IEEE Open Journal of the Communications Society

Received 13 August 2025; revised 12 October 2025; accepted 5 November 2025. Date of publication 12 November 2025;
date of current version 21 November 2025.

Digital Object Identifier 10.1109/OJCOMS.2025.3631799

HiSO-CoMA: Hierarchical Self-Optimizing Framework for O-RAN Slicing Using Cooperative Multiple Agent Deep Reinforcement Learning

OHOOD SABR[1] (Graduate Student Member, IEEE), GEORGES KADDOUM[1] (Senior Member, IEEE), AND KULJEET KAUR[1,2,3] (Member, IEEE)

[1]Department of Electrical Engineering, École de Technologie Supérieure, University of Quebec, Montreal, QC H3C 1K3, Canada
[2]School of Engineering, Applied Science and Technology (SEAST), Canadian University Dubai, Dubai, UAE
[3]Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura 140401, India

CORRESPONDING AUTHOR: O. SABR (e-mail: ohood.sabr.1@ens.etsmtl.ca)

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN-2021-04013.

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.

INDEX TERMS 6G, beamforming optimization, deep reinforcement learning, inter-RAN slicing, intra-RAN slicing, multiple radio resource allocation, power allocation, zero touch networks.

I. INTRODUCTION

FUTURE networks are anticipated in a completely autonomous manner, eliminating the need for human intervention [1]. 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 [1]. 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 [1], [2]. 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 [3].

NS is a promising technology that enables the creation of isolated virtual logical networks on top of a physical operator network [4]. 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 [2]. 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 [5]. 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 [6]. 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 [3] or inter-slice [7] 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.

A. 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 [8] 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 [9], [10], [11] to manage the power and beamforming of eMBB and uRLLC in multiple-input single-output (MISO) systems based on iterative algorithms. These studies [9], [10], [11] 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 [12]. An example can be found in [12], 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 [13], 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, [14] 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 [15] 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, [16] 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 [6], [17], [18], [19], [20], [21], [22], [23], 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. [7] proposed a scheme that jointly manages both power and bandwidth, providing a more integrated solution for resource allocation. These studies (*e.g.*, [6], [7], [17], [18], [19], [20], [21], [22], [23]) 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 [7], [21], [23], where the authors designed a similar algorithm for both MISO and MIMO systems. However, the strategy used in these studies (*e.g.*, [6], [7], [17], [18], [19], [20], [21],

[22], [23]), 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 [24], 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 [4], 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, [25] 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.

B. 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.,* [4], [6], [7], [17], [18], [19], [20], [21], [22], [23], [24]). 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.,* [4], [6], [17], [18], [19], [20], [21], [22], [23], [24], [25]) 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) [26]. 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.,*[4], [6], [17], [18], [19], [20], [22], [24]) without exploring how these algorithms can be adapted for multi-antenna systems, which are expected to dominate next-generation networks [27]. 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 [28], along with other limitations highlighted in [29] 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.,* [6], [17], [18], [19], [20], [24]). Furthermore, majority of existing studies have focused on optimizing resource allocation at the intra-slicing level (*e.g.,* [3], [8], [9], [10], [11], [12], [13], [14], [15], [16]), 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 [30], 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.

C. 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.

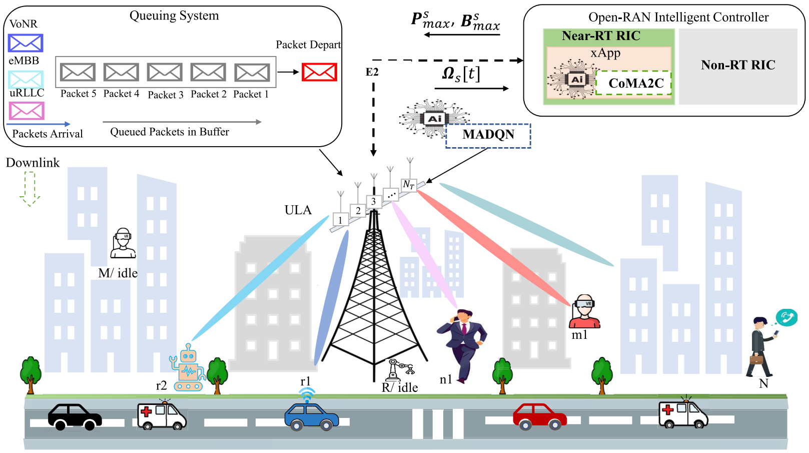


FIGURE 1. System model of downlink RSMA with heterogeneous inter- and intra-RAN slicing.

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 I-B, 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 [7] and traditional algorithms to test the adaptability and reliability of the proposed framework.

D. ORGANIZATION

The remainder of this paper is organized as follows. Section II details the system model and problem formulation. Section III elaborates on the proposed hierarchical slef-optimization framework. Numerical evaluations and discussion are provided in Section IV. Finally, Section V concludes this paper and outlines the future research directions.

II. 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.

A. 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 (see Fig. 1). The E2 interface enables seamless communication between the RIC and RAN components, thus enabling the exchange of data/information

TABLE 1. SLAs for admitted slices.

No.	Slice	$\Re_s^{\text{Min}}$	L_s^{Max}	SSR_s^{Th} [19]
1	VoNR	51kbps [20]	10ms [20]	$\geq 95\%$
2	eMBB	15 Mbps	10ms [20]	$\geq 95\%$
3	uRLLC	10Mbps [20]	3ms	$\geq 95\%$

and coordinated management [31]. 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) [32]. 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 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 [19], [25]. 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:

$$\Omega_s[t] = \sum_{u_s \in \mathcal{U}_s} \Phi_{u_s}; \forall s \in \mathcal{S}. \tag{1}$$

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 [25]. 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 [33]. 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 ($\Re_s^{\text{Min}}$) for the users of each slice, as listed in Table 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

$$d_{\psi_{u_s}} = \begin{cases} 1, \text{ if } \Re_{u_s} \geq \Re_s^{\text{Min}} \;\&\; l_{\psi_{u_s}} \leq L_s^{\text{Max}} \\ 0, \text{ otherwise} \end{cases}, \tag{2}$$

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 [34]. 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

$$l_{\psi_{u_s}} = D_{\text{Trans}} + D_{\text{Queuing}}, \tag{3}$$

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 [33]. 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 [6]. 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

$$\text{SSR}_s[t] = \frac{\sum_{u_s \in \mathcal{U}_s} \sum_{\psi_{u_s} \in \Phi_{u_s}} d_{\psi_{u_s}}}{\sum_{u_s \in \mathcal{U}_s} |\Phi_{u_s}|}; \forall s \in \mathcal{S} \tag{4}$$

