{Z}_k \in \mathcal{CN}(0, \sigma^2)$ at the k^{th} user.

The SINR of k^{th} in S^u at time t is expressed as follows:

$$\gamma_k(t) = \frac{\left|\mathbf{h}_k^H(t)\mathbf{w}_k(t)\right|^2}{\underbrace{\sum_{j=1, j\neq k}^{K}\left|\mathbf{h}_k^H(t)\mathbf{w}_j(t)\right|^2}_{\text{intra-slice interference}} + \underbrace{\sigma_k^2}_{\text{noise power}}}. \tag{2.11}$$

As mentioned in Section 2.4, our main focus is on the reliability aspect of uRLLC. Therefore, we consider a pivotal metric for assessing connection robustness, represented by the instantaneous SINR, as described in (Bana, Xu, De Carvalho & Popovski, 2018). This metric is characterized by the outage probability for each communication link. This probability depends on whether the instantaneous SINR falls below a certain threshold (γ_{Th}) (Hunter, Sanayei & Nosratinia, 2006; Xue *et al.*, 2024b). The main reason for using the outage probability as an appropriate evaluation metric for uRLLC is that packet errors typically occur when the downlink SINR requirements are not satisfied (Bana *et al.*, 2018). Hence, this outage probability is mathematically represented as:

$$P_{\text{out}}^{\text{device}}(t) = \Pr(\gamma_k(t) < \gamma_{\text{Th}}), \tag{2.12}$$

where $\gamma_k(t)$ is expressed in Eq. (2.11) and $\gamma_{\text{Th}} = 2^R - 1$.

For Rayleigh fading, $\gamma_k(t)$ follows an exponential distribution. Therefore, the outage probability in the context of Rayleigh fading (Hunter *et al.*, 2006; Adamu *et al.*, 2023) can be expressed as follows:

$$P_{\text{out}}(t) = \Pr(\gamma_k(t) < \gamma_{\text{Th}}) = 1 - \exp\left(-\frac{\gamma_{\text{Th}}}{\Gamma}\right), \tag{2.13}$$

where Γ indicates the average of SINR.

To achieve the goal of the proposed scheme in terms of ensuring high reliability for uRLLC users, the objective function is formulated to minimize the outage probability, expressed as

$$\mathcal{F}_2: \quad \underset{\mathbf{W}^u(t)}{\text{minimize}} \quad \sum_{k=1}^{K} P_{\text{out}} \tag{2.14}$$

subject to

$$C1: \|\mathbf{w}_k\|^2 \geq 0; \quad \forall\, k \in \mathcal{S}^u,$$
$$C2: \sum_{k=1}^{K} \|\mathbf{w}_k\|^2 \leq P^u_{\max},$$
$$C3: \sum_{k=1}^{K} \|\mathbf{w}_k\|^2 + P^e_{\max} \leq P_{\text{total}},$$

where $\mathbf{W}^u(t) = [\mathbf{w}_{k1}(t), \mathbf{w}_{k2}(t), \ldots, \mathbf{w}_K(t)]$. In Eq. (2.14), constraint C1 ensures that the allocated power to each user is non negative value. Constraint C2 guarantees that the power allocated to all users in the uRLLC slice does not exceed the slice budget. C3 ensures that the total power allocated to the uRLLC and eMBB slices does not exceed the total power budget of the BS.

The beamformer vectors are decomposed as $\mathbf{w}_v = \sqrt{p_v}\bar{\mathbf{w}}_v$ for all v users, where $p_v = \|\mathbf{w}_v\|^2$ are the power allocation coefficients and $\bar{\mathbf{w}}_v$ represent the normalized beamforming directions. These variables serve as the optimization variables in $\mathcal{F}_1$ and $\mathcal{F}_2$, where jointly selecting the optimal p_v and $\bar{\mathbf{w}}_v$ ensures the maximization of $\mathcal{F}_1$ and the minimization of $\mathcal{F}_2$.

In this study, we consider beamforming based on a codebook technique, as defined in modern wireless standards (Cheng & Pesavento, 2013). The codebook technique consists of a finite number of predefined beamforming vectors, where each vector directs the beam in a specific direction (Cheng & Pesavento, 2013). Consequently, in this approach, normalized beamforming vectors are selected from a predefined codebook matrix, which remains unchanged throughout the duration of the communication. The detailed design of the adopted codebook technique is elaborated on in Section 2.6.2.

$\mathcal{F}1$ is a non-convex problem, categorized as NP-hard (Ge *et al.*, 2020). In addition to this, $\mathcal{F}2$ also shares this non-convex nature. Given the complexities and requirements of ZTNs in 6G networks, traditional methods face significant challenges in dynamically changing environments

such as NS, owing to their high computational demands, in addition to the other drawbacks mentioned in Section 2.3. To overcome these difficulties and propose a solution that paves the way toward ZT operations in RAN slicing, we utilize a DRL framework to effectively address $\mathcal{F}1$ and $\mathcal{F}2$. In the following section, we present our solution utilizing the DRL framework.

2.6 Proposed Multi-Agent DRL-Based Power Allocation & Beam Optimization Across Multiple Slices

This section introduces the proposed scheme by discussing the motivation behind its design, followed by a comprehensive description of its implementation.

2.6.1 Motivation Behind Design and Methodology

We develop our proposed solution to address $\mathcal{F}_1$ and $\mathcal{F}_2$ for multiple slices within a DRL framework. There are various RL and DRL algorithms; however, in this study, we focus on the DQL algorithm, which is a model-free algorithm known for its effectiveness, even in the absence of knowledge about the dynamics of the environment (Ge *et al.*, 2020). The main advantage of this model-free algorithm is its ability to continuously improve the policy using a trial-and-error strategy (Ge *et al.*, 2020). To deal with the diverse requirements and dynamic nature of RAN slicing, we approach the NS problem by treating each slice as an independent network in our design. Specifically, we consider each admitted slice in the BS as an autonomous agent.

The motivation behind designing an independent agent for each slice stems from the challenge of the increase in size of the state-action space, as discussed in (Zangooei *et al.*, 2023). A single DRL agent may face this challenge if it is used to manage multiple slices simultaneously. In this study without loss of generality, we consider only two slices for simplicity. However, in the context of NS and real deployment, a BS can host many slices. Managing multiple network slices simultaneously is one of the biggest challenges for an agent. This challenge can lead to a slowdown in the decision-making processes. To address this issue and improve the scalability of our solution, we adopt a strategic approach. Instead of allowing the state-action space to expand as more network slices are added, we opt for a solution that involves increasing the number of

agents, with each agent dedicated to managing a specific network slice. This approach not only mitigates the challenge associated with the increase in the size of the state-action space but also accelerates convergence. Additionally, reducing the action space will contribute to enhance the overall system's scalability.

2.6.2 MDP Formulation of Multiple DQL Agents

Generally, RL is composed of an environment and one or more agents. In RL, the agent-environment interaction is modeled as a MDP framework, which comprises four components $(S, A, R, \mathbb{P})$ (Ge *et al.*, 2020). S refers to a set of states, where the $s_t \in S$ represents the state of an agent at time step t. A is a set of actions, where the $a_t \in A$ represents the action of an agent at time step t. R is the reward function, where the $r_t \in R$ indicates immediate reward of the agent at time step t. $\mathbb{P}$ is the state transition probability from s_t to the the next state s' after executing a in s; *i.e.,* $\mathbb{P}(s' \mid s, a)$ (Ge *et al.*, 2020).

We reformulate $\mathcal{F}_1$ and $\mathcal{F}_2$ into a MDP framework by defining the S, A, and R for multi-agent (eMBB agent and uRLLC agent), as follows.

2.6.2.1 Action space

The downlink beamformer is a continuous optimization problem; however, the DQL algorithm supports discrete action spaces. In order to address this issue, we decompose the beamformer into two parts: power and beam direction. Then we discretized the power using a set of available discrete power levels, while we considered utilizing the codebook technique to discretize the beam directions. The setup of the action space for a multiagent is illustrated as follows:

$$\mathbf{w}_v(t) = \sqrt{p_v(t)}\bar{\mathbf{w}}_v(t), \quad v \in \{m, k\}, \tag{2.15}$$

where $p_v(t) = \|\mathbf{w}_v(t)\|^2$ indicates the transmit power of BS to the eMBB and uRLLC users in time slot t, and $\bar{\mathbf{w}}_v(t) = \frac{\mathbf{w}_v(t)}{\|\mathbf{w}_v(t)\|}$ represents the direction of the transmit beam (normalized beamforming). In the eMBB and uRLLC slices, the power budget of the slice is denoted as

$P_{\text{budget}} \in \{P^e_{\max}, P^u_{\max}\}$, which is discretized into N_{L} transmit power levels. These levels are distributed uniformly within the range from 0 to the maximum transmit power $p_{\max}$, as defined in Eq. (2.16). Additionally, we discretize the beam directions by designing a matrix based on the codebook technique denoted as $\mathbf{C}_{\text{book}} = \{\mathbf{c_0}, \mathbf{c_1}, \ldots, \mathbf{c}_{B_{\text{code}}-1}\} \in \mathbb{C}^{N_T \times B_{\text{code}}}$, where B_{code} indicates the size of the codebook and $B_{\text{code}} \geq N_T$ (Ge *et al.*, 2020). Each vector $\mathbf{c}$ in $\mathbf{C}_{\text{book}}$ corresponds to a specific beam pattern (direction) in the range $[0, 2\pi)$ for $\bar{\mathbf{w}}_v(t)$. We follow the codebook design strategy outlined in (Ge *et al.*, 2020), which is mathematically represented in Eq. (2.17). This equation is implemented in Matlab to generate a codebook matrix.

At each time slot, each agent optimizes the transmit power level and beam direction of all users in the system by selecting a power level $p_v(t)$ from P_{budget} and a code $\mathbf{c}(t)$ from the $\mathbf{C}_{\text{book}}$. Hence, the size of the action space is $N_L \times B_{\text{code}}$ and the available actions for the agent are represented as a set of all possible action combinations denoted by $A \in \{\mathcal{S}^e, \mathcal{S}^u\}$, where each action involves $(p, \mathbf{c})$. Therefore, we define a_t as

$$a_t = \{(p, \mathbf{c}),\ p \in \mathcal{P}_{\text{budget}},\ \mathbf{c} \in C_{\text{book}}\}, \tag{2.16}$$

where

$$\mathcal{P}_{\text{budget}} = \{0, \frac{1}{N_{\text{L}} - 1} p_{\max}, \frac{2}{N_{\text{L}} - 1} p_{\max}, \ldots, p_{\max}\}, \text{and}$$
$$\mathbf{C}_{\text{book}} = \{\mathbf{c_0}, \mathbf{c_1}, \ldots, \mathbf{c}_{B_{\text{code}}-1}\}.$$

$\mathbf{C}_{\text{book}}$ is expressed as follows:

$$\mathbf{C}_{book}[n_t, b_c] = \frac{1}{\sqrt{N_T}} \exp\left(j \frac{2\pi}{F} \left\lfloor \frac{n_t \bmod \left(b_c + \frac{B_{\text{code}}}{2}, B_{\text{code}} \right)}{B_{\text{code}}/F} \right\rfloor \right), \tag{2.17}$$

where $(\cdot)$ denotes the modulo operation, $\lfloor \cdot \rfloor$ represents the floor operation, and F indicates the number of distinct phase settings available for each antenna element. In the matrix $\mathbf{C}_{\text{book}}$, each element (n_t, b_c) is a complex number, representing the phase shift for a specific antenna n_t.

More importantly, the accuracy and beam width are both directly affected by the number of antennas in the array. Employing more antennas results in a more focused and accurate beam, which increases the gain.

2.6.2.2 State space

For each slice, a set of states are defined to inform the agent about the current status of its slice, enabling it to make the right decision in terms of jointly optimizing the power allocation and beam direction. Therefore, the states of the eMBB agent in the system at t time step can be defined as:

$$s_t^e = \left\{ \mathbf{p}_m(t-1), \mathbf{I}_c^m(t-1), \boldsymbol{\xi}_m(t-1), \boldsymbol{G}_m(t) \right\}. \tag{2.18}$$

where $\mathbf{p}_m(t-1)$ is the set of the previous transmit power levels, $\mathbf{I}_c^m(t-1)$ is the set of the previous beam direction indexes, $\boldsymbol{\xi}_m(t-1)$ is the set of the previous achievable rates, and $\boldsymbol{G}_m(t)$ is the set of the equivalent channel gains; wherein each $G_m(t) = \left|\mathbf{h}_m^H(t)\bar{\mathbf{w}}_m(t-1)\right|^2$.

Moreover, the states of the uRLLC agent in the system at t time step can be defined as:

$$s_t^u = \left\{ \mathbf{p}_k(t-1), \mathbf{I}_c^k(t-1), \boldsymbol{P}_{out}^k(t-1), \boldsymbol{G}_k(t) \right\}. \tag{2.19}$$

where $\mathbf{p}_k(t-1)$ is the set of the previous transmit power levels, $\mathbf{I}_c^k(t-1)$ is the set of the previous beam direction indexes, $\boldsymbol{P}_{out}^k(t-1)$ is the set of the previous outage probabilities, and $\boldsymbol{G}_k(t)$ is the set of the equivalent channel gains; wherein each $G_k(t) = \left|\mathbf{h}_k^H(t)\bar{\mathbf{w}}_k(t-1)\right|^2$.

2.6.2.3 Reward

After each agent performs an action in its respective NS, the NS provides the agent with a reward or penalty. In our design, two reward functions are tailored to meet the QoS requirements of the corresponding slices. In Eq. (2.20), the reward function for the eMBB agent is formulated. Here,

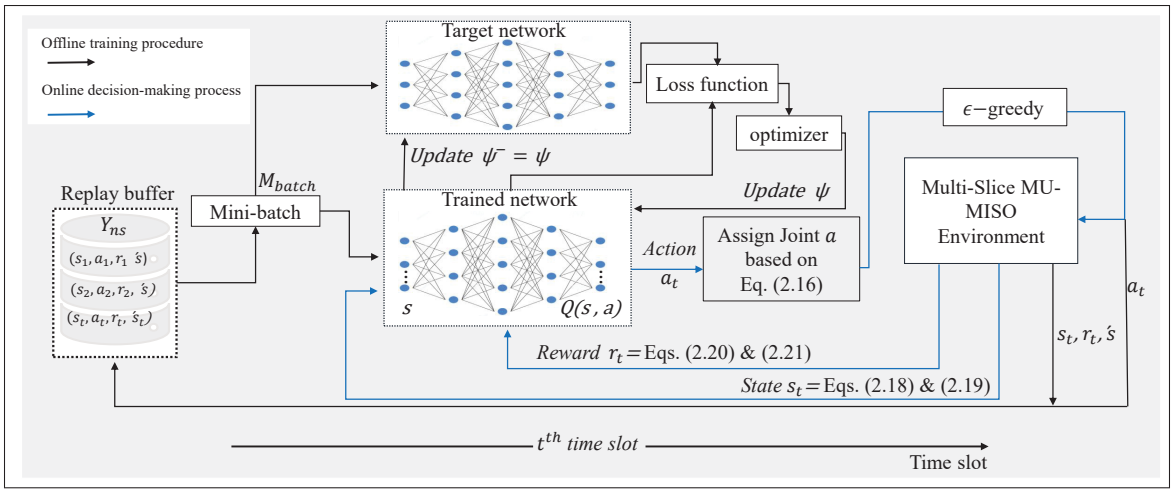

Figure 2.2 Architecture of the proposed PABSO-DRL scheme for multi-slice MISO systems

the agent receives a positive reward for each action resulting in a rate exceeding the minimum rate ξ_{min}; otherwise, it is penalized.

$$r_t^e = \begin{cases} \sum_{m=1}^{M} \xi_m(t), & \text{if } \xi_m(t) \geq \xi_{min} \\ 0, & \text{otherwise.} \end{cases} \tag{2.20}$$

The reward function for the uRLLC agent is designed according to Eq. (2.21), aiming to minimize the outage probability for each uRLLC user at time step t.

$$r_t^u = \sum_{k=1}^{K} 1 - P_{\text{out}}(t). \tag{2.21}$$

2.6.3 Design and Train Multiple DQL Agents for Multi-Slices

To address the multiple MDP models in the previous section, we design and train multiple independent agents based on the DQL theory (Ge *et al.*, 2020). Fig. 2.2 illustrates the design of the proposed PABSO-DRL scheme. In the proposed scheme, the agents operate autonomously

to maintain QoS of their respective slices. Particularly, each agent strives to find a policy (π) that maximizes the expected reward. In DQL algorithm, the action-value function denoted by $Q_\pi(s,a)$ can be utilized to estimates the expected value of the total reward when selecting a specific action a in a given state s under a certain π, which is mathematically represented as (Ahmed & Hossain, 2019).

$$Q_\pi(s,a) = \mathbb{E}\left[R_t \mid s_t = s, a_t = a, \pi\right], \tag{2.22}$$

where $R_t = \sum_{\tau=0}^{\infty} \gamma^\tau r^{(t+\tau+1)}$ represents the expected reward function (Ahmed & Hossain, 2019). Here, $\gamma \in (0,1)$ denotes the discount factor that evaluates the importance of future rewards relative to current rewards and r_t denotes the value of the reward at time slot t. The optimal action-value function $Q^*(s,a) = \max_\pi Q^\pi(s,a)$ represents the maximum achievable action value when considering π for a in a given s. The Bellman equation can be used to express the optimal action-value function as follows (Ahmed & Hossain, 2019).

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q^*(s',a') | s_t = s, a_t = a\right], \tag{2.23}$$

During the learning process, each agent follows an ϵ-greedy strategy, to select a_t, maintaining the trade-off between exploring new actions (exploration) and exploiting ones with known high Q-value (exploitation) (Ge *et al.*, 2020). In other words, the ϵ-greedy strategy helps the agent determine whether to investigate new actions to improve its knowledge or to exploit its existing knowledge by selecting actions expected to yield the highest reward.

More precisely, the ϵ-greedy strategy selects a according to the following rule (Ge *et al.*, 2020):

$$a_t = \begin{cases} \text{random action } (a); \text{ with probability } \epsilon \\ \arg\max_{a \in \mathcal{A}} \{Q(s,a)\}\,; \text{ with probability } 1-\epsilon \end{cases}, \tag{2.24}$$

where, ϵ indicates the exploration rate, which gradually reduces according to $\epsilon(t) = \max \epsilon_{\min}$, $(1 - \lambda_\epsilon)\epsilon(t-1)$ (Ge *et al.*, 2020). At the beginning of the learning process, the ϵ is set with an initial high exploration probability $\epsilon(0)$ to encourage extensive exploration. As the agent gained more experience through interactions with the environment, ϵ gradually decreases according to the rate of decay (λ_ϵ), eventually reaching a minimum exploration probability ($\epsilon_{\min}$). This reduction in ϵ encourages the agent to move towards exploitation over time, thereby depending more on the acquired policy to make decisions.

In the proposed scheme, the architecture of each agent consists of a deep Q network (DQN), which includes two DNNs, with the objective of approximating the Q-value function. The DNNs receive a set of s as input and produce Q-values for all feasible actions as output. The architecture of the DNNs is the same, but their weights are different. The first network has weights (ψ) and is called the trained Q-network, while the second is the target Q-network with weights (ψ^-). The target Q-network is designed to produce the target Q-value, which is then used to formulate the loss function during the training process. The ψ^- of the target network is updated every T steps to match the ψ of the trained network.

Each agent in the system utilizes a replay buffer strategy, to save its experiences that are generated through the interaction process with the environment. The experience of each agent at step t is represented as a tuple (s, a, r, s'), and is stored in its corresponding replay buffer (Y_{ns}). Hence, this leads the agent to enhance its DQN and stabilize learning by using previous experiences, preventing the agent from becoming biased towards recent experiences and avoiding overfitting of the DQN model (Filali *et al.*, 2022). The learning process begins when there are sufficient experiences in the Y_{ns} of each agent. During the training process, each agent randomly selects a mini-batch of experiences (M_{batch}) from its own Y_{ns} to train the trained Q-network. The purpose of the training procedure is to reduce the prediction error between the target Q-value and the estimated Q-value by minimizing the mean squared error (MSE) loss function (F^{loss}) at each training step t. This is defined by (Ge *et al.*, 2020):

Algorithm 2.1 Pseudocode for the training phase

: For each agent in the system, establish a replay memory Y_{ns} with a size limit of X.
: For each agent, set up two DNNs: trained Q-network with ψ and target Q-network with ψ^{-}.
: Set the initial exploration rate ϵ for each agent.
: **for** *each training episode* **do**
  : **for** *each step* **do**
    : **for** *each agent* **do**
      : Get $s \in \{s_t^e, s_t^u\}$ using Eqs. (2.18) and (2.19).
      : Agent selects a_t based on ϵ greedy policy.
      : Execute joint a_t using Eq. (2.16) and receive $r_t \in \{r_t^e, r_t^u\}$ and moves to s'.
      : Save the experience (s_t, a_t, r_t, s') in Y_{ns}.
      : Sample random M_{batch} from Y_{ns}.
      : Calculate target Q-value.
      : Calculate loss between the trained network and the target network according to Eq. (2.25).
      : Update the ψ of trained DQN using gradient descent.
      : Update the target network parameters $\psi^{-} = \psi$ every 100 steps.
      : Repeat until convergence.
    end for
  end for
end for

$$F^{loss}(\psi) = \frac{1}{2} \sum_{\langle s,a,r,s' \rangle \in Y_{ns}} (V^T - Q(s, a; \psi))^2, \tag{2.25}$$

where $V^T = r + \gamma \max_{a'} Q(s', a'; \psi^{-})$ represents the target value, and $Q(s, a; \psi)$ is the estimated Q-value provided by trained network with weights ψ. The MSE loss function is employed by each agent in the proposed scheme, playing a pivotal role in evaluating model performance and optimizing training procedures. Following that, the stochastic gradient descent optimizer is applied to minimize Eq. (2.25) and update the ψ of the trained network, utilizing the back-propagation technique.

The training procedure of the proposed scheme is presented in Algorithm 2.1. To mitigate the drawbacks of centralized training in dynamic environments (Ge *et al.*, 2020), we adopt

a distributed training mode in the proposed scheme. The implementation of the proposed scheme involves two main phases: an offline training procedure and an online decision-making process. The offline training procedure, illustrated by the black line in Fig. 2.2, focuses on training each agent's DNN using the dataset collected during the interaction between the agents and environment. In contrast, the online phase, depicted by the blue line in Fig. 2.2, involves real-time action selection based on the trained models. We describe the process of both phases below.

The initial steps in the training process include: (1) creating the NS environment parameters; (2) initializing the DQN of each agent; (3) generating the end-users, including their locations on the grid and the services they require (*e.g.,* eMBB or uRLLC). Subsequently, the DQN iterates through episodes. At each time step of each episode, the locations of users and channel coefficients (*i.e.,* small-scale fading) and other parameters are updated for each agent. Specifically, during each time step t, each agent observes its current $s \in \{s_t^e, s_t^u\}$ from the environment and chooses a joint a from its set of possible actions combination A. During the initial 200 training steps, the proposed scheme selects actions randomly to accumulate experiences. Following this, it adopts the ϵ-greedy strategy in Eq. (2.24) for the remaining episodes. In line with this, each agent constructs its joint action, computes its reward $r \in \{r_t^e, r_t^u\}$ as defined by Eqs. (2.20) and (2.21), and then transitions to s'. After that, each agent sends its experience (s, a, r, s') to be stored in its own Y_{ns}. Once a sufficient number of experiences have been stored, each agent selects a random M_{batch} of 32 samples from its own Y_{ns}. In the proposed scheme, the Y_{ns} plays a vital role in boosting the sample efficiency of the algorithm by enabling data re-usability; furthermore, it enhances the stability of training. Following that, the loss function is calculated based on Eq. (2.25). After the computation of the loss function, each agent executes an optimizer to modify the parameters of the trained Q-network. Then, at fixed intervals, corresponding to 100 time steps in our implementation, the parameters of the target Q-network are updated by copying the parameters from the trained Q-network, and the whole process continues until the convergence. After completing the training procedure for each DQN, which is conducted offline,

the trained DQNs are utilized for online decision-making. In this phase, each agent selects a based on the current s, using Eq. (2.24).

2.7 Experiment Results

In this section, we evaluate the performance of the proposed PABSO-DRL scheme. We conducted simulations using various traditional baseline algorithms and state-of-the-art under different scenarios. The baselines are designed to jointly support power allocation and beam direction management (PBM) for multiple users of different services. These included equal allocation (EQ-PBM), random allocation (RA-PBM), greedy allocation (GR-PBM), and exhaustive search (ES-PBM), all simulated under imperfect CSI. Additionally, we conducted comprehensive comparisons with the state-of-the-art schemes (Tang *et al.*, 2019b; Slalmi *et al.*, 2021b) under both perfect and imperfect CSI conditions to thoroughly evaluate the robustness and efficiency of our proposed scheme.

2.7.1 Simulation Setup

In our scenario, we consider a single cell where a multi-antenna BS is located at the center, and single-antenna users (eMBB and uRLLC) are randomly distributed within the cell. The cell radius is set to 500m, and the maximum power budget of the BS is set to 1W, as suggested in (Setayesh *et al.*, 2022). Table 2.2 presents the complete set of simulation parameters. For the simulation of the wireless channels between the BS and v, the path loss attenuation, represented by β_v, depends on the distance d_v between the BS and v, (expressed in kms). More precisely, β_v is expressed as $120.9 + 37.6 \log_{10}(d_v)$ dB. Furthermore, the log-normal shadowing is set to 8 dB (Ge *et al.*, 2020). The AWGN power, σ^2, is set to -114 dBm (Filali *et al.*, 2022). Herein, the parameter, $\mathbf{g}_v(t)$ is modeled as a first-order complex Gauss-Markov process, consistent with the Jakes fading model (Ge *et al.*, 2020). The scenario also incorporates four paths with an angular spread of 3° (Ge *et al.*, 2020). Given that the positions of the users in the system are randomly initialized, the nominal DoD of the channels is defined by the azimuth. In our setup, each agent is trained using a DQN consisting of an input layer, an output layer, and two fully

Table 2.2 Main parameters

Parameter	**Value**
b^e of eMBB slice	5 MHz
$\mathcal{P}_{\text{budget}}$ of uRLLC slice	27 dBm
$\mathcal{P}_{\text{budget}}$ of eMBB slice	27 dBm
Minimum rate (ξ_{min})	4 Mbps
SINR threshold (γ_{Th})	10 dB
Total power budget at the BS	1 W
Number of antennas (N_{T})	4, 8, 16, 20
Number of codes (B_{code})	8, 16, 32, 40
Correlation coefficient (ρ)	0.64 (Ge *et al.*, 2020)
Time slot interval	20 ms (Ge *et al.*, 2020)

Table 2.3 Training parameters

Parameter	**Value**
Layers	{Input layer, Hidden Layer, Output}
Number of hidden layer	2
Number of neurons of each hidden layer	64 and 32
Activation function	ReLU
Replay memory size	500
Batch size	32
Initial learning rate α	5e-4
Optimizer	RMSProp
Discount factor γ	0.5
Initial exploration probability $\epsilon(0)$	0.6
Min exploration probability ϵ_{min}	0.01
Decay-rate λ_ϵ	1×10^{-4}
Update ψ with ψ^-	Every 100 time steps

connected hidden layers. The hyperparameters used in the setup of each DQN agent, illustrated in Table 2.3, are consistent with those outlined in (Ge *et al.*, 2020).

The experimental phase of this work was conducted on a computer equipped with a 64-bit operating system, an x64-based processor, and 64 GB of RAM. The software used is the PyTorch framework and Matlab.

2.7.2 Performance Evaluation Against Traditional Benchmark Algorithms

2.7.2.1 Average Rate of eMBB vs. Number of Time Slots

To test the proposed scheme in terms of satisfying the minimum rate constraint for eMBB users, ξ_{min} is set to 4Mbps in the proposed system model as defined in (Setayesh *et al.*, 2022). Further, we considered a ULA with 4 antennas (N_T) and a codebook size (B_{code}) of 8. Therefore, the BS can technically form $\mathcal{G}$ beams, which is less than or equal to the number of N_T in the ULA. In line with this, the testing results of this scenario are presented in Fig. 2.3. We observe that introducing imperfections to the channel has an impact on the performance of the DQL algorithm. This is because the BS, with imperfect knowledge about the communication channel, cannot form a beam precisely directed towards a specific user or allocate the right power, leading to a reduction in the data rate. The eMBB agent is trained for a total of 80,000 time slots under both perfect and imperfect CSI conditions. We observed that the agent improves the performance of the eMBB slice as the number of training time slots increases, demonstrating the effectiveness of the proposed training algorithm. Specifically, when the training reaches approximately 50,000 time slots, the agent gains sufficient valuable experience to begin exploiting better actions, leading to effective learning and stable performance. Notably, the imperfection does not affect the convergence of the DRL algorithm, which achieves a fairly stable state at 50,000 time slots, similar to the convergence observed under perfect channel estimation. Moreover, we observe that, under imperfect CSI, DQL achieves the second best performance among all the considered strategies, after the ES-PBM. Although ES-PBM outperforms the DQL algorithm under imperfect CSI, it is unrealistic and computationally expensive. Additionally, it requires a significant amount of time to find the optimal solution. It is noted that the complexity of the ES-PBM algorithm increases with the size of the action space, rendering it unfeasible in large dimensional environments. This is demonstrated in the radio resource management of RAN slicing, where the number of variables or dimensions in the optimization problem is substantial. This makes it challenging for the ES-PBM algorithm to effectively navigate the action space, and find optimal solutions efficiently without sacrificing complexity and computational cost. In

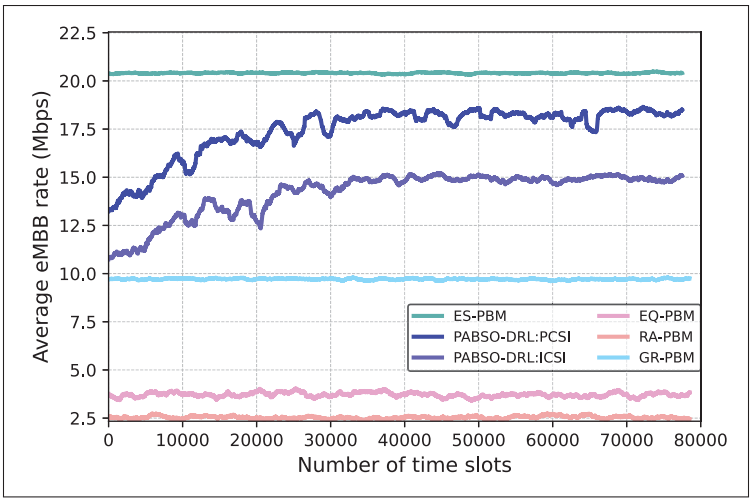

Figure 2.3 Average rate of eMBB
Note: PCSI and ICSI denote perfect and imperfect CSI, respectively

addition, it is not applicable to dynamic environments, such as intra-network slicing, because it lacks the ability to learn from trial and error or adapt to the dynamic conditions of NS.

2.7.2.2 Outage Probability of uRLLC vs. Number of Time Slots

Fig. 2.4 illustrates the performance of the PABSO-DRL scheme by minimizing the outage probability for uRLLC under both perfect and imperfect CSI. In this scenario, we consider the BS to have $N_T = 8$ antennas, and B_{code} is set to 16. In addition, N_{L} is set to 5 levels. In our analysis, we assume that the uRLLC slice serves four users, each requiring a minimum SINR of 10 dB (Mei *et al.*, 2021) for a reliable connection. The proposed scheme targets an enhancement of reliability through the minimization of the outage probability. This constraint relies on the SINR, with a reduction in outage probability leading to an increased SINR, as shown in Fig. 2.5. A higher SINR enhances reliability by facilitating timely packet delivery and reducing retransmission times through lower bit error rates (BER). This leads to enhanced link reliability and decreased chances of connection failures. From Fig. 2.4, it is evident that PABSO-DRL achieves a performance close to the performance of ES-PBM in terms of outages, even in scenarios with imperfect CSI. Furthermore, PABSO-DRL outperforms the other baseline

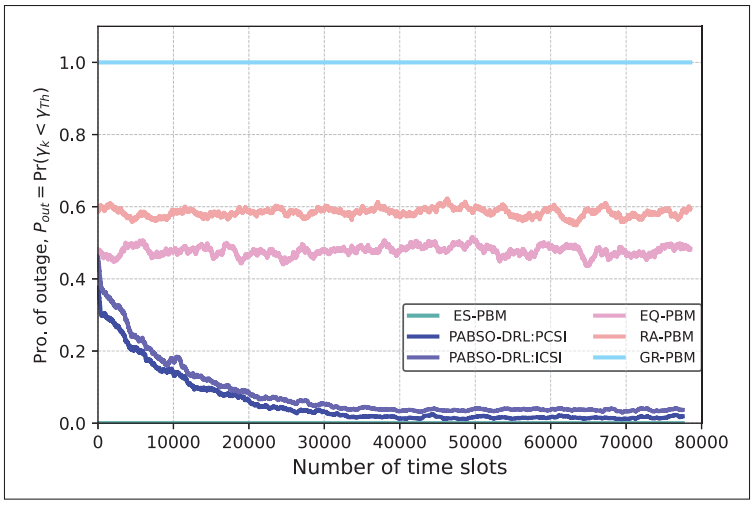

Figure 2.4 Outage probability of uRLLC

algorithms. Simulation results for the ES-PBM are conducted with only 2 users and 4 antennas due to its inability to handle the scenario with 8 antennas and 4 users. This clearly demonstrates that the ES-PBM lacks the capability to deal with large environments such as NS. The training procedure of the uRLLC agent in the proposed scheme is similar to that of the eMBB agent in terms of the total number of time slots and convergence. We observed that the agent improves the connection reliability of the uRLLC slice as the number of training time slots increases, by decreasing the outage probability, and it shows stable performance at approximately 50,000 time slots under both perfect and imperfect CSI.

2.7.3 Performance Evaluation Against State-of-the-art Schemes

We implemented two schemes from the literature, which we named iterative slicing resource allocation scheme 1 (ISRA-S1) (Tang *et al.*, 2019b) and scheme 2 (ISRA-S2) (Slalmi *et al.*, 2021b). ISRA-S1 is based on SCA and SDR, and ISRA-S2 is based on SCP method. Both SCA and SCP are local optimization methods. ISRA-S1 and ISRA-S2 are adapted to satisfy our system model requirements in Python and tested under perfect and imperfect CSI. The ISRA-S1 manages two levels of resource allocation in RAN: inter-slicing and intra-slicing. Since our focus is on intra-slicing, we used the intra-slicing aspect of the ISRA-S1 scheme. The ISRA-S1 and

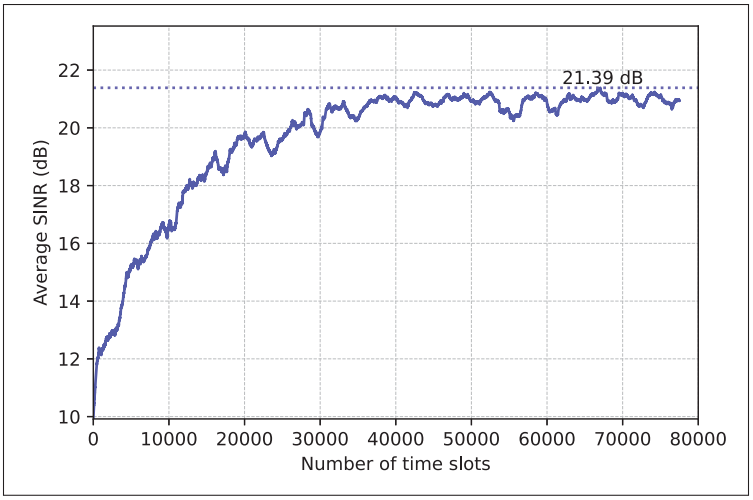

Figure 2.5 SINR vs. number of time slots under ICSI

ISRA-S2 are selected as a benchmark due to several similarities: joint optimization for power and beamforming, consideration of similar services (eMBB and uRLLC), support for orthogonal slicing among slices in MISO systems, and some similarities in optimization parameters. The adopted beamforming technique is not explicitly mentioned in (Tang *et al.*, 2019b) and (Slalmi *et al.*, 2021b); therefore, we assume and implement the codebook technique in ISRA-S1 and ISRA-S2 to facilitate the comparison process. Figs. 2.6 and 2.7 illustrate the performance of PABSO-DRL compared to ISRA-S1 and ISRA-S2 in eMBB and uRLLC services under perfect and imperfect CSI, respectively. From Figs. 2.6 and 2.7, we observe that the PABSO-DRL consistently achieves the best performance in terms of maximizing the data rate for eMBB users and minimizing the outage probability for uRLLC users, under both perfect and imperfect CSI. However, while the ISRA-S1 scheme performs well under perfect CSI for both services, its performance degrades under imperfect CSI. Based on the obtained results, it is evident that each agent in the PABSO-DRL scheme works to find a better power level and beam direction for each user in the system by learning and adapting over time, capable of approaching near optimal even in the presence of the imperfection, while satisfying the QoS constraints for the heterogeneous services. In contrast, the ISRA-S1 and ISRA-S2 schemes struggle to achieve a good QoS under imperfect conditions. This is due to the nature of the iterative algorithms, which require access to real-time and global CSI (Ge *et al.*, 2023); thereby, their performance is highly impacted by

the uncertainty of the channel. These results demonstrate the adaptability of the latest DRL techniques and their advantages over traditional iterative algorithms in RAN slicing, as noted in the literature (Abbas *et al.*, 2024; Xiong *et al.*, 2019; Mei *et al.*, 2021).

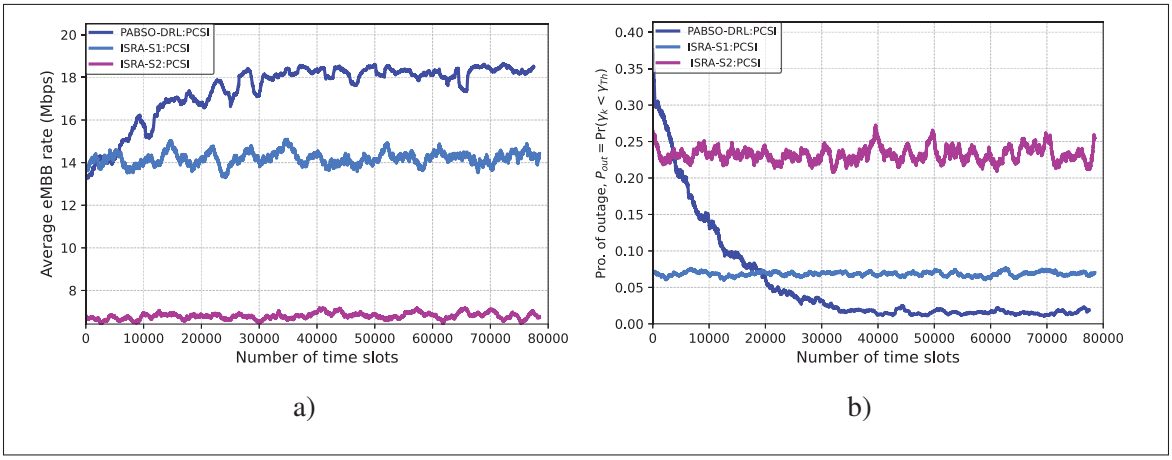

Figure 2.6 Comparison with ISRA-S1 and ISRA-S2 schemes under PCSI: (a) Average rate of eMBB and (b) Outage probability of uRLLC

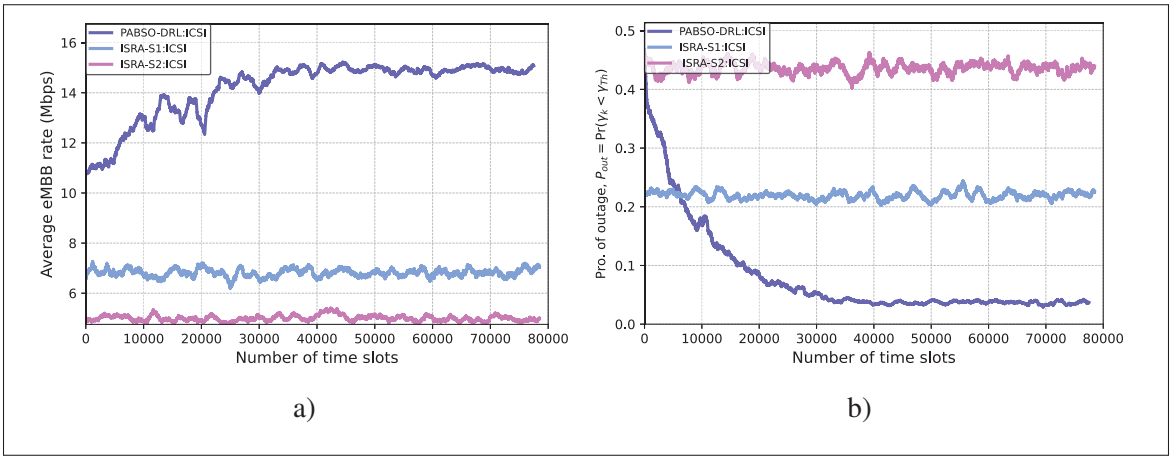

Figure 2.7 Comparison with ISRA-S1 and ISRA-S2 schemes under ICSI: (a) Average rate of eMBB and (b) Outage probability of uRLLC

2.7.4 Impact of Codebook Size on Proposed Scheme

Fig. 2.8 sheds light on the impact of B_{code} on the performance of the PABSO-DRL scheme. We conducted simulations with three different scenarios (A, B, C), each of which has three different

Table 2.4 Simulation parameters of scenarios A, B, and C

Scenario	N_{T}	B_{code}	N_{L}	Number of users per slice
A	4	4, 8, 16	5	2
B	8	8, 16, 32	5	4
C	12	12, 24, 48	5	6

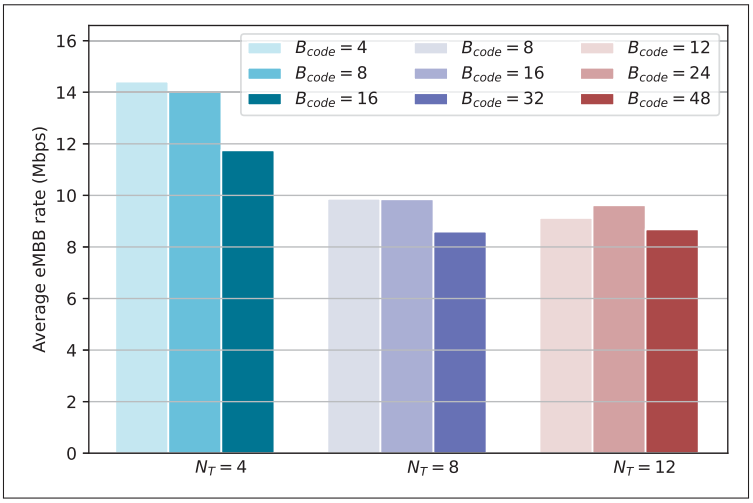

Figure 2.8 Average rate for eMBB slice with different codebook sizes under ICSI

B_{code} sizes. For simulation and investigation purposes, the testing focuses only on the eMBB slice. The simulation parameters are presented in Table 2.4. As depicted in Fig. 2.8, when B_{code} is set to 4 in scenario A, which is equal to N_T, each antenna has four possible directions, resulting in good performance. Similarly, maintaining four antennas while setting B_{code} to 8 also achieves good performance. However, when B_{code} is increased to 16, which is four times greater than N_T, the performance of the proposed scheme decreases. Similar results for scenarios B and C are shown in Fig. 2.8. This emphasizes the importance of selecting an appropriate B_{code} relative to the available N_T in the ULA. Accordingly, optimal results are achieved when B_{code} is equal to or double N_T, ensuring sufficient beam directions for each antenna. Beyond this, excessive complexity may arise. This is evident, from the cases where B_{code} = (16, 32, 48), that the DQN

faces challenges in determining the optimal strategy. Hence, it's vital to set the appropriate B_{code} for N_T in multi-antenna systems to achieve optimal performance while avoiding unnecessary complexity.

2.7.5 Scalability Assessment of PABSO-DRL Scheme

In our proposed study, we select N_T as a metric to assess the flexibility and scalability of the proposed PABSO-DRL scheme. This choice is driven by the fact that increasing N_T increases the size of the ULA and action space. Furthermore, it facilitates an increase in the number of beams that can serve more users. To accomplish this, we examine various numbers of antennas, outline three scenarios to evaluate the proposed scheme, and compare its performance with three traditional baseline algorithms, all of which operate under imperfect CSI. It is important to highlight that the ES-PBM struggled to handle these scenarios because of its inability to effectively search high-dimensional environments, as well as its time-consuming nature. Fig. 2.9 illustrates the aforementioned scenarios with the detailed parameters provided in Table 2.5.

It is evident from the simulation results that the proposed PABSO-DRL scheme is more adaptable to varying antenna sizes than other strategies. Furthermore, Fig. 2.9 illustrates the impact of the action space size on the overall eMBB average rate. In the first scenario, with an action space combination size of 40,960,000, the achieved average rate is better than that of the second and third scenarios, which have the action space combination sizes of 655,360,000 and 1,600,000,000, respectively. This indicates that as the NS size increases (*i.e.,* the number of users and antennas and optimizing parameters, such as power and beam), the performance of the proposed PABSO-DRL gradually decreases. This is because the state and action spaces also increase with the NS size, requiring the DQN to explore more space to find the optimal action policy. Consequently, more exploration is required for larger state-action spaces (Ahmed & Hossain, 2019). However, the degradation in the performance of our proposed scheme is insignificant because of the adopted distributed learning approach, which allows independent agents to manage one slice at a time, making the action space more manageable. Despite this reduction in performance, our proposed scheme continues to outperform other traditional approaches.