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:

$$\text{SSR}_{u_s}[t] = \frac{\sum_{\psi_{u_s} \in \Phi_{u_s}} d_{\psi_{u_s}}}{|\Phi_{u_s}|}; \forall u_s \in \mathcal{U}_s \tag{5}$$

B. 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 [35]. Accordingly, the downlink channel vector $\left(\mathbf{h}_{u_s}\right)$ between the BS and user u_s at time slot t is expressed as follows:

$$\mathbf{h}_{u_s}[t] = \sqrt{\frac{\beta_{u_s}}{L}} \mathbf{A}\left(N_T, \theta_{u_s}, \Delta\right) \mathbf{g}_{u_s}; \;\; u_s \in \{n, m, r\}, \tag{6}$$

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:

$$\mathbf{A}\left(N_T, \theta_{u_s}, \Delta\right) = \left[\alpha_1(N_T, \theta_1) \ldots \alpha_L(N_T, \theta_L)\right] \in \mathbb{C}^{N_T \times L}, \tag{7}$$

where θ_{u_s} denotes the direction of departure (DoD) from the BS toward user u_s, while Δ represents angular spread. In Eq. (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:

$$\alpha_l(N_T, \theta_l) = \left[1, e^{j2\pi \frac{d^n}{\lambda} \cos\theta_2} \ldots e^{j2\pi \frac{d^n}{\lambda}(N_T - 1)\cos\theta_l}\right]^T, \tag{8}$$

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 [35]. 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 [36]. Due to the block-fading, the channel remains constant within each time slot, but changes independently from one slot to the another.

C. 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 [37]. 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 [30]. The main advantage of RSMA lies in its flexibility in managing interference, allowing it to be partially decoded and partially treated as noise [15]. 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 $(G_{u_s}^c)$ and private $(G_{u_s}^p)$ 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 $G_{u_s}^p$ 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 [30].

$$\mathbf{x}[t] = \mathbf{w}_c s_c + \sum_{u_s=1}^{U_s} \mathbf{w}_{u_s} s_{u_s}; \;\; u_s \in \{n, m, r\}, \tag{9}$$

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.

D. 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.

$$\begin{aligned} y_{u_s} = {} & \underbrace{\mathbf{h}_{u_s}^H[t]\mathbf{w}_c[t]s_c(t)}_{\text{common message}} + \underbrace{\mathbf{h}_{u_s}^H[t]\mathbf{w}_{u_s}[t]s_{u_s}[t]}_{\text{private message}} \\ & + \sum_{i=1, i \neq u_s}^{U_s} \underbrace{\mathbf{h}_{u_s}^H[t]\mathbf{w}_i[t]s_i[t]}_{\text{intra interference}} \\ & + \underbrace{\mathbb{I}_{BS}^t}_{\text{HWI at ULA}} + \underbrace{\mathbb{Z}_{u_s}[t]}_{\text{AWGN at } u_s} + \underbrace{\mathbb{I}_{u_s}^r}_{\text{Self-distortion}} \; ; \forall u_s \in \mathcal{U}_s, \end{aligned} \tag{10}$$

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. (10) represents the intra-interference experienced by the u_sth user. It is also assumed that all the users are subjected to additive white Gaussian noise (AWGN) with noise variance σ^2, where $\mathbb{Z}_{u_s} \in \mathcal{CN}\left(0, \sigma^2\right)$ denotes the noise at the user u_s at time slot t. Furthermore, $\mathbb{I}_{BS}^t \sim \mathcal{CN}\left(\mathbf{0}, \boldsymbol{d}^t\right)$ 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 [26].

$$\mathbf{d}^t = \kappa^t \cdot \text{diag}\left(|\omega_1|^2, |\omega_2|^2, \ldots, |\omega_{N_T}|^2\right), \tag{11}$$

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 [26]

$$d^r = \kappa^r |\mathbf{h}_{u_s}^H \mathbf{w}|^2, \tag{12}$$

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 $\left(\Gamma^c_{u_s}\right)$ and private $\left(\Gamma^p_{u_s}\right)$ stream at time t are given as follows.

$$\Gamma^c_{u_s}[t] = \frac{|\mathbf{h}^H_{u_s}[t]\mathbf{w}_c[t]|^2}{\underbrace{\sum_{i=1}^{U_s} |\mathbf{h}^H_{u_s}[t]\mathbf{w}_i[t]|^2}_{\text{Interference}} + \underbrace{\kappa^t \sum_{\tau=1}^{N_T} |h_\tau w_\tau|^2}_{\text{HWI at ULA}} + \underbrace{\kappa^r \left|\mathbf{h}^H_{u_s}[t]\mathbf{w}_c[t]\right|^2}_{\text{Self-distortion}} + \underbrace{\sigma^2_{u_s}}_{\text{AWGN}}} \quad ; \forall u_s \in \mathcal{U}_s \tag{13}$$

$$\Gamma^p_{u_s}[t] = \frac{|\mathbf{h}^H_{u_s}[t]\mathbf{w}_{u_s}[t]|^2}{\sum_{i=1, i\neq u_s}^{U_s} |\mathbf{h}^H_{u_s}\mathbf{w}_i|^2 + \underbrace{\kappa^t \sum_{\tau=1}^{N_T} |h_\tau w_\tau|^2}_{\text{HWI at ULA}} + \underbrace{\kappa^r \left|\mathbf{h}^H_{u_s}[t]\mathbf{w}_{u_s}[t]\right|^2}_{\text{Self-distortion}} + \underbrace{\sigma^2_{u_s}}_{\text{AWGN}}} \quad ; \forall u_s \in \mathcal{U}_s \tag{14}$$

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 $r_c[t] = \min_{u_s \in \mathcal{U}_s}\{\log_2\left(1 + \Gamma^c_{u_s}\right)\}$. The instantaneous rate for decoding private stream at user u_s is below.

$$r^p_{u_s}[t] = \log_2\left(1 + \Gamma^p_{u_s}\right); \forall u_s \in \mathcal{U}_s. \tag{15}$$

We assume that r_c is shared by the users within each slice such that $\sum\limits_{u_s \in U_s} \mathcal{X}^c_{u_s}$, 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:

$$\Re_{u_s}[t] = \mathcal{X}^c_{u_s} + r^p_{u_s}. \tag{16}$$

E. 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. (17). A higher utility value indicates better QoS performance for the network slices. In Eq. (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 [6]. In this context, η is defined as $\eta[t] = \frac{\sum_{s\in\mathcal{S}} \sum_{u_s \in \mathcal{U}_s} \Re_{u_s}}{\mathbb{B}^T}$, 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.