Table 2.5 Simulation parameters for the three simulation scenarios

Scenario	N_{T}	B_{code}	N_{L}	Minimum number of M	Size of action space (N_{L} * B_{code})
1^{st}	8	16	5	4	80^4
2^{nd}	16	32	5	4	160^4
3^{th}	20	40	5	4	200^4

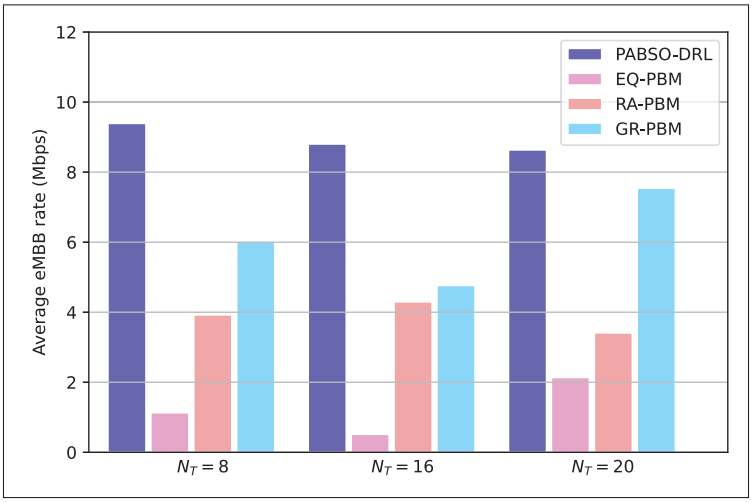

Figure 2.9 Average eMBB rate under ICSI vs. number of antennas for different management schemes

Fig. 2.10 shows the performance of the PABSO-DRL scheme compared to the ISRA-S1 and ISRA-S2 schemes in terms of the average data rate under imperfect CSI with ULA of varying sizes. The PABSO-DRL scheme consistently achieves the best performance, maintaining consistent average eMBB rates of approximately 8-9 Mbps with 8, 16, and 20 antennas. Although both iterative optimization approaches managed to handle different ULA sizes, these methods are not as efficient as DRL. Since the ISRA-S1 and ISRA-S2 schemes are simulated under conditions similar to those of the PABSO-DRL scheme, the observed performance differences can be attributed to the nature of the schemes' design methodologies. DRL algorithms used in the PABSO-DRL scheme enable it to dynamically adapt to changing environmental conditions and manage power allocation and beam direction based on learned policies, effectively handling the

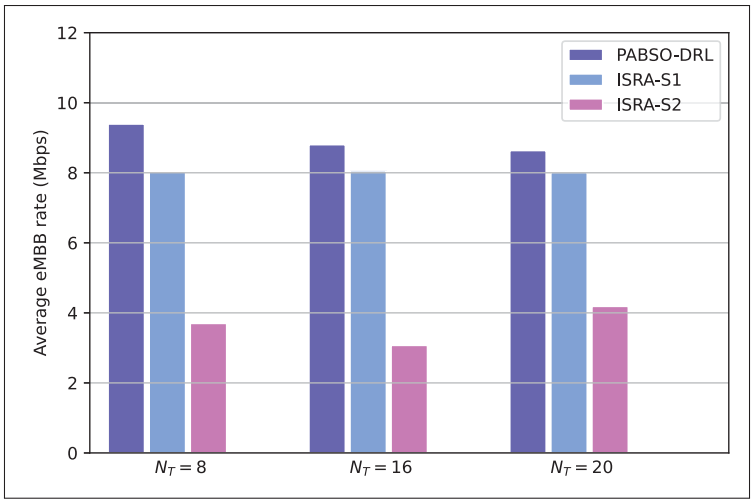

Figure 2.10 Comparison with state-of-the-art schemes

complexity arising from the increasing size of the ULA. Conversely, ISRA-S1 employs SCA, whereby the initial non-convex problem is approximated using a sequence of convex problems (Tang *et al.*, 2019b). According to the results, SCA seems not as efficient as DRL in capturing all the complexities of the RAN slicing environment. In contrast to DRL, ISRA-S2 is based on the SCP method, which concentrates on solving convex approximations iteratively; however, it has limitations when it comes to addressing the dynamicity and complexity of the RAN slicing environment. Overall, PABSO-DRL consistently outperforms state-of-the-art schemes and traditional baseline approaches across all tested ULA sizes. This shows the potential of the PABSO-DRL scheme as a promising scheme for multi-slice multi-antenna systems. Despite the ISRA-S1 scheme achieving decent performance with varying sizes of ULAs, we believe that iterative algorithms are generally not the optimal choice to enable ZT operations in RAN slicing, which requires high adaptability, dynamism and scalability. This is due to various limitations inherent in their iterative nature, as discussed in Section 2.8, rather than just their performance level.

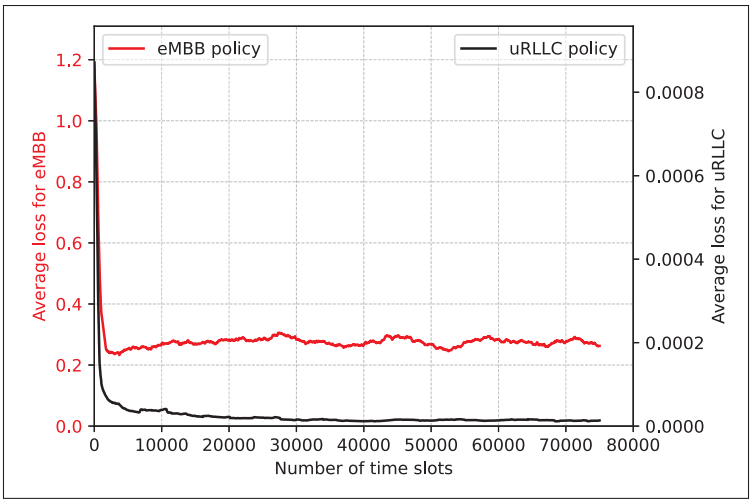

Figure 2.11 Training loss for PABSO-DRL scheme

2.7.6 Loss Functions for Proposed Multi-Agents

Fig. 2.11 depicts the evolution of the loss function throughout the training of the eMBB and uRLLC policies. Initially, as agents explore various actions randomly, the loss gradually decreases. With continued training, each agent enhances its decision-making based on the Q-value function, leading to improved performance reliability. Eventually, the loss stabilizes at its minimum value, signifying a precise estimate of the Q-value.

2.7.7 Effect of Hyperparameter of DRL on Proposed Scheme

Our proposed scheme is based on DRL algorithm; the performance of this kind of algorithm is impacted by several hyperparameters during the training process. One of the most crucial hyperparameters is the learning rate, which regulates how quickly the DQN learns from data. The effectiveness and efficiency of the training are directly affected by the learning rate, making it challenging to adjust. A small learning rate may lead to longer training times, but unstable conditions can arise if it is set too high (Ahmed & Hossain, 2019). To determine the ideal learning rate, we examined the performance of the proposed algorithm across varying learning rates (α= 0.0005, 0.005, 0.05, 0.5). Fig. 2.12 illustrates the average eMBB rate of the proposed scheme under different α. We observe that α= 0.0005 demonstrates more stability, whereas

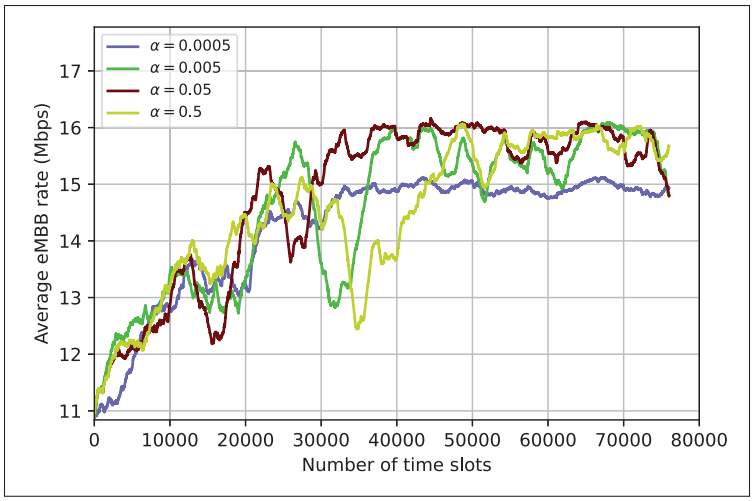

Figure 2.12 Convergence of PABSO-DRL scheme with various learning rates

the three others α slightly improve the average rate, but impacts the convergence stability. Therefore, α should be carefully selected to ensure a good trade-off between the performance and convergence stability.

Table 2.6 Comparative analysis of PABSO-DRL scheme with the existing schemes based on MISO

REF	Optimization Parameters		Channel		Slice Type		Intra Interference		Optimization Problem (Performance evaluation)	Method	Scalability of ULA
	Power	Beam	Perfect	Imperfect	eMBB	uRLLC	eMBB	uRLLC			
(Ghanem *et al.*, 2020)	**Yes**	No	Yes	**Yes**	No	Yes	No	**Yes**	Maximize the weighted sum throughput	Iterative algorithm	8
(Elsayed & Erol-Kantarci, 2020)	No	**Yes**	Yes	No	Yes	Yes	**Yes**	**Yes**	Maximize sum rate for eMBB & Minimize latency for uRLLC	Online clustering algorithm	Not declared
(Tang *et al.*, 2019a)	**Yes**	No	Yes	No	Yes	Yes	No	No	Minimize the total power consumption	Iterative algorithm	2
(Tang *et al.*, 2019b)	**Yes**	**Yes**	Yes	No	Yes	Yes	No	No	Maximize operator's revenue	Iterative algorithm	2
(Ginige *et al.*, 2020b)	**Yes**	**Yes**	Yes	No	Yes	Yes	**Yes**	No	Maximizing the number of admitted eMBB users	Iterative algorithm	4
(Slalmi *et al.*, 2021b)	**Yes**	**Yes**	Yes	No	Yes	Yes	**Yes**	No	Maximizing the number of admitted eMBB users	Iterative algorithm	Not declared
Ours	**Yes**	**Yes**	Yes	**Yes**	Yes	Yes	**Yes**	**Yes**	Maximize rate for eMBB & Minimize outage for uRLLC	**DRL algorithms**	**20+**

2.8 Discussion

After presenting and analyzing our results, this section discusses and connects our findings with existing research. The obtained results confirm that our scheme effectively determines the optimal power levels and beam directions for each user in the system. This ensures QoS for eMBB and uRLLC users even when the channel undergoes fluctuations and imperfections. Based on the obtained results, the proposed PABSO-DRL scheme proved to be effective in learning

two different policies with high accuracy, as shown in Fig. 2.11. These policies were designed to guarantee the long-term QoS requirements of each slice within the system. Furthermore, our testing sheds light on the efficiency of the proposed scheme-based multi-agent. Employing dedicated agents to handle each slice in the system not only makes the action space manageable for each agent but also provides the system with the adaptability to add more slices in the future. In addition, the proposed scheme is capable of handling large ULAs, which are expected to become increasingly prevalent in future scenarios of multi-antenna systems. This scheme was tested with 20 antennas and showed potential to handle more.

Table 2.6 provides a comparative summary between our work and state-of-the-art schemes (Ghanem *et al.*, 2020; Elsayed & Erol-Kantarci, 2020; Tang *et al.*, 2019a,b; Ginige *et al.*, 2020b; Slalmi *et al.*, 2021b) in terms of several key design functionalities of each scheme. These functionalities include multi-resource optimization capability, channel condition consideration, heterogeneous service support, intra-interference management, performance evaluation metrics, and scalability. For instance, the scheme proposed in (Ghanem *et al.*, 2020) focuses on optimizing power to maximize system throughput specifically for uRLLC users, considering both perfect and imperfect CSI. However, it does not support eMBB service. In contrast, the scheme in (Elsayed & Erol-Kantarci, 2020) supports heterogeneous services and considers beam optimization, but it does not manage power allocation and assumes perfect CSI, which may impact the system's performance significantly in real-world deployment scenarios. Additionally, the scheme proposed in (Tang *et al.*, 2019a) optimizes power allocation for heterogeneous services under perfect CSI but does not consider beamforming optimization. Moreover, the schemes proposed in (Tang *et al.*, 2019b; Ginige *et al.*, 2020b; Slalmi *et al.*, 2021b) support the optimization of both the power and beam direction for multi-service scenarios, but perfect CSI is assumed. Furthermore, in terms of the consideration of intra-interference, which can be considered a confounding parameter for the performance of the schemes, we can notice that only our scheme and (Elsayed & Erol-Kantarci, 2020) account for the impact of intra-interference in both services, while other schemes consider it only in one service. In addition, all existing

schemes listed in Table 2.6 utilize a maximum of 8 antennas. There is no evidence that these schemes will continue to perform well if the number of antennas in the ULA is increased.

Notably, most existing schemes (Ghanem *et al.*, 2020; Elsayed & Erol-Kantarci, 2020; Tang *et al.*, 2019a,b; Ginige *et al.*, 2020b; Slalmi *et al.*, 2021b) have been designed based on iterative optimization algorithms to approach near-optimal solutions. However, NS environments are characterized by their dynamic nature, frequent channel fluctuations, significant user mobility, and diverse QoS. In such a dynamic setting, the power allocation and beam direction must be rapidly adjusted to accommodate the instantaneous needs of each slice, particularly as we target ZT operations. Unfortunately, these iterative methods lack the capacity to adapt to dynamic conditions. There is a lack of ability to learn from past training experiences to make informed decisions in real time, impose heavy computational demands, and rely heavily on timely access to CSI. This means that if there are changes in channel conditions, it is imperative to quickly collect updated channel data and re-execute the algorithms (Ge *et al.*, 2023). The question here is how long will re-execution from scratch take? Whereas we are looking for ultra-low latency and ultra-high reliability for future applications. Consequently, we believe that these algorithms are impractical for realizing the ZT operations in RAN slicing.

From the above analysis, we observe that our study fills a significant gap in the current literature on RAN slicing. This novel approach addresses the lack of existing research by utilizing DRL to simultaneously optimize beam and power allocation for multiple users with different QoS requirements in a multi-slice MISO system, under both perfect and imperfect CSI conditions. Hence, the proposed scheme based on DRL can be considered the best approach to lay the groundwork for ZT operation in RAN slicing.

2.9 Conclusion

In this work, we investigated the joint power allocation and beam optimization problem in the eMBB and uRLLC services, in a MU-MISO system. We presented the PABSO-DRL scheme, which utilizes a multi-DQL agent to dynamically serve users with varying QoS requirements.

The proposed scheme is designed to enable each agent to adapt its policy to the dynamic environmental variations while considering channel imperfections. Specifically, it aims to dynamically optimize the power allocation and beam directivity for users across multiple slices, thereby ensuring high data rates for eMBB slices and high reliability for uRLLC slices. The obtained results reflect that the proposed DRL algorithm can learn necessary policies for each agent and gradually select optimal strategies to maximize long-term data rates while minimizing outages. Consequently, numerical simulation results also validate the superiority of our proposed scheme over existing benchmarks, demonstrating its scalability for large-scale antenna array systems.

Future work in this research involves implementing non-orthogonal slicing and designing intelligent inter-slice management to complement intra-slice approaches and adjust total budgets on a large time scale.

Acknowledgement

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), with application number RGPIN-2021-04013.

CHAPTER 3

A SECURE MULTI-RADIO RESOURCE SCHEME USING COOPERATIVE DRL AGENTS FOR HETEROGENEOUS INTER-RAN SLICING UNDER HARDWARE IMPAIRMENTS

Ohood Sabr[1], Kuljeet Kaur[1,2], and Georges Kaddoum[1]

[1] Department of Electrical Engineering, École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, QC H3C 1K3, Canada
[2] Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, 140401, Punjab, India

Paper published in *IEEE Internet of Things Journal*, July 2025

3.1 Abstract

Network slicing (NS) is an innovative technology that shapes the architecture of sixth-generation (6G) networks, allowing each slice to meet specific quality of service (QoS) requirements. Managing radio resources across dense heterogeneous slices with diverse demands presents significant challenges, especially under the zero-touch network (ZTN) paradigm. To address this concern, in the present study, we propose a secure self-optimizing (SO) scheme to manage multiple radio resources (power and bandwidth) across heterogeneous slices on the inter-slice level in open-RAN (O-RAN). The main goal of the proposed scheme is to maximize spectral efficiency while ensuring a service-level agreement (SLA) for each slice. The problem is formulated as a partially observed Markov decision process (POMDP) and then solved using a cooperative multi-actor critic (CoMA2C), named SO-CoMA2C. We further enhance the learning process for each agent using long short-term memory (LSTM) networks. Also, we integrate the advanced encryption standard (AES) into the DRL framework to secure communication between the NS environment and agents to ensure secure resource allocation. The proposed approach considers both ideal and non-ideal hardware impairments, in addition to the fluctuating traffic loads, thereby enhancing the practical relevance and robustness of the solution. Extensive simulations under various conditions demonstrate the superiority of SO-CoMA2C compared

to state-of-the-art benchmarks. Importantly, the integration of AES introduces only minimal overhead, resulting in a 0.13% increase in training time and a 2.3% increase in memory usage.

The detailed abbreviations and definitions used in the study are listed in Table 3.1.

3.2 Introduction

Owing to the creation of isolated logical networks on a shared physical infrastructure, network slicing (NS) has revolutionized the architecture of mobile networks (Habibi *et al.*, 2023). NS made its debut in fifth-generation (5G) communication systems, where the conceptual framework is defined by the following three key logical network types: (i) enhanced mobile broadband (eMBB), which offers subscribers high data rates; (ii) ultra-reliable and low-latency communication (uRLLC), designed to meet industrial demands of applications such as telemedicine and automated driving; and (iii) massive machine-type communications (mMTC), which support a vast number of Internet-of-things (IoT) devices that do not require large data payloads (Shao *et al.*, 2021). At present, the development of NS continues within the sixth generation (6G), adopting a wider range of applications (Habibi *et al.*, 2023). Thus, massive and highly diverse NS has become an essential aspect of 6G networks (Chergui *et al.*, 2021). However, due to the high volume of data traffic, which is expected to experience a sharp growth in the future (Shao *et al.*, 2021), these heterogeneous services place tremendous pressure on current mobile networks. Furthermore, 6G networks need to handle massive slices covering multiple technological domains, including radio access networks (RAN), edge computing, clouds, and cores (Chergui *et al.*, 2021). This presents significant challenges for network orchestration and management, particularly in terms of sustainability and scalability (Chergui *et al.*, 2021). To address these challenges, ETSI proposed the concept of zero-touch network (ZTN) framework, designed as a next-generation management system aimed at fully automating all operational duties and functions. This is achieved through artificial intelligence (AI) and, more specifically, deep reinforcement learning (DRL) algorithms, which play a crucial role in enabling self-managing capabilities, and thus minimize operating expenses and reduce the potential for human error (Benzaid & Taleb, 2020). Managing radio resources in the RAN

Table 3.1 List of acronyms used in the study

Acronym	Definition
5G	Fifth-generation
6G	Sixth generation
A2C	Advantage actor–critic
AES	Advanced encryption standard
AI	Artificial intelligence
BS	Base station
CBC	Cipher block chaining
DQN	Deep Q-network
D3QN	Dueling double DQN
DDPG	Deep deterministic policy gradient
DNNs	Deep neural networks
DRL	Deep reinforcement learning
eMBB	Enhanced mobile broadband
ECB	Electronic codebook
EXP3	Exponential-weight algorithm for exploration and exploitation
FCFS	First-come-first-serve policy
GAN-DDQNs	Generative adversarial networks and deep distributional Q-networks
GCM	Galois counter mode
HW	Hardware
HWIs	Hardware impairments
IoT	Internet of Things
LSTM	Long short-term memory
MADRL	Multi-agent DRL
MARL	Multi-agent reinforcement learning
MDP	Markov decision process
MIMO	Massive multiple-input multiple-output
MISO	Multiple-input single-output
MMFPC	Max-min fairness power control
mMTC	Massive machine-type communications
NS	Network slicing
OFDMA	Orthogonal frequency-division multiple access
O-RAN	Open-RAN
POMDP	Partially observable Markov decision process
PRBs	Physical resource blocks
QoS	Quality of service
RAN	Radio access networks
RBs	Resource blocks
RNNs	Recurrent neural networks
RR	Round-robin
RRM	Radio resource management
SDN	Software-defined networking
SDNR	Instantaneous signal-to-distortion and noise ratio
SISO	Single-input, single-output
SLA	Service-level agreement
SO	Self-optimizing
SSR	Satisfaction of the SLA ratio
TD	Temporal difference
TL	Transfer learning
ULA	Uniform linear arrays
uRLLC	Ultra-reliable and low-latency communication
VoNR	Voice over new radio
ZT	Zero-touch
ZTN	Zero-touch network

domain is particularly challenging as compared to other slicing domains, which is due to the to dynamic nature and the heterogeneous demands of applications/services (Azimi *et al.*, 2022b) of the RAN domain. In NS, each logical NS is allocated a set quantity of resources that are then distributed to its subscribers according to connection requirements (Shao *et al.*, 2021). To effectively allocate limited resources while meeting a variety of needs, resource management in NS should be performed at two levels: within a slice (intra-slicing) and among slices (inter-slicing). To date, much of the literature has focused on radio resource management (RRM) on the intra-slicing level. However, few studies have explored RRM on the inter-slicing level, highlighting the need for further research.

3.2.1 Related Works and Motivation

As the primary focus of the present study is inter-RAN slicing management, in what follows, we briefly review previous literature on this topic, and discusses open challenges that motivate this study.

Most previous research on inter-RAN slicing focused on bandwidth allocation in single-input, single-output (SISO) systems. For instance, in (Li, Wang, Zhao, Guo & Zhang, 2020a), the authors proposed an inter-slicing algorithm to manage the bandwidth across various slices based on the traffic load of each slice. The proposed algorithm aimed to maximize the utility function by leveraging the advantages of the long short-term memory (LSTM)-advantage actor–critic (A2C) algorithm.

Similarly, in (Hua, Li, Zhao, Chen & Zhang, 2020), the authors introduced an inter-slicing allocation algorithm to manage the bandwidth among heterogeneous slices to maximize system utility. The proposed solution was designed using a combination of generative adversarial networks and deep distributional Q-networks (GAN-DDQNs). In another relevant study (Nagib, Abou-Zeid & Hassanein, 2024), an inter-slicing algorithm was explicitly proposed for bandwidth management among heterogeneous slices in an Open-RAN (O-RAN) architecture, with the goal of minimizing system latency. This solution was designed using DRL and transfer learning

(TL). Furthermore, in (Wang, Bai & Xie, 2024a), the authors introduced an algorithm that used the dueling double DQN (D3QN) to efficiently manage physical resource blocks (PRBs) across slices, to reduce PRB utilization.

In another relevant study (Filali *et al.*, 2022), a multi-level RAN management scheme to manage resource blocks (RBs) was proposed. The first level, represented by the software-defined networking (SDN) controller, allocated RBs to multiple gNodeBs using an exponential-weight algorithm for exploration and exploitation (EXP3). On the second level, the gNodeB allocation level, RBs were distributed among the users of eMBB and uRLLC based on a DRL algorithm to maximize the total sum rate. However, it remained unclear how the EXP3 algorithm could adjust the RBs at the SDN controller for multiple gNodeBs by only knowing the previous allocation and set of gNodeBs in the system as a state. This raised the question of how the controller knows the exact needs of each gNodeB. In a related effort, (Azimi *et al.*, 2022a), the authors proposed an inter-slicing scheme based on LSTM to estimate the required resources (power and bandwidth) for each slice. The goal of the proposed scheme was to maximize energy efficiency. Similarly, in (Yan, Ng, Ke & Lam, 2023), the authors proposed inter-slicing for massive multiple-input multiple-output (MIMO) systems to manage the bandwidth among various slices based on A2C. Finally, in (Zhang, Xu, Zhang & Jiang, 2022b), a management scheme was proposed for jointly managing bandwidth and base station (BS)-user association for eMBB and uRLLC on the inter-slicing level to maximize spectrum reuse and ensure service-level agreement (SLA) for eMBB and uRLLC based on the A2C algorithm and deep deterministic policy gradient (DDPG) algorithm.

Despite the contributions of the aforementioned studies, several challenges remain unaddressed, as discussed in the following analysis. In NS, to ensure the efficient use of limited resources based on the DRL algorithm, it is essential to allocate resources among heterogeneous slices according to the traffic demand of each slice (Li *et al.*, 2020a; Hua *et al.*, 2020; Nagib *et al.*, 2024; Yan *et al.*, 2023; Wang *et al.*, 2024a; Shao *et al.*, 2021; Chang *et al.*, 2023; Sabr, Kaur & Kaddoum, 2025b). However, allocating radio resources to each NS based on fluctuating traffic demands is a more desirable and efficient approach; it introduces substantial security challenges. An

example of this approach was introduced in several previous studies (Li *et al.*, 2020a; Hua *et al.*, 2020; Nagib *et al.*, 2024; Yan *et al.*, 2023; Chang *et al.*, 2023; Sabr *et al.*, 2025b) where the traffic loads were treated as states in DRL algorithms. In fact, the state can significantly influence accuracy of the decisions made by the DRL agent, as the agent performs actions based on the current state of the RAN slicing environment to optimize the resources. State information provides the agent with the latest status of the NS, thus allowing the optimization of resources to be updated accordingly. If an agent receives incorrect information or an inaccurate state, it may make misguided decisions, negatively affecting the entire system. Such incorrect states can result from attacks designed to compromise the system or manipulate the state, causing the agent to make poor decisions, such as allocating excessive resources to a slice beyond its actual needs. These vulnerabilities frequently stem from malicious behavior and are common in real-world applications. Despite extensive research on DRL, NS, and privacy-preserving machine learning, to date, no study has attempted to integrate these two approaches within the realm of RAN slicing.

Furthermore, previous studies (*e.g.,* (Li *et al.*, 2020a; Hua *et al.*, 2020; Nagib *et al.*, 2024; Yan *et al.*, 2023; Filali *et al.*, 2022; Wang *et al.*, 2024a; Zhang *et al.*, 2022b; Chang *et al.*, 2023; Sabr *et al.*, 2025b)) have extensively relied on a single DRL agent to manage the inter-RAN slicing level. However, this approach can encounter a significant challenge known as the curse of dimensionality, which arises from the expanded state-action space on the inter-slicing level, particularly in 6G and future networks that are expected to support a massive number of slices and handle diverse resources. As pointed out in a previous study (Azari, Ozger & Cavdar, 2019), large dimensionality and complexity are the biggest obstacles to employing intelligent RRM.

In addition, most existing schemes have solely focused on bandwidth allocation without considering power management. Another limitation of these schemes is that they fail to discuss whether their design strategies can accommodate more future resources or include additional slices, thereby revealing a critical gap in scalability.

Moreover, prior schemes were designed under the assumption that the hardware (HW) conditions of the transmitter and receiver are optimal. However, HW impairments (HWIs) such as amplifier amplitude nonlinearity, quantization errors, and phase noise frequently affect the transmitter and receiver HW of wireless devices, thereby degrading the performance of all realistic implementations (Björnson, Hoydis, Kountouris & Debbah, 2013). This issue is particularly critical for multi-antenna systems that use large uniform linear arrays (ULA) of antennas. Susceptibility to HWIs increases when each antenna element is equipped with low-cost HW. While compensation algorithms can mitigate the impact of these impairments, complete elimination is impossible (Björnson *et al.*, 2013).

Importantly, despite mMTC being one of the essential slices in 5G and beyond networks, previous studies have largely overlooked the management of mMTC slice on the inter-slicing level. Therefore, it is crucial to consider mMTC management in the design of inter-slicing management schemes to achieve more comprehensive management strategies for future networks. Finally, while previous existing studies developed inter-slicing algorithms for SISO systems, there is no evidence to indicate that these algorithms can effectively function with multi-antenna systems, which are anticipated to be widely used in future applications.

To the best of our knowledge, none of the previous studies addressed these gaps. Therefore, in this study, we propose a self-optimizing (SO) scheme to manage multiple radio resources using a cooperative multi-agent DRL (MADRL) framework for heterogeneous inter-RAN slicing (voice over new radio (VoNR), eMBB, uRLLC, and mMTC) while also considering future scalability. This study also investigates the impact of HWIs on the performance of the allocation scheme. Moreover, the proposed approach offers a solution to ensure secure communication between cooperative agents and their environments in multiple-input single-output (MISO) systems. Thus, the proposed model is more general and reliable than those proposed in (Shao *et al.*, 2021; Li *et al.*, 2020a; Hua *et al.*, 2020; Nagib *et al.*, 2024; Yan *et al.*, 2023; Filali *et al.*, 2022; Wang *et al.*, 2024a; Zhang *et al.*, 2022b; Chang *et al.*, 2023), in terms of its ability to manage multiple radio resources, its scalability in handling the management of a set of heterogeneous services, and its secure allocation under realistic conditions such as channel variation, fluctuating traffic

loads, and hardware distortions. Table 3.2 summarizes our contributions in comparison to state-of-the-art inter-RAN slicing management approaches.

This research is essential because it plays a critical role in optimizing the use of limited radio resources in the inter-RAN slicing domain. The proposed allocation scheme is traffic aware, enabling the efficient distribution of resources among heterogeneous services with diverse quality of service (QoS) requirements. Specifically, it ensures that slices with light traffic avoid over-provisioning (overflow of resources) and reduce waste, whereas slices with heavy traffic receive sufficient resources to maintain high QoS and avoid under-provisioning. Importantly, the strategy also incorporates a security-aware design that helps prevent attacks that could bias allocation or disrupt the overall resource management process. By combining traffic awareness with security mechanisms, this study contributes to more reliable, efficient, and robust RAN slicing, which is a key requirement for future mobile networks where the environment is ZTN.

3.2.2 Research Questions

This research aims to address the following research questions (RQ):

- **RQ1:** What DRL techniques and learning frameworks can be adopted to effectively automate RRM and address scalability and dynamicity challenges of inter-RAN slicing?
- **RQ2:** How could we effectively secure the communication between the DRL algorithm and a heterogeneous NS environment?
- **RQ3:** How do HWIs affect the QoS of heterogeneous network slices?

3.2.3 Contributions

In this context, to realize the vision of ZTN in NS and provide customers with a seamless network experience, the present study investigates a real-time inter-slice resource management scheme aligned with the ZTN framework using fluctuations in traffic load within the RAN to manage limited resources across heterogeneous slices efficiently. The main contributions of this work are summarized as follows:

- To effectively manage radio resources across heterogeneous services at the inter-slicing level of O-RAN, we propose **SO** scheme based on **co**operative **m**ultiple **A2C** algorithms, named the SO-CoMA2C scheme. It aims to maximize long-term spectral efficiency while ensuring consistent satisfaction of the SLA ratio (SSR) for admitted services. Unlike state-of-the-art algorithms that focus only on a single radio resource (bandwidth), the proposed scheme enables the simultaneous management of multiple heterogeneous radio resources.
- To align with real-world scenarios where the DRL agent has an incomplete view of the wireless environment, the problem is formulated as a partially observable Markov decision process (POMDP), while taking into consideration the dynamic nature of the environment due to fluctuating traffic loads and user mobility.
- To ensure future management scalability and address complexity challenges—particularly for NS in 6G and beyond, we design the architecture of the proposed scheme based on a fully distributed cooperative learning framework that more effectively tackles scalability issues. In our approach, multiple agents collaborate to form zero-touch inter-slicing agent farm, with each agent responsible for managing specific radio resources across heterogeneous slices. This collaborative structure, built on a distributed learning model, optimizes system performance and mitigates the state-action explosion problem common in single-agent systems. By decentralizing the learning process and enabling cooperation between agents, our approach provides a promising solution to the limitations of single-agent DRL models that dominate the design of state-of-the-art algorithms (*e.g.,* (Li *et al.*, 2020a; Hua *et al.*, 2020; Nagib *et al.*, 2024; Yan *et al.*, 2023; Wang *et al.*, 2024a; Shao *et al.*, 2021; Chang *et al.*, 2023; Sabr *et al.*, 2025b)).
- To analyze how unwanted noise from non-ideal HW affects SSR and the overall system performance, we consider the effects of HW distortion at both the transmitter and receiver in the proposed system model. This is unlike the state-of-the-art algorithms that rely on ideal HW models.
- To address the significant security issues overlooked by state-of-the-art algorithms, as discussed in Section 3.2.1, the proposed scheme integrates the Advanced Encryption Standard (AES) algorithm into the MDP framework. This integration enables encrypted

states, enhances allocation security, and prevents third parties from manipulating the MDP states during communication between the environment and its cooperative agents. To the best of our knowledge, no prior work has addressed this gap in the NS domain.

- To effectively benchmark the proposed SO-CoMA2C scheme and comprehensively evaluate its performance in heterogeneous NS environments, we compare it with state-of-the-art schemes. This assessment considers both ideal and non-ideal HW, non-secure and secure DRL states, as well as varying numbers of users and slices. The evaluation results confirm the robustness and adaptability of the proposed scheme to scale across diverse scenarios in an inter-RAN slicing environment.

3.2.4 Organization

The proposed system model and problem formulation are introduced in Section 3.3. Section 3.4 describes the proposed scheme and its implementation. The simulation results are described in Section 3.5. Section 3.6 concludes.

Table 3.2 Comparing our contributions to the state-of-the-art research on inter-RAN-slicing management

Ref.	Optimized Parameter	Performance Metrics	User Mobility	Method	System	CL	mMTC	HWI at ULA	HWI at receiver	Scalability	Secure Allocation
(Li *et al.*, 2020a)	Bandwidth	Utility	Yes	S-DRL	SISO	No	No	No	No	No	No
(Qi, Hua, Li, Zhao & Zhang, 2019)	Bandwidth	Utility	No	S-DRL	SISO	No	No	No	No	No	No
(Hua *et al.*, 2020)	Bandwidth	Utility	No	S-DRL	SISO	No	No	No	No	No	No
(Nagib *et al.*, 2024)	Bandwidth	latency	No	S-DRL	SISO	No	No	No	No	No	No
(Zhang *et al.*, 2022b)	User association & bandwidth allocation	Utility	No	S-DRL	SISO	No	No	No	No	No	No
(Chang *et al.*, 2023)	Bandwidth	Utility	Yes	S-DRL	SISO	No	No	No	No	No	No
(Yan *et al.*, 2023)	Bandwidth	Utility	Yes	S-DRL	MIMO	No	No	No	No	No	No
(Filali *et al.*, 2022)	Bandwidth	Sum data rates	No	S-DRL	SISO	No	No	No	No	No	No
(Azimi *et al.*, 2022a)	Power & Bandwidth	Energy efficiency	No	DL	SISO	No	No	No	No	No	No
This work	Power & Bandwidth	Spectral efficiency	Yes	**M-DRL**	MISO	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

* CL: Cooperative learning, S-DRL: Single DRL agent, M-DRL: Multiple DRL agents.

3.3 System Model and Problem Formulation

This section presents the system model in detail, including the NS model, the associated mathematical formulations, and the definition of the optimization problem.

3.3.1 Network Slicing Model

In this study, we investigate orthogonal frequency-division multiple access (OFDMA) downlinks of a heterogeneous radio slice scenario in MISO systems, where we consider a BS equipped with a ULAs of antennas $(\mathbb{T} > 1)$, controlled by a near real-time (RT) RAN intelligent controller (RIC) via the E2 interface. E2 open interface that connects the BS to near-RT RIC (Groen *et al.*, 2024b) (see Fig. 3.1). The near-RT RIC is an essential element of the O-RAN architecture, serving a vital role in controlling and optimizing radio resources (Lin, Lin & Chen, 2022). RIC hosts AI algorithms running through applications known as xApps (Groen *et al.*, 2024b), microservice-based applications. They are key to enabling near-real-time control and optimization of RAN functions, typically operating with latency constraints on the order of 10 milliseconds to 1 second. This study considers that the BS hosts all essential slices and operates concurrently, including eMBB, VoNR, uRLLC, and mMTC. The set of essential network slices is denoted by $C = \{1, 2, \ldots, C\}$, and each slice corresponds to a specific type of service. The sets of eMBB, VoNR, uRLLC, and mMTC users with a single antenna are denoted by $\mathcal{E} = \{1, 2, \ldots, e, \ldots, \mathbb{E}\}$, $\mathcal{O} = \{1, 2, \ldots, o, \ldots, \mathbb{O}\}$, $\mathcal{R} = \{1, 2, \ldots, r, \ldots, \mathbb{R}\}$, and $\mathcal{M} = \{1, 2, \ldots, m, \ldots, \mathbb{M}\}$, respectively. Therefore, the set of users in the system is expressed as $\mathcal{U}_c = \{\mathbb{E}, \mathbb{O}, \mathbb{R}, \mathbb{M}\} = \{1, 2, \ldots, U_c\}$. The total number of users is $U_c = \mathbb{E} + \mathbb{O} + \mathbb{R} + \mathbb{M}$, where u_c denotes the user belonging to slice c.

In the proposed system, we assume that each user u_c sends a request to subscribe to the service of slice c . Therefore, each slice c in the system receives a set of requests, denoted by $\mathcal{Q}_c = \{1, 2, \ldots, \mathbb{Q}_c\}$, where $\mathbb{Q}_c$ is the total number of requests made by users belonging to slice c. Furthermore, we assume that $\mathbb{Q}_c$ are sent by authorized users and approved by their corresponding slices. In response to the user's requests (q_c) for a particular service, the BS provides the user with data traffic demand corresponding to that requested service. The data traffic of each slice is represented by α_c. The total traffic demand, α_{total}, of the system can be expressed as shown below.

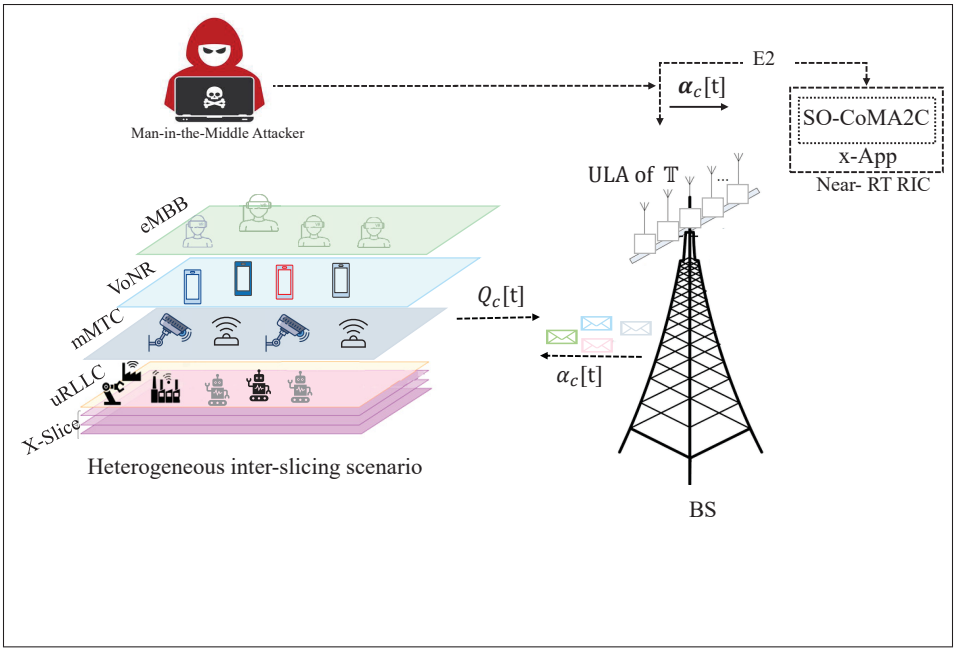

Figure 3.1 System model for heterogeneous RAN-slicing scenario

$$\alpha_{\text{total}}[t] = \sum_{c=1}^{C} \alpha_c[t], \tag{3.1}$$

For simulation purposes, we designed a descriptive traffic model to simulate α_c of each slice. The designed traffic model represents traffic for each user in the form of data packets. Let Φ_{u_c} denote the set of data packets transmitted from the BS to user u_c. Within this set, ψ_{u_c} represents an individual data packet where $\psi_{u_c} \in \Phi_{u_c}$. Thus, the data traffic for each slice can be expressed as follows.

$$\alpha_c[t] = \sum_{u_c \in U_c} \Phi_{u_c}, \forall c \in C, \tag{3.2}$$

The main objective of the designed traffic model is to capture statistical properties and patterns of NS traffic, including packet size and distribution of inter-arrival time, where the traffic of each slice follows a specific distribution (Hua *et al.*, 2020) (see Table 3.3).

Table 3.3 Overview of key settings for traffic generation by slice

Slice	Distribution of Inter-arrival Time per User (Hua *et al.*, 2020)	Packet Size	User's Buffer at BS
VoNR	Uniform distribution between 0 and 160 milliseconds	40 Byte (Nagib *et al.*, 2024)	5
eMBB	Pareto distribution with the mean of 6 milliseconds and the maximum of 12.5 milliseconds	250 Byte (Nagib *et al.*, 2024)	5
uRLLC	Exponential distribution with an average time of 180 milliseconds	32 Byte (Anand & de Veciana, 2018)	5
mMTC	Poisson distribution with mean of λ	85 Byte (Ratasuk, Mangalvedhe, Bhatoolaul & Ghosh, 2017)	5
X-Slice	Uniform distribution between 0 and 90 milliseconds	200 Byte	5

In general, when data traffic arrives at the BS for transmission to users within each slice, it is initially routed to a buffer designated for each specific type of user according to the requested service. The data are then delivered according to the first-come-first-serve (FCFS) policy (Mei *et al.*, 2021). We consider that each user has a queue buffer at the BS with a limited capacity of five packets.

In the context of NS, QoS is typically evaluated using metrics such as the data rate, packet latency, and transmission reliability, which are critical for evaluating adherence to SLAs (Zhang, Pan, Xu, Zhang & Jiang, 2022a). In the present study, we consider the data rate and packet latency as the primary metrics to evaluate the SLA compliance for each slice. Specifically, we define pre-established SLAs criteria by establishing thresholds for the minimum data rate (ξ_c^{Min}) and the maximum allowable latency (L_c^{Max}) for users of each slice (see Table 3.4). Consequently, we introduce a binary variable $d_{\psi_{u_c}} \in \{0, 1\}$, where $d_{\psi_{u_c}} = 1$ indicates that packet $\psi_{u_c} \in \Phi_{u_c}$ has been successfully received by user u_c in slice c (see Eq. (3.3)).

$$d_{\psi_{u_c}} = \begin{cases} 1, & \text{if } \xi_{u_c} \geq \xi_c^{\text{Min}} \ \& \ l_{\psi_{u_c}} \leq L_c^{\text{Max}} \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

where $l_{\psi_{u_c}}$ is the transmission delay of the packet ψ_{u_c} for user u_c in slice c.

Overall, in wireless communication, a packet experiences a variety of delays, such as transmission duration, receiver processing delay, queuing delay at the BS, and the time required to perform additional retransmissions, as needed (Anand & de Veciana, 2018). In this study, we consider

Table 3.4 SLA for admitted slices

Slice	ξ_c^{Min} (Hua *et al.*, 2020)	L_c^{Max} (Hua *et al.*, 2020)	SSR_c^{Th} (Li *et al.*, 2020a)
VoNR	51kbps	10ms	≥ 95%
eMBB	100 Mbps	10ms	≥ 95%
uRLLC	10Mbps	1ms	≥ 95%
mMTC	4Mbps	9ms (Ratasuk *et al.*, 2017)	≥ 95%
X-Slice	6Mbps	10ms	≥ 95%

the following two delaying factors to represent the delay in heterogeneous slices: queuing time (D_{Queuing}) and transmission time (D_{Trans}). While D_{Queuing} is influenced by the scheduling policy and is related to the waiting time for packets in the queue, D_{Trans} is determined by the instantaneous data rate and reflects how quickly the data are transmitted over the network. Therefore, $l_{\psi_{u_c}}$ is the sum of these two elements given below.

$$l_{\psi_{u_c}} = D_{\text{Trans}} + D_{\text{Queuing}}, \tag{3.4}$$

In the proposed system, whenever $l_{\psi_{u_c}}$ exceeds the defined L_c^{Max}, the network drops the packet in accordance with standard network protocols (Zhang *et al.*, 2022a).

We define the SSR to measure the satisfaction ratio of slice users with QoS. The SSR for users served by slice c is determined by the ratio of successfully transmitted packets to the total number of transmitted packets for c. Hence, the average SSR for each slice can be expressed as shown below.

$$\text{SSR}_c = \frac{\sum_{u_c \in \mathcal{U}_c} \sum_{\psi_{u_c} \in \Phi_{u_c}} d_{\psi_{u_c}}}{\sum_{u_c \in \mathcal{U}_c} \left|\Phi_{u_c}\right|}, \tag{3.5}$$

where $|\Phi_{u_c}|$ represents the total number of packets that send from the BS to user u_c in slice c .

According to Shannon's formula, two essential radio resources –namely, power and bandwidth– are crucial to ensure reliable communication. If these resources are not effectively optimized, are not in line with the requirements of future applications/services, communication performance

can be degraded. Therefore, we design a multiple-resource allocation scheme that automates the management of power and bandwidth across heterogeneous services, ensuring that diverse SLA requirements are met for each slice in the system.

Let the total power, $\mathbb{P}^T$ and total bandwidth $\mathbb{B}^T$ of the BS be distributed among admitted slices based on the traffic demand of each slice over a large time scale that fluctuates over time. The assigned power and bandwidth budgets for each slice are denoted by P^c (in dbm) and B^c (in MHz), respectively. We adopt an orthogonal spectrum-sharing strategy to prevent interference among admitted slices. To this end, we introduce a binary decision variable $\Lambda_{B^c}(t)$ which serves as an indicator function. Specifically, if bandwidth B^c is allocated to slice c during time slot t, then $\Lambda_{B^c}(t) = 1$; otherwise, $\Lambda_{B^c}(t) = 0$.

On the intra-slicing level, B^c is partitioned into a set of RBs denoted by F, where each f is characterized by a bandwidth denoted by b. To maintain orthogonality in downlink transmissions among users served by the same slice $c \in C$, each RB $f \in F$ must be exclusively assigned to one user $u_c \in U_c$. Therefore, let the assignment of a RB $f \in F$ to associated user $u_c \in U_c$ in slice $c \in C$ be represented by binary variable $a^f_{u_c}$, where

$$a^f_{u_c} = \begin{cases} 1, & \text{if } f \in \mathcal{F} \text{ is assigned to } u_c \in \mathcal{U}_c, c \in C \\ 0, & \text{otherwise.} \end{cases} \tag{3.6}$$

The received signal of user u_c in slice c at time t is computed as shown in Eq. (3.7).

$$y_{u_c}[t] = \underbrace{\mathbf{h}^H_{u_c}[t]\mathbf{w}_{u_c}[t]z_{u_c}[t]}_{\text{desired signal}} + \underbrace{\mathbb{I}^t_{BS}}_{\text{HWI at ULA}} + \underbrace{\mathbb{N}_{u_c}[t]}_{\text{noise}} + \underbrace{\mathbb{I}^r_{u_c}}_{\text{Self-distortion}}, \tag{3.7}$$

where $\mathbf{h}_{u_c} \in \mathbb{C}^{\mathbb{T}\times 1}$ represents the complex channel vector from the BS to user u_c. Furthermore, $\mathbf{w}_{u_c} \in \mathbb{C}^{\mathbb{T}\times 1}$ denotes the beamformer of the BS. z_{u_n} is the data symbol for user u_c, with $\mathbb{E}\{|z_{u_c}|^2\} = 1$, while $\mathbb{I}^t_{BS} \sim \mathcal{CN}(\mathbf{0}, \mathbf{d^t})$ represents the distortion at the ULA of the BS. The

additive white Gaussian noise (AWGN) at user u_c is denoted as $\mathbb{N}_{u_c} \in \mathcal{CN}(0, \sigma^2)$, in addition to the distortion at receiver $\mathbb{I}^r_{u_c} \sim \mathcal{CN}(0, d^r)$.