$$\begin{aligned}
\mathcal{OF}: \quad & \underset{\substack{\mathbf{P}^s_{\max}, \mathbf{B}^s_{\max} \\ \hat{\mathbf{w}}_c, \hat{\mathbf{w}}_{u_s} \\ \|\mathbf{w}_c\|^2, \|\mathbf{w}_{u_s}\|^2}}{\text{maximize}} \quad \alpha^\eta.\eta + \sum_{s\in\mathcal{S}} \mathfrak{B}_s \cdot SSR_s & (17)\\
\text{s.t.} \quad & \text{C1}: P^s_{\max} \geq \lambda^s_p; \quad \forall s \in \mathcal{S}, \\
& \text{C2}: \quad \sum_{s\in\mathcal{S}} P^s_{\max} = \mathbb{P}^T; \quad \forall t, \\
& \text{C3}: \quad B^s_{\max} \geq \Lambda^s_b; \quad \forall s \in \mathcal{S}, \\
& \text{C4}: \quad \sum_{s\in\mathcal{S}} B^s_{\max} = \mathbb{B}^T; \quad \forall t, \\
& \text{C5}: \quad \boldsymbol{\Omega}_s = (\Omega_{s1}, \ldots, \Omega_S); \quad \forall s \in \mathcal{S}, \\
& \text{C6}: \quad SSR_s \geq \text{SSR}^{\text{Th}}_s; \quad \forall s \in \mathcal{S}, \\
& \text{C7}: \|\mathbf{w}_{u_s}\|^2 \geq 0; \quad \forall u_s \in \mathcal{U}_s, \\
& \text{C8}: \|\mathbf{w}_c\|^2 + \sum_{n=1}^{N} \|\mathbf{w}_n\|^2 \leq P^n_{\max}, \\
& \text{C9}: \|\mathbf{w}_c\|^2 + \sum_{m=1}^{M} \|\mathbf{w}_m\|^2 \leq P^m_{\max}, \\
& \text{C10}: \|\mathbf{w}_c\|^2 + \sum_{r=1}^{R} \|\mathbf{w}_r\|^2 \leq P^r_{\max}, \\
& \text{C11}: \quad SSR_{u_s} \geq \text{SSR}^{\text{Th}}_s; \quad \forall u_s \in \mathcal{U}_s, \\
& \text{C12}: \quad \hat{\mathbf{w}}_{u_s} \in [0,\ 2\pi), \\
& \text{C13}: \quad \sum_{u_s \in U_s} \mathcal{X}^c_{u_s} \leq r_c; \quad \forall u_s \in \mathcal{U}_s, \forall s \in \mathcal{S} \\
& \text{C14}: \quad \mathcal{X}^c_{u_s}, r^p_{u_s} \geq 0; \quad \forall u_s \in \mathcal{U}_s, \forall s \in \mathcal{S}.
\end{aligned}$$

In Eq. (17), C1 guarantees that a minimum power $\left(\lambda^s_p\right)$ is allocated per slice. C2 defines the total power allocated to all the slices, which is equal to the total system power $\left(\mathbb{P}^T\right)$. C3 ensures that each slice is allocated a minimum bandwidth $\left(\Lambda^s_b\right)$. C4 defines the total bandwidth allocated to all the slices, which is equal to the $\mathbb{B}^T$. C5 defines Ω_s as the traffic model for slice s. C6 ensures that the SSR of each slice meets or exceeds the predefined threshold $\left(\text{SSR}^{\text{Th}}_s\right)$. 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 $\left(\text{SSR}^{\text{Th}}_s\right)$. 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 [38]. 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 [39]. Furthermore, traditional approaches lack self-learning capabilities, which are essential for the autonomous self-optimization expected in ZTNs scenarios [3].

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 [24], [40]. In Section III, 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.

III. 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.

A. OVERVIEW

To address the problem in Eq. (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 [4], while the intra-slice level is reformulated as an MDP, as discussed in Sections III-B and III-C.

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 [41]. 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 [42].

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) [4]. 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.

B. 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.

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 $\left(\mathbf{s}_{g_\sigma}\right)$ of each $g_\sigma \in \mathcal{G}_\sigma$ at $t \in \mathcal{T}^{\text{long}}$ is given by

$$\mathbf{s}_{g_\sigma}[t] = \mathbf{\Omega}_s = \left[\Omega_{s_1}, \Omega_{s_2}, \ldots, \Omega_S\right]; \quad \forall g_\sigma \in \mathcal{G}_\sigma, \tag{18}$$

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. (19) and Eq. (20), respectively.

$$\mathbf{a}_g^P[t] = \left\{\left(P_{\max}^{s_1}, P_{\max}^{s_2}, \ldots, P_{\max}^{S}\right) \in \mathbf{P}_{\max}^s \mid P_{\max}^s \geq \lambda_p^s, \right. \\ \left. \mathbf{a}_g^P \in \mathcal{A}_g^P, \right\} \tag{19}$$

$$\mathbf{a}_g^B[t] = \left\{\left(B_{\max}^{s_1}, B_{\max}^{s_2}, \ldots, B_{\max}^{S}\right) \in \mathbf{B}_{\max}^s \mid B_{\max}^s \geq \Lambda_b^s, \right. \\ \left. \mathbf{a}_g^B \in \mathcal{A}_g^B, \right\} \tag{20}$$

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 [32]. 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:

$$\Delta^{\Omega_s}[t] = \frac{|\Omega_s(t) - \Omega_s(t-1)|}{\Omega_s(t-1)} \times 100, \tag{21}$$

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

$$\Delta_{\max}[t] = \max\left\{\Delta^{\Omega_{s_1}}[t], \Delta^{\Omega_{s_2}}[t], \ldots, \Delta^{\Omega_S}[t]\right\}, \tag{22}$$

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

$$\tau^{\text{CoMA2C}} = \begin{cases} \text{True}; & \text{if } \Delta_{\max}[t] > \nabla^{\text{Th}} \\ \text{False}; & \text{if } \Delta_{\max}[t] \leq \nabla^{\text{Th}} \end{cases} \tag{23}$$

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.

3) GLOBAL REWARD

After agents of CoMA2C scheme perform their chosen actions, the NS environment sends them a team reward $\left(r_{g_\sigma}\right)$ according to Algorithm 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.