The distortion noise variance at the ULA is given below (Björnson *et al.*, 2013).

$$\mathbf{d}^t = \kappa^t \cdot \text{diag}(|w_1|^2, |w_2|^2, \ldots, |w_{(\mathbb{T})}|^2), \quad (3.8)$$

where $\kappa^t \geq 0$ is the distortion level, $\text{diag}(\cdot, \ldots, \cdot)$ indicates a diagonal matrix, and w is the element of $\mathbf{w}$. The variance of the distortion noise at user u_c in slice c is given by Eq. (3.9) (Björnson *et al.*, 2013).

$$d^r = \kappa^r |\mathbf{h}^H \mathbf{w}|^2, \quad (3.9)$$

where $\kappa^r \geq 0$ denotes the distortion level at user u_c.

To simplify the analysis, we assume that all antenna elements in the ULA experience the same level of (κ^t). In addition, all users are assumed to experience the same level of (κ^r).

To analyze the impact of HWI in the proposed system model, we assume that OFDMA provides perfect isolation. Hence, the instantaneous signal to distortion and noise ratio (SDNR) of user u_c in slice c at t time is given using the below equation.

$$\Gamma_{u_c}[t] = \frac{\left|\mathbf{h}^H_{u_c}[t]\mathbf{w}_{u_c}[t]\right|^2}{\underbrace{\kappa^t \sum_{\tau=1}^{\mathbb{T}} |h_\tau w_\tau|^2}_{\text{HWI at ULA}} + \underbrace{\kappa^r \left|\mathbf{h}^H_{u_c}[t]\mathbf{w}_{u_c}[t]\right|^2}_{\text{Self-distortion}} + \underbrace{\sigma^2_{u_c}}_{\text{noise}}} \quad (3.10)$$

where $\sigma^2_{u_c}$ is the noise power at u_c, and the channel coefficient is modeled as shown in Eq. (3.11) (Zhang *et al.*, 2022c).

$$\mathbf{h}_{u_c}[t] = \sqrt{\beta_{u_c}}\, \mathbf{g}_{u_c} \quad (3.11)$$

where β_{u_c} is the large-scale fading that includes pathloss and shadowing, and $\mathbf{g}_{u_c}$ is the small-scale fading vector.

The achievable data rate (ξ_{u_c}) of the u_c user belonging to a slice $c \in C$ is defined as follows:

$$\xi_{u_c}[t] = \begin{cases} b \ \log_2\left(1 + \Gamma_{u_c}[t]\right); \text{for long packet transmission;} \\ b \log_2\left(1 + \Gamma_{u_c}[t]\right) - \sqrt{\frac{D_{u_c}}{K_{u_c}}}\, Q^{-1}(\epsilon); \text{for short} \\ \text{packet transmission} \end{cases} \tag{3.12}$$

where $Q^{-1}(\cdot)$ is the inverse of the Gaussian Q-function (Yan *et al.*, 2023), K_{u_c} is the length of the packet, ϵ is the transmission error probability, and $D_{u_c} = \frac{1}{\left(1+\Gamma_{u_c}[t]\right)^2}$ represents the characteristic of the channel dispersion. In this study, short packet transmission is used for latency-sensitive services such as uRLLC, which operate under strict delay constraints. In contrast, long packet transmission is adopted for services such as eMBB, mMTC, and VoNR, which transmit larger payloads and operate under more relaxed latency requirements (e.g., 10 ms).

In this study, our aim is to find the optimal power and bandwidth allocation budgets that maximize spectral efficiency (η) while meeting the SSR_c for each slice. Here, η is given by

$$\eta[t] = \frac{\sum_{c \in C} \sum_{u_c \in \mathcal{U}_c} \xi_{u_c}}{\mathbb{B}^T}. \tag{3.13}$$

3.3.2 Optimization Problem Formulation

This subsection defines the optimization objective and outlines the associated system constraints.

$$\mathcal{OF}: \quad \underset{\mathbf{P}^c_{\max}, \mathbf{B}^c_{\max}}{\text{maximize}} \quad \eta[t] \tag{3.14}$$

subject to

$$\text{C1:} \quad P^c_{\max} \geq \lambda^c_p; \quad \forall c \in C,$$

$$\text{C2:} \quad \sum_{c \in C} P^c_{\max} = \mathbb{P}^T; \quad \forall t,$$

$$\text{C3:} \quad B^c_{\max} \geq \lambda^c_b; \quad \forall c \in C,$$

$$\text{C4:} \quad \sum_{c \in C} B^c_{\max} = \mathbb{B}^T; \quad \forall t,$$

$$\text{C5:} \quad \sum_{c \in C} \Lambda_{B^c}(t) \leq 1; \quad \forall t,$$

$$\text{C6:} \quad \boldsymbol{\alpha}_{\mathbf{c}} = (\alpha_{c1}, \ldots, \alpha_C),$$

$$\alpha_c \sim \text{Certain Traffic Model}; \quad \forall c \in C,$$

$$\text{C7:} \quad d_{\psi_{u_c}} \in \{0, 1\}; \quad \forall u_c \in \mathcal{U}_c, \forall c \in C,$$

$$\text{C8:} \quad SSR_c \geq \text{SSR}^{\text{Th}}_c; \quad \forall c \in C,$$

$$\text{C9:} \quad \|\mathbf{w}_{u_c}\|^2 \geq 0; \quad \forall u_c \in \mathcal{U}_c, \forall c \in C,$$

$$\text{C10:} \quad \sum_{u_c \in U_c} \|\mathbf{w}_{u_c}\|^2 \leq P^c_{\max}; \quad \forall c \in C,$$

$$\text{C11:} \quad \sum_{u_c \in \mathcal{U}_c} a^f_{u_c} \leq 1; \quad \forall f \in \mathcal{F}, \forall c \in C,$$

$$\text{C12:} \quad \sum_{u_c \in U_c} \sum_{f \in \mathcal{F}} a^f_{u_c} \leq B^c_{\max}; \forall c \in C.$$

In Eq. (3.14), constraint C1 ensures minimum allocated power (λ^c_p) per slice. C2 represents the total power allocated to each slice and equals the total system power . C3 ensures the minimum allocated bandwidth (λ^c_b) per slice. C4 represents the total allocated bandwidth for each slice and is equal to the total system bandwidth. C5 guarantees that slices are isolated from one another. In C6, each α_c is a traffic model for each slice in the system. C7 ensures the SLA of each slice. Furthermore, C1 and C3 ensure that each slice receives a minimum allocation of resources on the inter-level, thereby preventing the possibility of allocating zero resource budgets, which could lead to the removal of the slice from the system in the ZTN. This measure is crucial to maintain ZTN environment. Next, C8 ensures that the SSR of each slice exceeds or

is equal to a predefined threshold. Constraint C9 ensures that the power allocated to each user is non-negative. C10 guarantees that the power allocated to users in each slice does not exceed the slice budget. C11 ensures that each RB is assigned to one user at a time to prevent interference and maintain orthogonality. C12 ensures that the sum of the allocated bandwidths for the users of each slice does not exceed the total budget of that slice.

The challenges in addressing the $\mathcal{OF}$ problem arise from the following two main factors. First, joint optimization involves the allocation of power and bandwidth across heterogeneous network slices. The management of multiple resources operates at two levels– intra-slicing and inter-slicing– which add significant complexity to the problem. Second, over time, the varying traffic demand, which cannot be known in advance owing to the nature of the traffic model, makes it nearly impossible to develop an accurate mathematical model (Zhang *et al.*, 2022b) for practical systems. The dynamic characteristics of RAN slicing make DRL methods particularly effective in addressing these challenges (Filali *et al.*, 2022).

Therefore, in Section 3.4, we decompose the $\mathcal{OF}$ problem into two subproblems, each representing a level of management in the RAN. The first is inter-slicing, which is represented by C1-C8 and is solved using MADRL. The second level is intra-slicing, and here we use traditional algorithms, including the max-min fairness power control (MMFPC) algorithm (Björnson, Demir *et al.*, 2024, Ch. 6), to allocate and manage power within each slice while ensuring that constraints C9-C10 are satisfied. Meanwhile, the round-robin (RR) algorithm is used to distribute the bandwidth among active users within each slice, as in (Nagib *et al.*, 2024), also ensuring compliance with C11-C12.

3.4 Towards Heterogeneous Inter-RAN Slicing RRM Based on Cooperative MADRL

This section presents our proposed approach to solving the formulated resource allocation problem Eq. (3.14) in a computationally efficient manner, making it suitable for practical use as an inter-slicing scheme in ZTN environments.

3.4.1 Motivation for Cooperative MADRL

This subsection discusses the motivation behind the design approach and the selected DRL algorithm.

We propose SO scheme for the inter-slicing level based on cooperative MADRL framework. The cooperative learning framework is one of the most important types of learning in multi-agent reinforcement learning (MARL) (Shi *et al.*, 2022), where a set or team of independent agents dynamically interacts with a shared environment to determine the optimal policy (Tan *et al.*, 2022). The structure of the proposed inter-RAN slicing scheme is shown in Fig. 3.2.

The motivation behind designing the proposed scheme based on cooperative MADRL stems from the increasing complexity of managing resources on the inter-RAN slicing level as we move towards next-generation networks, particularly 6G and beyond. This complexity arises from the need to allocate various radio resources, as well as other critical resources such as CPU, memory, and storage, across multiple network slices. In addition, 6G is expected to support a large number of slices, with no predefined limits on the number of slices in future networks. Accordingly, managing such scenarios through static allocation would be an invalid option, leading to wasted resources. Moreover, concurrently managing multiple radio resources on the inter-slicing level or adding more slices expands the action space for DRL agents, making efficient resource management even more challenging. Specifically, a significant increase in the number of potential states and actions to explore can considerably hinder the performance of DRL systems, making them more complex and less efficient. Therefore, the scalability of existing inter-slicing management schemes is a central concern. To address these challenges, we propose SO scheme based on a distributed cooperative MADRL framework that focuses on managing resources on the inter-slicing level.

The proposed scheme, which is designed to be adaptable, initially focuses on optimizing multiple radio resources while remaining sufficiently flexible to accommodate additional resource types, as needed. In our scheme, we consider an agent for managing each radio resource. Therefore, each agent acts in accordance with its own policy. This design flexibility ensures that the

system can evolve into a comprehensive SO agent farm capable of addressing future resource management needs on the inter-slicing level. The proposed approach can significantly reduce the action space as compared to a single agent that must explore all possible actions. This approach also enhances scalability of the system, making it better suited for handling future network expansion (Wang *et al.*, 2024b).

Various DRL algorithms can be employed to design cooperative-learning frameworks. However, inspired by the performance of the A2C algorithm reported in (Li *et al.*, 2020a), we selected A2C for the design of the proposed scheme. A2C has proven beneficial in a number of real-world applications, such as power control, due to its ability to find optimal policies using low-variance gradient estimations, good convergence properties, and support for continuous action spaces (Grondman, Busoniu, Lopes & Babuska, 2012).

Furthermore, designing the proposed scheme so that all agents work cooperatively and train in a distributed manner offers significant benefits. This design significantly reduces the risk of a complete system failure that can occur if a single agent is responsible for allocating all resources simultaneously. By contrast, in the proposed approach, the failure of a single agent does not affect the overall system. Although the agents operate in a cooperative mode, they maintain independent architectures. As a result, a failure in any one agent does not lead to complete system failure, since each agent is responsible for its own tasks.

3.4.2 Markov Model for the heterogeneous Network Slices

In this subsection, the specific definitions of state, action, and reward are introduced for the cooperative agents.

We model Eq. (3.14) as the POMDP model, where DRL agents have incomplete or limited information about the state of the environment. While agents receive information about the traffic load for each slice in the proposed scheme, a complete view of the heterogeneous inter-RAN slicing state (including channel gain, level of distortion, number of users per slice, and overall

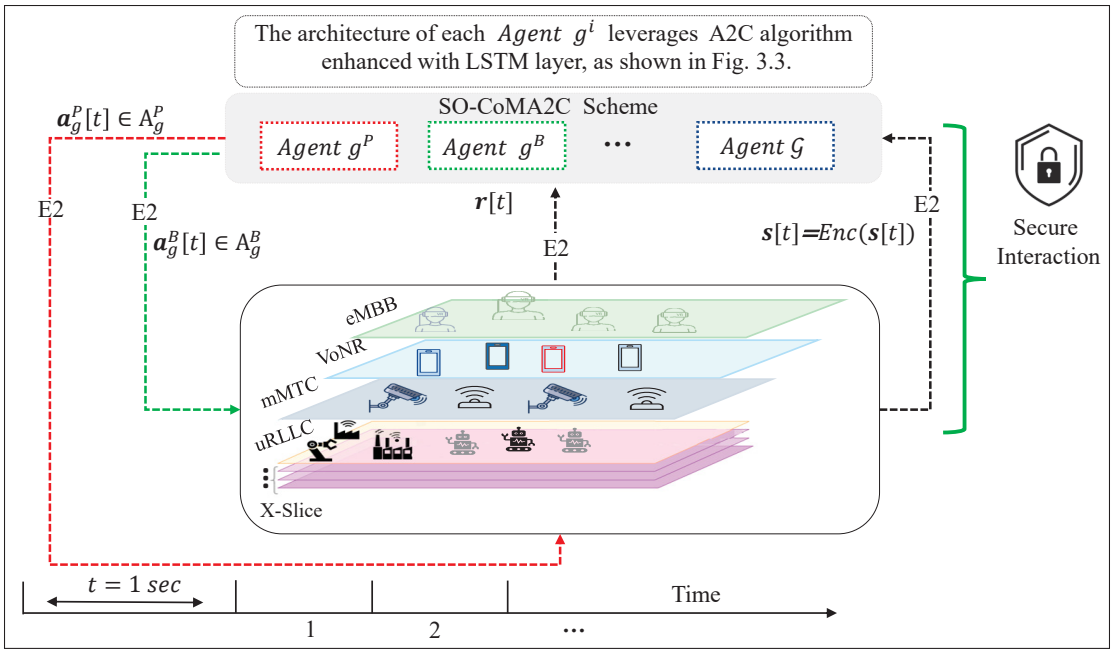

Figure 3.2 Architecture of the proposed secure SO-CoMA2C scheme for heterogeneous slices MISO systems

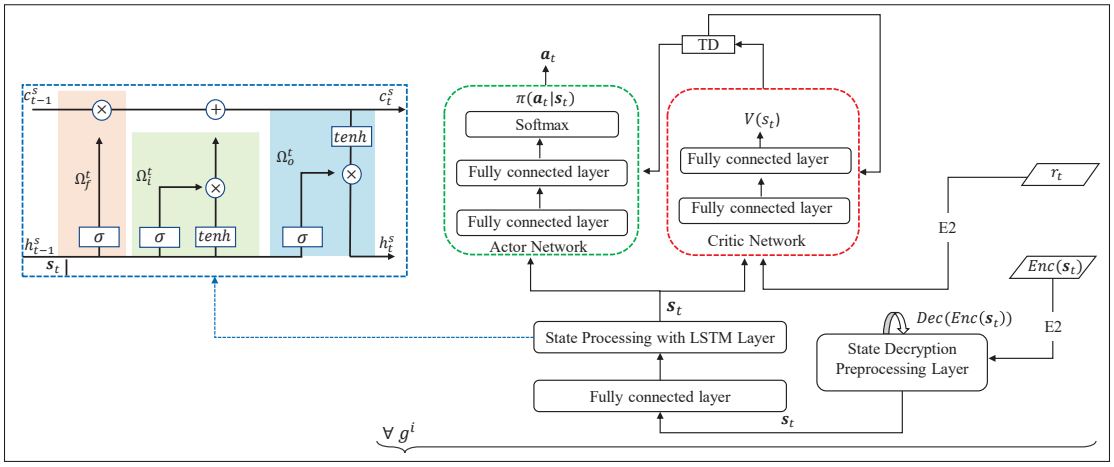

Figure 3.3 Learning network architecture of each agent in the proposed SO-CoMA2C scheme

SSR) remains inaccessible. This better reflects real-world conditions, where agents often operate under partial observability.

We denote the set of agents in the SO-CoMA2C scheme by $\mathcal{G}$, which is given by

$$\mathcal{G} = \{g^i \ \forall i \in \{P, B\}\}, \tag{3.15}$$

where i refers to the type of radio resource allocated by agent g^i, while P and B denote the power and bandwidth allocation, respectively. All agents simultaneously learn in the same environment. In the proposed scheme, each agent is defined by three fundamental elements that adhere to the MDP framework: state space, action space, and reward function (Filali *et al.*, 2022). These elements work together, and their relationship is governed by the agent's objectives. The environment is defined by a set of heterogeneous network slices, with xApp serving as the container executing the proposed scheme in the near-RT. Communication between the agents and the shared NS environment is facilitated by the E2 interface.

3.4.2.1 Action Space ($\mathcal{A}$)

In the proposed scheme, each agent operates within its own action space. Specifically, the agent responsible for managing power among the heterogeneous slices has an action space denoted by $\mathcal{A}_g^P$. By contrast, the agent in charge of the bandwidth allocation operates within the action space represented by $\mathcal{A}_g^B$. Action ($\mathbf{a}_t$) of g^P involves a vector of power budgets, which can be mathematically represented as follows:

$$\mathbf{a}_g^P[t] = \{(P_{\max}^{c_1}, P_{\max}^{c_2}, P_{\max}^{c_3}, P_{\max}^{c_4}) \in \mathbf{P}_{\max}^c \mid \lambda_p^c \leq P_{\max}^c \leq \mathbb{P}^T, \ \mathbf{a}_g^P \in \mathcal{A}_g^P, \} \tag{3.16}$$

where $\mathcal{A}_g^P$ represents all possible action combinations between λ_p^c and $P_{\max}$ for the admitted slices.

On the other hand, $\mathbf{a}_t$ of g^B involves a set of bandwidth budgets represented below.

$$\mathbf{a}_g^B[t] = \{[B_{\max}^{c_1}, B_{\max}^{c_2}, B_{\max}^{c3}, B_{\max}^{c4}] \in \mathbf{B}_{\max}^c \mid \lambda_b^c \leq B_{\max}^c \leq \mathbb{B}^T, \mathbf{a}_g^B \in \mathcal{A}_g^B, \} \tag{3.17}$$

where $\mathcal{A}_g^B$ represents all possible action combinations between λ_b^c and $B_{\max}$. Therefore, the total joint action space of all agents is calculated as $\mathcal{A} = \sum_{g=1}^{\mathcal{G}} \mathcal{A}_g^i$.

For RIC to effectively manage radio resources on the inter-slicing level, it is crucial to have up-to-date information about the status of each slice on the intra-slicing level. More specifically, it is essential to understand the traffic demand that the BS plans to send in the downlink, as well as the QoS or SSR for each slice. Consequently, the proposed approach incorporates this information into the state and reward of each DRL agent.

3.4.2.2 State Space ($\mathcal{S}$)

The state provides the agent with insights about heterogeneous NS environment. In our scenario, the network state for each $g^i \in \mathcal{G}$ consists of traffic demand of each slice, which is defined as.

$$\mathbf{s}[t] = \boldsymbol{\alpha} = \left[\alpha_{c_1}, \alpha_{c_2}, \alpha_{c_3}, \alpha_{c_4}, \alpha_C\right] \quad \forall g^i \in \mathcal{G}, \tag{3.18}$$

As discussed in Section 3.2.1, the state is one of the most critical pieces of information exchanged between the heterogeneous NS environment and DRL agent. However, the state of the system can be vulnerable to external manipulations, such as man-in-the-middle, network intrusion and eavesdropping attacks (Groen *et al.*, 2024a), which can alter and manipulate data as they pass through the E2 interface. Numerous studies, including (Groen *et al.*, 2024a), identified data passing through the E2 interface as a major threat to O-RAN interfaces, since E2 serves as the primary link between RT-RIC and RAN. Accordingly, any kind of manipulation can impact the integrity of DRL decisions and ultimately cause a decline in system performance, leading to an inefficient use of limited resources.

Therefore, to ensure the secure allocation of radio resources on the inter-slicing level in the proposed system, we consider security of the state space to prevent state manipulations and to ensure secure communication between the environment and its agents. Specifically, the proposed scheme encrypts state information on the environment side and sends an encrypted state to

the agent over the E2 interface. This modification ensures that the original state data are not accessible to any third party that might intercept communication over the E2 interface. The encrypted states are then decrypted at the x-App.

To this end, we adopt a cryptographic algorithm known as AES. AES is a high-speed symmetric encryption technique, where a single, shared key is used to encrypt and decrypt data between the sender and the receiver (Groen *et al.*, 2024a; Lu & Tseng, 2002). AES offers numerous benefits over other encryption methods, such as flexibility, high efficiency, and robust security (Ananya, Nikhitha, Arjun & Gowda, 2023). It is now frequently used as the standard encryption algorithm for several applications (Ananya *et al.*, 2023), more details on architectures of AES, and mathematical encryption procedure can be found in (Lu & Tseng, 2002).

Accordingly, we use AES-128 to perform encryption of the state on the environment side. AES supports several modes of operation, including electronic codebook (ECB), cipher block chaining (CBC), and Galois counter mode (GCM), each offering distinct trade-offs between security and computational efficiency (Chen, 2024). Our primary objective is to ensure the confidentiality of the state vector during transmission within the DRL framework, where both performance and resource efficiency are critical factors. Therefore, we require an encryption mode that maintains a reasonable balance between security and computational cost. Guided by the results of prior research, particularly (Chen, 2024) which highlights that CBC offers lower encryption and decryption overheads compared to more complex modes such as GCM. Although GCM provides strong security guarantees and built-in authentication, it incurs higher computational overhead (Chen, 2024). Conversely, ECB is computationally efficient but is widely regarded as insecure, even with longer key lengths (Chen, 2024). Therefore, the AES-CBC mode is adopted in the proposed scheme, as it provides a trade-off between security and computational efficiency, making it a practical choice for lightweight, distributed DRL framework operating in ZTN environments.

In this study, the AES encryption algorithm is implemented in Python, leveraging PyCryptodome, a robust and widely adopted cryptographic library. It provides efficient built-in functions for

symmetric encryption, hashing, and secure key generation, thereby streamlining the development of secure encryption solutions (Pasarelski, Angelov, Postagian & Sadinov, 2023).

The encrypted state is represented by Eq. (3.19), while the decrypted state is represented by Eq. (3.20).

$$\text{Enc}(\mathbf{s}[t]) = \text{Enc}\left([\alpha_{VoNR}, \alpha_{eMBB}, \alpha_{uRLLC}, \alpha_{mMTC}]\right) \quad \forall g^i \in \mathcal{G}, \tag{3.19}$$

where $\text{Enc}(\cdot)$ denotes the encryption function applied to state vector.

$$\text{Dec}\left(Enc(\mathbf{s}[t])\right) = \text{Dec}\left(\text{Enc}\right.$$
$$\left.\left([\alpha_{VoNR}, \alpha_{eMBB}, \alpha_{uRLLC}, \alpha_{mMTC}]\right)\right)) = \mathbf{s}[t]; \quad \forall g^i \in \mathcal{G}, \tag{3.20}$$

where $\text{Dec}(\cdot)$ denotes the decryption function applied to the encrypted state, which returns the original state vector.

3.4.2.3 Reward ($\mathcal{R}$) Function

The reward function plays a vital role in guiding the agent's behavior. In the proposed scheme, the reward (r_t) for each agent is formulated based on a combination of η and the specific requirements of SSR_c. The design incorporates three penalty terms: (1) ∇^{η}, a proportional penalty applied when all SSR_c constraints are met but η falls below the defined threshold η^{Th}. Based on recent literature where (Yan *et al.*, 2023) employs 200 bps/Hz and (Li *et al.*, 2020a) uses 280 bps/Hz as threshold values, extensive experimentation with various threshold values across this range, identified 250 bps/Hz as yielding optimal performance in our multi-slice environment; (2) ∇^{uRLLC}, a proportional penalty is applied when the SSR of the uRLLC slice is not satisfied; and (3) $\nabla^{\mathcal{T}}$, a comparatively larger negative penalty (*i.e.,* -10) is applied when any one of VoNR, eMBB, mMTC fails to meet its predefined SSR requirement. The scaling factors 0.1 and 10, used in Lines 4 and 9 of Algorithm 3.1, respectively, are chosen in accordance with

those described in (Yan *et al.*, 2024). Consequently, each agent receives a collaborative reward. The calculation of the reward for each $g^i \in \mathcal{G}$ is defined by Algorithm 3.1.

Algorithm 3.1 Compute $\mathcal{R}$ based on SSR_c & η

Input: SSR_c, SSR_c^{Th}, η
if SSR_{VoNR} **&** SSR_{eMBB} **&** $\text{SSR}_{mMTC} \geq \text{SSR}_c^{Th}$ **then**
  if $\text{SSR}_{uRLLC} \geq \text{SSR}_c^{Th}$ **then**
    if $\eta < \eta^{Th}$ **then**
      $r[t] \leftarrow \nabla^{\eta} \leftarrow 10 + (\eta - \eta^{Th}) \times 0.1$
    else
      $r[t] \leftarrow 10$
    end if
  else
    $r[t] \leftarrow \nabla^{uRLLC} \leftarrow (\text{SSR}_{uRLLC} - 0.8) \times 10$
  end if
else
  $r[t] \leftarrow \nabla^{\mathcal{T}} \leftarrow -10$
end if
Output: $r[t]$

3.4.3 Overview of the A2C Algorithm

To solve the POMDP problem formulated in the previous section, we use cooperative multiple A2C agents. Each A2C agent is a temporal difference (TD) learning algorithm whose architecture consists of the following two core components: the actor network and critic network (Grondman *et al.*, 2012), where each network is implemented in the form of a deep neural networks (DNNs), as shown in Fig. 3.3. This kind of algorithm is a hybrid approach that incorporates two approaches: policy-based (Paczolay & Harmati, 2020), which is represented by the actor, in charge of learning a policy (π) to maximize the expected reward, and value-based (Paczolay & Harmati, 2020), which is represented by the critic aims to estimate an accurate value function for evaluating the actor's actions (Kouchaki & Marojevic, 2022). The idea behind this hybrid approach is to address the shortcomings of each approach when applied separately (Grondman *et al.*, 2012).

In short, the actor network's role of each agent is to explore the action space and make decisions based on the existing π, whereas the critic network of each agent evaluates the decisions made

by the actor (Kouchaki & Marojevic, 2022). Similar to the other DRL algorithms, at each time step, the agent observes its current state $s_t \in \mathcal{S}$ and selects action $a_t \in \mathcal{A}$. Then, the agent receives reward r_t so as to transition to the next state s_{t+1}. The total accumulated reward (expected return) at time slot t is defined by $R_t = \sum_{t=0}^{\infty} \Upsilon^t r_t$, where Υ ($0 \leq \Upsilon \leq 1$) is the discount factor, a hyperparameter that emphasizes the significance of future rewards (Sun & Zhang, 2022). The actor of each agent in charge of learning the best π to maximize R_t, which is determined using the state-value function $V(s)$, as well as the action-value function $Q(s, a)$. $V(s)$, which estimates the average expected return when the agent is in s, is mathematically expressed as $V(s) = \mathbb{E}[R^t_{g^i}|s_t = s]$, whereas $Q(s, a)$, which calculates the expected return (total future rewards) to choose a an in s at time step t, is mathematically represented as $Q(s, a) = \mathbb{E}[R^t_{g^i}|s_t = s, a_t = a]$. The weights of the actor and critic networks are denoted by $\psi^a_{g^i}$ and $\psi^c_{g^i}$, respectively. The actor is trained using the policy gradient method with respect to policy parameters $\psi^a_{g^i}$, denoted by $\nabla_{\psi^a_{g^i}} \log \pi(a_t|s_t; \psi^a_{g^i})\delta(s_t)$, where ∇ is the gradient with respect to $\psi^a_{g^i}$ and $\delta(s_t)$ is the advantage function indicating the advantage of executing a_t at s_t is defined as

$$\delta_t = \underbrace{r_{t+1} + \Upsilon V\left(s_{t+1}; \psi^c_{g^i}\right)}_{Q(s_t, a_t)} - V\left(s_t; \psi^c_{g^i}\right). \tag{3.21}$$

The critic can effectively estimate the value function under a particular π using δ_t (Zhang, Yu, Fu, Yan & Wang, 2018).

3.4.4 LSTM-Enhanced A2C Algorithm

Similarly to real-world MADRL contexts, where agents frequently have only a limited view of the overall state, which results in an incomplete understanding of their environment (Tan *et al.*, 2022), the A2C agents in the proposed scheme operate in a partially observed heterogeneous inter-RAN slicing environment. This type of setting complicates making the right decisions in real-world deployments, particularly when traffic dynamics are unpredictable for heterogeneous

services. This uncertainty makes it problematic to perform timely identification of the feasible solutions. Hence, the traditional A2C algorithm may struggle to effectively address these challenges.

To address this concern, the proposed approach incorporates a hybrid architecture for each agent by combining DNN and LSTM layer. Specifically, in our implementation, the actor and critic networks of each agent share a common LSTM layer, as shown in Fig. 3.3. This design enhances the learning reliability by leveraging the strengths of LSTMs. LSTMs are particularly effective with regard to managing sequential information and maintaining state awareness, while tackling the issue of vanishing gradients that affects conventional recurrent neural networks (RNNs). The LSTM layer enables the proposed system to retain and extract relevant features from the historical traffic demand, thereby significantly improving the network's decision-making capabilities based on past traffic patterns. The structure of an LSTM cell is composed of the following two types of states: the first is the cell state (c_t^s), which is the long-term memory, and second is the hidden state (h_t^s), which represents short-term memory (Chang *et al.*, 2023). Furthermore, there are three gates in LSTM: the forget gate (Ω_f^t), which determines whether the data from the prior cell state (c_{t-1}^s) should be retained or discarded; the input gate (Ω_i^t), which is used to measure the significance of the new information carried by the input; and, finally, the output gate (Ω_o^t), which uses the current input and the previous hidden state to calculate the next hidden state output. Each has its own weight (Θ), bias (Ξ), and sigmoid (σ) activation function (Chang *et al.*, 2023). Equation (3.22) can be used to express LSTM gates:

$$\begin{aligned} \Omega_f^t &= \sigma\left(\Theta_f \cdot [h_{t-1}, \mathbf{s}_t] + \Xi_f\right) \\ \Omega_i^t &= \sigma\left(\Theta_i \cdot [h_{t-1}, \mathbf{s}_t] + \Xi_i\right) \\ \Omega_o^t &= \sigma\left(\Theta_o \cdot [h_{t-1}, \mathbf{s}_t] + \Xi_o\right) \end{aligned} \tag{3.22}$$

Both c_t^s and h_t^s can be determined as follows:

$$
\begin{aligned}
c_t^s &= \Omega_f^t \cdot c_{t-1}^s + \Omega_i^t \cdot tanh\left(\Theta_c\left[\mathbf{s}_t, h_{t-1}^s\right] + \Xi_c\right) \\
h_t^s &= \Omega_o^t \cdot tanh\left(c_t^s\right)
\end{aligned}
\tag{3.23}
$$

3.4.5 Implementation of the Proposed SO-CoMA2C Scheme

The architecture of the proposed SO-CoMA2C is illustrated in Fig. 3.2. It is mainly composed of th following three core components: LSTM, cooperative multiple A2C agents, and AES algorithm. Considering the fundamental principles and components of A2C and LSTM algorithms explained in Sections 3.4.3 and 3.4.4, the architecture of each agent in the proposed scheme is designed.

Algorithm 3.2 describe our proposed secure SO-CoMA2C scheme. The decision-making process of the proposed scheme starts with interactions between the agents and the NS environment on a large timescale (each 1 second). At each time step, the environment communicates with the agents by providing information regarding the traffic load for each slice in the downlink transmission, represented as a state vector in Eq. (3.18). Subsequently, according to Eq. (3.19), the state vector is converted from plain text to ciphertext (encrypted), where the AES algorithm is applied to secure NS states. This process involves applying a set of mathematical operations to the state vector including SubBytes, ShiftRows, MixColumns, and AddRoundKey. After receiving the encrypted state, the architecture of each agent is designed with a custom preprocessing layer that executes state decryption before passing it to the LSTM and DNN layers, according to Eq. (3.20), where the inverse procedure is performed to convert the state from ciphertext to plaintext. We design the encryption of the state on the environment side and decryption on the agent side as a custom preprocessing layer. This approach keeps the network simpler and allows for learning from the decrypted state, thus reducing the time that could otherwise be needed if the decryption were done inside the DNNs layers. Next, the actor network of each agent performs actions based on obtain state in Eq. (3.20) and assigns a $\mathbf{P}_{\max}^c$ and $\mathbf{B}_{\max}^c$ in line with Eq. (3.16) and (3.17), respectively. Afterwards, the $P_{\max}^c$ of each slice is distributed among active users of each slice according to the MMFPC algorithm, whereas the $B_{\max}^c$ of each slice is distributed among the

active users of each slice based on the RR algorithm. Subsequently, the BS checks whether the SLA of each slice satisfies the SSR threshold. Based on the overall performance of the system and the SSR of each slice, the environment sends a reward to each agent according to Algorithm 3.1. The critic network of each agent in the proposed scheme observes the reward obtained and the new state resulting from the environment's response. Then, the TD Error is evaluated by the critic, as it is essential to the computation of the loss. To enhance the exploration capabilities, we incorporate entropy regularization into the cost function of actor network. The actor loss function can be represented as shown in Eq. (3.24) (Li *et al.*, 2020a).

$$\mathcal{L}^{Actor} = -[\delta_t(\mathbf{s}_t; \Psi^c_{g^i}) \underbrace{\log \pi(\mathbf{a}_t|\mathbf{s}_t; \Psi^a_{g^i}))}_{\text{Action log probability}} + \phi \mathbb{E}(\pi(\mathbf{a}_t|\mathbf{s}_t; \Psi^a_{g^i}))] \tag{3.24}$$

where $\mathbb{E}(\pi(\mathbf{a}_t|\mathbf{s}_t; \Psi^a_{g^i})$ denotes the entropy term encourages exploration of actor network during the learning process. Parameter ϕ controls the strength of the entropy regularization. A larger ϕ encourages more exploration (higher entropy), while a smaller ϕ makes the policy more deterministic. The critic network's loss function (Li *et al.*, 2020a) is expressed as

$$\mathcal{L}^{Critic} = \underbrace{(r_t + \Upsilon V(\mathbf{s}_{t+1}; \Psi^c_{g^i}) - V(\mathbf{s}_t; \Psi^c_{g^i}))^2}_{TD \text{ error / Advantage}}. \tag{3.25}$$

Finally, the actor and critic parameters are iteratively updated based on Eq. (3.26) and (3.27), respectively.

$$\Psi^a_{g^i}[t+1] \leftarrow \Psi^a_{g^i} + \varrho^a_{g^i} \underbrace{\nabla \mathcal{L}^{\text{Actor}}(\Psi^a_{g^i})}_{\text{Gradient of actor loss}}, \tag{3.26}$$

$$\Psi^c_{g^i}[t+1] \leftarrow \Psi^c_{g^i} + \varrho^c_{g^i} \nabla(\delta_t)^2. \tag{3.27}$$

where $\varrho^a_{g^i}$ and $\varrho^c_{g^i}$ represent the learning rates for the actor and critic, respectively.

Algorithm 3.2 Secure SO-CoMA2C Scheme

Initialization:

- $\forall c \in C$: initialize:
  Queue buffer, Latency buffer,
  Traffic model, User location,
  User speed, SSR_c^{Th}.
- $\forall g^i \in \mathcal{G}$ initialize Actor's network with $\Psi_{g^i}^a$ critic's network with $\Psi_{g^i}^c$,
  Learning rates of actor $\varrho_{g^i}^a > 0$ &
  the critic $\varrho_{g^i}^c > 0$.

for *iteration* $i = 1$ **to** $\mathcal{F}^{\mathcal{I}}$ **do**
  for $t = 1$ **to** T **do**
    for *agent* $g = 1$ **to** $\mathcal{G}$ **do**
      NS environment sends $Enc(s_t)$ of C according to Eq. (3.19) to $\mathcal{G}$;
      Get the $Enc(\mathbf{s}_t)$ and perform state decryption at each $g^i \in \mathcal{G}$ according to Eq. (3.20) ;
      Each $g^i \in \mathcal{G}$ performs $a_t = (\mathbf{a}_g^P, \mathbf{a}_g^B) \in \{\mathcal{A}_g^P, \mathcal{A}_g^B\}$ to allocate $\mathbf{P}_{\max}^c$ and $\mathbf{B}_{\max}^c$ in line with Eqs. (3.16) and (3.17), respectively;
      Manage $P_{\max}$ & $B_{\max}$ among active users of each $c \in C$ based on MMFPC and RR, respectively;
      Calculate SLA, SSR requirements for each $c \in C$ & system η;
      Calculate the reward based on Algorithm 3.1, observe the next state s_{t+1};
      Critic of each agent calculates TD error by Eq. (3.21);
      Calculate actor loss and critic loss according to Eq. (3.24) and (3.25), respectively;
      Update the parameters of actor network $\Psi_{g^i}^a$ by (3.26) and critic network $\Psi_{g^i}^c$ by (3.27) per learning steps;
      Obtain the new the state vector: $s_t = s_{t+1}$;
      Repeat until completion of $\mathcal{F}^{\mathcal{I}}$ iterations
    end for
  end for
end for

Output: Best policy $\pi^* \left(\mathbf{a}(t) \mid \mathbf{s}(t), \forall g^i \in \mathcal{G}\right)$.

3.5 Experiment Results and Discussion

In this section we evaluate the effectiveness of the proposed SO-CoMA2C scheme in managing multiple radio resources across heterogeneous slices on the inter-slicing level, in a time-varying and partially observable environment, under both ideal and non-ideal HW conditions.

Table 3.5 Experiments parameters

Parameter	**Value**
Total users (U_c)	400 (default), 800, 1200
r_c (m)	40 (Li *et al.*, 2020a)
Probability of users in each slice	VoNR =1, eMBB= 2, , uRLLC = 3 and mMTC= 4
User movement	1 m/s (VoNR), 4m/s (eMBB), 8 m/s (uRLLC and mMTC) (Li *et al.*, 2020a)
Size of ULA	16,32,48, 64 (default) antennas
Total available power ($\mathbb{P}^T$)	50 dBm
Total available bandwidth ($\mathbb{B}^T$)	10 MHz
Minimum power per slice (λ_p^c)	10 dBm
Minimum bandwidth per slice (λ_b^c)	1 MHz
The transmission error probability (ϵ)	0.00001 (Yan *et al.*, 2024)
Optimizers	Adam
Actor learning rate ($\varrho_{g^t}^{a_t}$)	0.005
Critic learning rate ($\varrho_{g^t}^{c_t}$)	0.008
Discount factor (Υ)	0.99
Slice SSR threshold (SSR_c^{Th})	95%
Level of distortion (κ^t(and (κ^r)	2e-8-2e-6
Maximum latency ($L_c^{\max}$)	10 ms (VoNR), 10 ms (eMBB), 1 ms (uRLLC), 9 ms (mMTC)
Noise power ($\sigma_{u_c}^2$)	-174 dBm
Path loss model (β)	$120.9 + 37.6 \log_{10}(\text{dis})$
LSTML layer	1 LSTM (shared) layer with 64 neurons
Entropy rate	0.01
Actor network	Fully connected layer [32,2]
Critic network	Fully connected layer [32,2]
Simulation time	10,000 time slots, with a duration of 1 second for each slot

3.5.1 Simulation Setup

We consider an O-RAN scenario with four heterogeneous slices (*i.e.,* VoNR, eMBB, uRLLC, mMTC) hosted by a single BS within a cell radius r_c (m) (see Fig. 3.1). Users, with varying QoS requirements, are randomly distributed within the cell. The total number of users is distributed across four services: VoNR, eMBB, uRLLC, and mMTC, with probabilities of 1, 2, 3, and 4, respectively. The highest number of users is assigned to the mMTC, as this service is designed to support massive connectivity. To accurately simulate the dynamic and highly fluctuating nature of the RAN slicing environment, we consider user mobility within each slice in the proposed system. For the sake of simplicity, we assume that users within the same slice exhibit similar movement patterns, including the velocity distribution and direction, similar to the model previously proposed in (Li *et al.*, 2020a). Table 3.5 provides an overview of the full network configurations used in our simulations. All numerical experiments are conducted using Python 3.11 with Tensor Flow on 11th Gen Intel(R) Core(TM) i9-11900 PC with 64 GB of RAM.

Table 3.6 Summary of baseline methods

Baseline	Algorithm / Method	Learning Architecture	Purpose of Comparison
SO-CoMDQN	LSTM-based DQN	Cooperative multi-agent	To assess the performance of alternative DRL algorithms compared to the proposed SO-CoMA2C scheme in the context of heterogeneous inter-RAN slicing.
SO-SA2C (Li *et al.*, 2020a)	LSTM-based A2C	Centralized single-agent	To evaluate our cooperative distributed scheme against a state-of-the-art single centralized DRL agent.
MR-HA	Traditional	Non-learning algorithm	To evaluate the effectiveness of adaptive DRL resource management compared to traditional hard slicing, which is a common static method in the literature.

3.5.2 Benchmarks

To evaluate the performance of the proposed SO-CoMA2C scheme, we consider three distinct baseline schemes. These include an alternative multi-agent DRL algorithm, a centralized learning framework, and a non-intelligent traditional method. Table 3.6 provides a detailed summary of each baseline, and further descriptions are given below.

- **SO-CoMDQN**: This scheme is specifically a SO inter-slicing scheme, and its design consists of multiple DQN agents to jointly optimize power and bandwidth. For a fair comparison, SO-CoMDQN was designed to be identical to SO-CoMA2C in terms of the training style and agent architecture. Specifically, each agent in the SO-CoMDQN integrates an LSTM layer, similar to the setup in SO-CoMA2C.
- **SO-SA2C**: This state-of-the-art scheme (Li *et al.*, 2020a), is referred to as SO single A2C (SO-SA2C) scheme. While the SO-SA2C scheme was originally designed using an LSTM-based A2C framework to manage the bandwidth among heterogeneous slices, we adapt it in this work so as to meet the requirements of our system model.

Table 3.7 Simulation parameters of scenarios A, B, and C

Scenario	Status of the State	Status of HW
A	Non-secure	Ideal HW
B	Non-secure	Non-ideal
C	Secure	Non-ideal

- **MR-HA**: This is a multiple radio resource based hard allocation (MR-HA) scheme, which is one of the traditional state-of-the-art approaches used to evenly distribute radio resources across heterogeneous slices on the inter-slicing level.

3.5.3 Impact of AES and HWIs on Model Performance

To analyze the impact of HWIs and secure states by AES algorithm on the proposed SO-CoMA2C scheme, we evaluate its convergence and performance under three scenarios—A, B, and C. The simulation parameters corresponding to each scenario are provided in Table 3.7, where in Scenario A, the proposed scheme is evaluated under ideal HW conditions with non-secure allocation. On the other hand, Scenario B examines the proposed scheme in the presence of HWIs and non-secure allocation. Finally, Scenario C evaluates the scheme under both secure allocation and HWIs. Identical hyperparameters and environmental conditions are used across all the scenarios, with the only variations being the presence of HWIs and state encryption. From Fig. 3.4, we observe that the proposed SO-CoMA2C scheme achieves excellent performance in scenario A, showing perfect and rapid convergence within 10,000 learning steps. In Scenario B, we found that the HWIs at the ULA and self-distortion significantly affected the performance of the proposed scheme. However, this does not affect convergence. In Scenario C, the encrypted state of the cooperative A2C-LSTM does not influence the performance or convergence of the proposed scheme. The similarity in performance and convergence patterns observed in the results of scenarios B and C show identical performances in maximizing spectral efficiency. Both implementations successfully converged after 8,000 learning steps and maintained stable performance throughout the remaining training period. This indicates that AES encryption

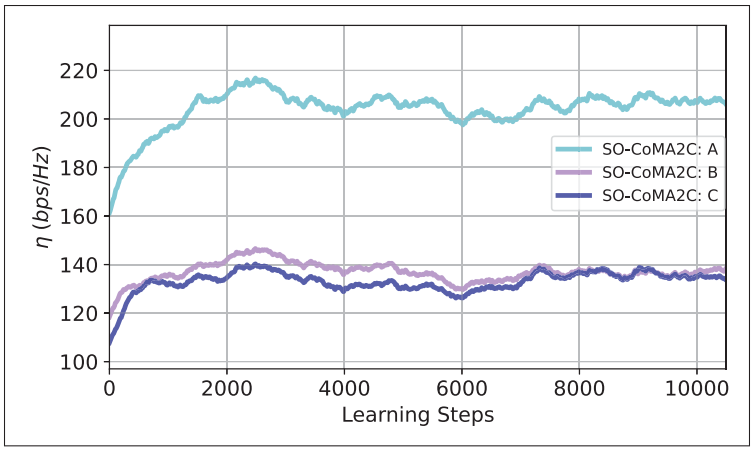

Figure 3.4 Performance of the proposed algorithm under Scenarios A, B, and C

preserves the mathematical structure and raw data of the state vector while rendering them unreadable to adversaries or any entity without the decryption key. This indicates that the implementation of the proposed approach ensures that the decryption process restores the exact original values without any loss of precision before passing the state to the DNNs of the SO-CoMA2C. From the perspective of DNNs, this means that it processes identical data in both implementations, and the encryption/decryption process is essentially transparent to the learning algorithm, resulting in nearly identical outcomes. Importantly, these results demonstrate that NS algorithms can incorporate AES protection with minimal performance penalties. This favorable performance-security tradeoff suggests that network operators can implement robust state encryption to protect against the threat scenarios outlined in Sections 3.2.1 and 3.4.2.2, with minimal performance degradation. Hence, we can confidently argue that the proposed scheme, integrated with AES, enhances secure interaction between the cooperative agents and the NS environment while maintaining the efficiency of the allocation process.

Table 3.8 Overheads of proposed scheme under HWIs

Overhead Metric	**Without Enc**	**With Enc**	**Increase (%)**
Training time (sec)	16277.80 s	16298.84 s	0.13%
Memory usage (MB)	690.02 MB	705.84 MB	2.3%

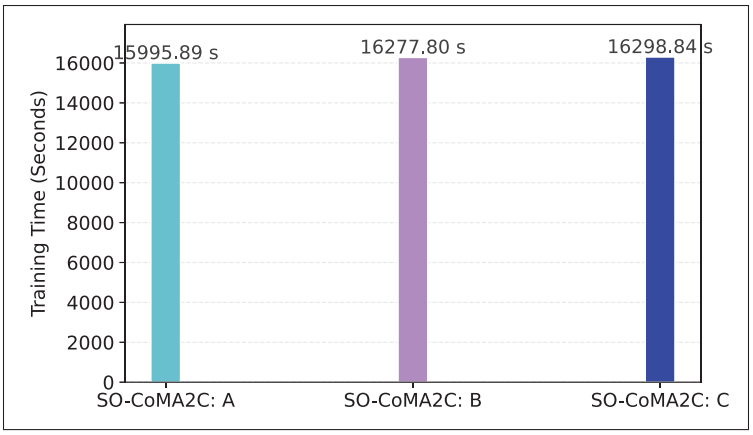

Figure 3.5 Training time of the proposed algorithm under Scenarios A, B, and C

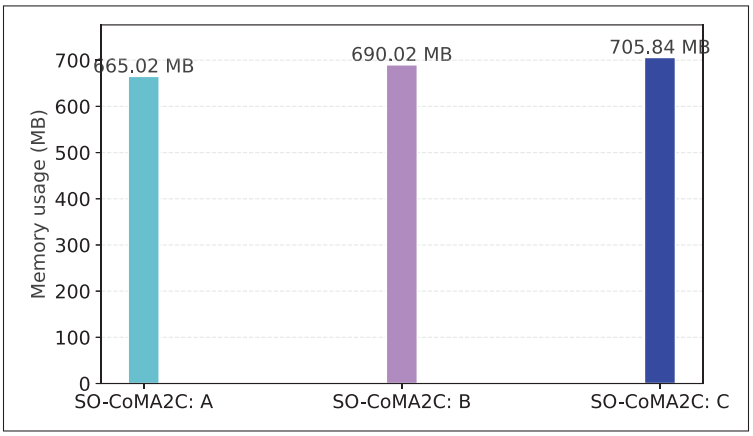

Figure 3.6 Memory usage of the proposed scheme under Scenarios A, B, and C

3.5.4 Impact of AES and HWIs on Computational Overhead

To further evaluate the impact of integrating AES into the proposed scheme in terms of overhead, this section analyzes the training time and memory usage across three different scenarios. Figs. 3.5 and 3.6 illustrate the the related results, respectively. It can be observed that when the system operates based on scenario A, the training time is the shortest and the memory usage is the lowest. As in Scenario A, the proposed scheme runs under optimal conditions with no external distortions or encryption overhead. On the other hand, when the proposed algorithm operates

under Scenario B, the training time and memory usage increase compared with Scenario A. This indicates that the distortions introduce additional processing overhead, causing a noticeable delay in the overall training time, although this scenario does not involve any encryption or decryption operations. In Scenario C, the training time and memory usage increased among the three scenarios. This means that the secure state introduces extra computational overhead to ensure data protection and integrity owing to the encryption and decryption operations. The results show that HWIs alone (non-secure state) increase training time. However, the most substantial delay and memory consumption occur when both the HWIs and security measures are involved. As shown in Table 3.8, integrating AES encryption into the SO-CoMA2C scheme led to only a 0.13% increase in training time and a 2.3% increase in memory usage. This demonstrates that the enhanced security comes at a negligible cost, preserving the overall efficiency of the proposed scheme.

3.5.5 Impact of AES with Various Key Sizes on Model Performance

Our experiment is conducted primarily using AES-128 key size. However, to examine the impact of various key sizes on the performance and convergence of the proposed scheme, we evaluated it using 192-bit and 256-bit key sizes while maintaining the CBC mode. From Fig. 3.7, it can be observed that a variety of key sizes did not significantly impact the performance of the proposed scheme. However, Fig. 3.8 illustrates that memory usage increases linearly with key size. This is because a larger key size corresponds to a higher level of security, which in turn demands more rounds of transformation. For example, AES-128 requires 10 rounds, AES-192 needs 12 rounds, and AES-256 requires 14 rounds (Groen *et al.*, 2024a). Although AES-256 offers both enhanced security and better performance, our experiment demonstrates that there is a trade-off between the security level and the associated overhead. Therefore, selection of an appropriate key size in NS systems depends on the desired level of security and acceptable resource constraints.

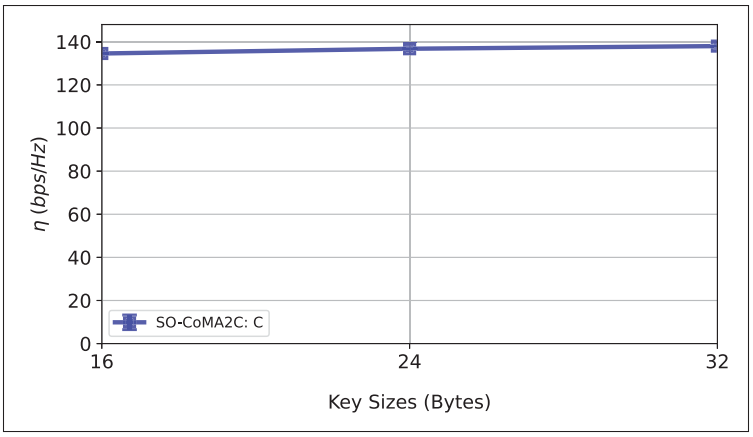

Figure 3.7 Performance of the proposed algorithm under various AES key sizes

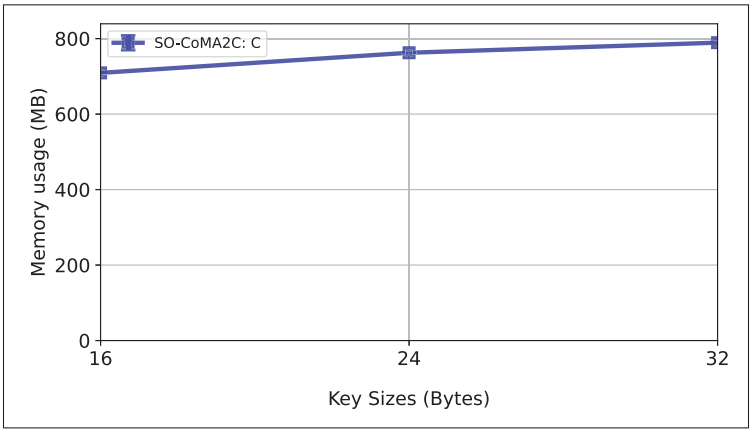

Figure 3.8 Memory usage of the proposed algorithm under various AES key sizes

3.5.6 Comparative Analysis with Baseline Methods

Figure. 3.9 shows that the proposed SO-CoMA2C outperforms the benchmark schemes in terms of maximizing spectral efficiency under both ideal and non-ideal HW conditions. This finding highlights effectiveness of the management strategy proposed in SO-CoMA2C scheme, which assigns power allocation management to one agent and bandwidth allocation to another, thereby enabling cooperative teamwork to achieve shared objectives. This approach efficiently manages resources and more effectively addresses action-space challenges than alternative

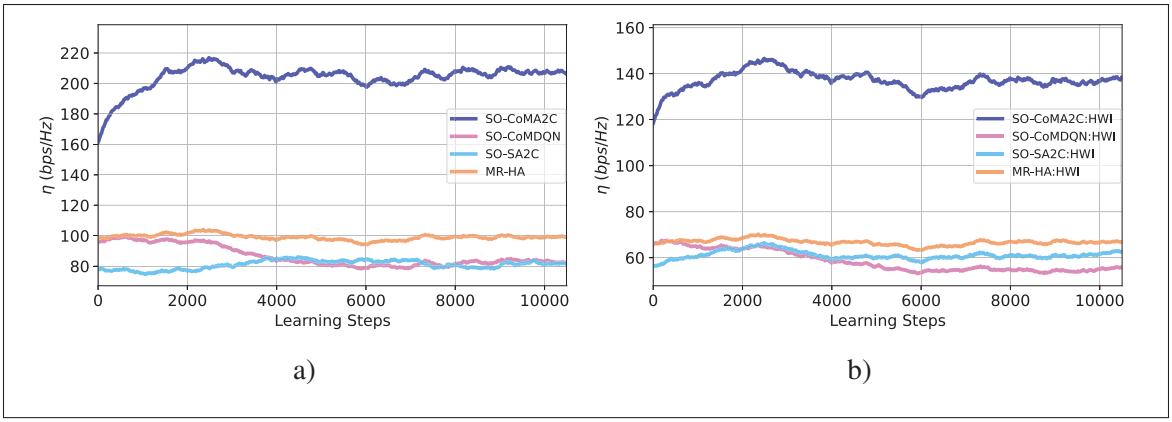

Figure 3.9 Comparison of spectral efficiency with benchmarks schemes under (a) ideal HW and secure state; (b) non-ideal HW and secure state

schemes. MR-HA scheme, on the other hand, achieves the second-best performance owing to its limited action space, which consists of three equal actions based on the number of slices in the system. However, its performance remains unaffected by traffic load fluctuations or the dynamic nature of RAN slices caused by user mobility, as this allocation strategy does not adhere to the MDP framework. Furthermore, MR-HA's inability to learn from interactions or adapt to the challenges of a RAN environment makes it unsuitable for inter-slicing management in a ZTN environment. Its reliance on a predefined set of equal actions, regardless of the traffic load on each slice, could lead to significant resource wastage. The proposed SO-CoMA2C scheme also outperforms the SO-CoMDQN scheme, despite their similar designs. Meanwhile, the SO-SA2C scheme demonstrates poor performance, highlighting the limitations of single-agent approaches in managing multi-radio resources and handling high traffic fluctuations across admitted slices. Finally, while non-ideal HW conditions significantly affect performance of all allocation schemes, the proposed SO-CoMA2C scheme still outperforms the others, which further demonstrates its robustness and effectiveness. Table 3.9 quantifies the performance gaps between SO-CoMA2C and benchmark schemes under ideal and non-ideal HW conditions

Table 3.9 Performance gaps of benchmark schemes relative to SO-CoMA2C under ideal and non-ideal HW

Scheme	η (Ideal HW)	PG (Ideal) (%)	η (Non-ideal HW)	PG (Non-ideal) (%)
SO-CoMA2C	206 bps/HZ	-	138 bps/HZ	-
SO-CoMDQN	81.9 bps/HZ	60.24%	56 bps/HZ	59.42%
SO-SA2C	81 bps/HZ	60.67%	62 bps/HZ	55%
MR-HA	100 bps/HZ	51.45%	67 bps/HZ	51.44%

* PG: Performance Gap.

3.5.7 Convergence Performance of the Proposed Scheme

Generally, a DRL agent aims to intelligently learn an optimal policy that maximizes the system reward or minimizes the system loss function (Yang & Xie, 2020). Therefore, the convergence of the proposed SO-CoMA2C scheme is evaluated using two key indicators: the training loss (Fig. 3.10) and the average reward (Fig. 3.11).

Figure 3.10 shows the convergence of the proposed scheme, measured by the rate at which the actor and critic loss functions decrease over time during the training of the SO-CoMA2C agents (e.g., Eq. (3.24) and Eq. (3.25)). As observed in Fig. 3.10, the losses of both the actor and critic networks decrease as training progresses. Specifically, a reduction in the average actor loss indicates that the actor networks are learning to take actions that maximize global reward, as shown in Fig. 3.11. Meanwhile, a decrease in the average critic loss indicates that the critic networks are becoming more accurate in evaluating those actions and gradually approaching the optimal Q-network. These trends demonstrate the stability and learning effectiveness of the proposed scheme, even in the presence of various confounding factors such as user mobility, HWIs, and fluctuating traffic loads.

Figure 3.11 illustrates the cumulative average rewards and convergence behavior of the proposed secure SO-CoMA2C scheme under both ideal and non-ideal HWI conditions. As the number of training iterations increases, the cumulative reward also improves, indicating that the proposed scheme progressively learns to allocate resources more efficiently and reaches a stable

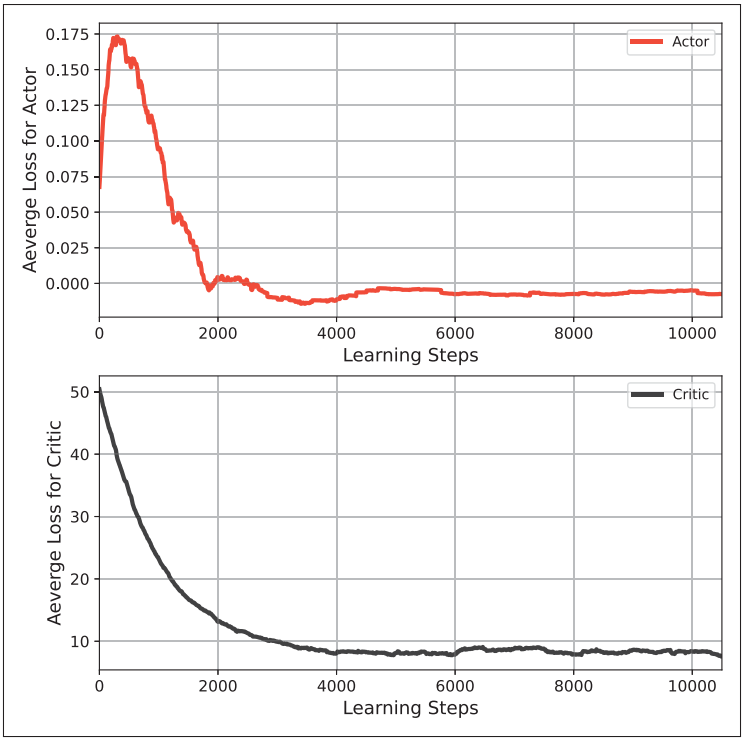

Figure 3.10 Average training loss of actor and critic networks

performance level. Under ideal conditions, the scheme begins to exploit better actions after approximately 8,000 training steps, suggesting that it has gained sufficient experience. In contrast, the average reward is noticeably affected by HWIs. Nevertheless, despite the distortions introduced by HWIs, the proposed scheme continues to learn and optimize the total reward, showing no significant degradation in the learning process or convergence behavior.

3.5.8 Performance Evaluation of Satisfying the SSR

Figure. 3.12a shows that all allocation schemes achieve comparable performance in satisfying the average SSR for VoNR, eMBB, and uRLLC under ideal HW conditions. However, under non-ideal HW conditions, the performance of all schemes decreases (see Fig. 3.12b). In particular, HWIs affect eMBB, where significant degradation can be observe across all management schemes. Despite this, the proposed scheme outperforms the others, achieving an average SSR

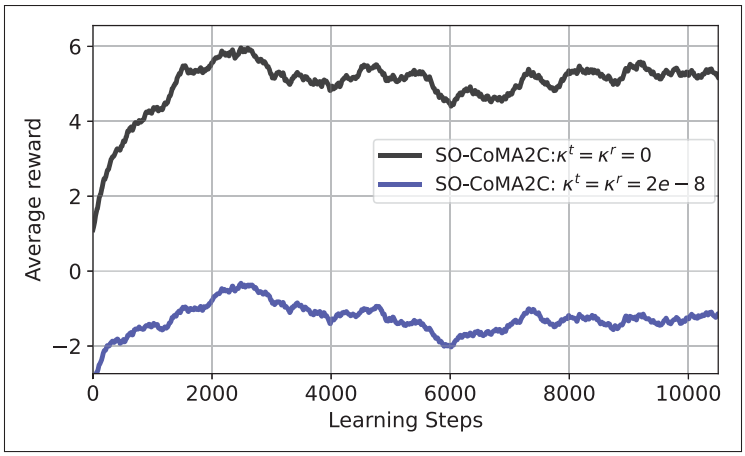

Figure 3.11 Average reward over the training under ideal and non-ideal HW

of 0.90 eMBB, whereas other DRL schemes fail to exceed 0.62. According to the achieved SSRs under HWIs, the proposed scheme effectively learns the optimal policy and achieves near-optimal QoS for all admitted services without sacrificing the performance of any service. However, although MR-HA demonstrates good performance, it is not a viable solution for 6G and beyond networks. Its major drawbacks are its lack of adaptability and self-learning capabilities. MR-HA scheme relies on the equal allocation of resources among all slices, regardless of their actual demand (traffic load). This means that, even if a particular slice experiences low traffic load and requires fewer resources, it receives the same allocation as slices under high demand. Consequently, resources that can be better used by high-demand slices are wasted on low-demand slices. This inefficiency can lead to performance issues, where slices that truly need more resources may not receive enough, which adversely affects overall system effectiveness. To achieve dynamic and efficient resource management required for future networks, more sophisticated and adaptive methods would be needed.

3.5.9 Impact of Varying HWIs on the Proposed Scheme

Increasing levels of HWIs lead to a noticeable decline in system performance. As illustrated in Fig. 3.13, the spectral efficiency of the system decreases with an increase of the level of HWIs.

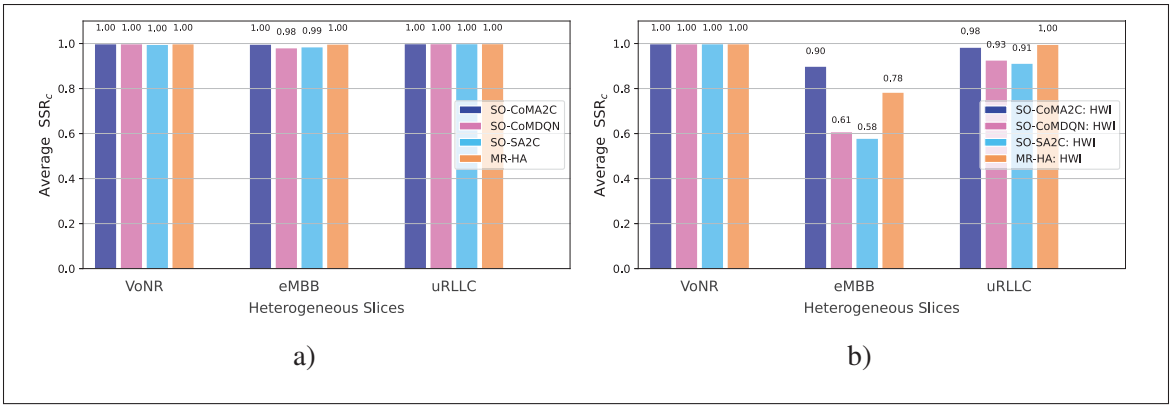

Figure 3.12 Average SSR of heterogeneous slices across different allocation schemes under: (a) ideal HW; (b) non-ideal HW

Similarly, as can be seen in Fig. 3.14, SSR gradually declines with increasing distortion levels, particularly for eMBB and uRLLC services. This is so because higher distortion levels result in lower data rates for eMBB users, causing the system to transmit larger packets over multiple time slots, which, in turn, affects latency requirements of the slice. A similar effect was observed for uRLLC services, which have highly stringent latency requirements. Despite employing a multi-antenna system, the impact of HW quality remains significant, as there is a linear relationship between the distortion level and the transmit power of each antenna. The observed performance degradation caused by HWI can also be linked to the sensitivity of traditional algorithms, which are governed by the intra-slicing level. This inherent sensitivity reduces their effectiveness and, as a result, adversely affects the overall performance of the proposed inter-slicing scheme. Yet, despite these impacts, we observed that the proposed scheme achieves high performance in ensuring QoS for VoNR, as well as for eMBB and uRLLC, regardless of their stringent QoS constraints.

3.5.10 Impact of ULAs Size on the Proposed Scheme

To analyze the impact of the number of antennas on the overall system performance and SSR of each service, following [16, 32, 48, 64], we fix the κ^t and $\kappa^r = 2e - 8$ and vary the size of the ULA as shown in Fig. 3.15 and 3.16, respectively. In detail, Fig. 3.15 illustrates the impact of

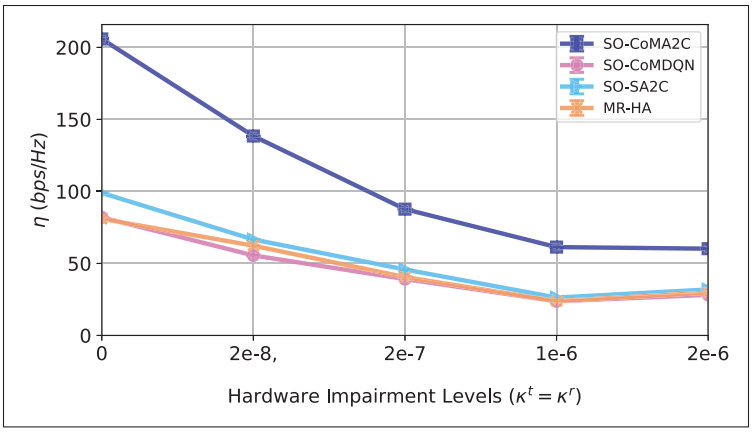

Figure 3.13 Impact of different levels of distortion on spectral efficiency

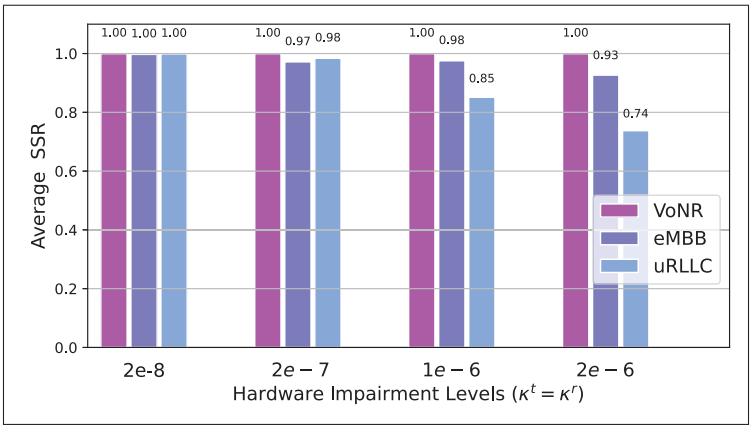

Figure 3.14 Impact of different levels of distortion on average SSR of the proposed scheme

different ULA sizes on spectral efficiency for various allocation schemes. The results of our analysis reveal a linear relationship between the number of antennas and improvements in spectral efficiency across all allocation schemes. Interestingly, the SO-CoMA2C scheme consistently demonstrates superior performance, highlighting its effectiveness in resource optimization. On the other hand, Fig. 3.16 shows that the average SSR for VoNR and eMBB remain stable across all ULA sizes. However, the average SSR for uRLLC degrads when the ULA size was less than 32. This suggests that a system with a ULA of $\mathbb{T} < 32$ will significantly affect the performance of uRLLC applications, because this type of service has stringent requirements

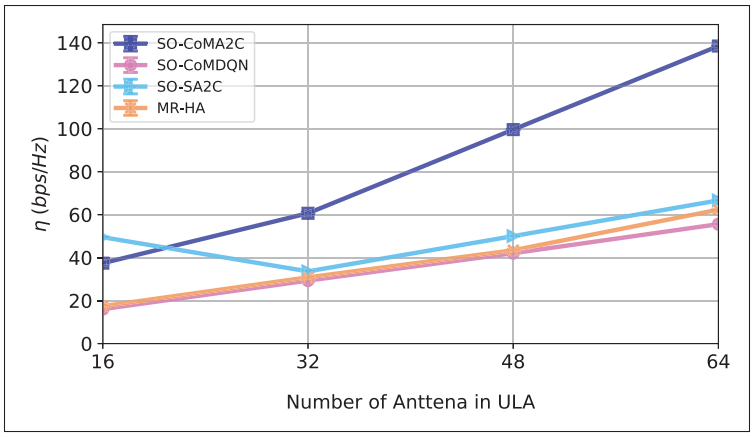

Figure 3.15 Spectral efficiency of the system with various allocation schemes vs. number of antennas under HWIs

(see Table 3.4). Overall, the SSR for all services could degrade further when increase the level of HWIs increase (see Fig. 3.14) while reduce the size of ULAs, as larger ULAs can boost the data rates which play crucial role in satisfying the SLAs of the NS. According to the results, VoNR and eMBB maintain good performance even with smaller ULA sizes. This is so because their QoS requirements, in terms of latency, are less demanding than uRLLC, which are more significantly affected by antenna reduction. This finding highlights the critical role of multiple-antenna systems in supporting future applications and services.

In order to reduce cost and complexity of future systems while ensuring the required QoS for various services and applications, a practical solution for the implementation of the proposed scheme in real-world deployments is to use high-quality HW only for uRLLC services. Alternatively, a more practical approach would be to ensure that the ULA is equipped with a large number of antennas (i.e., $\mathbb{T} > 64$), but with simpler HW resolution (κ^t and κ^r) $<= 2e - 6$. This could help to meet the stringent requirements. Finally, we believe that the proposed scheme can deliver better performance with large ULAs, even when simpler HW components are used. This approach can help to reduce the costs, power consumption, and size, thereby facilitating deployment of the proposed scheme in real-world scenarios.

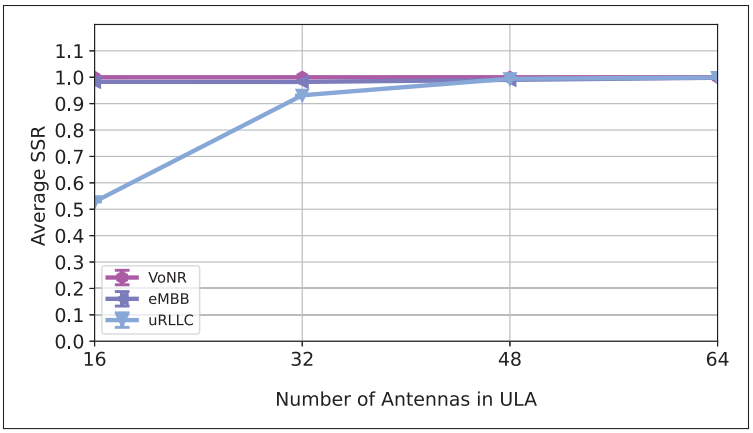

Figure 3.16 Comparison of average SSR for heterogeneous services vs. number of antenna under HWIs

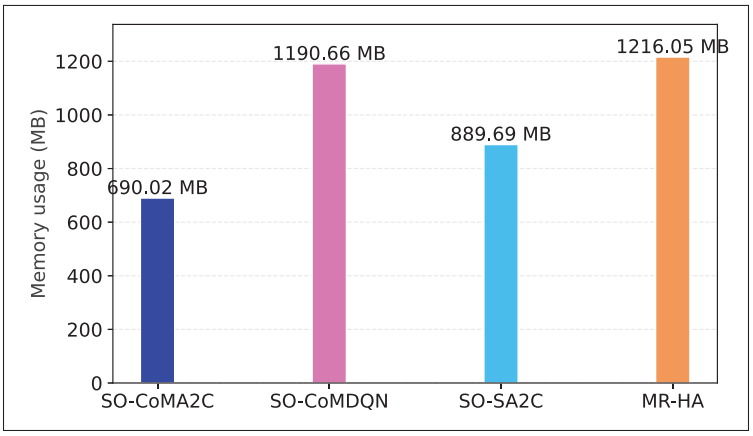

Figure 3.17 Memory consumption of the proposed SO-CoMA2C scheme vs. benchmark schemes under HWIs

3.5.11 Performance Evaluation in Terms of Memory Usage

To evaluate the efficiency of the proposed SO-CoMA2C scheme, we analyze its memory usage during the learning process under HWIs conditions and compare it with benchmark schemes. As shown in Fig. 3.17, the benchmark approaches consume significantly more memory than SO-CoMA2C. Their high memory consumption and computational complexity limit their ability to manage large state spaces effectively, making them less suitable for real-world applications

such as large-scale NS systems with multiple radio resources. In contrast, SO-CoMA2C achieves lower memory usage, even with the integration of the AES algorithm—without sacrificing performance, unlike the benchmark schemes which consume more memory without such integration. This efficient use of resources allows SO-CoMA2C to support more agents and handle more complex environments, highlighting its scalability in dynamic, resource-constrained network scenarios.

3.5.12 Performance Evaluation in Terms of Scalability

To assess the scalability of the proposed SO-CoMA2C scheme in terms of accommodating more users, we evaluate its performance under varying user loads, ranging from 400 to 1200 users. The number of users is selected as a key factor to measure scalability because it directly impacts network traffic, mobility density, and the complexity of resource management. An increase in users not only raises the traffic demand on each NS, but also makes the environment more dynamic and complex due to denser mobility. Fig. 3.18 shows that the sum spectral efficiency increases as the number of users grows. Notably, the proposed SO-CoMA2C scheme manages this growth efficiently, without negatively impacting the learning process or convergence behavior.

We further evaluate the scalability of the proposed scheme in terms of supporting additional NS. Specifically, we analyze how the same available system resources are distributed among five heterogeneous services by introducing mMTC and an additional slice, referred to as X-Slice (added for testing purposes), into the NS system. Fig. 3.19 illustrates the performance of the proposed scheme under different numbers of NS: $C = 3$ (VoNR, eMBB, and uRLLC), $C = 4$ (VoNR, eMBB, uRLLC, and mMTC), and $C = 5$ (VoNR, eMBB, uRLLC, mMTC, and X-Slice). Introducing additional slices, such as mMTC and X-Slice, increases system complexity due to their own QoS requirements, traffic load fluctuations, and user mobility characteristics. Our results reveal that while the addition of slices reduces the spectral efficiency of the proposed scheme due to the resource budget being shared among more slices, it does not affect the convergence of the the proposed scheme.

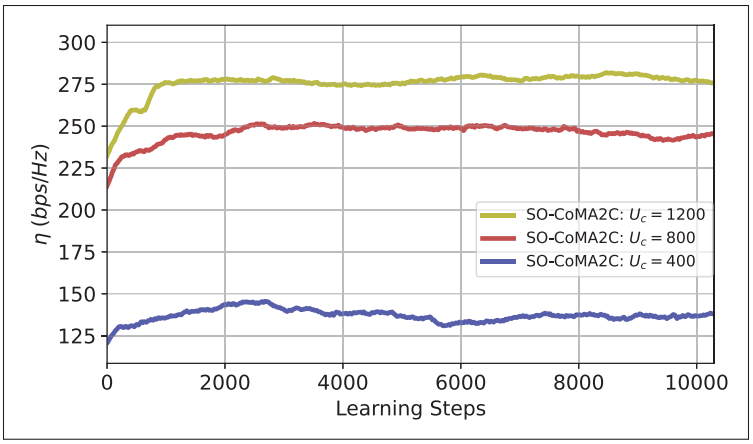

Figure 3.18 Performance of SO-CoMA2C for different numbers of users under HWIs

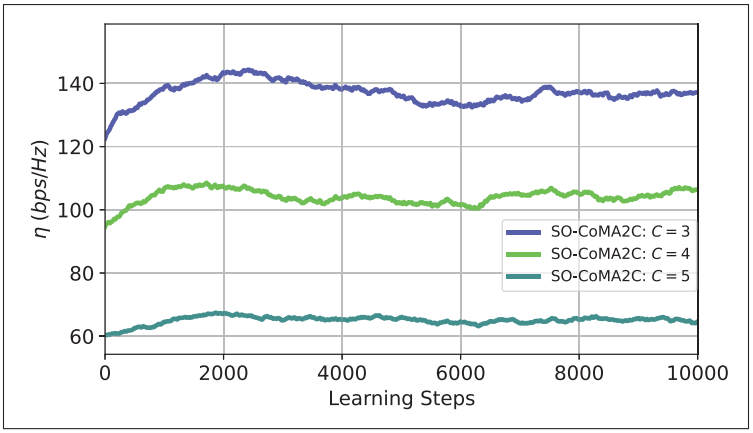

Figure 3.19 Sum spectral efficiency of the proposed scheme under HWIs for different numbers of slices: $C = 3$, $C = 4$, and $C = 5$, with corresponding action space sizes of $|A| = [2100, 3]$, $|A| = [2100, 4]$, and $|A| = [2520, 5]$, respectively

In addition to the scalability evaluation discussed earlier, we further assess the performance of the proposed SO-CoMA2C scheme under varying action space sizes. For this evaluation, U_c is set to 400, a ULA with 64 antennas under HWIs. We consider four different action space sizes: $|A| = 2100^4$, $|A| = 350^4$, $|A| = 150^4$, $|A| = 50^4$. As shown in Fig. 3.20, the best performance is achieved when the action space cardinality is set to $|A| = 150^4$ and

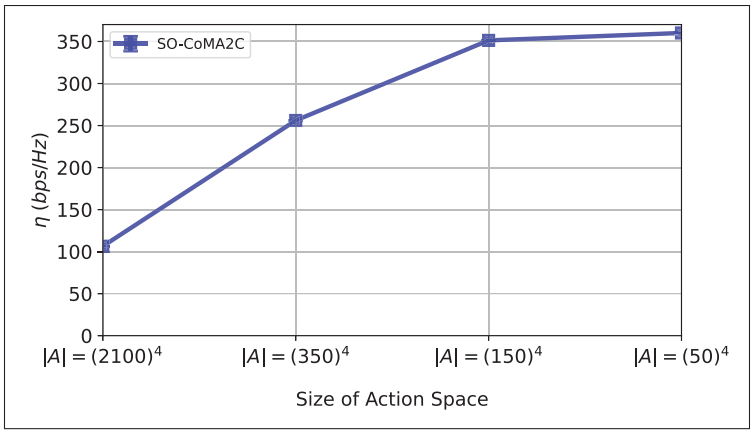

Figure 3.20 Sum of η versus the cardinality of the action space under HWIs

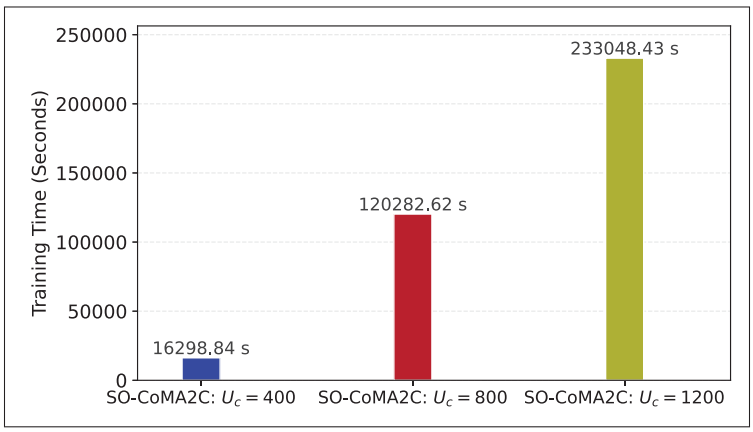

Figure 3.21 Training time versus number of users under HWIs

$|A| = 50^4$. This improvement is due to the fact that a decrease in action space size results from an increase in the radio resources allocated per action in the action combination. This allocation

Table 3.10 Overview of action space size

Ref.	No.Slice	Resources	Action space Size
(Zhang *et al.*, 2022a) (Filali *et al.*, 2022)	2	S-RR	Not declare
(Yan *et al.*, 2023) (Nagib *et al.*, 2024)(Shao *et al.*, 2021),(Li *et al.*, 2020a)	3	S-RR	Not declare
(Chang *et al.*, 2023)	3	S-RR	$\|A\| = [1128, 3]$
(Qi *et al.*, 2019)	3	S-RR	$\|A\| = 1176$
This work	**5**	**M-RR**	$\|A\| =$ **[2100, 4]**, $\|A\| =$ **[2520, 5]**

* S-RR:single radio resource, M-RR: multiple radio resources

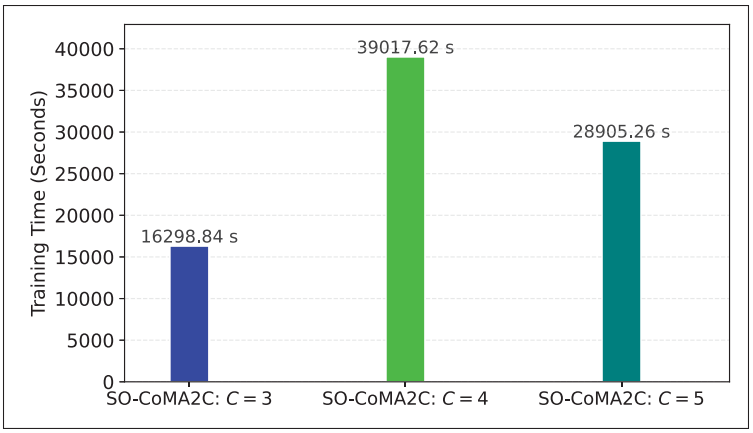

Figure 3.22 Training time versus number of slices under HWIs

is governed by the granularity values used in the system, specifically $\Delta = 1, 2, 3$, and 4 MHz. In other words, a larger granularity results in a smaller action space. Consequently, each slice reserves more resources per action, enabling the agent to explore a more compact action space and converge to an optimal policy more efficiently than when navigating a larger action space. However, our experiment was conducted with ($|A| = 2100^4$) and ($|A| = 2520^5$) to validate the proposed scheme under large and worst-case action-space expansion. As shown in Table 3.10, the proposed scheme operates in the largest action space as compared to the state-of-the-art schemes. This expansion in action space is attributed to its support for multiple radio resources across five heterogeneous slices. The results reveal that the proposed SO-CoMA2C scheme efficiently searches the action space and learns the optimal policy. Furthermore, we believe that the cooperative learning approach, which drives the proposed scheme, offers significant potential with regard to mitigating the state-action dimensionality challenge and thereby enhances the scalability of the proposed scheme. The idea of simulating the proposed scheme in a large action space not only tests the model's ability to scale with increasing slices or resource types, but also validates the robustness of the DRL algorithm in learning optimal or near-optimal policies under challenging conditions, such as HWIs and expanded action spaces. Unlike the traditional single-agent DRL approaches, our modular design uses separate A2C-based LSTM agents for each resource type. This decomposition simplifies the learning task for each agent, making the

system more scalable and adaptable as new resource types are introduced. Through cooperative learning, agents coordinate their decisions effectively, enabling the scheme to handle complex and large-scale network-slicing scenarios. As a result, the proposed architecture is well suited to support the demands of future 6G networks, where managing diverse resources across numerous slices is essential.

3.5.13 Impact of System Scalability on Training Time

To investigate the impact of system scalability on training time, two scenarios are considered. First, we evaluate the effect of increasing the number of users while keeping the number of slices fixed at $C = 3$. As shown in Fig. 3.21, the results demonstrate a linear relationship between the number of users and training time, indicating that the computational cost increases proportionally as the system scales with the number of users.

Second, we investigate how increasing the number of slices affects training time, while keeping the number of users fixed at 400. As the number of slices increases, the state-action space expands. As shown in Fig. 3.22, a notable increase in training time is observed at $C = 4$, when the mMTC slice is introduced. This increase in the training time at $C = 4$ can be attributed to multiple factors. First, the expanded state-action space inherently increases the complexity of learning. Second, this may be partially due to the characteristics of the mMTC slice and the nature of its traffic. The mMTC traffic is characterised by Poisson-distributed inter-arrival, resulting in bursty user traffic. These traffic characteristics lead to a more dynamic system behavior and frequent changes in resource demands, which complicate multi-agent learning coordination. Consequently, training time increases.

Notably, when the X-Slice is added at C = 5, training time decreases significantly to 28,905.26 seconds, despite the expanded state-action space. This unexpected improvement can be attributed to two complementary factors: (1) X-Slice traffic is more stable based on uniform inter-arrival times and (2) the number of users (e.g., 400) is redistributed across all slices whenever a new slice is added. For example, at $C = 5$, the slices are assigned user allocation probabilities of 1,

2, 3, 5, and 4 for VoNR, eMBB, uRLLC, mMTC, and X-Slice, respectively. Compared to the allocation at $C = 4$, where the mMTC had a higher number of users, this redistribution resulted in a reduced number of users assigned to the mMTC slice. This redistribution minimises the load on the mMTC slice, potentially mitigating the impact of the high traffic variance. Overall, these results show that training complexity is more sensitive to the nature of traffic fluctuations, such as Poisson-driven mMTC, than to the number of slices or dimensional expansion alone, highlighting the need for further exploration and consideration in future research.

3.5.14 Computational Complexity of the Proposed Scheme

The computational complexity of the proposed SO-CoMA2C scheme primarily depends on the training cost of the DNNs associated with each agent (Li *et al.*, 2023). In the proposed scheme, the actor and critic networks of each agent consist of J^a and H^c fully connected layers, respectively. Each j-th layer of the actor network contains N_j^a neurons. Therefore, the training complexity of the actor network is $O\left(\sum_{j=1}^{J^a-1} N_j^a N_{j+1}^a\right)$. Similarly, each h-th layer of the critic network contains N_h^c neurons. Thus, the training complexity of the critic network is $O\left(\sum_{h=1}^{H^c-1} N_h^c N_{h+1}^c\right)$. Therefore, the overall computational complexity of the proposed scheme is $O\left(\mathcal{G} \times \mathcal{F}^{\mathcal{I}} \left(\sum_{j=1}^{J^a-1} N_j^a N_{j+1}^a + \sum_{h=1}^{H^c-1} N_h^c N_{h+1}^c\right)\right)$, where, $\mathcal{F}^{\mathcal{I}}$ denotes the number of training iterations, and $\mathcal{G}$ represents the total number of resource agents involved in training. In our proposed scheme, increasing the number of agents $\mathcal{G}$ does not impact each agent's computational complexity. Consequently, the overall computational cost scales linearly with both the number of agents $\mathcal{G}$, and the number of training iterations $\mathcal{F}^{\mathcal{I}}$.

3.5.15 Synthesis of Findings and Prospects for Future Research

In this section, drawing on the experimental results presented and discussed above, we answer the research questions.

Regarding **RQ1**, findings demonstrates that the A2C-based cooperative learning framework is one of the most effective and promising methods for automating RRM in inter RAN slicing,

as it tackles the critical challenges of inter-RAN slicing. As concerns **RQ2**, we find that secure communication between the DRL algorithm and a heterogeneous NS environment can be effectively achieved through the integration of the AES algorithm into the DRL system. This integration ensures a secure exchange of encrypted data, enhancing both confidentiality and integrity of communications within NS. In addition, our research highlights this type of integration between the DRL and security algorithms, which can pave the way for new research on NS that would combine automation and security, both of which are essential requirements in 6G and beyond networks. With regard to **RQ3**, our results show that HWIs in ULAs, coupled with end device HWIs, can significantly impact performance of resource-allocation algorithms. These impairments can also affect the SSR or SLA of each slice, highlighting the need for further exploration of these effects to improve robustness and efficiency of resource-allocation strategies.

Finally, in the design of the proposed secure SO-CoMA2C scheme, we assume that cryptographic keys are securely distributed prior to secure communication over the E2 interface. However, this assumption must be carefully revisited in real-world deployments, particularly in ZT environments. In practice, key distribution faces several challenges. Without a robust key management and distribution policy, attackers may exploit vulnerabilities through key substitution attacks, where one key type is replaced with another, resulting in incorrect cryptographic operations or the intended disclosure of sensitive data. Additionally, adaptive adversaries capable of intercepting, replaying, or tampering with key exchanges may cause key desynchronization, which can degrade the learning performance or destabilize the entire system. These attacks could result in critical failures, such as mismatched encryption keys between agents and controllers, leading to communication breakdowns or misinformed decisions. As a future direction, integrating lightweight cryptographic protocols and leveraging decentralized trust mechanisms, such as blockchain-based key management or quantum key distribution, may support secure key exchanges. Ultimately, cryptographic key management and distribution should be approached with the same rigor as the selection of the right cryptographic algorithms for RAN slicing

domain. This will enhance the integration of cryptographic algorithms with the DRL framework, thereby improving the resilience of secure DRL frameworks in 6G network slicing.

3.6 Conclusion

In this study, we investigated the dynamic resource management problem of joint power and bandwidth allocation in heterogeneous inter-RAN slicing. In order to maximize long-term average spectral efficiency while satisfying the QoS requirements for each slice, we transformed the original optimization problem into a POMDP framework. Considering the high dimensionality of the action and state spaces, as well as potential vulnerabilities and attacks that could impact resource management schemes in ZTN, we proposed a secure SO-CoMA2C scheme. The proposed SO-CoMA2C scheme is based on cooperative MADRL framework and is designed to efficiently manage limited radio resources in response to fluctuating traffic loads of diverse services. Communication between the RIC and NS environments was secured through the integration of the AES algorithm into the DRL framework. Our experimental results revealed that SO-CoMA2C can effectively automate management of multi-radio resources in highly fluctuating and unpredictable environments, such as inter-RAN slicing, even in the presence of various sources of distortion, thus outperforming benchmark schemes. Our findings also highlighted that not all DRL algorithms are suitable for inter-RAN slicing. Based on this evidence, we conclude that the proposed SO-CoMA2C scheme is a promising approach that enables automation and can provide an intelligent and secure resource management strategy for inter-RAN slicing.

In future research, it would be necessary to carefully examine the impact of HW distortions on the QoS of future applications and services, particularly uRLLC and eMBB slices.

Acknowledgement

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), with application number RGPIN-2021-04013.

CHAPTER 4

HISO-COMA: HIERARCHICAL SELF-OPTIMISING FRAMEWORK FOR O-RAN SLICING USING COOPERATIVE MULTIPLE AGENT DEEP REINFORCEMENT LEARNING

Ohood Sabr[1], Georges Kaddoum[1], and Kuljeet Kaur[1,2,3]

[1] Department of Electrical Engineering, École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, QC H3C 1K3, Canada
[2]Canadian University Dubai, City Walk, Dubai, United Arab Emirates
[3] Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, 140401, Punjab, India

Paper published in *IEEE Open Journal of the Communications Society*, November 2025

4.1 Abstract

Network slicing (NS) is a cornerstone technology for sixth-generation (6G) networks, enabling the support of heterogeneous services with diverse quality-of-service (QoS) requirements. However, existing radio access network (RAN) slicing schemes often rely on single-level resource allocation, limiting their adaptability to the dynamic nature of RAN and the efficient use of limited radio resources. This leads to challenges in satisfying service-level agreements (SLAs). Moreover, effective hierarchical slicing that operates under fluctuating traffic loads, and hardware impairments for multiple antenna systems remains a challenge. To address these issues, we propose a hierarchical self-optimization framework aimed at maximizing both the long-term QoS and the spectral efficiency. Specifically, the proposed framework consists of two slicing management schemes: a cooperative multiple actor-critic (CoMA2C) scheme to manage the power and bandwidth among heterogeneous slices on a large scale. Concurrently, a multi-agent deep Q-network (MADQN) scheme manages the power and beamforming for active users within each slice on a small time scale, accounting for hardware impairments, user mobility, traffic fluctuations, and channel variations. The DQN and A2C algorithms are employed in the design of the proposed schemes owing to their proven effectiveness in real-time decision-making in dynamic environments. Furthermore, a promising scheme based on rate-splitting multiple access (RSMA) is investigated for heterogeneous services. Simulation

results showcase the effectiveness of our proposed framework, demonstrating its ability to satisfy SLAs for heterogeneous services while reducing network overhead and outperforming existing state-of-the-art approaches.