C. 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.

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 [43]. Specifically, the state $\left(s_{j_s}\right)$ of each agent at time $t \in \mathcal{T}^{\text{short}}$ is defined as

$$s_{j_s}[t] = \left\{\|\mathbf{w}_c\|^2[t-1], \Gamma_{u_s}^c[t-1], \mathcal{X}_{u_s}^c[t-1], \|\mathbf{w}_{u_s}\|^2[t-1]\right.$$

Algorithm 1: Calculate team reward for $\mathcal{G}_\sigma$.

```
Input: $SSR_s = \{SSR_{\text{VoNR}}, SSR_{\text{eMBB}}, SSR_{\text{uRLLC}}\}, \eta$
If $SSR_{\text{VoNR}}$, $SSR_{\text{eMBB}}$ and $SSR_{\text{uRLLC}} \geq \text{SSR}_s^{\text{Th}}$ Then
    If $\eta < 100$ Then
        $r_{g_\sigma}[t] \leftarrow 10$
    Else
        $r_{g_\sigma}[t] \leftarrow 10 + 0.1 \cdot (\eta - 100)$
    End If
Else If $SSR_{\text{uRLLC}} < \text{SSR}_s^{\text{Th}}$ Then
    $r_{g_\sigma}[t] \leftarrow 10 \cdot (SSR_{\text{uRLLC}} - 0.7)$ [19]
Else
    $r_{g_\sigma}[t] \leftarrow -2 \cdot (1 - \min(SSR_{\text{VoNR}}, SSR_{\text{eMBB}}))$
End If
Output: $r_{g_\sigma} \forall g_\sigma \in \mathcal{G}_\sigma$
```

$$\Gamma^p_{u_s}[t-1], I^{\hat{\mathbf{w}}_{u_s}}[t-1], r^p_{u_s}[t-1], \left|\mathbf{h}^H_{u_s}[t]\hat{\mathbf{w}}_{u_s}[t]\right|^2\Big\}; \quad \forall u_s \in \mathcal{U}_s, \forall s \in \mathcal{S}. \tag{24}$$

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 $|\mathbf{h}^H_{u_s}[t]\hat{\mathbf{w}}_{u_s}[t]|^2$ denote the previous power, SDNR, beam direction index, rate, and the equivalent channel gain for the private stream respectively.

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 $P^s_{\text{max}} \in \{P^n_{\text{max}}, P^m_{\text{max}}, P^r_{\text{max}}, \ldots, P^S_{\text{max}}\}$ 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) [30], [44] 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 $B_{\text{code}} \geq N_T$ [3]. 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 [3], and it is also applied in [35]. The total number of available actions is defined as $N^s_L \times B_{\text{code}}$, 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. (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.

$$\mathcal{A}^s = \{(p^c, p^p, \mathbf{c}), \;\; p^c, p^p \in \mathcal{P}, \;\; \mathbf{c} \in \mathbf{C}_{\text{book}}\}, \tag{25}$$

where

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

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 [29]. 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. (26). To ensure stability during training, r_{j_s} is clipped to prevent extreme values from destabilizing the learning process.

$$r_{j_s}[t] = \text{clip}\big(\eta_{u_s} \cdot \vartheta_{u_s} \cdot \delta_{u_s}, -\mu, \mu\big), \tag{26}$$

In Eq. (26), η_{u_s} represents the spectral efficiency of u_s in $s \in \mathcal{S}$ and given by

$$\eta_{u_s} = \frac{\mathfrak{R}_{u_s}[t]}{b_{u_s}}, \tag{27}$$

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. (26) are defined as follows:

$$\vartheta_{u_s} = \begin{cases} 1; & \text{if } \text{SSR}_{u_s} \geq \text{SSR}^{Th}_s \\ \max\Big(0.1, \frac{\text{SSR}_{u_s}}{\text{SSR}^{Th}_s}\Big); & \text{if } \text{SSR}_{u_s} < \text{SSR}^{Th}_s \end{cases}, \tag{28}$$

$$\delta_{u_s} = \begin{cases} 1; & \text{if } \mathfrak{R}_{u_s} \geq \mathfrak{R}^{\text{Min}}_s \\ \max\Big(0.1, \frac{\mathfrak{R}_{u_s}}{\mathfrak{R}^{\text{Min}}_s}\Big); & \text{if } \mathfrak{R}_{u_s} < \mathfrak{R}^{\text{Min}}_s \end{cases}. \tag{29}$$

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.

D. CHALLENGES

Overall, solving a twin-timescale MDP presents significant challenges [25]. It is important to note that the problem in Eq. (17) is especially difficult to solve using MADRL due to the following challenges.

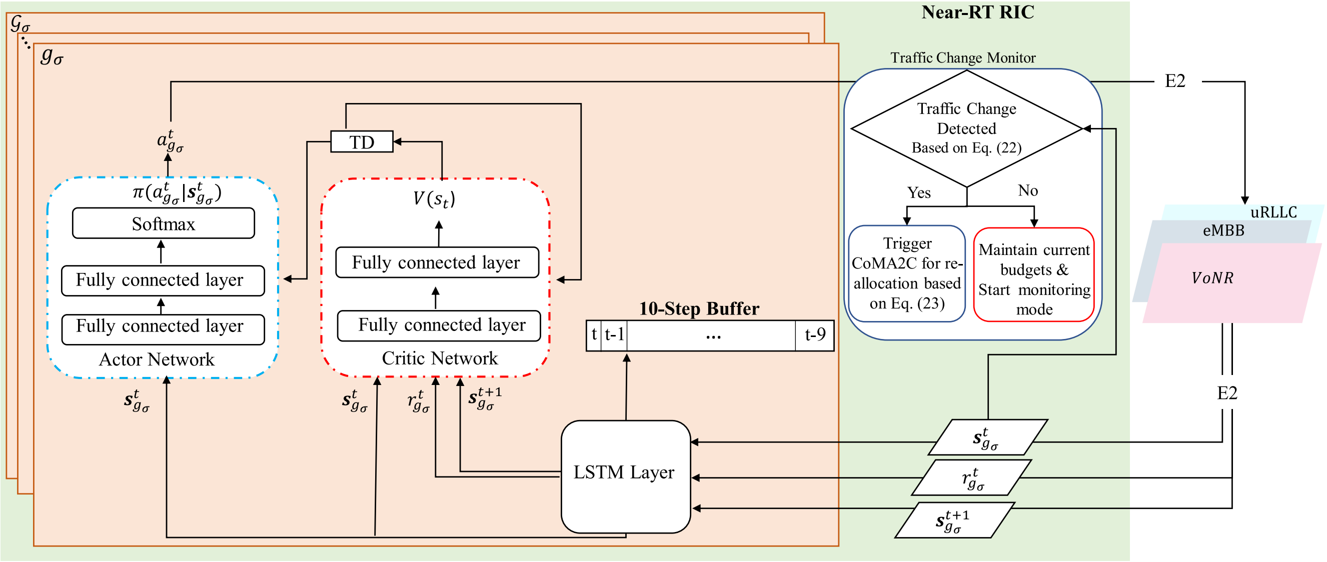


FIGURE 2. An illustration of the CoMA2C scheme for joint resource management among heterogeneous inter-RAN slicing.