4.2 Introduction

Future networks are anticipated in a completely autonomous manner, eliminating the need for human intervention (Yang *et al.*, 2024b). In this context, zero-touch networks (ZTNs) represent a state-of-the-art paradigm shift towards completely automated and intelligent network management. ZTNs utilize machine learning (ML) and artificial intelligence (AI) to improve operational effectiveness, facilitate smart decision-making, and guarantee efficient resource allocation (Yang *et al.*, 2024b). The networking and communication scientific community anticipates that AI/ML approaches will play a critical role in fully automating the management and orchestration of the sixth generation (6G) of mobile networks. Specifically, deep reinforcement learning (DRL) algorithms are known for their ability to automate and optimize complex sequential decision-making tasks, effectively addressing challenging NP-hard problems by interacting with the environment without requiring prior knowledge about the system (Yang *et al.*, 2024b; Rezazadeh, Chergui & Verikoukis, 2021b). Generally, ZTNs consist of four essential functional classes: self-configuration, self-optimization, self-protection, and self-healing. These classes work together seamlessly to achieve complete automation and provide a fully zero-touch (ZT) operational environment. The capabilities of ZTNs are underpinned by a set of enabling technologies powered by AI, among which NS stands out as a key component (Sabr *et al.*, 2025a).

NS is a promising technology that enables the creation of isolated virtual logical networks on top of a physical operator network(Setayesh *et al.*, 2022). AI-driven slicing is envisioned as a viable solution for automating demand-aware resource management and orchestration (MANO) as well as enhancing the capabilities of heterogeneous beyond 5G (B5G) communication systems (Rezazadeh *et al.*, 2021b). However, several concerns surrounding NS in next generation networks remain yet unresolved. Among these, inter- and intra-slice coordination as well as

dynamic resource allocation pose substantial issues. These include sharing radio resources per slice, managing the priority of slices/users, complex traffic management, and overloads (Debbabi *et al.*, 2022). Unlike legacy networks, NS requires resource management at two levels: inter- and intra-slice. The inter-slice level is responsible for managing resources across different slices, whereas the intra-slice level manages resources within each individual slice (Shao *et al.*, 2021). Managing radio resources at these two levels is a complex task, yet it is essential to ensure efficient resource utilization and lay the groundwork for complete ZT operations in radio access network (RAN). Despite extensive efforts in the RAN slicing domain, most available solutions focus exclusively on intra-slice (Sabr *et al.*, 2025a) or inter-slice (Sabr *et al.*, 2025b) management, with very few studies investigating both levels simultaneously.

Therefore, to achieve the vision of ZT NS, this study investigates hierarchical radio resource management (RRM) framework for RAN slicing domain in next-generation networks. The proposed framework facilitates efficient resource distribution among heterogeneous services with stringent and diverse quality of service (QoS) requirements. Specifically, it ensures that slices with light traffic loads avoid excessive resource allocation and waste, whereas slices with heavy traffic loads receive adequate resources to maintain a high QoS. The proposed framework also ensures that radio resources are managed and optimized within each service. By integrating inter- and intra-slice resource management and introducing strategies to mitigate overheads, this study contributes to more reliable, efficient, and robust RAN slicing, which is an essential requirement for future mobile networks operating in ZT environments.

4.2.1 Related works

The literature on RAN RRM typically falls into two main threads. **The first line** of research pertains to developing RRM algorithms to manage intra-slice radio resources. For instance, the algorithm proposed in (Zhao *et al.*, 2022) jointly optimizes enhanced mobile broadband (eMBB) and ultra-reliable low-latency communications (uRLLC) bandwidth and power allocation based on the Lyapunov Drift method. Another example involves the algorithms proposed in (Tang *et al.*, 2019a; Ginige *et al.*, 2020b; Slalmi, Chaibi, Saadane & Chehri, 2021a) to manage the

power and beamforming of eMBB and uRLLC in multiple-input single-output (MISO) systems based on iterative algorithms. These studies (Tang *et al.*, 2019a; Ginige *et al.*, 2020b; Slalmi *et al.*, 2021a) have relied on orthogonal multiple-access techniques to share resources among and within services. Recent studies have been conducted on intra-RAN slicing based on rate-splitting multiple access (RSMA) for single or multiple slices; RSMA is emerged as a crucial multiple access scheme for 6G that offering significant improvements in data rates by partially decoding interference while treating the remaining interference as noise (Tan, Si, Chen, Li & Lv, 2024). An example can be found in (Tan *et al.*, 2024), where the authors jointly optimized beamforming and rate using an iterative algorithm, and examined the application of RSMA for uRLLC in cell-free massive multi-input and multi-output (CF-mMIMO) systems. RSMA was applied in another relevant study (Taskou & Rasti, 2024), where the authors proposed a DRL algorithm to manage RB allocation and power control for eMBB and uRLLC in single-input single-output (SISO) systems. Furthermore, (Huang, Wong & Schober, 2023) proposed an algorithm to manage power and beamforming in MISO systems for virtual reality (VR) applications. This algorithm leverages RSMA and an intelligent reflecting surface to support VR applications. Further, the authors of (Dizdar, Mao, Xu, Zhu & Clerckx, 2021b) investigated the performance of RSMA for eMBB and uRLLC by optimizing beamforming through zero-forcing (ZF) precoders for private streams and a random beamformer for the common stream, along with the rate of the common message. In this context, the power assigned to private precoders is equally distributed among private streams, whereas the power allocated to common precoders is equally distributed among common streams. Finally, (Li, Zhang, Guo & Yuan, 2024) proposed an RSMA beamforming scheme for uRLLC in an MISO system, which was designed based on an iterative algorithm.

The second line of research focuses on developing RRM algorithms to manage radio resources among heterogeneous services at both inter- and intra-RAN slicing levels simultaneously. Examples of this include the algorithms proposed in (Nagib *et al.*, 2024; Qi *et al.*, 2019; Li *et al.*, 2020a; Hua *et al.*, 2020; Yan *et al.*, 2024; Shao *et al.*, 2021; Chang *et al.*, 2023; Yan *et al.*, 2023), where the authors designed algorithms based on a single DRL agent to manage the bandwidth

among heterogeneous slices (eMBB, uRLLC, and voice over new radio (VoNR)) based on the traffic demand of each service, while traditional algorithms were applied within each slice to distribute the allocated bandwidth among its users. Building on this approach, Sabr *et al.* (Sabr *et al.*, 2025b) proposed a scheme that jointly manages both power and bandwidth, providing a more integrated solution for resource allocation. These studies (*e.g.,* (Nagib *et al.*, 2024; Qi *et al.*, 2019; Li *et al.*, 2020a; Hua *et al.*, 2020; Yan *et al.*, 2024; Shao *et al.*, 2021; Chang *et al.*, 2023; Yan *et al.*, 2023; Sabr *et al.*, 2025b)) have relied on orthogonal frequency-division multiple access (OFDMA) to share resources among and within services. In addition, these algorithms target SISO systems, except for (Yan *et al.*, 2024; Sabr *et al.*, 2025b; Yan *et al.*, 2023), where the authors designed a similar algorithm for both MISO and MIMO systems. However, the strategy used in these studies (*e.g.,* (Nagib *et al.*, 2024; Qi *et al.*, 2019; Li *et al.*, 2020a; Hua *et al.*, 2020; Yan *et al.*, 2024; Shao *et al.*, 2021; Chang *et al.*, 2023; Yan *et al.*, 2023; Sabr *et al.*, 2025b)), applying DRL at the inter level and traditional algorithms at the intra level, is suboptimal for automating RAN slicing management. More specifically, it lacks synchronization in learning between the two heterogeneous methods, which hinders optimal resource efficiency. To address this gap, very few studies have investigated DRL for hierarchical RRM, to manage radio resources at the two slicing levels for eMBB and uRLLC in SISO systems. An example of this approach can be found in (Filali *et al.*, 2022), where the authors presented a two-time-scale RAN slicing algorithm to manage the bandwidth of eMBB and URLLC services. On a large time scale, the software-defined networking (SDN) controller assigns radio resources to gNodeBs. Each gNodeB then distributes its resources to the end users of the eMBB and URLLC slices. The proposed algorithm adopts OFDMA to avoid both inter-slice and intra-slice interference, with designs at both levels based on DRL algorithms. Another example of a multiple-level management for SISO systems is (Setayesh *et al.*, 2022), where the authors used DRL to manage power and bandwidth at the inter-slice level while applying deep learning (DL) to control resources at the intra-slice level for eMBB and URLLC. In this context, OFDMA was used to mitigate the interference. Similarly, (Mei *et al.*, 2021) proposed a two-layer control mechanism for eMBB and uRLLC based on DRL algorithms. The upper layer was designed as a slice configuration to set the guaranteed bit rate and maximum bit rate for the users in each slice

using the double deep Q-network (DQN) algorithm. The lower layer was designed to manage the power and bandwidth of users in both slices using the deep deterministic policy gradient (DDPG) algorithm.

4.2.2 Motivation

Although previous studies have explored RAN slicing RRM, a significant gap remains in the existing literature regarding the excessive overhead introduced by state-of-the-art algorithms that manage both the inter- and intra-slice levels (*e.g.,* (Sabr *et al.*, 2025b; Nagib *et al.*, 2024; Qi *et al.*, 2019; Li *et al.*, 2020a; Hua *et al.*, 2020; Yan *et al.*, 2024; Shao *et al.*, 2021; Chang *et al.*, 2023; Yan *et al.*, 2023; Setayesh *et al.*, 2022; Filali *et al.*, 2022)). These algorithms typically operate on large timescales, often every second, without assessing whether such adjustments are truly necessary. This fixed-time resource adjustment not only creates unnecessary overhead but also depletes valuable resources and reduces the overall network efficiency. This research gap is further compounded by the fact that existing algorithms (*e.g.,* (Nagib *et al.*, 2024; Qi *et al.*, 2019; Li *et al.*, 2020a; Hua *et al.*, 2020; Yan *et al.*, 2024; Shao *et al.*, 2021; Chang *et al.*, 2023; Yan *et al.*, 2023; Setayesh *et al.*, 2022; Filali *et al.*, 2022; Mei *et al.*, 2021)) were designed based on idealized assumptions, such as perfect hardware (HW) and interference-free conditions. While these assumptions simplify the analysis, they overlook the significant impact of hardware impairments (HWIs), particularly when using low-cost massive antennas such as uniform linear arrays (ULAs) (Björnson *et al.*, 2013). In real-world wireless communication systems, HWIs and interference are key confounding factors that can dramatically affect performance. Therefore, it is crucial to evaluate state-of-the-art schemes under realistic conditions, as many prior studies have overlooked these important limitations. Furthermore, the majority of the previous studies on multiple-level RRM have focused on SISO systems (*e.g.,*(Nagib *et al.*, 2024; Qi *et al.*, 2019; Li *et al.*, 2020a; Hua *et al.*, 2020; Shao *et al.*, 2021; Chang *et al.*, 2023; Setayesh *et al.*, 2022; Filali *et al.*, 2022)) without exploring how these algorithms can be adapted for multi-antenna systems, which are expected to dominate next-generation networks (Zhang, Sun, Rong, Yu & Lu, 2015). This limits the applicability of previous algorithms to next-generation wireless technologies,

which heavily rely on multi-antenna systems to enhance performance. Notably, prior DRL-based studies on inter-RAN slicing have adopted centralized single-DRL agent, leading to large observation spaces, slower convergence, and higher memory requirements (Liao, Shen, Wu & Feng, 2024), along with other limitations highlighted in (Dubey, Singh & Mishra, 2025) that make this approach unsuitable for handling NS. This approach will face scalability and training complexity challenges, particularly as the number of resources and radio slices increase. Although a few studies have considered the management of RAN at both levels, most concentrate on a single radio resource (bandwidth) (*e.g.,* (Shao *et al.*, 2021; Nagib *et al.*, 2024; Qi *et al.*, 2019; Li *et al.*, 2020a; Hua *et al.*, 2020; Filali *et al.*, 2022)). Furthermore, majority of existing studies have focused on optimizing resource allocation at the intra-slicing level (*e.g.,* (Sabr *et al.*, 2025a; Zhao *et al.*, 2022; Tang *et al.*, 2019a; Ginige *et al.*, 2020b; Slalmi *et al.*, 2021a; Tan *et al.*, 2024; Taskou & Rasti, 2024; Huang *et al.*, 2023; Dizdar *et al.*, 2021b; Li *et al.*, 2024)), whereas the inter-slicing resource budget remains fixed. This can lead to significant resource wastage, particularly in slices with low traffic demand, as resources are allocated regardless of the actual needs. Slices with high traffic may not receive the necessary resources, resulting in inefficiency. This fixed-budget approach highlights the need to optimize resources at both slicing levels simultaneously. Therefore, a more comprehensive approach is needed that considers both levels to enable ZT operations in future RAN architectures. Although there has been some research on RAN management using RSMA, which has shown significant potential to improve the performance of heterogeneous networks (Mao *et al.*, 2022), its application across both slicing levels remains unexplored. Therefore, to address the aforementioned issues, this study proposes a novel twin-timescale framework designed to tackle the complexities of multi-level RAN RRM, based on distributed, cooperative, multi-DRL agent. More specifically, the proposed framework adopts multiple actor-critic (A2C) algorithms to manage the inter-slice level, while multiple DQN algorithms are used to manage the intra-slice level. This framework contributes to the self-optimizing class of ZTNs, thereby laying the groundwork for ZT management in future networks.

4.2.3 Contributions

The key contributions of this study are summarized as follows:

- We propose a **hi**erarchical **s**elf-**o**ptimizing framework for managing heterogeneous network slices in the RAN domain, based on **co**operative **m**ultiple **a**gent DRL, referred to as HiSO-CoMA, which aims to maximize the long-term spectral efficiency and meet service level agreements satisfaction ratio (SSR) of diverse services. The proposed framework adopts the RSMA scheme to support the coexistence of heterogeneous services. To the best of our knowledge, RSMA has not been previously applied in this context.
- To accommodate time-varying network conditions and diverse QoS requirements in terms of data rate and latency, while ensuring efficient use of limited radio resources and smooth synchronization between the management levels, the problem is formulated as a twin-timescale resource-management problem. Specifically, it consists of two management levels: inter-slice (on a large timescale) and intra-slice (on a small timescale).
- Based on the fluctuating traffic load of heterogeneous services, the inter-slice level of the proposed framework allocates multiple radio resources (e.g., power and bandwidth) among these services to minimize waste in the system's radio resources. Meanwhile, intra-slice level management performs fine-grained control by allocating power, adjusting bandwidth, and optimizing beam directivity for active users within each service.
- To solve the large timescale problem in line with real world deployments, we reformulate it as a partially observable Markov decision process (POMDP) and solve it using a distributed cooperative multiple A2C (CoMA2C) scheme. Meanwhile, the small timescale problem is reformulated as a Markov decision process (MDP) and addressed using a distributed multi-agent DQN (MADQN) scheme.
- To address the overhead issues in state-of-the-art algorithms discussed in Section 4.2.2, we propose a novel mitigation strategy that enables the proposed framework to adjust resources at the inter-RAN slicing level, only when significant changes occur in the slice traffic load. This approach alleviates communication overhead, reduces the complexity of coordination

between the inter-slice (COMA2C) and intra-slice (MADQN) control policies, and ensuring efficient and real-time resource management.

- To examine the impact of unwanted noise from non-ideal HW on the heterogeneous QoS of future applications, we consider the effects of HWIs at the transmitter and self-distortion at the receiver in the proposed system model. Specifically, we investigate how HWIs influence the learning process of the proposed framework and affect its training time, both of which are critical for achieving reliable system performance.
- To extensively evaluate the proposed scheme in a heterogeneous inter- and intra-RAN slicing environment, we conducted a comprehensive set of simulations while taking into consideration user mobility, time-varying channels, fluctuating traffic loads, and HWIs. This evaluation included comparisons with both state-of-the-art (Sabr *et al.*, 2025b) and traditional algorithms to test the adaptability and reliability of the proposed framework.

4.2.4 Organization

The remainder of this chapter is organized as follows. Section 4.3 details the system model and problem formulation. Section 4.4 elaborates on the proposed hierarchical slef-optimization framework. Numerical evaluations and discussion are provided in Section 4.5. Finally, Section 4.6 concludes this work and outlines the future research directions.

4.3 System Model and Problem Formulation

This section elaborates on the system model, including the NS model, communication channel, and multiple access techniques. Finally, the objectives of the proposed optimization problem are defined.

4.3.1 Radio Slicing Scenario

We consider a heterogeneous inter- and intra-NS scenario in open RAN (O-RAN) architecture, where a single base station (BS) equipped with $N_T > 1$ transmit antennas serves users across

a set of services denoted by $\mathcal{S} = \{1, 2, \ldots, S\}$. The BS is remotely managed by RAN intelligent controller (RIC) via the E2 interface (refer to Fig. 4.1). The E2 interface enables seamless communication between the RIC and RAN components, thus enabling the exchange of data/information and coordinated management (M, Sadashivappa & Palepu, 2023). The RIC is a crucial component of O-RAN architecture and a key enabler of intelligent RRM and optimization. The RIC consists of the following two distinct components: (i) non-real-time (RT) RIC, which performs non-real-time tasks (usually beyond 1 s), and (ii) near-RT RIC, which runs software applications, known as xApps, and addresses real-time control and optimization, which is essential for making quick decisions within the RAN (usually from 10 ms to 1 s) (Ngo *et al.*, 2024b). For the sake of simplicity, we focus on three slices as a case study: VoNR, eMBB and uRLLC, denoted by S^n, S^m and S^r, respectively. The sets of users in slices S^n, S^m and S^r are denoted by $\mathcal{N} = \{1, 2, n, \ldots, N\}$, $\mathcal{M} = \{1, 2, m, \ldots, M\}$, and $\mathcal{R} = \{1, 2, r, \ldots, R\}$, respectively, where N, M and R represent the total numbers of VoNR, eMBB, and uRLLC users. All users are assumed to be equipped with a single antenna to ensure low hardware complexity. Therefore, the set of users in the system can be represented as $\mathcal{U}_s = \{1, 2, \ldots, U_s\}$. The total number of users is $U_s = N + M + R$, where the user belonging to slice s is denoted by $u_s \in \{n, m, r\}$. We assume that each user belongs to only one NS, based on the required services. In line with the heterogeneous requirements of future networks, each slice has its own QoS requirement based on the service level agreement (SLA) between the group of users and the service provider, as explained in Table. 4.1.

In the proposed scenario, we assume that each user u_s sends a request to the service of one of the admitted slices. Hence, each slice s in the system receives a set of requests, denoted by $\mathcal{Q}_s = \{1, 2, \ldots, Q_s\}$, where Q_s is the number of requests made by users belonging to slice s. Furthermore, we assume that the requests are sent by authorized users and approved by their corresponding slices. In response to user requests (q_s) for a certain service, the BS provides users with data traffic for the requested service. The data traffic of each slice is represented by Ω_s, and the system's total traffic demand, $\mathbf{\Omega}_{\text{total}}$, can be defined as $\mathbf{\Omega}_{\text{total}}[t] = \sum_{s=1}^{S} \Omega_s[t]$. We adopt a descriptive traffic model to mimic Ω_s for each NS, where the traffic model for users in

each slice is defined by specific inter-arrival time distributions, packet sizes, and buffer settings (Mei *et al.*, 2021; Li *et al.*, 2020a). The designed traffic model represents the traffic for each user as a data packet. Let Φ_{u_s} represent the set of data packets sent from the BS to user u_s. In this set, ψ_{u_s} denotes a single data packet where $\psi_{u_s} \in \Phi_{u_s}$. Therefore, the data traffic for each NS at time slot t is given by the following:

$$\Omega_s[t] = \sum_{u_s \in \mathcal{U}_s} \Phi_{u_s}; \forall s \in \mathcal{S}. \tag{4.1}$$

Typically, the data traffic originating from the core layer is sent to the BS, where it is first directed to a buffer assigned to each user type based on the requested service. Once the data reaches the BS, it is transmitted to users within their respective NSs. The first-come-first-serve (FCFS) strategy is then followed to deliver the data (Mei *et al.*, 2021). Hence, we assume that each user at the BS has a queue buffer with a configurable packet limit, denoted as Ξ_{u_s}. In the proposed system, user u_s is considered idle if its queue buffer is empty; otherwise, u_s is classified as active.

The QoS of each NS is typically evaluated using metrics that are essential for assessing adherence to SLAs, such as data rate, packet latency, and transmission reliability (Zhang *et al.*, 2022a). In this study, the main indicators used to assess each slice's SLA compliance are data rate and packet delay. Therefore, we establish thresholds for the maximum permitted latency (L_s^{Max}) and the minimum data rate ($\mathfrak{R}_s^{\text{Min}}$) for the users of each slice, as listed in Table 4.1. Thus, we define a binary variable $d_{\psi_{u_s}} \in \{0, 1\}$, where $d_{\psi_{u_s}} = 1$ implies that user u_s in slice s successfully received a packet $\psi_{u_s} \in \Phi_{u_s}$, yielding

Table 4.1 SLAs for Admitted Slices

No.	Slice	$\mathfrak{R}_s^{\text{Min}}$	L_s^{Max}	SSR_s^{Th} (Li *et al.*, 2020a)
1	VoNR	51kbps (Hua *et al.*, 2020)	10ms (Hua *et al.*, 2020)	$\geq 95\%$
2	eMBB	15 Mbps	10ms (Hua *et al.*, 2020)	$\geq 95\%$
3	uRLLC	10Mbps (Hua *et al.*, 2020)	3ms	$\geq 95\%$

$$d_{\psi_{u_s}} = \begin{cases} 1, & \text{if } \Re_{u_s} \geq \Re_s^{\text{Min}} \;\&\; l_{\psi_{u_s}} \leq L_s^{\text{Max}} \\ 0, & \text{otherwise} \end{cases}, \tag{4.2}$$

where $\Re_{u_s}$ denotes the instantaneous data rate for user u_s and $l_{\psi_{u_s}}$ is the transmission delay of packet ψ_{u_s} in slice s.

In general, in wireless communications, packets experience various sources of delay, such as propagation delays, receiver processing delay, queuing delay at the BS, and the time required for additional retransmissions (Anand & de Veciana, 2018). In this study, we examine two of these sources: queuing time (D_{Queuing}) and propagation time (D_{Trans}). The former is affected by the scheduling policy and is related to the waiting time for packets in the queue. In contrast, the latter depends on the instantaneous data rate and reflects how quickly the data is transmitted over the network. Therefore, $l_{\psi_{u_s}}$ is the sum of these two elements, yielding

$$l_{\psi_{u_s}} = D_{\text{Trans}} + D_{\text{Queuing}}, \tag{4.3}$$

In the proposed system, when $l_{\psi_{u_s}}$ exceeds the defined L_s^{Max}, the NS drops the packet according to standard network protocols (Zhang *et al.*, 2022a). From an empirical perspective, an effective resource-management strategy must guarantee the QoS of each NS. This means maximizing the traffic's successful transmission ratio to improve network efficiency (Shao *et al.*, 2021). Therefore, we include the SSR of each slice s (SSR_s) as a QoS measuring criteria. SSR_s is defined as the percentage of successfully transmitted and received packets, which can be written as

$$\text{SSR}_s[t] = \frac{\sum_{u_s \in \mathcal{U}_s} \sum_{\psi_{u_s} \in \Phi_{u_s}} d_{\psi_{u_s}}}{\sum_{u_s \in \mathcal{U}_s} \left|\Phi_{u_s}\right|}; \forall s \in \mathcal{S} \tag{4.4}$$

where $|\Phi_{u_s}|$ represents the total number of packets sent from the BS to user u_s in slice s.

To ensure fairness among the users, we evaluate how well the system satisfies the QoS requirements for each user in a particular slice by measuring the SSR at the user level as follows:

$$\mathrm{SSR}_{u_s}[t] = \frac{\sum_{\psi_{u_s} \in \Phi_{u_s}} d_{\psi_{u_s}}}{|\Phi_{u_s}|}; \forall u_s \in \mathcal{U}_s \tag{4.5}$$

4.3.2 Channel Model

To mimic the dynamics of the downlink multi-slice heterogeneous MISO channel and simulate the system under more realistic conditions, we adopt a flat-and-block fading channel (Ge *et al.*, 2020). Accordingly, the downlink channel vector ($\mathbf{h}_{u_s}$) between the BS and user u_s at time slot t is expressed as follows:

$$\mathbf{h}_{u_s}[t] = \sqrt{\frac{\beta_{u_s}}{L}}\mathbf{A}(N_T, \theta_{u_s}, \Delta)\mathbf{g}_{u_s};\ u_s \in \{n, m, r\}, \tag{4.6}$$

where β_{u_s} denotes the large-scale fading, which includes shadowing and path loss between the BS and user u_s in slice s. Here, the shadowing is modeled as a log-normal distribution with variance of σ_{sf}, and the path loss depends on the distance between the BS and user u_s in kilometers (ζ_{u_s}) as $120.9 + 37.6 \log_{10} \zeta_{u_s}$ dB. Moreover, the signal for user u_s propagates through L distinct paths. The matrix $\mathbf{A}(N_T, \theta_{u_s}, \Delta) \in \mathbb{C}^{N_T \times L}$ captures the antenna array response over the L paths as shown below:

$$\mathbf{A}(N_T, \theta_{u_s}, \Delta) = \begin{bmatrix} \alpha_1(N_T, \theta_1) & \dots & \alpha_L(N_T, \theta_L) \end{bmatrix} \in \mathbb{C}^{N_T \times L}, \tag{4.7}$$

where θ_{u_s} denotes the direction of departure (DoD) from the BS toward user u_s, while Δ represents angular spread. In Eq. (4.7), $\alpha_l(N_T, \theta_l) \in \mathbb{C}^{N_T \times 1}$ denotes the array response vector of the l^{th} path, given by the following:

$$\alpha_l(N_T, \theta_l) = \begin{bmatrix} 1, e^{j2\pi \frac{d^n}{\lambda} \cos\theta_2} & \dots & e^{j2\pi \frac{d^n}{\lambda}(N_T - 1)\cos\theta_l} \end{bmatrix}^T, \tag{4.8}$$

where λ denotes the wavelength of the downlink carrier wave, d^n denotes the distance between adjacent antennas, and θ_l is the DoD of the l^{th} path. Here, we assume that the DoDs of all the paths are uniformly distributed (Ge *et al.*, 2020). Finally, $\mathbf{g}_{u_s} \in \mathbb{C}^{L\times 1}$ denotes the small-scale fading vector from the BS to user u_s, which is modeled according to a first-order complex Gauss-Markov process, as discussed in (Wang, Sun, Xin, Liu & Xu, 2022). Due to the block-fading, the channel remains constant within each time slot, but changes independently from one slot to the another.

4.3.3 Multiple access techniques

To facilitate the coexistence of heterogeneous services, we adopt frequency division duplex (FDD) to provide inter-slice resource isolation and adaptively guarantee the QoS requirements of each NS, similar to (Tang *et al.*, 2019b). This approach can ensure slice isolation, which becomes increasingly critical as the number of services grows in future networks. Moreover, to enable resources sharing within each NS, we adopt RSMA, which is one of the most promising multiple access technique for 6G networks (Mao *et al.*, 2022). The main advantage of RSMA lies in its flexibility in managing interference, allowing it to be partially decoded and partially treated as noise (Dizdar *et al.*, 2021b). According to the RSMA strategy, the downlink messages intended for users are denoted as $G = \{G_1, \ldots, G_{u_s}, \ldots, G_{U_s}\}$, where G_{u_s} is divided into: common ($G^c_{u_s}$) and private ($G^p_{u_s}$) components. The common parts of all user messages within the same slice are combined and encoded into a common signal stream (s_c). On the other hand, the private parts $G^p_{u_s}$ are individually encoded into streams ($s_{u_s 1}, \ldots, s_{U_s}$). *BS* then sends the superimposed signal of its common stream s_c and private stream s_{u_s} simultaneously to the end users. Thus, the transmitted signal from BS to the users of slice s at time slot t using 1-layer downlink RSMA is given by (Mao *et al.*, 2022).

$$\mathbf{x}[t] = \mathbf{w}_c s_c + \sum_{u_s=1}^{U_s} \mathbf{w}_{u_s} s_{u_s};\ u_s \in \{n, m, r\}, \tag{4.9}$$

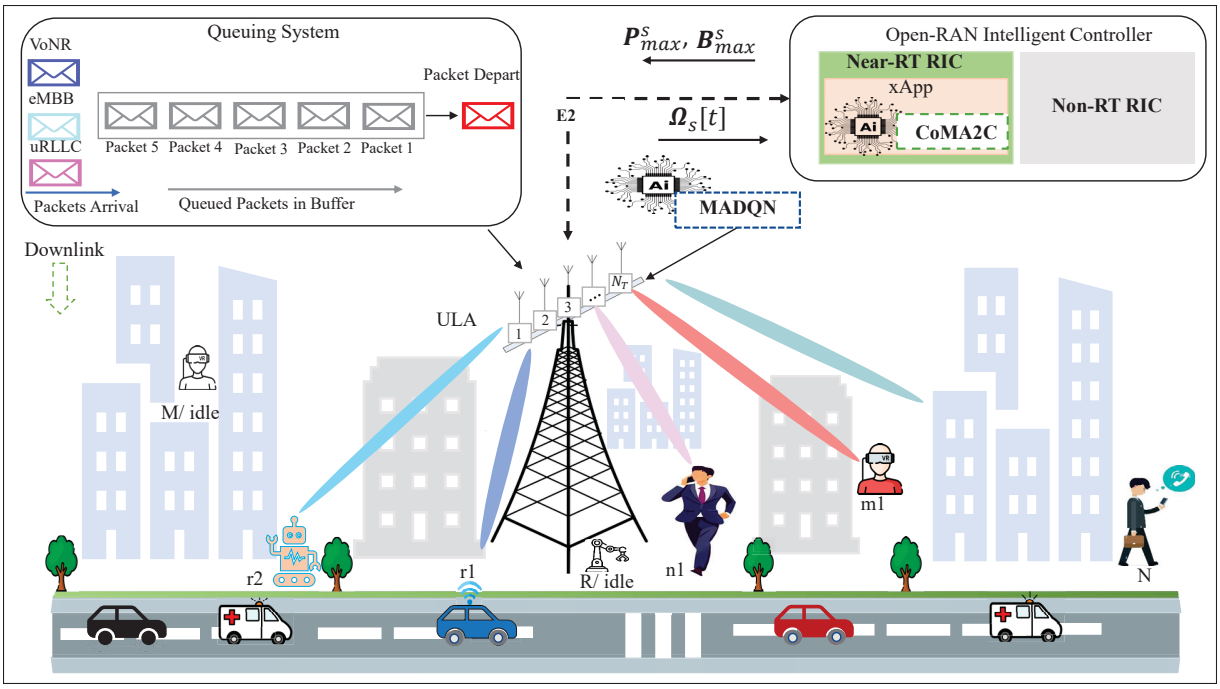

Figure 4.1 System model of downlink RSMA with heterogeneous inter- and intra-RAN slicing

where $\mathbf{w}_c \in \mathbb{C}^{N_T \times 1}$ and $\mathbf{w}_{u_s} \in \mathbb{C}^{N_T \times 1}$ are the beamforming vectors of the common and private messages, respectively.

4.3.4 Signal Model

The received signal of the user u_s at time slot t, where $u_s \in \{n, m, r\}$, is given by using the following equation.

$$y_{u_s} = \underbrace{\mathbf{h}_{u_s}^H[t]\mathbf{w}_c[t]s_c(t)}_{\text{common message}} + \underbrace{\mathbf{h}_{u_s}^H[t]\mathbf{w}_{u_s}[t]s_{u_s}[t]}_{\text{private message}} + \sum_{i=1,i\neq u_s}^{U_s} \underbrace{\mathbf{h}_{u_s}^H[t]\mathbf{w}_i[t]s_i[t]}_{\text{intra interference}} + \underbrace{\mathbb{I}_{BS}^t}_{\text{HWI at ULA}} + \underbrace{\mathbb{Z}_{u_s}[t]}_{\text{AWGN at } u_s} + \underbrace{\mathbb{I}_{u_s}^r}_{\text{Self-distortion}} ; \forall u_s \in \mathcal{U}_s, \quad (4.10)$$

where $\mathbf{h}_{u_s} \in \mathbb{C}^{N_T \times 1}$ denotes the complex channel vector between the BS and user u_s in a given slice s. Moreover, the superscript $(\cdot)^H$ denotes the Hermitian operator. The second term in Eq. (4.10) represents the intra-interference experienced by the $u_s{}^{\text{th}}$ user. It is also assumed that all the users are subjected to additive white Gaussian noise (AWGN) with noise variance σ^2, where $\mathbb{Z}_{u_s} \in \mathcal{CN}(0, \sigma^2)$ denotes the noise at the user u_s at time slot t. Furthermore, $\mathbb{I}_{BS}^t \sim \mathcal{CN}(\mathbf{0}, \mathbf{d^t})$ and $\mathbb{I}_u^r \sim \mathcal{CN}(0, d^r)$ denote the distortions at the ULA of the BS and the receiver, respectively.

The distortion noise variance at the ULA is represented mathematically as follows (Björnson *et al.*, 2013).

$$\mathbf{d}^t = \kappa^t \cdot \text{diag}(|\omega_1|^2, |\omega_2|^2, \ldots, |\omega_{N_T}|^2), \quad (4.11)$$

where $\kappa^t \geq 0$ the transmitter's distortion level, $\text{diag}(\cdot, \ldots, \cdot)$ is a diagonal matrix, and ω is the element of $\mathbf{w}$, where $\mathbf{w} \in \{\mathbf{w}_c, \mathbf{w}_{u_s}\}$. In this context, we consider that HWIs affect both the common and private messages for each user. The variance of the distortion noise at user u_s is given by (Björnson *et al.*, 2013)

$$d^r = \kappa^r |\mathbf{h}_{u_s}^H \mathbf{w}|^2, \quad (4.12)$$

where $\kappa^r \geq 0$ denotes the self distortion level at user u_s. We simplify the analysis by assuming that all antenna elements in the ULA undergo the same level of distortion κ^t. Furthermore, it is assumed that all users experience the same level of distortion κ^r.

Initially, each user decodes the common stream, and the interference from all private streams is regarded as noise. Then, each user uses successive interference cancellation (SIC) to remove the shared stream (s_c) from their signals. Users then decode the intended private stream, while considering the interference from the other private streams as noise. Consequently, the instantaneous signal to distortion and noise ratio (SDNR) of the common stream ($\Gamma^c_{u_s}$) and private ($\Gamma^p_{u_s}$) stream at time t are given as follows.

$$\Gamma^c_{u_s}[t] = \frac{|\mathbf{h}^H_{u_s}[t]\mathbf{w}_c[t]|^2}{\underbrace{\sum_{i=1}^{U_s}|\mathbf{h}^H_{u_s}[t]\mathbf{w}_i[t]|^2}_{\text{Interference}} + \underbrace{\kappa^t\sum_{\tau=1}^{N_T}|h_\tau w_\tau|^2}_{\text{HWI at ULA}} + \underbrace{\kappa^r\left|\mathbf{h}^H_{u_s}[t]\mathbf{w}_c[t]\right|^2}_{\text{Self-distortion}} + \underbrace{\sigma^2_{u_s}}_{\text{AWGN}}}; \forall u_s \in \mathcal{U}_s \tag{4.13}$$

$$\Gamma^p_{u_s}[t] = \frac{|\mathbf{h}^H_{u_s}[t]\mathbf{w}_{u_s}[t]|^2}{\underbrace{\sum_{i=1,i\neq u_s}^{U_s}|\mathbf{h}^H_{u_s}\mathbf{w}_i|^2}_{\text{Interference}} + \underbrace{\kappa^t\sum_{\tau=1}^{N_T}|h_\tau w_\tau|^2}_{\text{HWI at ULA}} + \underbrace{\kappa^r\left|\mathbf{h}^H_{u_s}[t]\mathbf{w}_{u_s}[t]\right|^2}_{\text{Self-distortion}} + \underbrace{\sigma^2_{u_s}}_{\text{AWGN}}}; \forall u_s \in \mathcal{U}_s \tag{4.14}$$

The original message for each user is then reconstructed by combining the decoded private message of that user with the decoded common message. To ensure that the common message can be successfully decoded by all users in the system, the achievable rate of the common part is calculated as $r_c[t] = \min_{u_s\in\mathcal{U}_s}\left\{\log_2\left(1+\Gamma^c_{u_s}\right)\right\}$. The instantaneous rate for decoding private stream at user u_s is below.

$$r^p_{u_s}[t] = \log_2\left(1+\Gamma^p_{u_s}\right); \forall u_s \in \mathcal{U}_s. \tag{4.15}$$

We assume that r_c is shared by the users within each slice such that $\sum_{u_s\in U_s}\mathcal{X}^c_{u_s}$, where $\mathcal{X}^c_{u_s}$ is the portion of the common stream's rate that is meant for user u_s. Thus, the total achievable rate for

each user u_s in slice s can be expressed as:

$$\mathfrak{R}_{u_s}[t] = \mathcal{X}^c_{u_s} + r^p_{u_s}. \tag{4.16}$$

4.3.5 Objective Function

The objective of this study is to maximize the long-term utility function $f(\cdot)$, which is defined as a combination of the weighted sum of the spectral efficiency (η) and the SSRs of the different services. The mathematical formulation of the objective function ($\mathcal{OF}$) is provided in Eq. (4.17). A higher utility value indicates better QoS performance for the network slices. In Eq. (4.17), the parameters α^η and $\mathfrak{B}_s = \{\mathfrak{B}_{s1}, \ldots, \mathfrak{B}_S\}$ denote the weights associated to η and the SSR_s of slices, respectively. These parameters reflect the relative importance of η and the SSR_s and can be tuned to meet specific system requirements (Shao *et al.*, 2021). In this context, η is defined as $\eta[t] = \frac{\sum_{s\in\mathcal{S}}\sum_{u_s\in\mathcal{U}_s}\mathfrak{R}_{u_s}}{\mathbb{B}^T}$, where $\mathbb{B}^T$ denotes the total available bandwidth.

To achieve optimal inter- and intra-RAN slicing control strategies that maximize the long-term objective, the system dynamically adjusts the power allocation $\mathbf{P}^s_{\max} = \{P^n_{\max}, P^m_{\max}, P^r_{\max}\}$, where $P^n_{\max}, P^m_{\max}, P^r_{\max}$ represent the power budgets for VoNR, eMBB, and uRLLC, respectively. Similarly, the bandwidth allocation is defined as $\mathbf{B}^s_{\max} = \{B^n_{\max}, B^m_{\max}, B^r_{\max}\}$, where $B^n_{\max}, B^m_{\max}, B^r_{\max}$ denote the bandwidth budgets for VoNR, eMBB, and uRLLC, respectively, at the inter-slice level. At the intra-slice level, the system further optimizes the beamformer vectors, which include the powers $\|\mathbf{w}_c\|^2$ and $\|\mathbf{w}_{u_s}\|^2$, along with their corresponding directions $\hat{\mathbf{w}}_c$, $\hat{\mathbf{w}}_{u_s}$. These optimizations are performed in compliance with radio resource constraints at both the inter- and intra-slice levels.

$$\begin{aligned} \mathcal{OF}: \quad & \underset{\substack{\mathbf{P}^s_{\max}, \mathbf{B}^s_{\max} \\ \hat{\mathbf{w}}_c, \hat{\mathbf{w}}_{u_s} \\ \|\mathbf{w}_c\|^2, \|\mathbf{w}_{u_s}\|^2}}{\text{maximize}} \quad \alpha^\eta.\eta + \sum_{s\in\mathcal{S}} \mathfrak{B}_s \cdot SSR_s \end{aligned} \tag{4.17}$$

$$
\begin{aligned}
\text{s.t.} \quad & \text{C1: } P^s_{\max} \geq \lambda^s_p; \quad \forall s \in \mathcal{S}, \\
& \text{C2: } \sum_{s \in \mathcal{S}} P^s_{\max} = \mathbb{P}^T; \quad \forall t, \\
& \text{C3: } B^s_{\max} \geq \Lambda^s_b; \quad \forall s \in \mathcal{S}, \\
& \text{C4: } \sum_{s \in \mathcal{S}} B^s_{\max} = \mathbb{B}^T; \quad \forall t, \\
& \text{C5: } \mathbf{\Omega}_s = (\Omega_{s1}, \ldots, \Omega_{sS}); \quad \forall s \in \mathcal{S}, \\
& \text{C6: } SSR_s \geq \text{SSR}^{\text{Th}}_s; \quad \forall s \in \mathcal{S}, \\
& \text{C7: } \|\mathbf{w}_{u_s}\|^2 \geq 0; \quad \forall u_s \in \mathcal{U}_s, \\
& \text{C8: } \|\mathbf{w}_c\|^2 + \sum_{n=1}^{N} \|\mathbf{w}_n\|^2 \leq P^n_{\max}, \\
& \text{C9: } \|\mathbf{w}_c\|^2 + \sum_{m=1}^{M} \|\mathbf{w}_m\|^2 \leq P^m_{\max}, \\
& \text{C10: } \|\mathbf{w}_c\|^2 + \sum_{r=1}^{R} \|\mathbf{w}_r\|^2 \leq P^r_{\max}, \\
& \text{C11: } SSR_{u_s} \geq \text{SSR}^{\text{Th}}_s; \quad \forall u_s \in \mathcal{U}_s, \\
& \text{C12: } \hat{\mathbf{w}}_{u_s} \in [0,\, 2\pi), \\
& \text{C13: } \sum_{u_s \in U_s} \mathcal{X}^c_{u_s} \leq r_c; \quad \forall u_s \in \mathcal{U}_s, \forall s \in \mathcal{S} \\
& \text{C14: } \mathcal{X}^c_{u_s}, r^p_{u_s} \geq 0; \quad \forall u_s \in \mathcal{U}_s, \forall s \in \mathcal{S}.
\end{aligned}
$$

In Eq. (4.17), C1 guarantees that a minimum power (λ^s_p) is allocated per slice. C2 defines the total power allocated to all the slices, which is equal to the total system power ($\mathbb{P}^T$). C3 ensures that each slice is allocated a minimum bandwidth (Λ^s_b). C4 defines the total bandwidth allocated to all the slices, which is equal to the $\mathbb{B}^T$. C5 defines $\mathbf{\Omega}_s$ as the traffic model for slice s. C6 ensures that the SSR of each slice meets or exceeds the predefined threshold (SSR^{Th}_s). C7 ensures that the power allocated to each user is non-negative. Constraints 8-10 guarantee that the power allocated to the users of each slice does not exceed the slice budget. C11 ensures that the SSR of each user in each slice is greater than a predefined threshold (SSR^{Th}_s). C12 defines

the antenna phase shift constraint. C13 defines the common rate constraint. C14 indicates that the common and private rates must not be negative.

The optimization problem $\mathcal{OF}$ is a non-convex problem classified as NP-hard and challenging to solve. The difficulty of solving $\mathcal{OF}$ stems from the following two main factors. First, the joint optimization of both inter- and intra-slice levels significantly increases computational complexity. Second, because of user mobility and the stochastic nature of the traffic model, traffic demand fluctuates over time and cannot be accurately predicted in advance. These challenges are further intensified in the context of ZTNs for 6G, which demand autonomous and intelligent decision-making in highly dynamic, dense, and heterogeneous RAN slicing environments. Addressing these complex requirements exposes the limitations of traditional optimization techniques. Methods such as genetic algorithms and heuristics frequently struggle with NP-hard problems, particularly in the dynamic and heterogeneous nature of RAN slicing. These approaches depend on approximate mathematical models that can not fully or accurately capture the complexity and dynamic nature of real-world environments. As NS complexity increases with more devices, services, and diverse QoS requirements, such methods become less scalable and efficient. Their high computational cost tends to yield suboptimal solutions in large-scale real-time scenarios (Zangooei *et al.*, 2023). While exhaustive search method could theoretically yield optimal solutions, it is computationally infeasible due to its exponential complexity with respect to the number of variables (Pala, Katwe, Singh, Clerckx & Li, 2024). Furthermore, traditional approaches lack self-learning capabilities, which are essential for the autonomous self-optimization expected in ZTNs scenarios (Sabr *et al.*, 2025a).

Overcoming these challenges requires a more adaptive and scalable approach— the one aligned with the requirements of ZT applications. Accordingly, we propose a cooperative MADRL approach to solve our $\mathcal{OF}$. DRL algorithms are well-suited for this task due to their ability to learn optimal policies through interaction with the environment and adaption to changing NSs conditions in real time, while maintaining computational efficiency (Filali *et al.*, 2022; Saha, Zangooei, Golkarifard & Boutaba, 2023). In Section 4.4, we decompose the problem $\mathcal{OF}$ into two subproblems, each corresponding to a specific level of management within the

RAN domain. The first subproblem addresses inter-slice, represented by constraints C1–C6, whereas the second focuses on intra-slice, represented by constraints C7–14. Both subproblems are solved using the MADRL approach.

4.4 Hierarchical RRM Framework Based on Cooperative Heterogeneous MADRL

This section presents an overview of the proposed solution, with details of the design and implementation of the proposed HiSO-CoMA framework.

4.4.1 Overview

To address the problem in Eq. (4.17) using the MADRL approach, we reformulate it as a twin-timescale MDP. More specifically, we reformulate the inter-slice level problem as a POMDP to align with real-world scenarios where the RIC has an incomplete view of the NS environments (Setayesh *et al.*, 2022), while the intra-slice level is reformulated as an MDP, as discussed in Sections 4.4.2 and 4.4.3.

To solve the POMDP, we employ cooperative multiple A2C agents to form the CoMA2C scheme. Each agent, denoted by $g_\sigma \in \mathcal{G}_\sigma$, is responsible for managing a specific type of radio resource $\sigma \in \Theta$, such as power or bandwidth, across heterogeneous services, including VoNR, eMBB, and uRLLC. For the intra-slice level, we use a set of DQN agents $\mathcal{J}_s$ to form a MADQN scheme. Here, a dedicated agent, denoted $j_s \in \mathcal{J}_s$, is assigned to manage the resources within each NS among its active users. Our design choices for the DRL algorithms are guided by the literature that consistently adopts A2C for inter-slice and DQN for intra-slice optimization.

The main benefit of adopting the MADRL approach is that it allows for the decomposition of high-dimensional state and action spaces compared to a single-agent DRL (Liu *et al.*, 2024b). This decomposition simplifies the complexity of the problem and provides a more efficient and scalable architecture that can be generalized to more resources and slices in the future. Furthermore, the proposed framework incorporates distributed learning at both management

levels, significantly reducing the signaling overhead compared to centralized learning, as highlighted in (Wang *et al.*, 2024b).

The CoMA2C scheme of the proposed framework operates on a large timescale, denoted $\mathcal{T}^{\text{long}} = \{1, 2, \ldots, T^{\text{long}}\}$, which represents a set of indices corresponding to long time slots. Each long time slot has a fixed duration of ΔT^{long} (e.g., 1 second) (Setayesh *et al.*, 2022). Conversely, the MADQN scheme operates on a small timescale at the intra-RAN slicing level, where each long time slot $t \in \mathcal{T}^{\text{long}}$ is divided into smaller time slots, $\mathcal{T}^{\text{short}} = \{1, 2, \ldots, T^{\text{short}}\}$, each with equal duration ΔT^{short} (e.g., 0.5 ms).

To ensure real-time performance and ZT management in RAN slicing, RRM at the inter-slice level must frequently synchronize with RRM at the intra-slice level. However, high traffic fluctuations and significant user mobility at the intra-slice level make this synchronization a complex task. Coordinating multiple resource management across RAN slicing levels requires continuous communication and updates, which can lead to excessive network overhead. To address this challenge, we propose a mitigation strategy to minimize network overhead while ensuring synchronization between the two-timescale operational models of the proposed framework, as detailed in the next section.

4.4.2 POMDP Formulation for CoMA2C at the Large Timescale

In this subsection, the specific definitions of state, action, and reward are introduced for agents that manage resources on a large time scale.

4.4.2.1 State

We assume that information is exchanged between the RIC and the BS via the E2 interface. This includes the number of network slices hosted by the BS and their corresponding traffic loads. As a result, the state ($\mathbf{s}_{g_\sigma}$) of each $g_\sigma \in \mathcal{G}_\sigma$ at $t \in \mathcal{T}^{\text{long}}$ is given by

$$\mathbf{s}_{g_\sigma}[t] = \mathbf{\Omega}_s = \left[\Omega_{s_1}, \Omega_{s_2}, ..., \Omega_S\right]; \quad \forall g_\sigma \in \mathcal{G}_\sigma, \tag{4.18}$$

4.4.2.2 Action

After the RIC observes the instantaneous traffic load for each NS, the cooperative agents—one responsible for power allocation ($g_\sigma = g_P$) and another for bandwidth allocation ($g_\sigma = g_B$) take action at $t \in \mathcal{T}^{\text{long}}$ to allocate the respective budgets to the heterogeneous NS. These actions are determined according to Eq. (4.19) and Eq. (4.20), respectively.

$$\mathbf{a}_g^P[t] = \{(P_{\max}^{s_1}, P_{\max}^{s_2}, ..., P_{\max}^{S}) \in \mathbf{P}_{\max}^s \mid P_{\max}^s \geq \lambda_p^s,\ \mathbf{a}_g^P \in \mathcal{A}_g^P, \} \tag{4.19}$$

$$\mathbf{a}_g^B[t] = \{(B_{\max}^{s_1}, B_{\max}^{s_2}, ..., B_{\max}^{S}) \in \mathbf{B}_{\max}^s \mid B_{\max}^s \geq \Lambda_b^s, \mathbf{a}_g^B \in \mathcal{A}_g^B, \} \tag{4.20}$$

where $\mathcal{A}_g^P$ and $\mathcal{A}_g^B$ represent all the feasible action combinations, including possible power allocation actions between λ_p^s and $P_{\max}^s$, as well as possible bandwidth allocation actions between Λ_b^s and $B_{\max}^s$, respectively.

The inter-slice management scheme of the proposed framework dynamically updates the resources of each NS based on its specific QoS requirement and traffic demand. Typically, the near-RT RIC functions within a time range of 10 ms to 1 s (Ngo *et al.*, 2024b). To align with near-RT constraints while minimizing overhead, the proposed CoMA2C scheme adjusts resource allocations only when significant traffic variations are detected across slices. To achieve this, the RIC continuously monitors traffic loads and evaluates at one second intervals. If a substantial variation is detected, resource reallocation is triggered; otherwise, the current configuration is maintained. The relative change in the traffic load Ω_s of each service at $t \in \mathcal{T}^{\text{long}}$ is calculated as:

$$\Delta^{\Omega_s}[t] = \frac{|\Omega_s(t) - \Omega_s(t-1)|}{\Omega_s(t-1)} \times 100, \tag{4.21}$$

We define the maximum change across all services at $t \in \mathcal{T}^{\text{long}}$ as

$$\Delta_{\max}[t] = \max\left\{\Delta^{\Omega_{s_1}}[t], \Delta^{\Omega_{s_2}}[t], ..., \Delta^{\Omega_S}[t]\right\}, \tag{4.22}$$

Here, $\Delta_{\max}[t]$ represents the maximum observed traffic change at time t. The decision to trigger inter-slice resource reallocation via CoMA2C (τ^{CoMA2C}) is then made based on a predefined threshold (∇^{Th}) as follows

$$\tau^{\text{CoMA2C}} = \begin{cases} \text{True}; & \text{if } \Delta_{\max}[t] > \nabla^{\text{Th}} \\ \text{False}; & \text{if } \Delta_{\max}[t] \leq \nabla^{\text{Th}} \end{cases} \tag{4.23}$$

If $\Delta_{\max} > \nabla^{\text{Th}}$, a significant change in traffic is inferred, this prompts the RIC to activate CoMA2C at the inter-slice level to reallocate resources accordingly. Otherwise, the system retains the current allocation, while continuing real-time monitoring at the inter-slice level and resource management at the intra-slice level. This strategy plays a key role in reducing overhead by limiting interactions between agents in the CoMA2C and MADQN schemes. These interactions are triggered only when substantial traffic variations occur, thereby avoiding unnecessary communication. Meanwhile, the system remains in real-time monitoring mode to effectively handle any sudden traffic fluctuations.

4.4.2.3 Global Reward

After agents of CoMA2C scheme perform their chosen actions, the NS environment sends them a team reward (r_{g_σ}) according to Algorithm 4.1, which represents feedback that measures how well the executed actions align with observed conditions. The reward function design considers four scenarios. In the first scenario (Lines 2–4), if the SSR_s of all services are greater than or equal to the predefined SSR_s^{Th} and the spectral efficiency is below 100 bps/Hz, the agent receives a scalar reward of 10. In the second scenario (Line 6), if all services are satisfied the SSR_s^{Th} and the spectral efficiency is greater than 100 bps/Hz, the agent receives a bonus reward proportional to the spectral efficiency. In the third scenario (Lines 8-9), if uRLLC does

not achieve its predefined SSR_s^{Th}, the agent receives a proportional reward based on uRLLC performance. Finally, in the fourth scenario (Line 11), a negative reward (penalty) is applied if either VoNR or eMBB—or both— falls below their respective SSR_s^{Th}. The penalty term in Line 11 uses a min operator to identify the worst-performing service, ensuring that the penalty is proportional to the most degraded service quality.

Algorithm 4.1 Calculate Team Reward for $\mathcal{G}_\sigma$

1 : **Input:** $SSR_s = \{SSR_{\text{VoNR}}, SSR_{\text{eMBB}}, SSR_{\text{uRLLC}}\}, \eta$
2 : **If** $SSR_{\text{VoNR}}, SSR_{\text{eMBB}}$ **and** $SSR_{\text{uRLLC}} \geq \text{SSR}_s^{\text{Th}}$ **Then**
3 : **If** $\eta < 100$ **Then**
4 : $r_{g_\sigma}[t] \leftarrow 10$
5 : **Else**
6 : $r_{g_\sigma}[t] \leftarrow 10 + 0.1 \cdot (\eta - 100)$
7 : **End If**
8 : **Else If** $SSR_{\text{uRLLC}} < \text{SSR}_s^{\text{Th}}$ **Then**
9 : $r_{g_\sigma}[t] \leftarrow 10 \cdot (SSR_{\text{uRLLC}} - 0.7)$ (Li *et al.*, 2020a)
10 : **Else**
11 : $r_{g_\sigma}[t] \leftarrow -2 \cdot (1 - \min(SSR_{\text{VoNR}}, SSR_{\text{eMBB}}))$
12 : **End If**
13 : **Output:** $r_{g_\sigma} \forall g_\sigma \in \mathcal{G}_\sigma$

4.4.3 MDP Formulation for MADQN at the Small Timescale

In this subsection, we introduce the state, action, and reward for the agents responsible for managing resources on a small timescale.

4.4.3.1 State

Due to the heterogeneous QoS requirements of each NS, each agent $j_s \in \mathcal{J}_s$ independently observes the state of its own NS. This state is constructed solely based on locally available information, enabling decentralized decision-making. This approach reduces signaling overhead and minimizes processing latency (Tran, Sharma, Ha, Chatzinotas & Woungang, 2023). Specifically, the state (s_{j_s}) of each agent at time $t \in \mathcal{T}^{\text{short}}$ is defined as

$$s_{j_s}[t] = \Big\{ \|\mathbf{w}_c\|^2[t-1], \Gamma^c_{u_s}[t-1], \mathcal{X}^c_{u_s}[t-1], \|\mathbf{w}_{u_s}\|^2[t-1] \\ \Gamma^p_{u_s}[t-1], I^{\hat{\mathbf{w}}_{u_s}}[t-1], r^p_{u_s}[t-1], \left|\mathbf{h}^H_{u_s}[t]\hat{\mathbf{w}}_{u_s}[t]\right|^2 \Big\}; \\ \forall u_s \in \mathcal{U}_s, \forall s \in \mathcal{S}. \quad (4.24)$$

where $\|\mathbf{w}_c\|^2[t-1]$, $\Gamma^c_{u_s}[t-1]$, and $\mathcal{X}^c_{u_s}[t-1]$ represent the previous power, SDNR, and rate for the common stream, respectively. Similarly, $\|\mathbf{w}_{u_s}\|^2[t-1]$, $\Gamma^p_{u_s}[t-1]$, $I^{\hat{\mathbf{w}}_{u_s}}[t-1]$, $r^p_{u_s}[t-1]$, and $\left|\mathbf{h}^H_{u_s}[t]\hat{\mathbf{w}}_{u_s}[t]\right|^2$ denote the previous power, SDNR, beam direction index, rate, and the equivalent channel gain for the private stream respectively.

4.4.3.2 Action

The aim of each $j_s \in \mathcal{J}_s$ within the MADQN scheme is to optimize the downlink power for both the private and common streams, as well as the beam directions for the private streams. To design the action space in discrete form and align it with the DQN algorithm, we discretize each NS's power budget $P^s_{\max} \in \{P^n_{\max}, P^m_{\max}, P^r_{\max}, ..., P^S_{\max}\}$ into N^s_{L} transmit power levels, uniformly distributed over the range from zero to the maximum transmit power $p^s_{\max}$. Additionally, we adopt the codebook technique to discretize the beam directions for the private stream, while random beamforming (RBF) (Dizdar, Mao & Clerckx, 2021a; Mao *et al.*, 2022) is considered for the common stream. To implement this, we design a matrix based on the codebook technique, denoted as $\mathbf{C}_{\text{book}} = \{\mathbf{c_0}, \mathbf{c_1}, \ldots, \mathbf{c}_{B_{\text{code}}-1}\} \in \mathbb{C}^{N_T \times B_{\text{code}}}$, where B_{code} denotes the size of the

codebook and $B_{\text{code}} \geq N_T$ (Sabr *et al.*, 2025a). Each vector $\mathbf{c} \in \mathbb{C}^{N_T \times 1}$ in $\mathbf{C}_{\text{book}}$ corresponds to a specific beam pattern (direction) within the range $[0, 2\pi)$ for $\hat{\mathbf{w}}_{u_s}$. The details of the codebook design procedure used in this work are similar to those presented in our previous work, as reported in (Sabr *et al.*, 2025a), and it is also applied in (Ge *et al.*, 2020). The total number of available actions is defined as $N_L^s \times B_{\text{code}}$, which is equal to the output dimension of the DQN. Thus, the available actions for each $j_s \in \mathcal{J}_s$ are represented as a set of all possible action combinations, denoted by $\mathcal{A}^s \in \{\mathcal{A}^{\text{VoNR}}, \mathcal{A}^{\text{eMBB}}, \mathcal{A}^{\text{uRLLC}}\}$. At each time step $t \in \mathcal{T}^{\text{short}}$, each agent $j_s \in \mathcal{J}_s$ takes an action $a_{j_s}[t] = (p^c, p^p, \mathbf{c}) \in \mathcal{A}^s$, as defined in Eq. (4.25), where $p^c = \|\mathbf{w}_c\|^2$ and $p^p = \|\mathbf{w}_{u_s}\|^2$. Simultaneously, the common beamforming vector $\hat{\mathbf{w}}_c$ is randomly generated according to the RBF strategy.

$$\mathcal{A}^s = \{(p^c, p^p, \mathbf{c}),\ p^c, p^p \in \mathcal{P},\ \mathbf{c} \in \mathbf{C}_{\text{book}}\}, \tag{4.25}$$

where

$$\mathcal{P} = \{0, \frac{1}{N_{\text{L}}^s - 1} p_{\max}^s, \frac{2}{N_{\text{L}}^s - 1} p_{\max}^s, \ldots, p_{\max}^s\}, \text{and}$$
$$\mathbf{C}_{\text{book}} = \{\mathbf{c_0}, \mathbf{c_1}, \ldots, \mathbf{c}_{B_{\text{code}}-1}\}.$$

4.4.3.3 Reward

Since each NS in the proposed system serves users with distinct QoS requirements, designing a reward function that accurately reflects the SLA of each NS represents another research challenge, as discussed in (Dubey *et al.*, 2025). Here, reward (r_{j_s}) plays a crucial role in guiding the $j_s \in \mathcal{J}_s$ towards an optimal policy, where a well-designed reward facilitates effective learning and policy convergence, and a poorly designed reward can hinder convergence and mislead the agent. Therefore, in the proposed system, each $j_s \in \mathcal{J}_s$ receives r_{j_s} at $t \in \mathcal{T}^{\text{short}}$ when its actions enhance the spectral efficiency, meet the minimum rate requirements, and satisfy the SSR, as

defined in Eq. (4.26). To ensure stability during training, r_{j_s} is clipped to prevent extreme values from destabilizing the learning process.

$$r_{j_s}[t] = \text{clip}\left(\eta_{u_s} \cdot \vartheta_{u_s} \cdot \delta_{u_s}, -\mu, \mu\right), \tag{4.26}$$

In Eq. (4.26), η_{u_s} represents the spectral efficiency of u_s in $s \in \mathcal{S}$ and given by

$$\eta_{u_s} = \frac{\mathfrak{R}_{u_s}[t]}{b_{u_s}}, \tag{4.27}$$

where b_{u_s} represents the bandwidth allocated to u_s. The terms ϑ_{u_s} and δ_{u_s} function as constraint violation penalties for QoS and minimum rate requirements, respectively. The parameters $-\mu, \mu$ denote the lower and upper clipping bounds, respectively. Both ϑ_{u_s} and δ_{u_s} in Eq. (4.26) are defined as follows:

$$\vartheta_{u_s} = \begin{cases} 1; & \text{if } \text{SSR}_{u_s} \geq \text{SSR}_s^{Th} \\ \max(0.1, \frac{\text{SSR}_{u_s}}{\text{SSR}_s^{Th}}); & \text{if } \text{SSR}_{u_s} < \text{SSR}_s^{Th} \end{cases}, \tag{4.28}$$

$$\delta_{u_s} = \begin{cases} 1; & \text{if } \mathfrak{R}_{u_s} \geq \mathfrak{R}_s^{\text{Min}} \\ \max(0.1, \frac{\mathfrak{R}_{u_s}}{\mathfrak{R}_s^{\text{Min}}}); & \text{if } \mathfrak{R}_{u_s} < \mathfrak{R}_s^{\text{Min}} \end{cases}. \tag{4.29}$$

The design of r_{j_s} is based on a multiplicative relationship that combines three components, ensures that each $j_s \in \mathcal{J}_s$ is incentivized to optimize all three key performance indicators simultaneously, while maintaining stable learning through appropriately scaled rewards.

4.4.4 Challenges

Overall, solving a twin-timescale MDP presents significant challenges (Mei *et al.*, 2021). It is important to note that the problem in Eq. (4.17) is especially difficult to solve using MADRL due to the following challenges.

Challenge 1 Coordinated multi-agent learning complexity: Implementing the CoMA2C scheme requires managing heterogeneous A2C agents that operate cooperatively across diverse network slices. These agents are required to learn concurrently within a shared environment, coordinating rewards and hyperparameters to achieve joint optimality. In this context, synchronizing agent convergence with training time is a core challenge in ensuring system stability and high performance.

Challenge 2 Synchronization between slicing levels: Another significant challenge lies in designing a hierarchical ZT framework that enables the simultaneous training and coordination of both inter- and intra-RAN slicing policies. While the inter-RAN policy allocates resources across slices, the intra-RAN policy manages the resources within each slice. Frequent user mobility, traffic fluctuations, and improper hyperparameter tuning can disrupt synchronization between these two levels, making it difficult to ensure aligned learning and convergence—ultimately threatening the stability and performance of the overall system.

4.4.5 Hierarchical MADRL framework for Solving $\mathcal{OF}$

The proposed framework is developed in two stages: **Stage I** introduces the CoMA2C scheme, with its functionality briefly depicted in Fig. 4.2, while **Stage II** presents the MADQN scheme, where the architecture of each agent in the MADQN scheme is depicted in Fig. 4.3. Together, these schemes form the proposed HiSO-CoMA framework, as shown in Fig. 4.4.

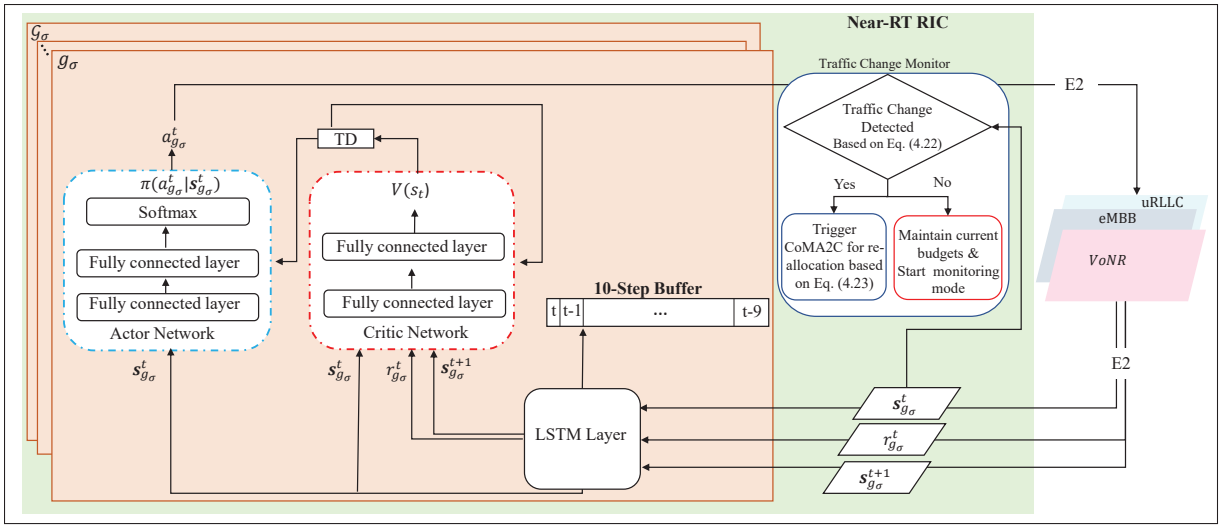

Figure 4.2 An illustration of the CoMA2C scheme for joint resource management among heterogeneous inter-RAN slicing

4.4.5.1 Stage I- Desgin and Learn inter-slice Policy

The architecture of each $g_\sigma \in \mathcal{G}_\sigma$ in the CoMA2C scheme is deployed with two separate deep neural networks (DNNs): the actor and the critic networks. The actor network represents a policy (π) responsible for exploring the action space in order to maximize the expected cumulative rewards ($\bar{R}_t$) from each state $\mathbf{s}^t_{g_\sigma}$ after taking action $a^t_{g_\sigma} \in \{\mathbf{a}^P_g, \mathbf{a}^B_g\}$ based on its current policy $\pi(a^t_{g_\sigma}|\mathbf{s}^t_{g_\sigma})$. Following interaction with the environment, the agent moves to the next state $\mathbf{s}^{t+1}_{g_\sigma}$ and receives a reward $r^t_{g_\sigma}$. At time slot t, the total accumulated reward is defined as $\bar{R}_t = \sum_{t=1}^{T^i} \Upsilon^{t-1}.r^t_{g_\sigma}$, typically estimated through the state-value function, where $\Upsilon \in [0, 1]$ represents the discount factor and T^i is the total number of time steps per iteration (Sun & Zhang, 2022). The critic network is responsible for estimating the state-value function ($V(\mathbf{s}) = \mathbb{E}[\bar{R}_t|\mathbf{s}^t_{g_\sigma} = \mathbf{s}]$), basically calculates the average anticipated return from state $\mathbf{s}$ and assesses the actor-optimized policy(Li *et al.*, 2020a). The DNN structures of the actor and critic networks leverage a common LSTM layer. The internal memory mechanism of an LSTM layer enables actor and critic DNNs to implicitly create the historical sequence of actions and observations needed to address the POMDP's challenges (Setayesh *et al.*, 2022). Algorithm 4.2 describes the CoMA2C scheme of our proposed framework. The initialization

stage of Algorithm 4.2 is defined in Lines 1-5, where Line 2 initializes A2C based LSTM for each $\sigma \in \Theta$ in the system. Line 3 defines the SLA criteria for each NS. Line 4, initializes $\Psi^a_{g_\sigma}$ to parameterize the actor neural network and $\Psi^c_{g_\sigma}$ to parameterize the critic neural network, and initialize the learning rates for the actor ($\varrho^a_{g_\sigma}$) and critic ($\varrho^c_{g_\sigma}$) networks, respectively. Finally, in Line 5, the tracker is initialized to monitor and track changes in the traffic loads $\mathbf{\Omega}_s$. The loop within Lines 6–21 involves the RIC's learning process, where at the beginning of each $t \in \mathcal{T}^{\text{long}}$ the actor of each agent observes $\mathbf{\Omega}_s$, represented by a state vector (see Eq. (4.18)). Next, the actor network of each $g_\sigma \in \mathcal{G}_\sigma$ takes actions based on the observed environmental state and assigns $\mathbf{P}^s_{\max}$ and $\mathbf{B}^s_{\max}$ in line with Eq. (4.19) and (4.20). Then, at the intra-slice management level, each $j_s \in \mathcal{J}_s$ receives its corresponding $P^s_{\max}$ as input and distributes it among the slice's active users by performing joint power allocation and beamforming optimization according to Algorithm 4.3. Meanwhile, $B^s_{\max}$ of each NS is distributed among the active users based on the round robin (RR) algorithm. Then, based on the achieved QoS for each NS, the NS environment shapes the team reward following Algorithm 4.1 for each $g_\sigma \in \mathcal{G}_\sigma$. Then, the temporal difference (TD) error, denoted by $\mathcal{E}$, which is essential for the computation of the loss, is evaluated by the critic network of each $g_\sigma \in \mathcal{G}_\sigma$, as given by Eq. (4.30) (Li *et al.*, 2020a).

$$\mathcal{E}_t = \underbrace{r^t_{g_\sigma} + \Upsilon V\left(\mathbf{s}^{t+1}_{g_\sigma}; \psi^c_{g_\sigma}\right)}_{Q(s_t, a_t)} - \underbrace{V\left(\mathbf{s}^t_{g_\sigma}; \psi^c_{g_\sigma}\right)}_{\text{current value function}} . \tag{4.30}$$

The actor loss function ($\mathcal{L}^{\text{Actor}}_{g_\sigma}$) (Li *et al.*, 2020a; Sun & Zhang, 2022) for each $g_\sigma \in \mathcal{G}_\sigma$ is given by

$$\mathcal{L}^{\text{Actor}}_{g_\sigma} = -\Big[\underbrace{\log \pi(a^t_{g_\sigma} | \mathbf{s}^t_{g_\sigma}; \psi^a_{g_\sigma})}_{\substack{\text{log probability of} \\ \text{action given a state}}} + \phi\, \mathbb{E}\left(\pi(a^t_{g_\sigma} | \mathbf{s}^t_{g_\sigma}; \psi^a_{g_\sigma})\right) \mathcal{E}_t\Big] \tag{4.31}$$

where $\mathbb{E}(\pi(a^t_{g_\sigma} | \mathbf{s}^t_{g_\sigma}; \psi^a)$ denotes the entropy regularization term added to the cost function to encourage exploration during the learning process. Here, ϕ controls the exploration rate. The loss function of the critic network ($\mathcal{L}^{Critic}_{g_\sigma}$) (Li *et al.*, 2020a) for each $g_\sigma \in \mathcal{G}_\sigma$ is expressed as

$$\mathcal{L}_{g_\sigma}^{Critic} = (\mathcal{E}_t)^2. \tag{4.32}$$

The parameters of the actor and critic networks are updated using gradients. This learning process continues for 200 learning steps, and then the system enters the monitoring mode, as detailed in Lines 19-22, where the system maintains the NS status in monitoring mode based on Eq. (4.22) without adjustment until the RIC observes that there is a change in $\mathbf{\Omega}_s$ according to Eq. 4.23 then the CoMA2C policy is triggered to adjust the resource among the heterogeneous services. Next, monitoring and adjusting process is repeated until convergence. During the training process of each $g_\sigma \in \mathcal{G}_\sigma$ in the proposed scheme, the dropout technique (Zhu & Tan, 2024) is applied to mitigate the risk of overfitting and enhance the generalization ability of the proposed model.

4.4.5.2 Stage II- Design and Learn intra-slice Policy

The architecture of each $j_s \in \mathcal{J}_s$ in the MADQN scheme of the proposed framework is designed based on the DQN algorithm adopted and detailed in (Sabr *et al.*, 2025a; Tran *et al.*, 2023) and the learning process illustrated in Algorithm 4.3. In Line 1 of Algorithm 4.3, we initialize the necessary setup for each NS, whereas in Line 2, we initialize $j_s \in \mathcal{J}_s$ to manage radio resources within each NS. Then, for each $j_s \in \mathcal{J}_s$, we define a replay buffer (Y_{j_s}) and two DNNs with identical architectures but different weights. The first DNN, referred to as the trained Q-network, is parametrized with weight ψ_{j_s}, while the second, the target Q-network, has weights $\psi_{j_s}^-$. An exploration rate (ϵ) and learning rate (α_{j_s}), are also defined, as explained in Lines 3-5 respectively. Lines 6-20 detail the learning process of each $j_s \in \mathcal{J}_s$, where during each $t \in \mathcal{T}^{\text{short}}$, $j_s \in \mathcal{J}_s$ observes its current state s_{j_s} and selects a joint action a_{j_s} from its $\mathcal{A}^s$. In line with this, each $j_s \in \mathcal{J}_s$ constructs its joint action according to ϵ-greedy strategy, given in Eq. (4.33) (Tran *et al.*, 2023).

$$a_{j_s}^t = \begin{cases} \text{random action } a_{j_s}, & \text{with probability } \epsilon, \\ \arg\max\limits_{a_{j_s} \in \mathcal{A}^s} \left\{ Q(s_{j_s}, a_{j_s}) \right\}, & \text{with probability } 1-\epsilon. \end{cases} \tag{4.33}$$

Algorithm 4.2 Pseudocode of CoMA2C Scheme

Initialization:
Initialize A2C-based LSTM; $\forall \sigma \in \Theta$.
Initialize SLA parameters; $\forall s \in \mathcal{S}$.
$\forall g_\sigma \in \mathcal{G}_\sigma$, initialize:
- Actor network: $\Psi^a_{g_\sigma}$.
- Critic network: $\Psi^c_{g_\sigma}$.
- Learning rates for actor: $\varrho^a_{g_\sigma} > 0$ and critic: $\varrho^c_{g_\sigma} > 0$.

Initialize NS state monitor to track traffic changes over time.
for *iteration* $i = 1$ **to** F_i **do**
  Learning Phase (First 200 Iterations):
  if $i < 200$ **then**
    for $t \in \mathcal{T}^{long}$ **do**
      for $g_\sigma \in \mathcal{G}_\sigma$ **do**
        Obtain state $\mathbf{s}_{g_\sigma}$ according to Eq. (4.18) from heterogeneous network slices.
        Perform the action $a^t_{g_\sigma} \in \{\mathbf{a}^P_g, \mathbf{a}^B_g\}$ to allocate power $\mathbf{P}^s_{\max}$ and bandwidth $\mathbf{B}^s_{\max}$ based on Eqs. (4.19) and (4.20).
        Manage $P^s_{\max}$ and $B^s_{\max}$ among active users in each $s \in \mathcal{S}$ at intra-RAN slicing by using MADQN scheme.
        Check SLA, SSR requirements for each $s \in \mathcal{S}$ and system performance η.
        Calculate the reward based on Algorithm 4.1 and move to the next state $\mathbf{s}^{t+1}_{g_\sigma}$.
        Critic calculates the corresponding TD error using Eq. (4.30).
        Calculate actor loss and critic loss using Eqs. (4.31) and (4.32), respectively.
        Update actor and critic networks:
        $\Psi^a_{g_\sigma}[t+1] \leftarrow \Psi^a_{g_\sigma} + \varrho^a_{g_\sigma} \nabla \mathcal{L}^{Actor}_{g_\sigma}(\Psi^a_{g_\sigma})$,
        $\Psi^c_{g_\sigma}[t+1] \leftarrow \Psi^c_{g_\sigma} + \varrho^c_{g_\sigma} \nabla (\mathcal{E}_t)^2$.
        $t = t + 1$
      end for
    end for
  end if
  else
    Monitoring and Learning Phase ($i > 200$)**:**
    Monitor $\mathbf{\Omega}_s$ changes based on Eq. (4.22).
    Trigger actor networks according to Eq. (4.23), to select or adjust the radio resources (Θ) based on current traffic loads.
    Repeat steps 6-18.
  end if
end for
Output: Best policy $\pi^*_{g_\sigma}\left(\mathbf{a}^t_{g_\sigma} \mid \mathbf{s}^t_{g_\sigma}, \forall g_\sigma \in \mathcal{G}_\sigma\right)$.

where the level of exploration is determined by the ϵ, which is decreased over time in order to lower the exploration rate as the learning advances.

Following this, each $j_s \in \mathcal{J}_s$ gets its reward r_{j_s} as defined in Eq. (4.26), and then transitions to a new state denoted by $s_{j_s}^{t+1}$. Subsequently, each $j_s \in \mathcal{J}_s$ sends its experience $(s_{j_s}, a_{j_s}, r_{j_s}, s_{j_s}^{t+1})$ to be stored in its replay buffer Y_{j_s}. Once a sufficient number of experiences have been stored, each $j_s \in \mathcal{J}_s$ selects a random mini-batches M_{batch} of 32 samples from its own Y_{j_s} to train its trained Q- network. The aim of training process is to minimize the loss function ($\mathcal{L}_{j_s}$), which is calculated as follows (Ge *et al.*, 2020):

$$\mathcal{L}_{j_s}(\psi_{j_s}) = \frac{1}{2M_{batch}} \sum_{\langle s_{j_s}, a_{j_s}, r_{j_s}, s_{j_s}^{t+1} \rangle \in Y_{j_s}} \underbrace{(V_{j_s}^{T}}_{\text{Target Q value}} \underbrace{- Q(s_{j_s}, a_{j_s}; \psi_{j_s}))^2}_{\text{Q value}}, \tag{4.34}$$

where $V_{j_s}^{T}(t) = r_{j_s}^{t} + \gamma \max_{a'_{j_s} \in \mathcal{A}^s} Q\left(s_{j_s}^{t+1}, a'_{j_s}; \psi_{j_s}^{-}\right)$ denotes the target value, determined through the target network (Tran *et al.*, 2023). Upon calculating $\mathcal{L}_{j_s}$, each agent uses an optimizer to adjust the parameters of the trained Q-network. Then, the parameters of the target Q-network are updated at predetermined intervals ($\mathbb{T}_{step}$) to mirror the training Q-network. This procedure is repeated until convergence.

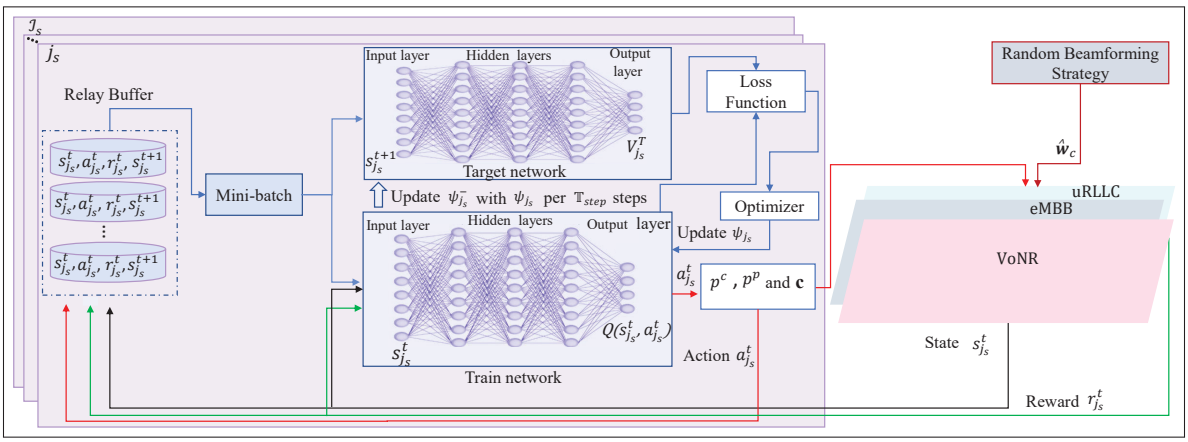

Figure 4.3 An illustration of the MADQN scheme for joint resource management in heterogeneous intra-RAN slicing

Algorithm 4.3 Pseudocode of MADQN Scheme

1 : NS Initialization:
  1.1: Queue buffer, Latency buffer, Traffic model; $\forall s \in \mathcal{S}$.
  1.2: User location, User velocity $\forall s \in \mathcal{S}$.
2: Initialize j_s; $\forall s \in \mathcal{S}$.
3: Establish Y_{j_s} with a size limit of $\aleph$; $\forall j_s \in \mathcal{J}_s$.
4: Initialize the set up of two DNNs: trained Q-network with ψ_{j_s} and target Q-network with $\psi^-_{j_s}$ $\forall j_s \in \mathcal{J}_s$.
5: Set the initial ϵ and α_{j_s}; $\forall j_s \in \mathcal{J}_s$.
6: **for** $j_s \in \mathcal{J}_s$ **do**
  7: **for** *each training episode* **do**
    8: Initialize the NS state $\forall j_s \in \mathcal{J}_s$.
    9: **for** $t \in \mathcal{T}^{short}$ **do**
      10: After receiving the corresponding $P^s_{\max}$ and $B^s_{\max}$ from the CoMA2C scheme.
      11: Get s_{j_s} using Eq. (4.24).
      12: Select a_{j_s} based on ϵ greedy policy in Eq. (4.33).
      13: Execute joint a_{j_s} using Eq. (4.25) and receive r_{j_s} as defined by Eq. (4.26) and moves to $s^{t+1}_{j_s}$.
      14: Save the experience $(s_{j_s}, a_{j_s}, r_{j_s}, s^{t+1}_{j_s})$ in Y_{j_s}.
      15: Randomly sample M_{batch} from Y_{j_s} for training.
      16: Calculate target Q-value.
      17: Determine the $\mathcal{L}_{j_s}$ between the trained network and the target network according to Eq. (4.34).
      18: Update the ψ_{j_s} of trained DQN by performing a gradient decent step.
      19: Update the target network parameters $(\psi^-_{j_s})$ as $\psi^-_{j_s} = \psi_{j_s}$ every $\mathbb{T}_{step} =$ 200 steps.
      20: Repeat until convergence.
    end for
  end for
end for
Output: Best policy $\pi^*_{j_s}\left(a_{j_s} \mid s_{j_s}, \forall j_s \in \mathcal{J}_s\right)$.

4.5 Experiments and Performance Evaluations

This section evaluates the performance of our HiSO-CoMA framework.

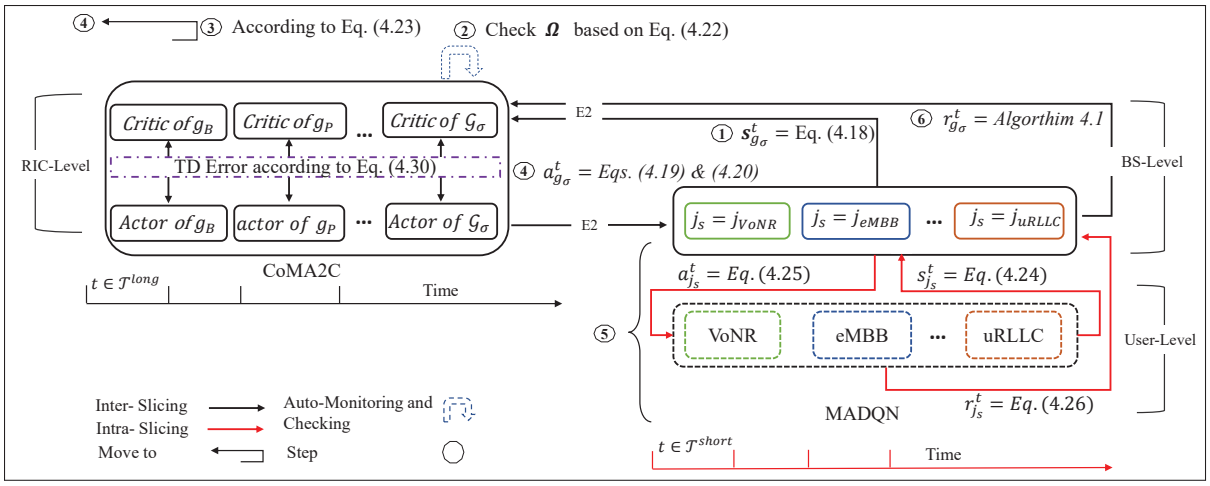

Figure 4.4 Schematic view of the proposed HiSO-CoMA framework for hierarchical heterogeneous multi-slice MISO systems

4.5.1 Simulation Settings

We consider the scenario shown in Fig. 4.1, where a single BS, controlled by RIC, hosts three heterogeneous slices. The users are distributed among the three services based on predefined slice probabilities, as shown in Table 4.2. The BS, with a coverage radius r, is located within a simulation area of 240 m × 240 m (Li *et al.*, 2020a). Users within the same slice are assumed to have similar mobility patterns, including both velocity and direction. When $u_s \in \mathcal{U}_s; \forall s \in \mathcal{S}$, reach the boundary of the simulation area, its direction is reflected, according to the mobility model in (Li *et al.*, 2020a). For simplicity, the transmission bandwidth for each slice is managed using the RR scheduling method (as bandwidth management at the intra-slice level is beyond the scope of this study).

All numerical experiments were conducted using Python 3.11 with TensorFlow on the 11th Gen Intel(R) Core(TM) i9-11900 PC with 64 GB of RAM, without GPU acceleration. The software frameworks used in this study include Spyder and MATLAB. Extensive simulations were performed to identify the best hyperparameter values for training the CoMA2C and MADRL schemes. The hyperparameters used in the setup of the CoMA2C and MADQN schemes are illustrated in Table 4.3.

Table 4.2 Main parameters and their descriptions

Parameter	Value
U_s	100
r (m)	40 (Li *et al.*, 2020a)
Probability of users in each NS	VoNR =1, eMBB= 2, and uRLLC = 3
User velocity	2 m/s, 6 m/s, 10 m/s and 14 m/s
Size of ULA	128
B_{code}	128
N_t^s	5
$\mathbb{P}^T$ and $\mathbb{B}^T$	60 dBm and 10 MHz (Li *et al.*, 2020a), respectively
Λ_p^s and Λ_b^s	15 dBm and 2 MHz, respectively
κ^t and κ^r	$1e^{-6}$, 0.0001, 0.001, 0.01, and 0.05
Ξ_{u_s}	5 packets (Yan *et al.*, 2023)
$\sigma^2_{u_s}$	-174 dBm
d^n	$\lambda/2$ (Ge *et al.*, 2020)
Packet size (Li *et al.*, 2020a)	40 Byte (VoNR), 300-1500 Byte (eMBB), and 32 Byte or (6.4, 12.8, 19.2, 25.6, 32) KByte (uRLLC)
∇^{Th}	10%
σ_{sf}	8 dB (Ge *et al.*, 2020)
L	4 (Ge *et al.*, 2020)
Δ	3° (Ge *et al.*, 2020)
$\mathfrak{B}_s$	[1, 2, 3] for VoNR, eMBB, and uRLLC, respectively
α^{η}	0.6
VoNR user inter-arrival distribution	Uniform distribution between 0 and 160 ms (Hua *et al.*, 2020)
eMBB user inter-arrival distribution	Pareto distribution with the mean of 6 ms and the maximum of 12.5 ms (Hua *et al.*, 2020)
uRLLC user inter-arrival distribution	Exponential distribution with an average time of 180 ms (Hua *et al.*, 2020)

Table 4.3 Training Parameters for CoMA2C and MADQN

CoMA2C Parameters	
Number of state elements	3
$\varrho^a_{g_\sigma}$ and $\varrho^c_{g_\sigma}$	5e-4 and 2e-3, respectively
Optimizer	RMSProp
Υ	0.999
Size of LSTM cells	One LSTM layer with 64 neurons
Entropy rate	0.01
Dropout rate	0.1
Actor network	Two fully connected layers, each with 32 units and ReLU activation
Critic network	Two fully connected layers, each with 32 units and ReLU activation
Simulation time	1,000 time slots
MADQN Parameters	
Replay memory size ($\aleph$)	1000
Optimizer	Adam
Discount factor (γ)	0.95
Update $\psi^-_{j_s} = \psi_{j_s}$	Every 200 time steps

* Other parameters of MADQN are similar to those in (Sabr *et al.*, 2025a).

4.5.2 Benchmark Algorithms

We validate the efficacy of our proposed HiSO-CoMA framework through comprehensive experimental analysis under various parameter settings. To this end, we compare our results to the following state-of-the-art (SOTA) and traditional schedulers.

- **SOTA scheduler:** This scheduler is designed to be identical to the proposed framework in structure and uses the same DRL algorithms to ensure a fair comparison. The only difference lies in the inter-slice resource allocation strategy, which follows the SOTA approach, where resources are allocated at every time step t, regardless of the actual need for the allocation.
- **RRA scheduler:** This scheduling approach uses a random allocation algorithm to manage both inter- and intra-RAN slicing.
- **GGA scheduler:** This scheduling approach employs a greedy allocation algorithm to manage both inter- and intra-RAN slicing.
- **EEA scheduler:** This scheduling approach utilizes an equal or hard allocation algorithm for both inter- and intra-RAN slicing.
- **SA2C-T scheduler (Sabr *et al.*, 2025b):** This scheduler is one of the most recent relevant state-of-the-art methods. It employs a single A2C-based LSTM to manage the power and bandwidth at the inter-slice level across heterogeneous slices, whereas traditional algorithms are used for resource management at the intra-slice level. SA2C-T was designed based on OFDMA and does not consider beamforming. To ensure a fair comparison and align with the state-of-the-art, we adapted the ZF technique, commonly used in the literature, to optimize the beam direction in SA2C-T.

4.5.3 Convergency Analysis of the proposed HiSO-CoMA

In the first experiment, the convergence of the proposed HiSO-CoMA framework is evaluated. Convergence is measured by the rate at which the loss function decreases over time during the training of the CoMA2C agents at the inter-RAN slicing level (e.g., Eq. (4.31) and Eq. (4.32)) and MADQN agents at the intra-RAN slicing level (Eq. (4.34)). Figure 4.5 shows the variations in training loss for both the inter- and intra-level control policies, where we can observe that

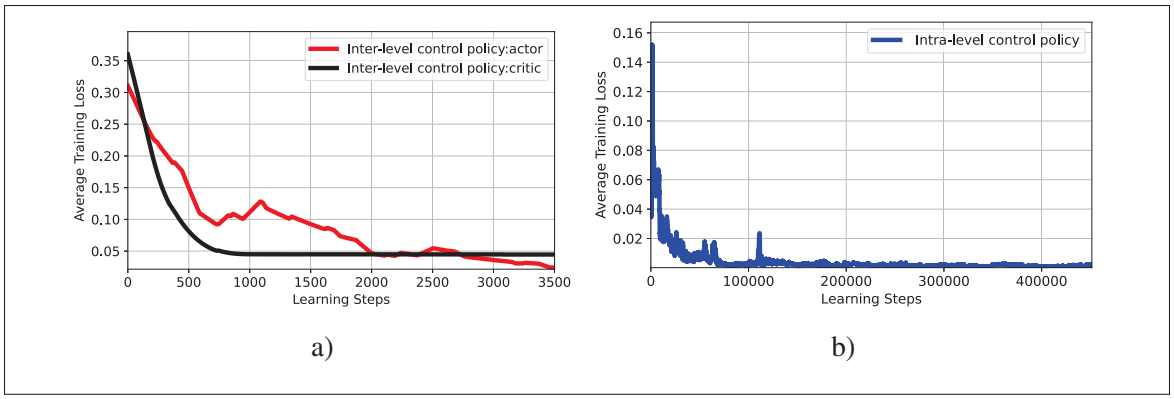

Figure 4.5 Convergence of the proposed framework under HWIs: a) Average training loss of the control policy for inter-RAN slicing; b) Average training loss of the control policy for intra-RAN slicing

the losses of both policies decrease as training progresses. This demonstrates the stability and learning effectiveness of the proposed framework, despite the presence of various confounding factors, such as user mobility, HWIs, and fluctuating traffic loads. In addition, this confirms that the proposed HiSO-CoMA framework effectively addresses the challenges of mis-convergence, synchronization and instability in learning the control strategy, as discussed in Section 4.4.4.