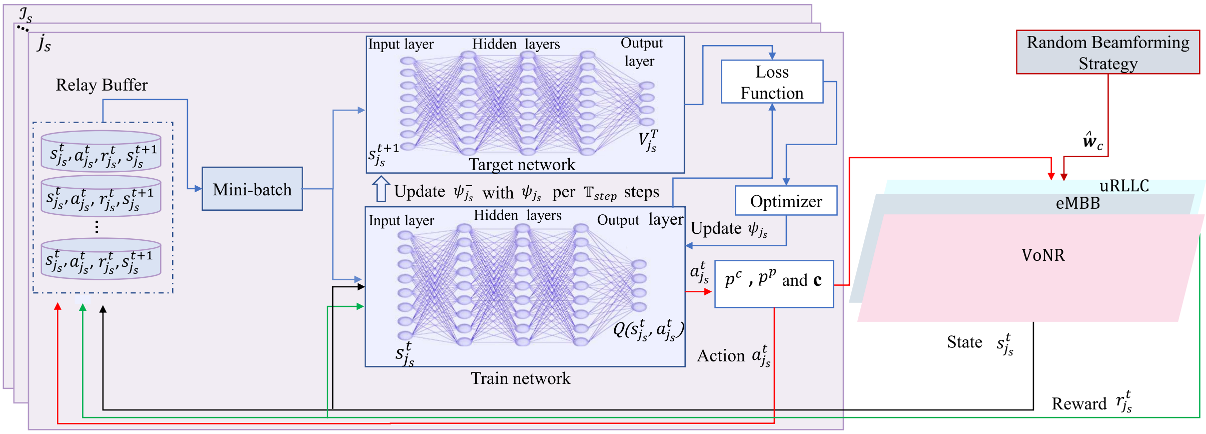


FIGURE 3. An illustration of the MADQN scheme for joint resource management in heterogeneous intra-RAN slicing.

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.

E. 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. 2, while **Stage II** presents the MADQN scheme, where the architecture of each agent in the MADQN scheme is depicted in Fig. 3. Together, these schemes form the proposed HiSO-CoMA framework, as shown in Fig. 4.

1) STAGE I—DESIGN 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 ($\bar{R}_t$)

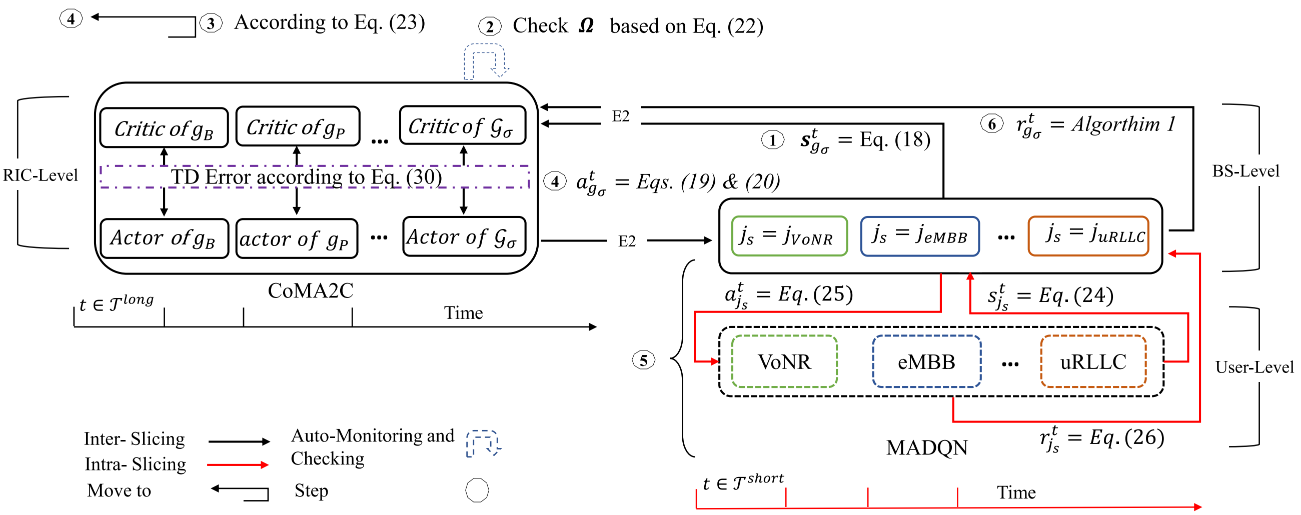

④
③ According to Eq. (23)
② Check $\boldsymbol{\Omega}$ based on Eq. (22)
RIC-Level
Critic of g_B
Critic of g_P
Critic of $\mathcal{G}_\sigma$
TD Error according to Eq. (30)
Actor of g_B
actor of g_P
Actor of $\mathcal{G}_\sigma$
CoMA2C
E2
① $\mathbf{s}^t_{g_\sigma}$ = Eq. (18)
④ $a^t_{g_\sigma}$ = Eqs. (19) & (20)
⑥ $r^t_{g_\sigma}$ = Algorthim 1
BS-Level
E2
$j_s = j_{VoNR}$
$j_s = j_{eMBB}$
$j_s = j_{uRLLC}$
$a^t_{j_s}$ = Eq. (25)
$s^t_{j_s}$ = Eq. (24)
⑤
VoNR
eMBB
uRLLC
User-Level
$r^t_{j_s}$ = Eq. (26)
MADQN
$t \in \mathcal{T}^{long}$
Time
$t \in \mathcal{T}^{short}$
Time
Inter- Slicing
Intra- Slicing
Move to
Auto-Monitoring and Checking
Step

FIGURE 4. Schematic view of the proposed HiSO-CoMA framework for hierarchical heterogeneous multi-slice MISO systems.

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 [45]. 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 [19]. 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 [4]. Algorithm 2 describes the CoMA2C scheme of our proposed framework. The initialization stage of Algorithm 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 $\boldsymbol{\Omega}_s$. 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 $\boldsymbol{\Omega}_s$, represented by a state vector (see Eq. (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. (19) and (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 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 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. (30) [19].

$$\mathcal{E}_t = \underbrace{r^t_{g_\sigma} + \Upsilon V\left(\mathbf{s}^{t+1}_{g_\sigma}; \psi^c_{g_\sigma}\right)}_{Q(s_t,a_t)} - \underbrace{V\left(\mathbf{s}^t_{g_\sigma}; \psi^c_{g_\sigma}\right)}_{\text{current value function}}. \tag{30}$$

The actor loss function ($\mathcal{L}^{\text{Actor}}_{g_\sigma}$) [19], [45] for each $g_\sigma \in \mathcal{G}_\sigma$ is given by

$$\mathcal{L}^{\text{Actor}}_{g_\sigma} = -\Big[\underbrace{\log\pi\left(a^t_{g_\sigma}|\mathbf{s}^t_{g_\sigma};\psi^a_{g_\sigma}\right)}_{\substack{\text{log probability of}\\ \text{action given a state}}} + \phi\,\mathbb{E}\left(\pi\left(a^t_{g_\sigma}|\mathbf{s}^t_{g_\sigma};\psi^a_{g_\sigma}\right)\right)\mathcal{E}_t\Big] \tag{31}$$

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 ($\mathcal{L}^{Critic}_{g_\sigma}$) [19] for each $g_\sigma \in \mathcal{G}_\sigma$ is expressed as

$$\mathcal{L}^{Critic}_{g_\sigma} = (\mathcal{E}_t)^2. \tag{32}$$

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. (22) without adjustment until the RIC observes that there is a change in $\boldsymbol{\Omega}_s$ according to Eq. (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 $g_\sigma \in$

Algorithm 2: Pseudocode of CoMA2C Scheme

```
1: Initialization:
2: Initialize A2C-based LSTM; $\forall\sigma \in \Theta$.
3: Initialize SLA parameters; $\forall s \in \mathcal{S}$.
4: $\forall g_\sigma \in \mathcal{G}_\sigma$, initialize:
   • Actor network: $\Psi^a_{g_\sigma}$.
   • Critic network: $\Psi^c_{g_\sigma}$.
   • Learning rates for actor: $\varrho^a_{g_\sigma} > 0$ and critic: $\varrho^c_{g_\sigma} > 0$.
5: Initialize NS state monitor to track traffic changes over time.
6: for iteration $i = 1$ to $F_i$ do
    Learning Phase (First 200 Iterations):
    7: if $i < 200$ then
        8: for $t \in \mathcal{T}^{long}$ do
            9: for $g_\sigma \in \mathcal{G}_\sigma$ do
                10: Obtain state $\mathbf{s}_{g_\sigma}$ according to Eq. (18) from heterogeneous network slices.
                11: 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. (19) and (20).
                12: 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.
                13: Check SLA, SSR requirements for each $s \in \mathcal{S}$ and system performance $\eta$.
                14: Calculate the reward based on Algorithm 1 and move to the next state $\mathbf{s}^{t+1}_{g_\sigma}$.
                15: Critic calculates the corresponding TD error using Eq. (30).
                16: Calculate actor loss and critic loss using Eqs. (31) and (32), respectively.
                17: Update actor and critic networks:
                $\Psi^a_{g_\sigma}[t+1] \leftarrow \Psi^a_{g_\sigma} + \varrho^a_{g_\sigma} \nabla \mathcal{L}^{Actor}_{g_\sigma}(\Psi^a_{g_\sigma})$,
                $\Psi^c_{g_\sigma}[t+1] \leftarrow \Psi^c_{g_\sigma} + \varrho^c_{g_\sigma} \nabla (\mathcal{E}_t)^2$.
                18: $t = t + 1$
            end
        end
    end
    else
        19: Monitoring and Learning Phase ($i > 200$):
        20: Monitor $\mathbf{\Omega}_s$ changes based on Eq. (22).
        21: Trigger actor networks according to Eq. (23), to select or adjust the radio resources ($\Theta$) based on current traffic loads.
        22: Repeat steps 6-18.
    end
end
Output: Best policy $\pi^*_{g_\sigma}\left(\mathbf{a}^t_{g_\sigma} \mid \mathbf{s}^t_{g_\sigma}, \forall g_\sigma \in \mathcal{G}_\sigma\right)$.
```