4.5.4 Evaluation of the Proposed HiSO-CoMA Vs. the SOTA

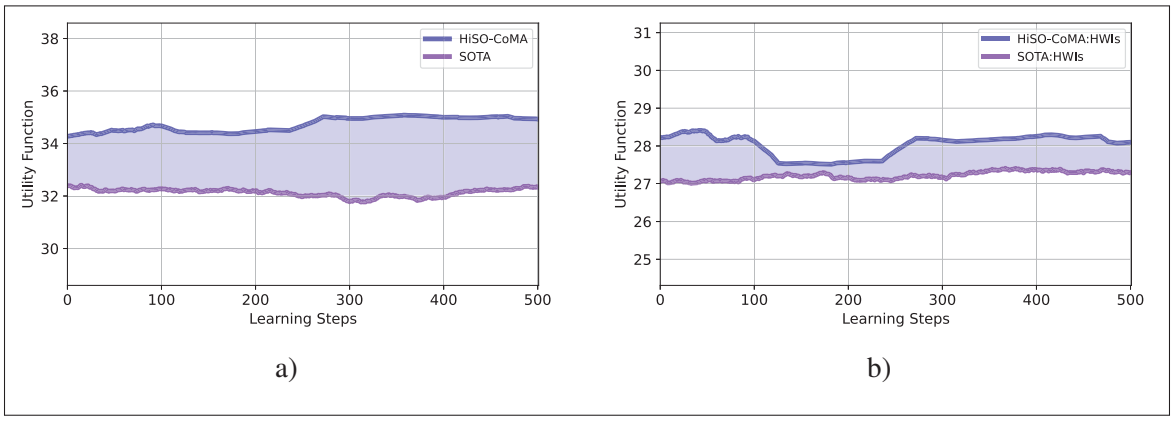

Figure 4.6 Utility function of the proposed framework vs. the SOTA approach under (a) ideal HW and (b) HWIs

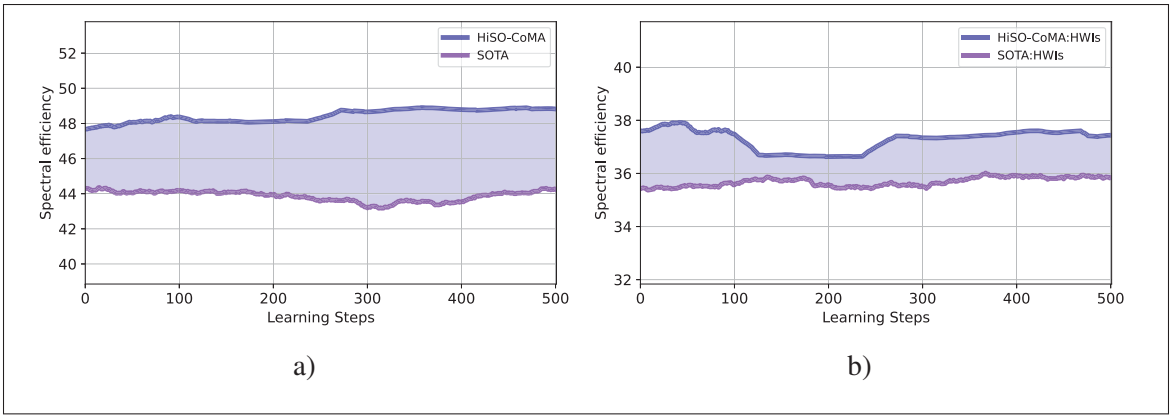

Figure 4.7 Spectral efficiency of the proposed framework vs. the SOTA approach under (a) ideal HW and (b) HWIs

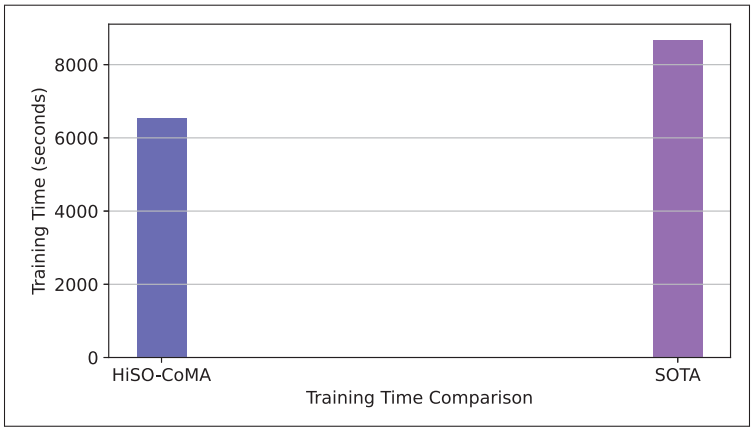

Figure 4.8 Training time of the proposed framework vs. the SOTA approach in the presence of HWIs

Figures 4.6 and 4.7 evaluate the performance of the proposed framework in terms of utility and spectral efficiency, respectively, under ideal and non-ideal HW conditions. We evaluate the proposed approach and compare it to the SOTA approach, in which the former, triggers the inter-RAN slicing policy only when a significant change in traffic demand is detected for admitted services, whereas in the latter, inter-slice resource allocation is performed at every learning step (*i.e.*, every 1 second). From Figs. 4.6 and 4.7, it can be observed that the proposed framework outperforms the SOTA approach for both performance metrics and under both ideal

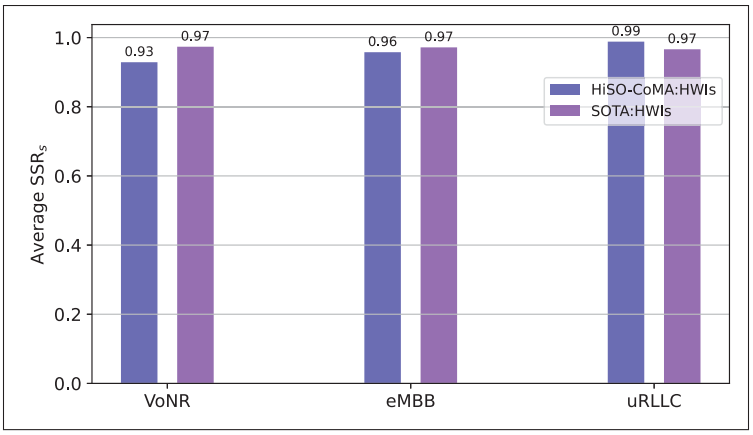

Figure 4.9 Average QoS of the proposed framework vs. the SOTA approach under HWIs

and non-ideal HW conditions. The SOTA approach's inferior performance can be attributed to its lack of traffic change detection; the agent continuously updates its policy, regardless of necessity. This leads to *policy churn*, a phenomenon in which beneficial policies are unnecessarily updated, destabilizing previously learned advantageous behaviors and degrading the overall policy's performance. In contrast, the proposed approach updates the resource allocation and learning policy only when needed, effectively stabilizing learning and improving resource allocations. These results highlight the effectiveness of the proposed framework in enhancing the overall learning process and ensuring more efficient resource management compared to the SOTA approach. Furthermore, we observe that HWIs affect the learning stability during initial time steps (0–300), as shown in Figs. 4.6b and 4.7b. However, this does not impact the overall convergence time, demonstrating the adaptability and robustness of the proposed framework under varying conditions.

Figure 4.8 illustrates the training time of the proposed HiSO-CoMA framework compared to the SOTA scheduler. As shown in the figure, the proposed framework significantly reduces training time relative to the SOTA scheduler. This improvement is due to the limited information exchange between slicing levels, such as states, rewards, and actions—since the CoMA2C scheme is triggered only when necessary, unlike the SOTA scheduler, which updates the system at every time step. This selective triggering reduces communication overhead, particularly in the

communication between CoMA2C and MADQN agents. Consequently, the proposed framework shows potential for enabling ZT operations in RAN slicing by offering promising management strategies.

Figure 4.9 shows that the proposed HiSO-CoMA framework achieves comparable QoS compared to the SOTA scheduler for eMBB and uRLLC slices. However, the SSR of VoNR decreases by 4% under the proposed framework, which highlights the need for instantaneous budget updates for the VoNR slice. This drop could be attributed to the nature of VoNR's traffic model, which follows a uniform inter-arrival time distribution (0–160ms). This distribution could generate irregular traffic variations that often remain below the 10% threshold required to trigger inter-slice budget reallocation. As a result, the RIC cannot initiate budget adjustments for VoNR if its traffic load fails to meet the predefined threshold. This could lead to gradual resource misalignment and a slight decrease in performance of VoNR service. Nevertheless, the VoNR slice still maintains an SSR above 90%, demonstrating the robustness of the proposed framework despite this potential limitation.

4.5.5 Performance of the proposed HiSO-CoMA .Vs Baselines

Figure 4.10 shows the performance of the proposed HiSO-CoMA framework in optimizing the objective function under both ideal and non-ideal HW conditions, and compares it with various resource allocation schedulers. It is observed that the proposed framework outperforms the baseline schedulers under both conditions. Unexpectedly, SA2C-T, which is based on heterogeneous optimization methods, exhibits the poorest performance among all evaluated approaches. This performance may be attributed to the heterogeneity of the applied methods, which likely results in a weak synchronization between the two allocation levels. Among traditional schedulers, RRA and EEA demonstrate strong performances; however, they are not suitable for real-world deployment, as their allocation strategies lack the smart policies needed for future applications, particularly in terms of adaptability and self-learning capabilities. In addition, their efficiency in meeting the requirements of NS in 6G is not as high as that of the proposed HiSO-CoMA framework, which maintains system utility above 30 and 25 under ideal

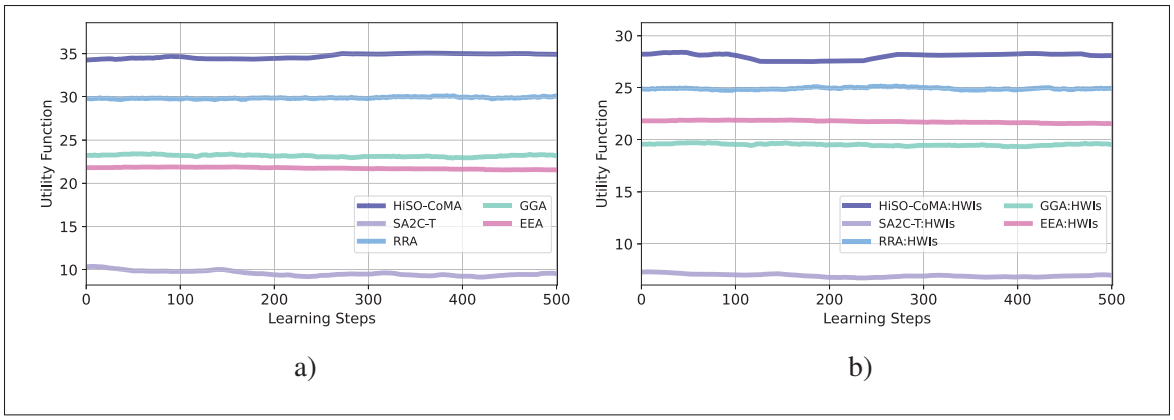

Figure 4.10 Utility function of the proposed framework vs. baselines under (a) ideal HW and (b) HWIs

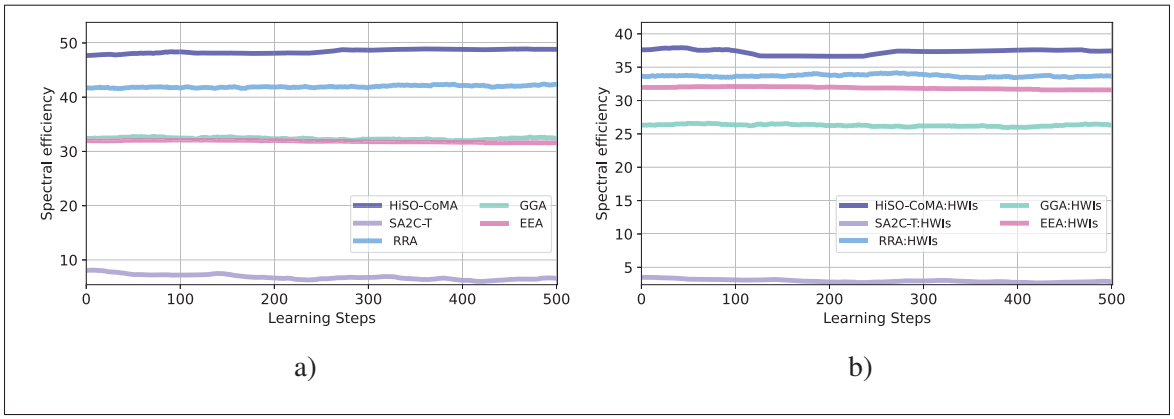

Figure 4.11 Spectral efficiency of the proposed framework vs. baseline schedulers under (a) ideal HW and (b) HWIs

and non-ideal HW conditions, respectively. Moreover, the proposed HiSO-CoMA framework employs smart strategies for managing limited resources, ensuring that allocation is based on the instantaneous needs of each slice and according to the actual traffic load, rather than relying on a random or equal resource distribution.

From Fig. 4.11, we observe that the proposed framework achieves the highest spectral efficiency of more than 40 bps/Hz under ideal conditions and 35 bps/Hz under HWI. This highlights the efficiency of the proposed framework in managing resources both among and within heterogeneous services.

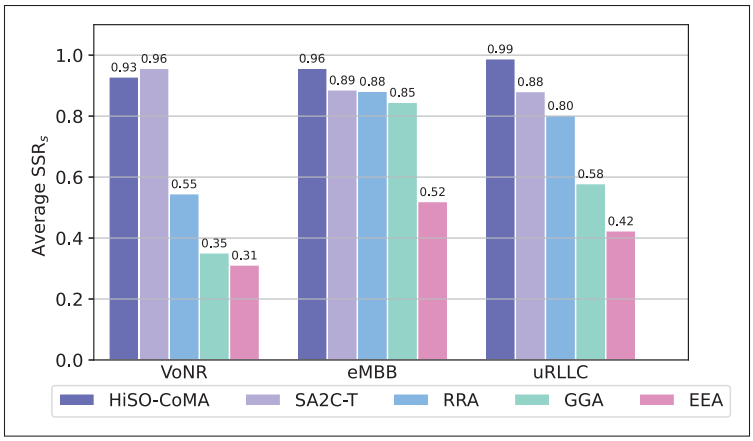

Figure 4.12 Average SSR for heterogeneous services under HWIs

Figure 4.12 shows the performance of the proposed framework in satisfying the QoS of heterogeneous slices under HWI conditions, and compared with various resource allocation schedulers. We can observe that the proposed framework outperforms the benchmarks in terms of maximizing QoS. Among the baseline schedulers, SA2C-T, which incorporates DRL in its design, achieves better performance than the GGA, RRA, and EEA schedulers. This demonstrates the ability of the DRL algorithm to learn from the assigned reward, which acts as a guiding signal for the agent to achieve the desired performance across heterogeneous services.

4.5.6 Impact of mobility on average QoS under HWIs

The effect of mobility on the average QoS for heterogeneous services using different allocation schedulers under HWIs is shown in Fig. 4.13. In this context, higher user speeds result in a highly dynamic environment, where fluctuations in channel conditions lead to a reduction in the SDNR, ultimately decreasing data rates. This, in turn, affects the transmission rates and increases packet latency. From Fig. 4.13, we observe that the proposed framework demonstrates excellent performance in maintaining strict SLAs across heterogeneous slices even under varying user velocities. SA2C-T achieves the second-best performance, providing comparable QoS for both VoNR and eMBB services, and outperforming GGA, RRA, and EEA schedulers across all service types. The outstanding performance of the proposed framework can be attributed to

several factors, including the reliable resource allocation across slices and robust synchronization in the learning process. Furthermore, the framework effectively addresses demand fluctuations and high user mobility at the intra-slice level by dynamically adjusting resources based on traffic load at both the inter- and intra-slice levels. Overall, the results highlight that to ensure efficient use of limited resources and enable a full automation in O-RAN, it is essential to manage the resource allocation across multiple slicing levels using DRL. We also note that schedulers leveraging DRL, either fully or partially, achieve better QoS performance compared to traditional schedulers.

4.5.7 Impact of Packet Size on HiSO-CoMA Vs. Baselines

To investigate the influence of packet size variations on utility and QoS, the eMBB slice is selected for testing due to its characteristic use of larger packet sizes compared to other slice types. To this end, we fix the minimum rate of the eMBB packet at 15 Mbps and vary the packet size, as shown in Figs. 4.14 and 4.15. It is observed that larger packet sizes result in degraded utility and QoS. This is due to the fact that large packets require extremely high data rates to be transmitted within one time slot. If the data rate is insufficient, the packet is divided into subframes, which also requires a high data-transmission rate. Otherwise, this increases packet latency, which in turn affects the SLA of the slice, leading to poor QoS. Despite this, we observe that the proposed framework outperforms all baseline schedulers in terms of maximizing the utility function across all the packet size range. This demonstrates the reliability and efficiency of the proposed framework compared to other approaches. However, we can observe that SA2C-T yields comparable results in terms of meeting the QoS requirements of eMBB for the considered packet sizes. This highlights the efficiency of the DRL-based scheduler compared to traditional schedulers, particularly in terms of adaptability to changing packet sizes and meeting the SLA of eMBB services.

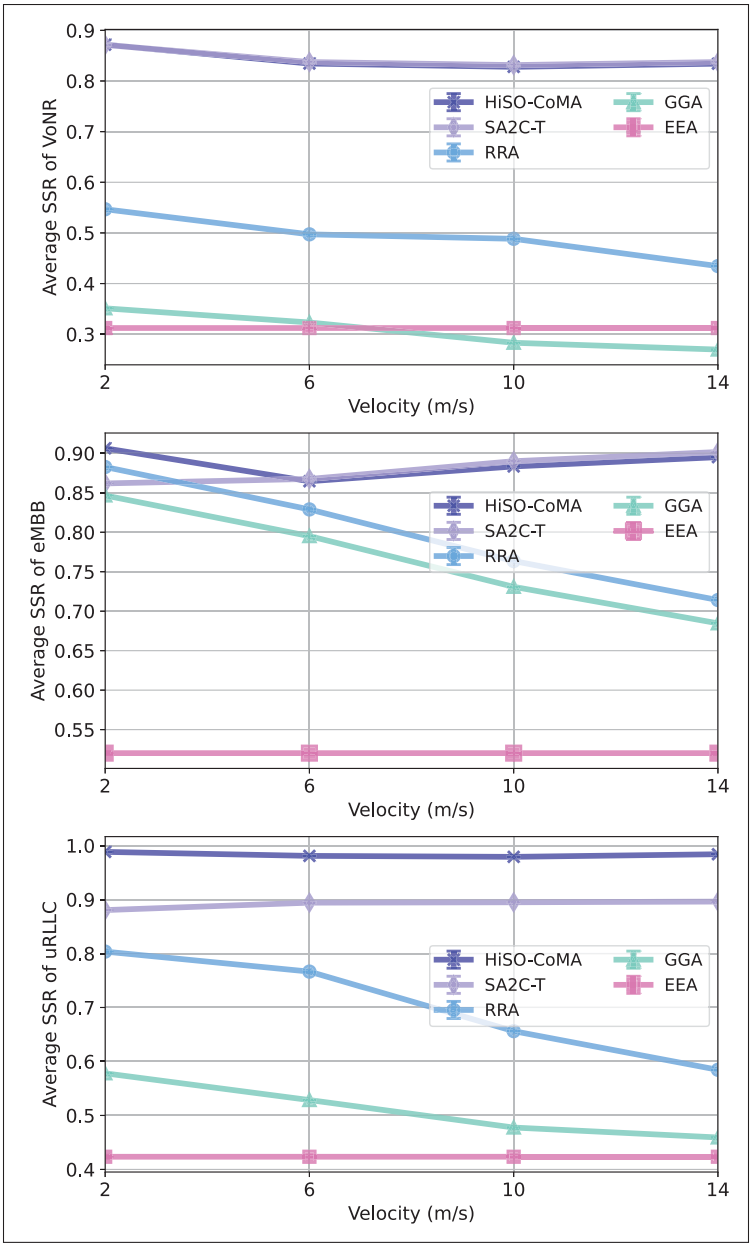

Figure 4.13 Impact of mobility on average QoS under various allocation schedulers

4.5.8 HiSO-CoMA framework under various HWIs

To evaluate the reliability of the proposed HiSO-CoMA framework against distortions, we test its performance under varying HWIs levels. As shown in Fig. 4.16, both the utility function and the spectral efficiency degrade as severity of HWIs increases. This highlights the effect

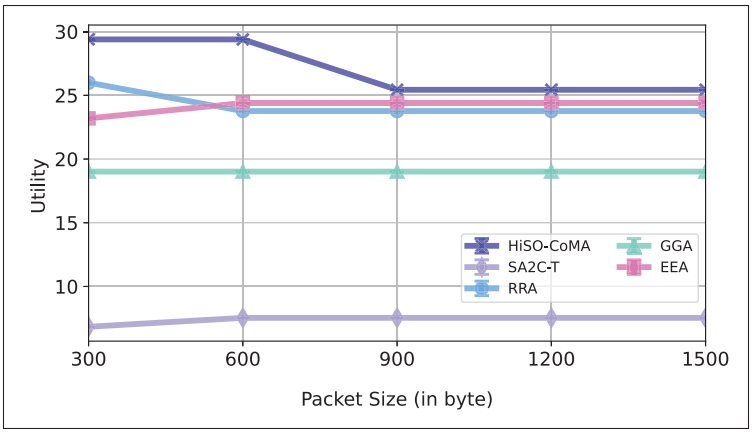

Figure 4.14 Utility of the proposed HiSO-CoMA framework vs. baselines strategies under HWIs and varying packet sizes for the eMBB slice

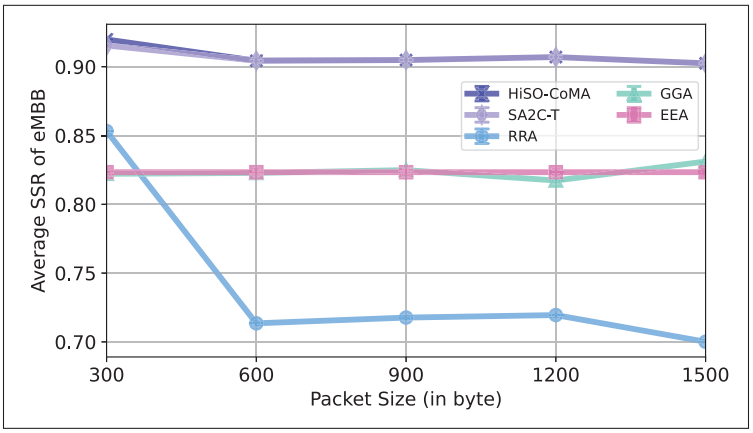

Figure 4.15 Average QoS of the eMBB slice for various packet sizes under HWIs

of hardware distortions on the performance of future applications. Nonetheless, the proposed framework consistently outperforms all baseline schedulers in maximizing utility and achieving higher spectral efficiency across varying HWI levels. More specifically, the results indicate that the framework maintains strong performance with hardware resolution levels of up to 0.05 at both the transmitter and receiver. However, for more severe impairments, integrating mitigation algorithms is essential to reducing the impact of HWI in real-world deployment scenarios. We also observe that, unexpectedly, SA2C-T scheduler exhibited the poorest performance among all

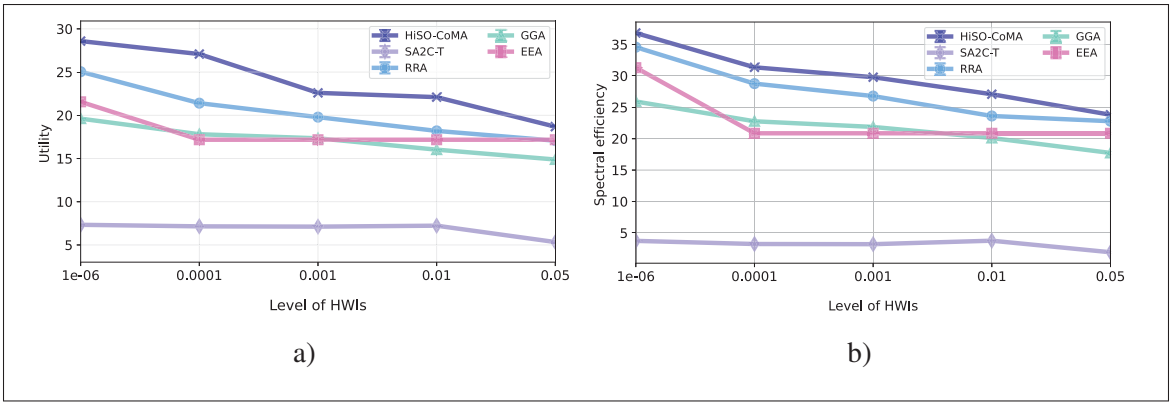

Figure 4.16 Utility and spectral efficiency of the proposed framework vs. baselines under various levels of HWIs

baseline methods. This may be attributed to the use of the ZF technique in its design, which appears to be more sensitive to HWIs compared to codebook-based techniques employed in the proposed framework and other benchmark schedulers.

Fig. 4.17 shows that the training time of the proposed framework increases gradually rather than abruptly when exposed to severe HWIs ($\kappa^t = \kappa^r = 0.05$). In this context, the training time relatively stable over the considered range of HWIs. This indicates that HWIs can affect the training time of DRL-based algorithms, highlighting the need for further investigation into mitigation techniques that could reduce their impact.

Finally, our findings confirm the effectiveness and resilience of the HiSO-CoMA framework, which consists of two management levels. The results demonstrate that the adaptation of RSMA significantly reduces interference for latency-sensitive services. Furthermore, intelligent power allocation and beamforming optimization, performed by MADQN agents, are dynamically adjusted based on channel conditions and service requirements. At the same time, overall resource budgets are optimally adjusted by CoMA2C based on slice traffic load. This learning occurs seamlessly, with strong coordination between the upper and lower management levels. This integration ensures that the system effectively maintains slice isolation as demand increases, while also guaranteeing that critical services, such as eMBB and URLLC, are allocated sufficient resources to meet their QoS requirements. Additionally, we found that the overall

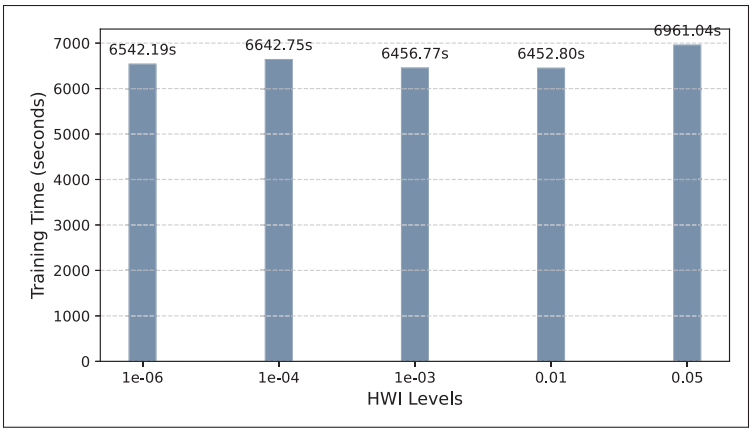

Figure 4.17 Training time of proposed HiSO-CoMA framework under various levels of HWIs

convergence of the proposed framework depends on the proper hyperparameters tuning for both learning schemes. Choosing the right hyperparameters is crucial for enabling effective synchronization of the learning processes across the management layers. This, in turn, ensures optimal resource allocation strategies that account for the relative importance of each service type. To identify suitable hyperparameters, we conducted an extensive empirical tuning process by thoroughly exploring various combinations of learning rates, discount factors, and neural network architectures. The tuning process was guided by continuous observation of reward trends and loss function stability over training episodes. This iterative process enables the selection of proper parameters that ensures stable convergence and optimal performance of the multi-level framework for heterogeneous network slicing.

4.6 Conclusion

In this study, we proposed an intelligent RAN slicing framework composed of two key schemes: CoMA2C for inter-slice management and MADQN for intra-slice management. Together, these schemes form a hierarchical self-optimizing framework. The proposed framework adopts RSMA as a promising technology to support the coexistence of heterogeneous services. Simulations conducted under various conditions, including user mobility, time-varying channels, and HWIs,

demonstrate that the proposed framework not only achieves stable convergence and superior performance but also offers an effective strategy for significantly reducing NS overhead, thereby enhancing overall QoS compared to baseline schemes.

In the future, we will focus on further developing a self-learning framework that can effectively integrate additional slices and radio resources at both inter-slice and intra-slice levels, while remaining robust against imperfect channel state information and hardware distortions. In addition, further research is needed to explore strategies that achieve a balance between performance improvements and the computational complexity inherent in DRL-based approaches.

CONCLUSION AND RECOMMENDATIONS

In this Chapter, we present the conclusions of the thesis (Section 5.1), followed by an outline of future research directions (Section 5.2).

5.1 Conclusions

The next generation of wireless systems aims to support a wide range of heterogeneous services and applications, each with diverse QoS requirements in terms of data rate, latency, and reliability. NS provides mobile networks with the architectural flexibility required to support these heterogeneous services in next-generation networks. In addition, DRL algorithms are expected to play a central role in simplifying network management and optimization by empowering the ZTNs paradigm as the primary management framework for future networks. More specifically, by providing ZT management that improves system performance through dynamic resource allocation and enables adaptive management and control in highly dynamic environments, ZTNs can play a significant role in enhancing operations in the RAN slicing domain. This, in turn, enables the system to ensure efficient resource utilization while satisfying diverse QoS demands in the presence of various network challenges.

In the context of ZT management for NS, in this thesis, we proposed a cooperative MADRL-based framework to support the coexistence of heterogeneous services in MU-MISO systems. Specifically, we designed, implemented, and evaluated self-optimization schemes for managing radio resources on both the inter- and intra-slice levels of RAN slicing in next-generation systems. Furthermore, we examined a hierarchical RRM framework within the O-RAN slicing architecture. The proposed schemes address several key challenges, including the coexistence of heterogeneous services with diverse SLA requirements, fluctuating traffic loads, and the need to meet constraints such as transmit power allocation, beamforming design, latency, reliability, and data rate targets within each slice, along with power and bandwidth allocation across slices.

In Chapter 2, we examined the orthogonal coexistence of eMBB and uRLLC services with heterogeneous QoS requirements. Specifically, we proposed a self-optimization scheme for intra-RAN slicing, named PABSO-DRL, designed to operate under ICSI and imperfect OFDMA conditions. The main objective of the PABSO-DRL scheme was to jointly manage power and beamforming for eMBB and uRLLC slices while maximizing the data rate for the former and minimizing the outage probability for the latter. The resulting multi-objective optimization problem was reformulated as an MDP and then effectively solved using a multi-agent DQN-based approach. Simulation results demonstrated that the proposed PABSO-DRL scheme provides a higher QoS for users in each slice than benchmark algorithms. Furthermore, the scheme was found to exhibit enhanced scalability and was designed to flexibly accommodate various sizes of ULA of antennas, which is essential for future wireless systems where massive antenna deployments are expected to become standard components of future networks.

Furthermore, in Chapter 3, we focused on inter-RAN slicing. In this context, we proposed a secure SO-CoMA2C scheme to manage power and bandwidth across heterogeneous services under fluctuating traffic loads and HWIs. The proposed scheme was designed to maximize spectral efficiency while satisfying diverse SLA requirements of each slice. The problem was modeled as a POMDP and then decomposed into subproblems, which were solved using a cooperative multi-agent A2C framework. To ensure secure resource allocation, the AES algorithm was integrated into each A2C agent to protect communication between NS environments from unauthorized access that could manipulate decision making when exchanging information over the E2 interface. Simulation results showed that, under both ideal and non-ideal HW conditions, the proposed secure SO-CoMA2C scheme outperforms benchmark algorithms, such as SO-CoMDQN, SO-SA2C, and MR-HA, in terms of maximizing spectral efficiency and meeting the SLA requirements of heterogeneous slices. The results also demonstrated substantial reductions in computational overhead, where the integration of AES introduced only a minimal cost, leading to a 0.13% increase in training time and a 2.3% increase in memory

usage. Furthermore, the findings highlighted the promising scalability of the proposed scheme, successfully supporting up to five slices and outperforming benchmark approaches, as well as its ability to effectively learn under different state–action spaces, demonstrating strong robustness. Taken together, these outcomes point to cooperative learning as a promising approach to achieve robust and efficient resource management across diverse service types in future RAN slicing architectures.

Finally, in Chapter 4, we addressed the coexistence of VoNR, eMBB, and uRLLC services in O-RAN by jointly considering resource management on both the inter- and intra-RAN slicing levels. On the inter-slice level, we analyzed the allocation of power and bandwidth, whereas on the intra-slice level, we focused, on power and beamforming optimization. To ensure effective slice coexistence, orthogonal multiple access was applied across slices, whereas RSMA was employed within each slice in the presence of HWIs. The problem was formulated to maximize a utility function that balances overall spectral efficiency and SLA satisfaction for each slice. We decoupled the inter-slicing and intra-slicing; then, the inter-slice resource management challenge was modeled as a POMDP, while the intra-slice problem was captured through an MDP. To address these challenges, we proposed a hierarchical framework named HiSO-CoMA. Simulation results revealed that the proposed framework outperforms the baseline methods by maximizing system utility, thus ensuring SLA satisfaction for all slices. It also learned effective power and beamforming strategies for active users even under HWIs levels of up to 0.5 resolution and with a minimum overheads. The obtained results shed light on the successful design of the proposed framework, which integrates different DRL algorithms assigned to distinct decision-making levels. For example, A2C forms the CoMA2C scheme on the inter-slice level, whereas DQN forms the MADQN scheme on the intra-slice level. These schemes operate in parallel, enabling agents at each level to concurrently learn and execute their policies, thereby effectively leveraging the distributed nature of wireless tasks. This capability provides a significant advantage for

achieving scalability in future DRL-driven resource management and demonstrates feasibility of enabling learning and synchronization among heterogeneous DRL agents.

Overall, the proposed decentralized MADRL schemes provide a robust foundation to cope with the heterogeneity and density of next-generation RAN slicing, showing promising potential for the application of DRL-based algorithms in future NS.

5.2 Future work

The results reported in the present thesis contribute to the development of intelligent next-generation O-RANs and demonstrated how cooperative MADRL can efficiently manage radio resources while meeting the SLA requirements of network-sliced services. Despite these contributions, several open challenges remain. In what follows, we present potential avenues for future investigations to further advance ZT in RAN slicing and enhance their real-world applicability.

5.2.1 Integrating Explainable AI for Improved RRM Transparency and Trust

In Chapters 2-4, we proposed RRM schemes designed to support heterogeneous services with diverse QoS requirements. A promising direction for future research is to develop an intelligent mechanism based on explainable AI (XAI) methods and integrate it to operate in parallel with the proposed schemes. Such an approach could enhance accuracy, reliability, transparency, interpretability, and fairness of DRL-based real-time decision making in ZTN environments.

5.2.2 Enhancing Buffer Security for Reliable Data Handling in RAN Slicing

In Chapters 3 and 4, we proposed a queuing system as part of the design of the secure SO-CoMA2C scheme and the HiSO-CoMA framework, where a limited-size buffer is used to store the data traffic of each user at the BS. A promising extension of this work would be to strengthen

buffer security at the BS by introducing mechanisms that verify authenticity of each incoming packet and prevent the injection of malicious or fake packets that could overflow the buffers of specific users within a given slice. This enhancement would ensure that the BS stores only legitimate packets, thereby preventing data loss and improving the overall system reliability.

5.2.3 Exploring Federated and Transfer Learning for Enhanced DRL Efficiency and Security

In Chapters 2 and 4, we proposed a DRL algorithm based on decentralized training, where each agent learns and updates its policy using the data associated with its corresponding slice. This work could be extended by exploring integration of federated learning to enhance the security and privacy of the training process and by employing transfer learning techniques to reduce training time and improve convergence speed.

5.2.4 Integrating Decode-and-Forward Protocols for Enhanced Coverage

Future studies could fruitfully explore the coverage issue in NS by integrating a decode-and-forward (DF) half-duplex protocol. This integration can reasonably be expected to enhance network coverage and more effectively support heterogeneous service requirements.

5.2.5 Intelligent and Dynamic Antenna Activation

In Chapter 4, we adopted ULA with a massive number of antennas to serve different types of users by jointly optimizing power allocation and beamforming for active users. A promising future research direction would be to extend this work towards a more intelligent and automated framework, where the BS dynamically activates antennas and forms beams based on the number of active users in the system rather than utilizing all antennas in the ULA at all times. Such an adaptive approach would reduce power consumption and ensure alignment between antenna activation and real-time network demands/load.

5.2.6 Enhancing Agent Communication and Reward Design

In this thesis, specifically in Chapters 3 and 4, we employed collaborative learning among multiple DRL agents to handle resource allocation across heterogeneous services. In future work, a promising research direction would be designing and implementing advanced communication protocols among agents to minimize conflicts during action selection and identify agents whose actions negatively affect system performance. Instead of relying on a single global reward, individual rewards could be assigned to each agent based on the degree to which its actions impact or improve the overall system performance.

5.2.7 Traffic Anomaly Detection for Slice Protection

In Chapters 3 and 4, we used specific traffic patterns to represent the traffic of each slice in the system. This work can be extended by designing a mechanism that will analyze the real-time traffic load of each slice and detect deviations from the predefined traffic profiles. If the observed traffic does not conform to the expected pattern, a self-protection module could be triggered to classify and discard suspicious packets, as they may indicate malicious activity intended to feed the system with fake data.

5.2.8 Intelligent Slice Admission Control

Finally, a promising avenue for future work is developing an intelligent multi-cell admission control mechanism capable of simultaneously assessing resource availability and QoS performance of ongoing slices. This mechanism determines whether a new slice request can be safely admitted without compromising the service-level guarantees of existing slices. Furthermore, in cases where a slice request cannot be admitted within a given cell, the system could be extended to recommend the most suitable neighboring cell with sufficient resources to accommodate the

request. This capability will facilitate seamless connectivity for slice tenants and their users, particularly in dynamic and resource-constrained RAN environments.

ANNEXE A

AUTHOR'S PUBLICATIONS

In the four years of Ph.D. study, the author contributed to the following published articles.

O. Sabr, G. Kaddoum and K. Kaur, "PABSO-DRL: Power and Beam Self-Optimization Scheme for Multiple Slices in MU-MISO Systems," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 2, pp. 4343-4358, May 2025, doi: 10.1109/TCE.2024.3504958.

O. Sabr, K. Kaur and G. Kaddoum, "A Secure Multi-Radio Resource Scheme Using Cooperative DRL Agents for Heterogeneous Inter-RAN Slicing Under Hardware Impairments," *IEEE Internet of Things Journal*, July 2025, doi: 10.1109/JIOT.2025.3593409.

O. Sabr, G. Kaddoum and K. Kaur, "HiSO-CoMA: Hierarchical Self-Optimising Framework for O-RAN Slicing Using Cooperative Multiple Agent Deep Reinforcement Learning," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 9632-9653, 2025, doi: 10.1109/OJCOMS.2025.3631799.

O. Sabr, K. Kaur and G. Kaddoum, "SOIS-A2C Scheme: Facilitating Management of Multi-Radio Resources in Heterogeneous Inter-RAN Slicing in the Presence of Hardware Impairments,"*ICC 2025 - IEEE International Conference on Communications, Montreal, QC, Canada, 2025, pp. 1414-1419*, doi: 10.1109/ICC52391.2025.11161109.

O. Sabr, G. Kaddoum and K. Kaur, "PoB-MDRL: Multi-Agent Deep Reinforcement Learning-Based Joint Power Allocation and Beamforming for Heterogeneous Services under Non-Ideal Network Conditions,"*The 39th Annual Canadian Conference on Electrical and Computer Engineering (CCECE 2026), Montreal, QC, Canada* (Accepted).

LIST OF REFERENCES

3GPP. (2013). *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study on scalable UMTS Frequency Division Duplex (FDD) bandwidth* (Report n°25.701). Retrieved from: http://www.3gpp.org.

Abbas, K., Nauman, A., Bilal, M., Yoo, J.-H., Hong, J. W.-K. & Song, W.-C. (2024). AI-Driven Data Analytics and Intent-Based Networking for Orchestration and Control of B5G Consumer Electronics Services. *IEEE Transactions on Consumer Electronics*, 70(1), 2155-2169.

Abbasi, M., Shahraki, A., Piran, M. J. & Taherkordi, A. (2021). Deep Reinforcement Learning for QoS provisioning at the MAC layer: A Survey. *Engineering Applications of Artificial Intelligence*, 102, 104234.

Abiko, Y., Saito, T., Ikeda, D., Ohta, K., Mizuno, T. & Mineno, H. (2020). Flexible Resource Block Allocation to Multiple Slices for Radio Access Network Slicing Using Deep Reinforcement Learning. *IEEE Access*, 8, 68183-68198. doi: 10.1109/ACCESS.2020.2986050.

Abood, M. S., Wang, H., He, D., Fathy, M., Rashid, S. A., Alibakhshikenari, M., Virdee, B. S., Khan, S., Pau, G., Dayoub, I., Livreri, P. & Elwi, T. A. (2023). An LSTM-Based Network Slicing Classification Future Predictive Framework for Optimized Resource Allocation in C-V2X. *IEEE Access*, 11, 129300-129310. doi: 10.1109/ACCESS.2023.3332225.

Abouaomar, A., Taik, A., Filali, A. & Cherkaoui, S. (2023). Federated Deep Reinforcement Learning for Open RAN Slicing in 6G Networks. *IEEE Communications Magazine*, 61(2), 126-132.

Adamu, P. U., López-Benítez, M. & Zhang, J. (2023). Hybrid Transmission Scheme for Improving Link Reliability in mmWave URLLC Communications. *IEEE Transactions on Wireless Communications*, 22(9), 6329-6340.

Agarwal, B., Irmer, R., Lister, D. & Muntean, G.-M. (2025). Open RAN for 6G Networks: Architecture, Use Cases and Open Issues. *IEEE Communications Surveys Tutorials*, 1-1. doi: 10.1109/COMST.2025.3562429.

Ahmed, K. I. & Hossain, E. (2019). A deep Q-learning method for downlink power allocation in multi-cell networks. *arXiv preprint arXiv:1904.13032*.

Alam, K., Habibi, M. A., Tammen, M., Krummacker, D., Saad, W., Renzo, M. D., Melodia, T., Costa-Pérez, X., Debbah, M., Dutta, A. & Schotten, H. D. (2025). A Comprehensive Tutorial and Survey of O-RAN: Exploring Slicing-Aware Architecture, Deployment Options, Use Cases, and Challenges. *IEEE Communications Surveys Tutorials*, 1-1. doi: 10.1109/COMST.2025.3598406.

Albrecht, S. V., Christianos, F. & Schäfer, L. (2024). *Multi-agent reinforcement learning: Foundations and modern approaches*. MIT Press.

Alcaraz, J. J., Losilla, F., Zanella, A. & Zorzi, M. (2023). Model-Based Reinforcement Learning With Kernels for Resource Allocation in RAN Slices. *IEEE Transactions on Wireless Communications*, 22(1), 486-501.

Aleyadeh, S. (2023). Towards Zero Touch Next Generation Network Management [PhD thesis]. Ontario,Canada.

Almeida, G. M., Bruno, G. Z., Huff, A., Hiltunen, M., Duarte, E. P., Both, C. B. & Cardoso, K. V. (2024). RIC-O: Efficient Placement of a Disaggregated and Distributed RAN Intelligent Controller With Dynamic Clustering of Radio Nodes. *IEEE Journal on Selected Areas in Communications*, 42(2), 446-459.

Almekhlafi, M., Arfaoui, M. A., Elhattab, M., Assi, C. & Ghrayeb, A. (2021). Joint Scheduling of eMBB and URLLC Services in RIS-Aided Downlink Cellular Networks. *2021 International Conference on Computer Communications and Networks (ICCCN)*, pp. 1-9. doi: 10.1109/ICCCN52240.2021.9522196.

Alwakeel, A. M., Alnaim, A. K. & Fernandez, E. B. (2019). Toward a Reference Architecture for NFV. *2019 2nd International Conference on Computer Applications Information Security (ICCAIS)*, pp. 1-6. doi: 10.1109/CAIS.2019.8769449.

Alwarafy, A., Abdallah, M., Ciftler, B. S., Al-Fuqaha, A. & Hamdi, M. (2021a). Deep reinforcement learning for radio resource allocation and management in next generation heterogeneous wireless networks: A survey. *arXiv preprint arXiv:2106.00574*.

Alwarafy, A., Albaseer, A., Ciftler, B. S., Abdallah, M. & Al-Fuqaha, A. (2021b). AI-Based Radio Resource Allocation in Support of the Massive Heterogeneity of 6G Networks. *2021 IEEE 4th 5G World Forum (5GWF)*, pp. 464-469. doi: 10.1109/5GWF52925.2021.00088.

Amin, R., Reisslein, M. & Shah, N. (2018). Hybrid SDN Networks: A Survey of Existing Approaches. *IEEE Communications Surveys Tutorials*, 20(4), 3259-3306.

Anand, A. & de Veciana, G. (2018). Resource Allocation and HARQ Optimization for URLLC Traffic in 5G Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 36(11), 2411-2421.

Ananya, B., Nikhitha, V., Arjun, S. & Gowda, N. C. (2023). Survey of applications, advantages, and comparisons of AES encryption algorithm with other standards. *International Journal of Computational Learning & Intelligence*, 2(2), 87–98.

Anıl Akyıldız, H., Faruk Gemici, , Hökelek, I. & Ali Çırpan, H. (2024). Hierarchical Reinforcement Learning Based Resource Allocation for RAN Slicing. *IEEE Access*, 12, 75818-75831. doi: 10.1109/ACCESS.2024.3406949.

Aouedi, O., Piamrat, K., Hamma, S. & Perera, J. M. (2022). Network traffic analysis using machine learning: an unsupervised approach to understand and slice your network. *annals of telecommunications*, 77(5), 297–309.

Arulkumaran, K., Deisenroth, M. P., Brundage, M. & Bharath, A. A. (2017). Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine*, 34(6), 26-38.

Arzo, S. T., Naiga, C., Granelli, F., Bassoli, R., Devetsikiotis, M. & Fitzek, F. H. P. (2021). A Theoretical Discussion and Survey of Network Automation for IoT: Challenges and Opportunity. *IEEE Internet of Things Journal*, 8(15), 12021-12045.

Awada, H., Berri, S. & Chorti, A. (2024). Learning-Based Resource Allocation for MBRLLC and Homogeneous Slices in 6G Networks. *2024 3rd International Conference on 6G Networking (6GNet)*, pp. 127-134. doi: 10.1109/6GNet63182.2024.10765787.