Algorithm 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 $\psi_{j_s}$ and target Q-network with $\psi^-_{j_s}$ $\forall j_s \in \mathcal{J}_s$.
5: Set the initial $\epsilon$ and $\alpha_{j_s}$; $\forall j_s \in \mathcal{J}_s$.
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. (24).
            12: Select $a_{j_s}$ based on $\epsilon$ greedy policy in Eq. (33).
            13: Execute joint $a_{j_s}$ using Eq. (25) and receive $r_{j_s}$ as defined by Eq. (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. (34).
            18: Update the $\psi_{j_s}$ 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
    end
end
Output: Best policy $\pi^*_{j_s}(a_{j_s} \mid s_{j_s}, \forall j_s \in \mathcal{J}_s)$.
```

$\mathcal{G}_\sigma$ in the proposed scheme, the dropout technique [46] is applied to mitigate the risk of overfitting and enhance the generalization ability of the proposed model.

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 [3], [43] and the learning process illustrated in Algorithm 3. In Line 1 of Algorithm 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 parameterized 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. (33) [43].

$$a^t_{j_s} = \begin{cases} \text{random action } (a_{j_s})\text{; with probability } \epsilon \\ \arg\max_{a_{j_s} \in \mathcal{A}^s}\left\{\mathcal{Q}(s_{j_s}, a_{j_s})\right\}\text{; with} \\ \text{probability} 1 - \epsilon, \end{cases} \tag{33}$$

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. (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 [35]:

$$\mathcal{L}_{j_s}(\psi_{j_s}) = \frac{1}{2M_{batch}} \sum_{\langle s_{j_s}, a_{j_s}, r_{j_s}, s_{j_s}^{t+1} \rangle \in Y_{j_s}} \underbrace{(V_{j_s}^{T}}_{\text{Target Q value}} - \underbrace{Q(s_{j_s}, a_{j_s}; \psi_{j_s}))^2}_{\text{Q value}}, \quad (34)$$

where $V_{j_s}^{T}(t) = r_{j_s}^{t} + \gamma \max_{a'_{j_s} \in \mathcal{A}^s} Q(s_{j_s}^{t+1}, a'_{j_s}; \psi_{j_s}^{-})$ denotes the target value, determined through the target network [43]. 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.

IV. EXPERIMENTS AND PERFORMANCE EVALUATIONS

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

A. SIMULATION SETTINGS

We consider the scenario shown in Fig. 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 2. The BS, with a coverage radius r, is located within a simulation area of 240 m × 240 m [19]. 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 [19]. 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 Core 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 MADQN schemes. The hyperparameters used in the setup of the CoMA2C and MADQN schemes are illustrated in Table 3.

TABLE 2. Main parameters and their descriptions.

Parameter	Value
U_s	100
r (m)	40 [19]
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_{L}^{s}	5
$\mathbb{P}^{T}$ and $\mathbb{B}^{T}$	60 dBm and 10 MHz [19], 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 [23]
$\sigma_{u_s}^2$	-174 dBm
d^n	$\lambda/2$ [35]
Packet size [19]	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 [35]
L	4 [35]
Δ	3° [35]
$\mathfrak{B}_s$	$[1, 2, 3]$ for VoNR, eMBB, and uRLLC, respectively
α^{η}	0.6
VoNR user inter-arrival distribution	Uniform distribution between 0 and 160 ms [20]
eMBB user inter-arrival distribution	Pareto distribution with the mean of 6 ms and the maximum of 12.5 ms [20]
uRLLC user inter-arrival distribution	Exponential distribution with an average time of 180 ms [20]

TABLE 3. Training parameters for CoMA2C and MADQN.

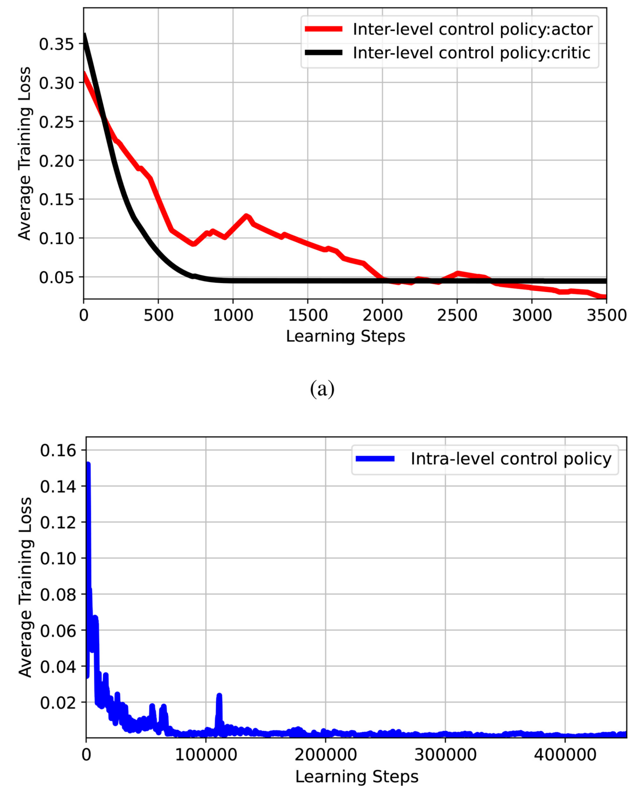
CoMA2C Parameters	
Number of state elements	3
$\varrho_{g_\sigma}^{a}$ and $\varrho_{g_\sigma}^{c}$	5e-4 and 2e-3, respectively
Optimizer	RMSProp