Azari, A., Ozger, M. & Cavdar, C. (2019). Risk-Aware Resource Allocation for URLLC: Challenges and Strategies with Machine Learning. *IEEE Communications Magazine*, 57(3), 42-48.

Azimi, Y., Yousefi, S., Kalbkhani, H. & Kunz, T. (2022a). Energy-Efficient Deep Reinforcement Learning Assisted Resource Allocation for 5G-RAN Slicing. *IEEE Transactions on Vehicular Technology*, 71(1), 856-871.

Azimi, Y., Yousefi, S., Kalbkhani, H. & Kunz, T. (2022b). Applications of Machine Learning in Resource Management for RAN-Slicing in 5G and Beyond Networks: A Survey. *IEEE Access*, 10, 106581-106612. doi: 10.1109/ACCESS.2022.3210254.

Aziz, B., Ariaudo, M. & Fijalkow, I. (2011). Critical carrier frequency offset for uplink OFDMA carrier allocation schemes without channel state information. *2011 41st European Microwave Conference*, pp. 709-712. doi: 10.23919/EuMC.2011.6101874.

Badini, N., Jaber, M., Marchese, M. & Patrone, F. (2024). User-Centric Satellite Handover for Multiple Traffic Profiles Using Deep Q-Learning. *IEEE Transactions on Aerospace and Electronic Systems*, 60(6), 8591-8604.

Bana, A.-S., Xu, G., De Carvalho, E. & Popovski, P. (2018). Ultra reliable low latency communications in massive multi-antenna systems. *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 188–192.

Barakabitze, A. A., Ahmad, A., Mijumbi, R. & Hines, A. (2020). 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. *Computer Networks*, 167, 106984.

Benzaid, C. & Taleb, T. (2020). AI-Driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions. *IEEE Network*, 34(2), 186-194.

Björnson, E., Demir, Ö. T. et al. (2024). Introduction to multiple antenna communications and reconfigurable surfaces.

Björnson, E., Hoydis, J., Kountouris, M. & Debbah, M. (2013). Hardware impairments in large-scale MISO systems: Energy efficiency, estimation, and capacity limits. *2013 18th International Conference on Digital Signal Processing (DSP)*, pp. 1-6. doi: 10.1109/ICDSP.2013.6622755.

Bugade, S., Tayal, U., Sharma, K., Gusain, B., Kumar, S. & Monika. (2021). Fifth Generation Networks - Issues and Challenges. *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, pp. 1-6. doi: 10.1109/ICAECA52838.2021.9675528.

Chakraborty, S. & Sivalingam, K. M. (2023). DRL-based admission control and resource allocation for 5G network slicing. *Sādhanā*, 48(3), 155.

Chang, C.-Y. & Nikaein, N. (2018). RAN Runtime Slicing System for Flexible and Dynamic Service Execution Environment. *IEEE Access*, 6, 34018-34042.

Chang, X., Ji, T., Zhu, R., Wu, Z., Li, C. & Jiang, Y. (2023). Toward an Efficient and Dynamic Allocation of Radio Access Network Slicing Resources for 5G Era. *IEEE Access*, 11, 95037-95050. doi: 10.1109/ACCESS.2023.3309294.

Chen, A. C. H. (2024). Evaluation of Advanced Encryption Standard Algorithms for Image Encryption. *2024 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES)*, pp. 1-6. doi: 10.1109/ICSSES62373.2024.10561385.

Chen, P., Wang, P. & Sun, J. (2011). Design and implement of the OFDM communication system. *2011 IEEE International Workshop on Open-source Software for Scientific Computation*, pp. 59-63. doi: 10.1109/OSSC.2011.6184695.

Chen, Q., Li, M., Yang, X., Alturki, R., Alshehri, M. D. & Khan, F. (2021). Impact of Residual Hardware Impairment on the IoT Secrecy Performance of RIS-Assisted NOMA Networks. *IEEE Access*, 9, 42583-42592. doi: 10.1109/ACCESS.2021.3065760.

Cheng, Y. & Pesavento, M. (2013). An Optimal Iterative Algorithm for Codebook-Based Downlink Beamforming. *IEEE Signal Processing Letters*, 20(8), 775-778.

Chergui, H., Blanco, L., Garrido, L. A., Ramantas, K., Kukliński, S., Ksentini, A. & Verikoukis, C. (2021). Zero-Touch AI-Driven Distributed Management for Energy-Efficient 6G Massive Network Slicing. *IEEE Network*, 35(6), 43-49.

Chergui, H., Ksentini, A., Blanco, L. & Verikoukis, C. (2022). Toward Zero-Touch Management and Orchestration of Massive Deployment of Network Slices in 6G. *IEEE Wireless Communications*, 29(1), 86-93.

Choudhary, A., Srivastava, G. & Jha, M. K. (2024). Technicalities of O-RAN for 5G and B5G. *2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS)*, 1, 1-5. doi: 10.1109/ICKECS61492.2024.10617378.

Clerckx, B., Mao, Y., Jorswieck, E. A., Yuan, J., Love, D. J., Erkip, E. & Niyato, D. (2023). A Primer on Rate-Splitting Multiple Access: Tutorial, Myths, and Frequently Asked Questions. *IEEE Journal on Selected Areas in Communications*, 41(5), 1265-1308.

Cui, J., Ding, Z. & Fan, P. (2018). Outage Probability Constrained MIMO-NOMA Designs Under Imperfect CSI. *IEEE Transactions on Wireless Communications*, 17(12), 8239-8255.

De Alwis, C., Porambage, P., Dev, K., Gadekallu, T. R. & Liyanage, M. (2024). A Survey on Network Slicing Security: Attacks, Challenges, Solutions and Research Directions. *IEEE Communications Surveys Tutorials*, 26(1), 534-570.

DEBBABI, F., JMAL, R., CHAARI, L., AGUIAR, R. L., GNICHI, R. & TALEB, S. (2022). Overview of AI-based Algorithms for Network Slicing Resource Management in B5G and 6G. *2022 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 330-335. doi: 10.1109/IWCMC55113.2022.9824988.

Debbabi, F., Jmal, R., Fourati, L. C. & Aguiar, R. L. (2022). An Overview of Interslice and Intraslice Resource Allocation in B5G Telecommunication Networks. *IEEE Transactions on Network and Service Management*, 19(4), 5120-5132.

Dizdar, O., Mao, Y. & Clerckx, B. (2021a). Rate-Splitting Multiple Access to Mitigate the Curse of Mobility in (Massive) MIMO Networks. *IEEE Transactions on Communications*, 69(10), 6765-6780.

Dizdar, O., Mao, Y., Xu, Y., Zhu, P. & Clerckx, B. (2021b). Rate-Splitting Multiple Access for Enhanced URLLC and eMBB in 6G: Invited Paper. *2021 17th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 1-6. doi: 10.1109/ISWCS49558.2021.9562192.

Djigal, H., Xu, J., Liu, L. & Zhang, Y. (2022). Machine and Deep Learning for Resource Allocation in Multi-Access Edge Computing: A Survey. *IEEE Communications Surveys Tutorials*, 24(4), 2449-2494.

Dubey, M., Singh, A. K. & Mishra, R. (2025). AI Based Resource Management for 5G Network Slicing: History, Use Cases, and Research Directions. *Concurrency and Computation: Practice and Experience*, 37(2), e8327.

Dulaj, K., Alhammadi, A., Shayea, I., El-Saleh, A. A. & Alnakhli, M. (2025). Harnessing Machine Learning for Intelligent Networking in 5G Technology and Beyond: Advancements, Applications and Challenges. *IEEE Open Journal of Intelligent Transportation Systems*, 6, 605-633. doi: 10.1109/OJITS.2025.3564361.

Ebrahimi, S., Bouali, F. & Haas, O. C. L. (2024). Resource Management From Single-Domain 5G to End-to-End 6G Network Slicing: A Survey. *IEEE Communications Surveys Tutorials*, 26(4), 2836-2866.

Elsayed, M. & Erol-Kantarci, M. (2020). Radio Resource and Beam Management in 5G mmWave Using Clustering and Deep Reinforcement Learning. *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1-6. doi: 10.1109/GLOBECOM42002.2020.9322401.

Feng, L., Zi, Y., Li, W., Zhou, F., Yu, P. & Kadoch, M. (2020). Dynamic Resource Allocation With RAN Slicing and Scheduling for uRLLC and eMBB Hybrid Services. *IEEE Access*, 8, 34538-34551. doi: 10.1109/ACCESS.2020.2974812.

Feriani, A. & Hossain, E. (2021). Single and Multi-Agent Deep Reinforcement Learning for AI-Enabled Wireless Networks: A Tutorial. *IEEE Communications Surveys Tutorials*, 23(2), 1226-1252.

Filali, A., Mlika, Z., Cherkaoui, S. & Kobbane, A. (2022). Dynamic SDN-Based Radio Access Network Slicing With Deep Reinforcement Learning for URLLC and eMBB Services. *IEEE Transactions on Network Science and Engineering*, 9(4), 2174-2187.

Friesen, M., Wisniewski, L. & Jasperneite, J. (2022). Machine Learning for Zero-Touch Management in Heterogeneous Industrial Networks - A Review. *2022 IEEE 18th International Conference on Factory Communication Systems (WFCS)*, pp. 1-8. doi: 10.1109/WFCS53837.2022.9779183.

Gallego-Madrid, J., Sanchez-Iborra, R., Ruiz, P. M. & Skarmeta, A. F. (2022). Machine learning-based zero-touch network and service management: A survey. *Digital Communications and Networks*, 8(2), 105–123.

Gangakhedkar, S., Cao, H., Ali, A. R., Ganesan, K., Gharba, M. & Eichinger, J. (2018). Use cases, requirements and challenges of 5G communication for industrial automation. *2018 IEEE international conference on communications workshops (icc workshops)*, pp. 1–6.

García, M. & Oberli, C. (2009). Intercarrier interference in OFDM: A general model for transmissions in mobile environments with imperfect synchronization. *EURASIP Journal on Wireless Communications and Networking*, 2009, 1–11.

Gavrilovska, L., Rakovic, V. & Denkovski, D. (2020). From cloud RAN to open RAN. *Wireless Personal Communications*, 113(3), 1523–1539.

Ge, J., Liang, Y.-C., Joung, J. & Sun, S. (2020). Deep Reinforcement Learning for Distributed Dynamic MISO Downlink-Beamforming Coordination. *IEEE Transactions on Communications*, 68(10), 6070-6085.

Ge, J., Liang, Y.-C., Zhang, L., Long, R. & Sun, S. (2023). Deep Reinforcement Learning for Distributed Dynamic Coordinated Beamforming in Massive MIMO Cellular Networks. *IEEE Transactions on Wireless Communications*, 1-1. doi: 10.1109/TWC.2023.3314930.

Ghanem, W. R., Jamali, V., Sun, Y. & Schober, R. (2020). Resource Allocation for Multi-User Downlink MISO OFDMA-URLLC Systems. *IEEE Transactions on Communications*, 68(11), 7184-7200.

Gharehgoli, A., Nouruzi, A., Mokari, N., Azmi, P., Javan, M. R. & Jorswieck, E. A. (2023). AI-Based Resource Allocation in End-to-End Network Slicing Under Demand and CSI Uncertainties. *IEEE Transactions on Network and Service Management*, 20(3), 3630-3651.

Ginige, N. U., Manosha, K. S., Rajatheva, N. & Latva-aho, M. (2020a). Admission control in 5G networks for the coexistence of eMBB-URLLC users. *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–6.

Ginige, N. U., Shashika Manosha, K. B., Rajatheva, N. & Latva-aho, M. (2020b). Admission Control in 5G Networks for the Coexistence of eMBB-URLLC Users. *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1-6. doi: 10.1109/VTC2020-Spring48590.2020.9129141.

Giordani, M., Polese, M., Mezzavilla, M., Rangan, S. & Zorzi, M. (2020). Toward 6G Networks: Use Cases and Technologies. *IEEE Communications Magazine*, 58(3), 55-61.

Groen, J., D'Oro, S., Demir, U., Bonati, L., Villa, D., Polese, M., Melodia, T. & Chowdhury, K. (2024a). Securing O-RAN Open Interfaces. *IEEE Transactions on Mobile Computing*, 1-13. doi: 10.1109/TMC.2024.3393430.

Groen, J., D'Oro, S., Demir, U., Bonati, L., Villa, D., Polese, M., Melodia, T. & Chowdhury, K. (2024b). Securing O-RAN Open Interfaces. *IEEE Transactions on Mobile Computing*.

Grondman, I., Busoniu, L., Lopes, G. A. D. & Babuska, R. (2012). A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), 1291-1307.

Gupta, A. & Jha, R. K. (2015). A Survey of 5G Network: Architecture and Emerging Technologies. *IEEE Access*, 3, 1206-1232. doi: 10.1109/ACCESS.2015.2461602.

Gupta, J. K., Egorov, M. & Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. *International conference on autonomous agents and multiagent systems*, pp. 66–83.

Habibi, M. A., Han, B., Fellan, A., Jiang, W., Sánchez, A. G., Pavon, I. L., Boubendir, A. & Schotten, H. D. (2023). Toward an Open, Intelligent, and End-to-End Architectural Framework for Network Slicing in 6G Communication Systems. *IEEE Open Journal of the Communications Society*, 4, 1615-1658. doi: 10.1109/OJCOMS.2023.3294445.

Hao, J., Yang, T., Tang, H., Bai, C., Liu, J., Meng, Z., Liu, P. & Wang, Z. (2024). Exploration in Deep Reinforcement Learning: From Single-Agent to Multiagent Domain. *IEEE Transactions on Neural Networks and Learning Systems*, 35(7), 8762-8782.

Hua, Y., Li, R., Zhao, Z., Chen, X. & Zhang, H. (2020). GAN-Powered Deep Distributional Reinforcement Learning for Resource Management in Network Slicing. *IEEE Journal on Selected Areas in Communications*, 38(2), 334-349.

Huang, R., Wong, V. W. & Schober, R. (2023). Rate-Splitting for Intelligent Reflecting Surface-Aided Multiuser VR Streaming. *IEEE Journal on Selected Areas in Communications*, 41(5), 1516-1535.

Hunter, T., Sanayei, S. & Nosratinia, A. (2006). Outage analysis of coded cooperation. *IEEE Transactions on Information Theory*, 52(2), 375-391.

Hurtado Sanchez, J. A., Casilimas, K. & Caicedo Rendon, O. M. (2022). Deep reinforcement learning for resource management on network slicing: A survey. *Sensors*, 22(8), 3031.

Iacoboaiea, O., Krolikowski, J., Houidi, Z. B. & Rossi, D. (2023). From Design to Deployment of Zero Touch Deep Reinforcement Learning WLANs. *IEEE Communications Magazine*, 61(2), 104-109.

Indoonundon, M. & Pawan Fowdur, T. (2021). Overview of the challenges and solutions for 5G channel coding schemes. *Journal of Information and Telecommunication*, 5(4), 460–483.

Irkiçatal, O. N., Yuksel, M. & Ceran, E. T. (2023). Deep Reinforcement Learning Aided Rate-Splitting for Interference Channels. *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, pp. 3735-3740. doi: 10.1109/GLOBECOM54140.2023.10437195.

Ishfaq Ahmed, K. & Hossain, E. (2019). A Deep Q-Learning Method for Downlink Power Allocation in Multi-Cell Networks. *arXiv e-prints*, arXiv–1904.

Javed, M. Y., Tervo, N. & Pärssinen, A. (2018). Inter-beam Interference Reduction in Hybrid mmW Beamforming Transceivers. *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 220-224. doi: 10.1109/PIMRC.2018.8580901.

Jiang, W., Han, B., Habibi, M. A. & Schotten, H. D. (2021). The Road Towards 6G: A Comprehensive Survey. *IEEE Open Journal of the Communications Society*, 2, 334-366.

Jin, W., Du, H., Zhao, B., Tian, X., Shi, B. & Yang, G. (2025). A comprehensive survey on multi-agent cooperative decision-making: Scenarios, approaches, challenges and perspectives. *arXiv preprint arXiv:2503.13415*.

Kantasewi, N., Marukatat, S., Thainimit, S. & Manabu, O. (2019). Multi Q-Table Q-Learning. *2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, pp. 1-7. doi: 10.1109/ICTEmSys.2019.8695963.

Khan, M. S., Din, I. U., Almogren, A. & Rodrigues, J. J. P. C. (2024). AI-Enhanced Secure Decision-Making in Ultra-Dense 6G Networks: An Optimized Context-Aware Multi-Attribute Utility Function. *IEEE Transactions on Consumer Electronics*, 1-1. doi: 10.1109/TCE.2024.3385828.

Kim, Y. & Lim, H. (2021). Multi-Agent Reinforcement Learning-Based Resource Management for End-to-End Network Slicing. *IEEE Access*, 9, 56178-56190. doi: 10.1109/ACCESS.2021.3072435.

Klaine, P. V., Imran, M. A., Onireti, O. & Souza, R. D. (2017). A survey of machine learning techniques applied to self-organizing cellular networks. *IEEE Communications Surveys & Tutorials*, 19(4), 2392–2431.

Kouchaki, M. & Marojevic, V. (2022). Actor-Critic Network for O-RAN Resource Allocation: xApp Design, Deployment, and Analysis. *2022 IEEE Globecom Workshops (GC Wkshps)*, pp. 968-973. doi: 10.1109/GCWkshps56602.2022.10008713.

Lee, N., Simeone, O. & Kang, J. (2012). The Effect of Imperfect Channel Knowledge on a MIMO System with Interference. *IEEE Transactions on Communications*, 60(8), 2221-2229.

Li, L., Tang, L., Liu, Q., Wang, Y., He, X. & Chen, Q. (2023). Handoff Control and Resource Allocation for RAN Slicing in IoT Based on DTN: An Improved Algorithm Based on Actor–Critic Framework. *IEEE Internet of Things Journal*, 10(15), 13370-13384.

Li, R., Zhao, Z., Sun, Q., I, C.-L., Yang, C., Chen, X., Zhao, M. & Zhang, H. (2018). Deep Reinforcement Learning for Resource Management in Network Slicing. *IEEE Access*, 6, 74429-74441. doi: 10.1109/ACCESS.2018.2881964.

Li, R., Wang, C., Zhao, Z., Guo, R. & Zhang, H. (2020a). The LSTM-Based Advantage Actor-Critic Learning for Resource Management in Network Slicing With User Mobility. *IEEE Communications Letters*, 24(9), 2005-2009.

Li, S. E. (2023). Deep reinforcement learning. In *Reinforcement Learning for Sequential Decision and Optimal Control* (pp. 365–402). Springer.

Li, T., Zhang, H., Guo, S. & Yuan, D. (2024). Robust Rate-Splitting and Beamforming for Ultra-Reliable and Low-Latency Communications. *IEEE Transactions on Wireless Communications*, 23(10), 15571-15585.

Li, X., Samaka, M., Chan, H. A., Bhamare, D., Gupta, L., Guo, C. & Jain, R. (2017). Network Slicing for 5G: Challenges and Opportunities. *IEEE Internet Computing*, 21(5), 20-27.

Li, X., Ni, R., Chen, J., Lyu, Y., Rong, Z. & Du, R. (2020b). End-to-End Network Slicing in Radio Access Network, Transport Network and Core Network Domains. *IEEE Access*, 8, 29525-29537. doi: 10.1109/ACCESS.2020.2972105.

Li, Y. (2017). Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.

Liao, P.-H., Shen, L.-H., Wu, P.-C. & Feng, K.-T. (2024). Multi-Agent Deep Reinforcement Learning for Energy Efficient Multi-Hop STAR-RIS-Assisted Transmissions. *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*, pp. 1-5. doi: 10.1109/VTC2024-Fall63153.2024.10758034.

Lin, S.-C., Lin, C.-H. & Chen, W.-C. (2022). Zero-Touch Network on Industrial IoT: An End-to-End Machine Learning Approach. *arXiv preprint arXiv:2204.12605*.

Liu, J., Ma, Y. & Tafazolli, R. (2024a). A Spatially Non-Stationary Fading Channel Model for Simulation and (Semi-) Analytical Study of ELAA-MIMO. *IEEE Transactions on Wireless Communications*, 23(5), 5203-5218.

Liu, Q., Choi, N. & Han, T. (2023). Deep Reinforcement Learning for End-to-End Network Slicing: Challenges and Solutions. *IEEE Network*, 37(2), 222-228.

Liu, W., Fu, Y., Guo, Y., Lee Wang, F., Sun, W. & Zhang, Y. (2024b). Two-Timescale Synchronization and Migration for Digital Twin Networks: A Multi-Agent Deep Reinforcement Learning Approach. *IEEE Transactions on Wireless Communications*, 23(11), 17294-17309.

Liu, Y., Ding, J. & Liu, X. (2020). A Constrained Reinforcement Learning Based Approach for Network Slicing. *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, pp. 1-6.

Liu, Y., Ma, T., Qin, X., Zhou, H. & Shen, X. (2024c). Reconfigurable RAN Slicing for Ultra-Dense LEO Satellite Networks via DRL. *IEEE Transactions on Cognitive Communications and Networking*.

Liyanage, M., Pham, Q.-V., Dev, K., Bhattacharya, S., Maddikunta, P. K. R., Gadekallu, T. R. & Yenduri, G. (2022). A survey on Zero touch network and Service (ZSM) Management for 5G and beyond networks. *Journal of Network and Computer Applications*, 103362.

Liyanagea, M., Phamb, Q.-V., Devc, K., Bhattacharyad, S., Reddy, P. K., Maddikuntad, T. R. G. & Yendurid, G. (2022). A Survey on Zero Touch Network and Service (ZSM) Management for 5G and Beyond Networks. *English, Journal of Network and Computer Applications*, 4, 103.

Lu, C.-C. & Tseng, S.-Y. (2002). Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. *Proceedings IEEE International Conference on Application- Specific Systems, Architectures, and Processors*, pp. 277-285. doi: 10.1109/ASAP.2002.1030726.

M, U. M., Sadashivappa, G. & Palepu, R. (2023). Integration of RIC and xApps for Open -Radio Access Network for Performance Optimization. *2023 7th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, pp. 1-4. doi: 10.1109/CSITSS60515.2023.10334159.

Mao, Y., Dizdar, O., Clerckx, B., Schober, R., Popovski, P. & Poor, H. V. (2022). Rate-Splitting Multiple Access: Fundamentals, Survey, and Future Research Trends. *IEEE Communications Surveys Tutorials*, 24(4), 2073-2126.

Marinova, S. & Leon-Garcia, A. (2024). Intelligent O-RAN Beyond 5G: Architecture, Use Cases, Challenges, and Opportunities. *IEEE Access*, 12, 27088-27114. doi: 10.1109/ACCESS.2024.3367289.

Marzouk, F., Radwan, A., Chi, H. R. & Barraca, J. P. (2023). Highly Flexible and Traffic Isolating RAN Slicing: A Consumer IoT-Based Use Case. *IEEE Transactions on Consumer Electronics*, 69(4), 709-718.

Mei, J., Wang, X., Zheng, K., Boudreau, G., Sediq, A. B. & Abou-Zeid, H. (2021). Intelligent Radio Access Network Slicing for Service Provisioning in 6G: A Hierarchical Deep Reinforcement Learning Approach. *IEEE Transactions on Communications*, 69(9), 6063-6078.

Meng, F., Chen, P., Wu, L. & Cheng, J. (2020). Power Allocation in Multi-User Cellular Networks: Deep Reinforcement Learning Approaches. *IEEE Transactions on Wireless Communications*, 19(10), 6255-6267.

Mienye, I. D. & Swart, T. G. (2024). A comprehensive review of deep learning: Architectures, recent advances, and applications. *Information*, 15(12), 755.

Mismar, F. B., Evans, B. L. & Alkhateeb, A. (2020). Deep Reinforcement Learning for 5G Networks: Joint Beamforming, Power Control, and Interference Coordination. *IEEE Transactions on Communications*, 68(3), 1581-1592.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540), 529–533.

Nagib, A. (2024). *A Trustworthy Deep Reinforcement Learning Framework for Slicing in Next-Generation Open Radio Access Networks*. (Ph.D. thesis, Queen's University (Canada)).

Nagib, A. M., Abou-Zeid, H. & Hassanein, H. S. (2024). Safe and Accelerated Deep Reinforcement Learning-Based O-RAN Slicing: A Hybrid Transfer Learning Approach. *IEEE Journal on Selected Areas in Communications*, 42(2), 310-325.

Ngo, M. V., Tran, N.-B.-L., Yoo, H.-M., Pua, Y.-H., Le, T.-L., Liang, X.-L., Chen, B., Hong, E.-K. & Quek, T. Q. (2024a). RAN Intelligent Controller (RIC): From open-source implementation to real-world validation. *ICT Express*, 10(3), 680-691.

Ngo, M. V., Yoo, H.-M., Pua, Y.-H., Le, T.-L., Liang, X.-L., Chen, B., Hong, E.-K., Quek, T. Q. et al. (2024b). RAN Intelligent Controller (RIC): From open-source implementation to real-world validation. *ICT Express*.

Nguyen, T. T., Nguyen, N. D. & Nahavandi, S. (2020). Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Transactions on Cybernetics*, 50(9), 3826-3839.

Ojaghi, B., Adelantado, F., Antonopoulos, A. & Verikoukis, C. (2022). Impact of Network Densification on Joint Slicing and Functional Splitting in 5G. *IEEE Communications Magazine*, 60(7), 30-35.

Paczolay, G. & Harmati, I. (2020). A New Advantage Actor-Critic Algorithm For Multi-Agent Environments. *2020 23rd International Symposium on Measurement and Control in Robotics (ISMCR)*, pp. 1-6. doi: 10.1109/ISMCR51255.2020.9263738.

Pala, S., Katwe, M., Singh, K., Clerckx, B. & Li, C.-P. (2024). Spectral-Efficient RIS-Aided RSMA URLLC: Toward Mobile Broadband Reliable Low Latency Communication (mBRLLC) System. *IEEE Transactions on Wireless Communications*, 23(4), 3507-3524.

Pasarelski, R., Angelov, K., Postagian, K. & Sadinov, S. (2023). Implementation and Analysis of a Customized Encryption Algorithm in 5G Networks for Educational Purposes. *2023 4th International Conference on Communications, Information, Electronic and Energy Systems (CIEES)*, pp. 1-5. doi: 10.1109/CIEES58940.2023.10378834.

Peng, H., Hsia, C.-C., Han, Z. & Wang, L.-C. (2024). A Generalized Delay and Backlog Analysis for Multiplexing URLLC and eMBB: Reconfigurable Intelligent Surfaces or Decode-and-Forward? *IEEE Transactions on Wireless Communications*, 23(5), 4049-4068.

Pennanen, H., Hänninen, T., Tervo, O., Tölli, A. & Latva-Aho, M. (2025). 6G: The Intelligent Network of Everything. *IEEE Access*, 13, 1319-1421. doi: 10.1109/ACCESS.2024.3521579.

Polese, M., Bonati, L., D'Oro, S., Basagni, S. & Melodia, T. (2023). Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. *IEEE Communications Surveys Tutorials*, 25(2), 1376-1411.

Qi, C., Hua, Y., Li, R., Zhao, Z. & Zhang, H. (2019). Deep Reinforcement Learning With Discrete Normalized Advantage Functions for Resource Management in Network Slicing. *IEEE Communications Letters*, 23(8), 1337-1341.

Qiao, K., Wang, H., Zhang, W., Yang, D., Zhang, Y. & Zhang, N. (2025). Resource Allocation for Network Slicing in Open RAN: A Hierarchical Learning Approach. *IEEE Transactions on Cognitive Communications and Networking*, 11(4), 2584-2600.

Rahmanian, G., Shahhoseini, H. S. & Pozveh, A. H. J. (2021). A Review of Network Slicing in 5G and Beyond: Intelligent Approaches and Challenges. *2021 ITU Kaleidoscope: Connecting Physical and Virtual Worlds (ITU K)*, pp. 1-8. doi: 10.23919/ITUK53220.2021.9662097.

Ratasuk, R., Mangalvedhe, N., Bhatoolaul, D. & Ghosh, A. (2017). LTE-M Evolution Towards 5G Massive MTC. *2017 IEEE Globecom Workshops (GC Wkshps)*, pp. 1-6. doi: 10.1109/GLOCOMW.2017.8269112.

Rezazadeh, F., Chergui, H., Christofi, L. & Verikoukis, C. (2021a). Actor-Critic-Based Learning for Zero-touch Joint Resource and Energy Control in Network Slicing. *ICC 2021 - IEEE International Conference on Communications*, pp. 1-6. doi: 10.1109/ICC42927.2021.9500265.

Rezazadeh, F., Chergui, H. & Verikoukis, C. (2021b). Zero-touch continuous network slicing control via scalable actor-critic learning. *arXiv preprint arXiv:2101.06654*.

Sabapathy, S., Maruthu, S., Kumar, L. S. & Jayakody, D. N. K. (2023). Outage Analysis of Sparse Vector Coding-Based Downlink Multicarrier NOMA for URLLC. *IEEE Internet of Things Journal*, 10(14), 12393-12400.

Sabr, O., Kaddoum, G. & Kaur, K. (2025a). PABSO-DRL: Power and Beam Self-Optimization Scheme for Multiple Slices in MU-MISO Systems. *IEEE Transactions on Consumer Electronics*, 71(2), 4343-4358.

Sabr, O., Kaur, K. & Kaddoum, G. (2025b). SOIS-A2C Scheme: Facilitating Management of Multi-Radio Resources in Heterogeneous Inter-RAN Slicing in the Presence of Hardware Impairments. *Proceedings IEEE ICC*, pp. 1414-1419. doi: 10.1109/ICC52391.2025.11161109.

Saha, N., Zangooei, M., Golkarifard, M. & Boutaba, R. (2023). Deep Reinforcement Learning Approaches to Network Slice Scaling and Placement: A Survey. *IEEE Communications Magazine*, 61(2), 82-87.

Salameh, A. I. & El Tarhuni, M. (2022). From 5G to 6G—challenges, technologies, and applications. *Future Internet*, 14(4), 117.

Sallent, O., Perez-Romero, J., Ferrus, R. & Agusti, R. (2017). On Radio Access Network Slicing from a Radio Resource Management Perspective. *IEEE Wireless Communications*, 24(5), 166-174.

Santos, J. F., Huff, A., Campos, D., Cardoso, K. V., Both, C. B. & DaSilva, L. A. (2025). Managing O-RAN Networks: xApp Development From Zero to Hero. *IEEE Communications Surveys Tutorials*, 1-1. doi: 10.1109/COMST.2025.3539687.

Setayesh, M. (2024). *Machine learning-based algorithms design for network slicing, federated learning, and 360° video streaming in wireless systems*. (Ph.D. thesis, University of British Columbia).

Setayesh, M., Bahrami, S. & Wong, V. W. (2020). Joint PRB and Power Allocation for Slicing eMBB and URLLC Services in 5G C-RAN. *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1-6.

Setayesh, M., Bahrami, S. & Wong, V. W. (2022). Resource Slicing for eMBB and URLLC Services in Radio Access Network Using Hierarchical Deep Learning. *IEEE Transactions on Wireless Communications*, 21(11), 8950-8966.

Shang, C., Sun, Y. & Luo, H. (2022). A Hybrid Deep Reinforcement Learning Approach for Dynamic Task Offloading in NOMA-MEC System. *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 434-442. doi: 10.1109/SECON55815.2022.9918560.

Shao, Y., Li, R., Hu, B., Wu, Y., Zhao, Z. & Zhang, H. (2021). Graph Attention Network-Based Multi-Agent Reinforcement Learning for Slicing Resource Management in Dense Cellular Network. *IEEE Transactions on Vehicular Technology*, 70(10), 10792-10803.

Shen, X., Gao, J., Wu, W., Lyu, K., Li, M., Zhuang, W., Li, X. & Rao, J. (2020). AI-Assisted Network-Slicing Based Next-Generation Wireless Networks. *IEEE Open Journal of Vehicular Technology*, 1, 45-66. doi: 10.1109/OJVT.2020.2965100.

Shi, D., Zhao, C., Wang, Y., Yang, H., Wang, G., Jiang, H., Xue, C., Yang, S. & Zhang, Y. (2022). Multi actor hierarchical attention critic with RNN-based feature extraction. *Neurocomputing*, 471, 79–93.

Shi, Z. (2021). Chapter 11-Emotion intelligence. In Shi, Z. (Ed.), *Intelligence Science* (pp. 437-463). Elsevier. doi: https://doi.org/10.1016/B978-0-323-85380-4.00011-7.

Shirzad, F. (2023). Network Slicing and Resource Allocation in Cloud Radio Access Networks. *Network*, 2023, 04–28.

Slalmi, A., Chaibi, H., Saadane, R. & Chehri, A. (2021a). Call Admission Control Optimization in 5G in Downlink Single-Cell MISO System. *Procedia Computer Science*, 192, 2502-2511. doi: https://doi.org/10.1016/j.procs.2021.09.019. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 25th International Conference KES2021.

Slalmi, A., Chaibi, H., Saadane, R. & Chehri, A. (2021b). Call Admission Control Optimization in 5G in Downlink Single-Cell MISO System. *Procedia Computer Science*, 192, 2502–2511.

Soltani, S., Amanloo, A., Shojafar, M. & Tafazolli, R. (2025). Intelligent Control in 6G Open RAN: Security Risk or Opportunity? *IEEE Open Journal of the Communications Society*, 6, 840-880. doi: 10.1109/OJCOMS.2025.3526215.

Sun, H., Liu, Y., Al-Tahmeesschi, A., Nag, A., Soleimanpour, M., Canberk, B., Arslan, H. & Ahmadi, H. (2025). Advancing 6G: Survey for Explainable AI on Communications and Network Slicing. *IEEE Open Journal of the Communications Society*, 6, 1372-1412. doi: 10.1109/OJCOMS.2025.3534626.

Sun, Y. & Zhang, X. (2022). A2C Learning for Tasks Segmentation with Cooperative Computing in Edge Computing Networks. *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 2236-2241. doi: 10.1109/GLOBECOM48099.2022.10000948.

Tan, F., Si, S., Chen, H., Li, S. & Lv, T. (2024). Rate Splitting Multiple Access Assisted Cell-Free Massive MIMO for URLLC Services in 5G and Beyond Networks. *IEEE Open Journal of the Communications Society*, 5, 6018-6032. doi: 10.1109/OJCOMS.2024.3459911.

Tan, X., Zhou, L., Wang, H., Sun, Y., Zhao, H., Seet, B.-C., Wei, J. & Leung, V. C. M. (2022). Cooperative Multi-Agent Reinforcement-Learning-Based Distributed Dynamic Spectrum Access in Cognitive Radio Networks. *IEEE Internet of Things Journal*, 9(19), 19477-19488.

Tang, J., Shim, B., Chang, T.-H. & Quek, T. Q. S. (2019a). Incorporating URLLC and Multicast eMBB in Sliced Cloud Radio Access Network. *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1-7. doi: 10.1109/ICC.2019.8761648.

Tang, J., Shim, B. & Quek, T. Q. S. (2019b). Service Multiplexing and Revenue Maximization in Sliced C-RAN Incorporated With URLLC and Multicast eMBB. *IEEE Journal on Selected Areas in Communications*, 37(4), 881-895.

Tariq, F., Khandaker, M. R. A., Wong, K.-K., Imran, M. A., Bennis, M. & Debbah, M. (2020). A Speculative Study on 6G. *IEEE Wireless Communications*, 27(4), 118-125.

Taskou, S. K. & Rasti, M. (2024). Resource Allocation for FeMBB and eURLLC Coexistence in RSMA-Based Cellular Networks. *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1-6. doi: 10.1109/WCNC57260.2024.10570774.

Tran, D.-D., Sharma, S. K., Ha, V. N., Chatzinotas, S. & Woungang, I. (2023). Multi-Agent DRL Approach for Energy-Efficient Resource Allocation in URLLC-Enabled Grant-Free NOMA Systems. *IEEE Open Journal of the Communications Society*, 4, 1470-1486. doi: 10.1109/OJCOMS.2023.3291689.

Viswanathan, H. & Mogensen, P. E. (2020). Communications in the 6G Era. *IEEE Access*, 8, 57063-57074.

Wan, A., Chang, Q., Khalil, A.-B. & He, J. (2023). Short-term power load forecasting for combined heat and power using CNN-LSTM enhanced by attention mechanism. *Energy*, 282, 128274.

Wang, C.-X., You, X., Gao, X., Zhu, X., Li, Z., Zhang, C., Wang, H., Huang, Y., Chen, Y., Haas, H., Thompson, J. S., Larsson, E. G., Renzo, M. D., Tong, W., Zhu, P., Shen, X., Poor, H. V. & Hanzo, L. (2023). On the Road to 6G: Visions, Requirements, Key Technologies, and Testbeds. *IEEE Communications Surveys Tutorials*, 25(2), 905-974.

Wang, H., Bai, Y. & Xie, X. (2024a). Deep Reinforcement Learning Based Resource Allocation in Delay-Tolerance-Aware 5G Industrial IoT Systems. *IEEE Transactions on Communications*, 72(1), 209-221.

Wang, J., Lan, Z., Sum, C.-S., Pyo, C.-W., Gao, J., Baykas, T., Rahman, A., Funada, R., Kojima, F., Lakkis, I., Harada, H. & Kato, S. (2009). Beamforming Codebook Design and Performance Evaluation for 60GHz Wideband WPANs. *2009 IEEE 70th Vehicular Technology Conference Fall*, pp. 1-6. doi: 10.1109/VETECF.2009.5379063.

Wang, W., Tang, L., Liu, T., He, X., Liang, C. & Chen, Q. (2024b). Toward Reliability-Enhanced, Delay-Guaranteed Dynamic Network Slicing: A Multiagent DQN Approach With an Action Space Reduction Strategy. *IEEE Internet of Things Journal*, 11(6), 9282-9297.

Wang, X., Sun, G., Xin, Y., Liu, T. & Xu, Y. (2022). Deep Transfer Reinforcement Learning for Beamforming and Resource Allocation in Multi-Cell MISO-OFDMA Systems. *IEEE Transactions on Signal and Information Processing over Networks*, 8, 815-829. doi: 10.1109/TSIPN.2022.3208432.

Wani, M., Kretschmer, M., Schröder, B., Grebe, A. & Rademacher, M. (2025). Open RAN: A Concise Overview. *IEEE Open Journal of the Communications Society*, 6, 13-28. doi: 10.1109/OJCOMS.2024.3430823.

Wei, Y., Yu, F. R., Song, M. & Han, Z. (2018). User Scheduling and Resource Allocation in HetNets With Hybrid Energy Supply: An Actor-Critic Reinforcement Learning Approach. *IEEE Transactions on Wireless Communications*, 17(1), 680-692.

Xiao, C., Zheng, Y. & Beaulieu, N. (2002). Second-order statistical properties of the WSS Jakes' fading channel simulator. *IEEE Transactions on Communications*, 50(6), 888-891.

Xiong, Z., Zhang, Y., Niyato, D., Deng, R., Wang, P. & Wang, L.-C. (2019). Deep Reinforcement Learning for Mobile 5G and Beyond: Fundamentals, Applications, and Challenges. *IEEE Vehicular Technology Magazine*, 14(2), 44-52.

Xue, J., Yu, K., Zhang, T., Zhou, H., Zhao, L. & Shen, X. (2024a). Cooperative Deep Reinforcement Learning Enabled Power Allocation for Packet Duplication URLLC in Multi-Connectivity Vehicular Networks. *IEEE Transactions on Mobile Computing*, 23(8), 8143-8157.

Xue, J., Yu, K., Zhang, T., Zhou, H., Zhao, L. & Shen, X. (2024b). Cooperative deep reinforcement learning enabled power allocation for packet duplication uRLLC in multi-connectivity vehicular networks. *IEEE Transactions on Mobile Computing*.

Yan, D., Ng, B. K., Ke, W. & Lam, C.-T. (2023). Deep Reinforcement Learning Based Resource Allocation for Network Slicing With Massive MIMO. *IEEE Access*, 11, 75899-75911.

Yan, D., NG, B. K., Ke, W. & Lam, C.-T. (2024). Multi-Agent Deep Reinforcement Learning Joint Beamforming for Slicing Resource Allocation. *IEEE Wireless Communications Letters*, 13(5), 1220-1224.

Yan, M., Feng, G., Zhou, J., Sun, Y. & Liang, Y.-C. (2019). Intelligent Resource Scheduling for 5G Radio Access Network Slicing. *IEEE Transactions on Vehicular Technology*, 68(8), 7691-7703.

Yang, H. & Xie, X. (2020). An Actor-Critic Deep Reinforcement Learning Approach for Transmission Scheduling in Cognitive Internet of Things Systems. *IEEE Systems Journal*, 14(1), 51-60.

Yang, L., El Rajab, M., Shami, A. & Muhaidat, S. (2023). Diving Into Zero-Touch Network Security: Use-Case Driven Analysis. *Authorea Preprints*.

Yang, L., El Rajab, M., Shami, A. & Muhaidat, S. (2024a). Enabling automl for zero-touch network security: Use-case driven analysis. *IEEE Transactions on Network and Service Management*, 21(3), 3555–3582.

Yang, L., Rajab, M. E., Shami, A. & Muhaidat, S. (2024b). Enabling AutoML for Zero-Touch Network Security: Use-Case Driven Analysis. *IEEE Transactions on Network and Service Management*, 21(3), 3555-3582.

Yang, L., Naser, S., Shami, A., Muhaidat, S., Ong, L. & Debbah, M. (2025). Toward Zero Touch Networks: Cross-Layer Automated Security Solutions for 6G Wireless Networks. *IEEE Transactions on Communications*, 73(9), 7650-7679.

Youssef, M.-J., Nour, C. A., Lagrange, X. & Douillard, C. (2021). A Deep Q-Learning Bisection Approach for Power Allocation in Downlink NOMA Systems. *IEEE Communications Letters*, 26(2), 316–320.

Youssef, M.-J., Nour, C. A., Lagrange, X. & Douillard, C. (2022). A Deep Q-Learning Bisection Approach for Power Allocation in Downlink NOMA Systems. *IEEE Communications Letters*, 26(2), 316-320.

Yungaicela-Naula, N. M., Vargas-Rosales, C., Pérez-Díaz, J. A. & Zareei, M. (2022). Towards security automation in software defined networks. *Computer Communications*, 183, 64–82.

Zangooei, M., Saha, N., Golkarifard, M. & Boutaba, R. (2023). Reinforcement Learning for Radio Resource Management in RAN Slicing: A Survey. *IEEE Communications Magazine*, 61(2), 118-124.

Zangooei, M., Golkarifard, M., Rouili, M., Saha, N. & Boutaba, R. (2024). Flexible RAN Slicing in Open RAN With Constrained Multi-Agent Reinforcement Learning. *IEEE Journal on Selected Areas in Communications*, 42(2), 280-294.

Zhang, F., Sun, S., Rong, B., Yu, F. R. & Lu, K. (2015). A Novel Massive MIMO Precoding Scheme for Next Generation Heterogeneous Networks. *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-6. doi: 10.1109/GLOCOM.2015.7417346.

Zhang, H., Liu, N., Chu, X., Long, K., Aghvami, A.-H. & Leung, V. C. M. (2017). Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges. *IEEE Communications Magazine*, 55(8), 138-145.

Zhang, H., Pan, G., Xu, S., Zhang, S. & Jiang, Z. (2022a). A Hard and Soft Hybrid Slicing Framework for Service Level Agreement Guarantee via Deep Reinforcement Learning. *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pp. 1-5. doi: 10.1109/VTC2022-Spring54318.2022.9860789.

Zhang, H., Xu, S., Zhang, S. & Jiang, Z. (2022b). Slicing Framework for Service Level Agreement Guarantee in Heterogeneous Networks—A Deep Reinforcement Learning Approach. *IEEE Wireless Communications Letters*, 11(1), 193-197.

Zhang, L., She, C., Ying, K., Li, Y. & Vucetic, B. (2023). Deep Reinforcement Learning for Improving Resource Utilization Efficiency of URLLC With Imperfect Channel State Information. *IEEE Wireless Communications Letters*, 12(10), 1796-1800.

Zhang, R., Xiong, K., Lu, Y., Gao, B., Fan, P. & Letaief, K. B. (2022c). Joint Coordinated Beamforming and Power Splitting Ratio Optimization in MU-MISO SWIPT-Enabled HetNets: A Multi-Agent DDQN-Based Approach. *IEEE Journal on Selected Areas in Communications*, 40(2), 677-693.

Zhang, S. (2019). An Overview of Network Slicing for 5G. *IEEE Wireless Communications*, 26(3), 111-117.

Zhang, Z., Xiao, Y., Ma, Z., Xiao, M., Ding, Z., Lei, X., Karagiannidis, G. K. & Fan, P. (2019). 6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies. *IEEE Vehicular Technology Magazine*, 14(3), 28-41.

Zhang, Z., Yu, F. R., Fu, F., Yan, Q. & Wang, Z. (2018). Joint offloading and resource allocation in mobile edge computing systems: An actor-critic approach. *2018 IEEE global communications conference (GLOBECOM)*, pp. 1–6.

Zhang, Z. & Tellambura, C. (2009). The effect of imperfect carrier frequency offset estimation on an OFDMA uplink. *IEEE Transactions on Communications*, 57(4), 1025-1030.

Zhang, Z., Zhang, D. & Qiu, R. C. (2020). Deep reinforcement learning for power system applications: An overview. *CSEE Journal of Power and Energy Systems*, 6(1), 213-225.

Zhao, Y., Chi, X., Qian, L., Zhu, Y. & Hou, F. (2022). Resource Allocation and Slicing Puncture in Cellular Networks With eMBB and URLLC Terminals Coexistence. *IEEE Internet of Things Journal*, 9(19), 18431-18444.

Zhou, G., Zhao, L., Zheng, G., Xie, Z., Song, S. & Chen, K.-C. (2023). Joint Multi-Objective Optimization for Radio Access Network Slicing Using Multi-Agent Deep Reinforcement Learning. *IEEE Transactions on Vehicular Technology*, 72(9), 11828-11843.

Zhou, H., Wang, X., Umehira, M. & Ji, Y. (2022). A Deep Reinforcement Learning based Analog Beamforming Approach in Downlink MISO Systems. *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pp. 1-6. doi: 10.1109/VTC2022-Spring54318.2022.9860777.

Zhu, L. & Tan, L. (2024). Task offloading scheme of vehicular cloud edge computing based on digital twin and improved a3c. *Internet of Things*, 26, 101192.

Zou, W., Cui, Z., Li, B., Zhou, Z. & Hu, Y. (2011). Beamforming codebook design and performance evaluation for 60GHz wireless communication. *2011 11th International Symposium on Communications Information Technologies (ISCIT)*, pp. 30-35. doi: 10.1109/ISCIT.2011.6089755.