Υ	0.999
Size of LSTM cells	One LSTM layer with 64 neurons
Entropy rate	0.01
Dropout rate	0.1
Actor network	Two fully connected layers, each with 32 units and 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 [3].

B. BENCHMARK ALGORITHMS

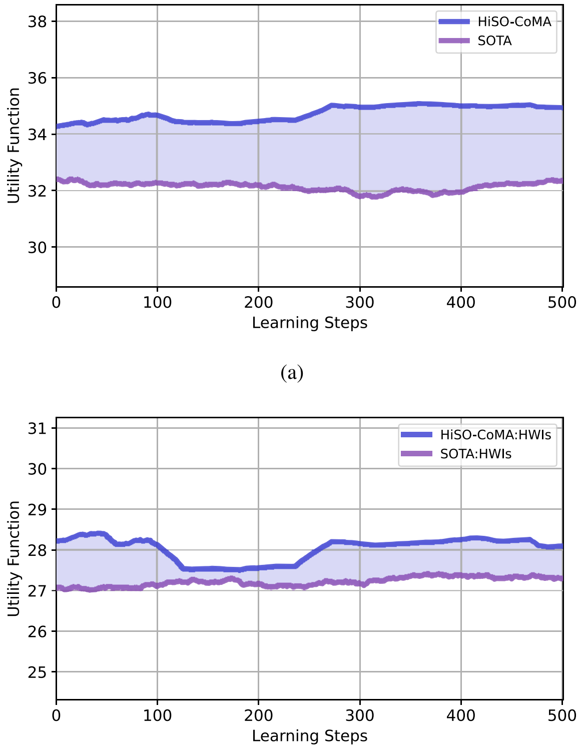
We validate the efficacy of our proposed HiSO-CoMA framework through comprehensive experimental analysis under various parameter settings. To this end, we compare our results to the following state-of-the-art (SOTA) and traditional schedulers.

- *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.



(a)

FIGURE 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.**



(a)

(b)

FIGURE 6. **Utility function of the proposed framework vs. the SOTA approach under (a) ideal HW and (b) HWIs.**

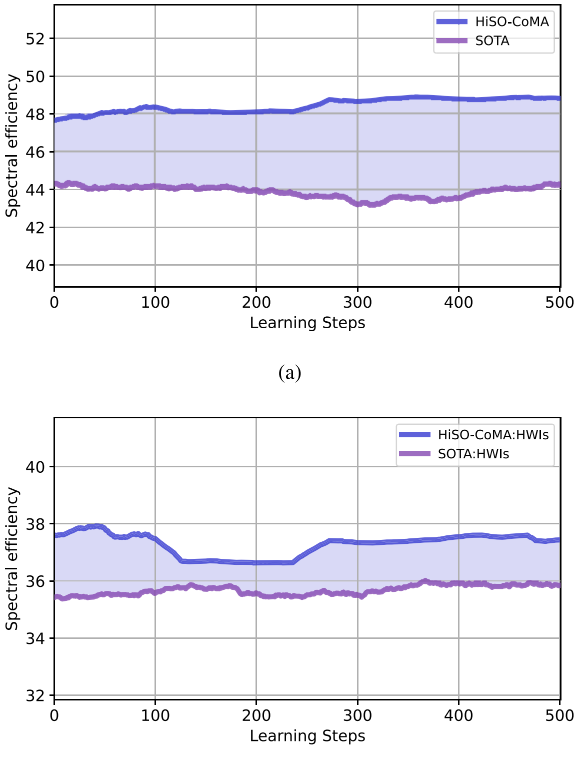
- *GGA scheduler:* This scheduling approach employs a greedy allocation algorithm to manage both inter- and intra-RAN slicing.
- *EEA scheduler:* This scheduling approach utilizes an equal or hard allocation algorithm for both inter- and intra-RAN slicing.
- *SA2C-T scheduler [7]:* This scheduler is 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.

C. 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. (31) and Eq. (32)) and MADQN agents at the intra-RAN slicing level (Eq. (34)). Figure 5 shows the variations in training loss for both the inter- and intra-level control policies, where we can observe that 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 III-D.

D. EVALUATION OF THE PROPOSED HiSO-CoMA VS. THE SOTA

Figures 6 and 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. 6 and 7, it can be observed that the proposed framework outperforms the SOTA approach for both performance metrics and under both ideal 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.



(a)

(b)

FIGURE 7. **Spectral efficiency of the proposed framework vs. the SOTA approach under (a) ideal HW and (b) HWIs.**

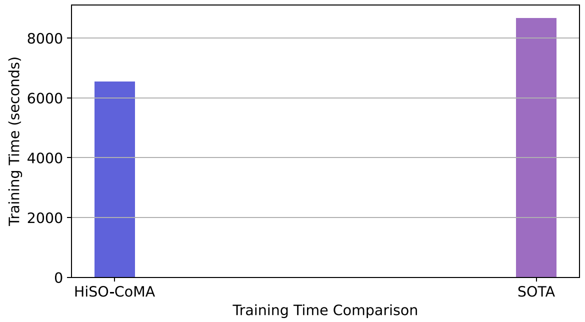


FIGURE 8. **Training time of the proposed framework vs. the SOTA approach in the presence of HWIs.**

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. 6(b) and 7(b). However, this does not impact the overall convergence time, demonstrating the adaptability and robustness of the proposed framework under varying conditions.

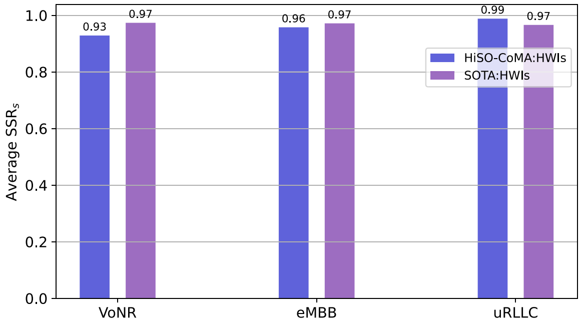


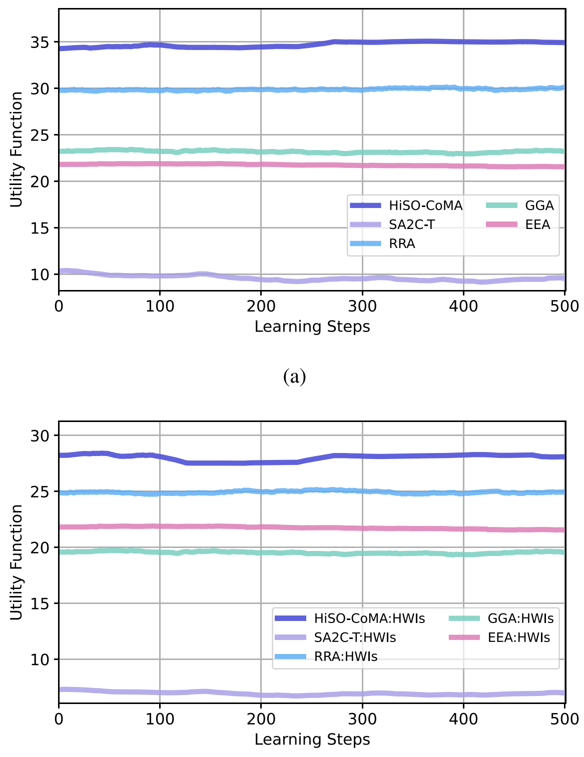
FIGURE 9. **Average QoS of the proposed framework vs. the SOTA approach under HWIs.**

Figure 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 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.

E. PERFORMANCE OF THE PROPOSED HiSO-CoMA.VS BASELINES

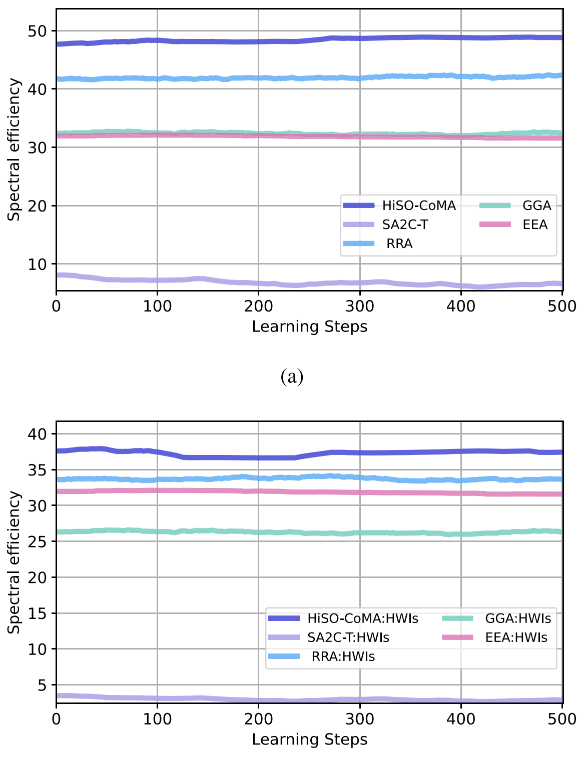
Figure 10 shows the performance of the proposed HiSO-CoMA framework in optimizing the objective function under both ideal and non-ideal HW conditions, and compares it with various resource allocation schedulers. It is observed that the proposed framework outperforms the baseline schedulers under both conditions. Unexpectedly, SA2C-T, which



(a)

(b)

FIGURE 10. Utility function of the proposed framework vs. baselines under (a) ideal HW and (b) HWIs.



(a)

(b)

FIGURE 11. Spectral efficiency of the proposed framework vs. baseline schedulers under (a) ideal HW and (b) HWIs.

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 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. 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 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.

F. 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. 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. 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

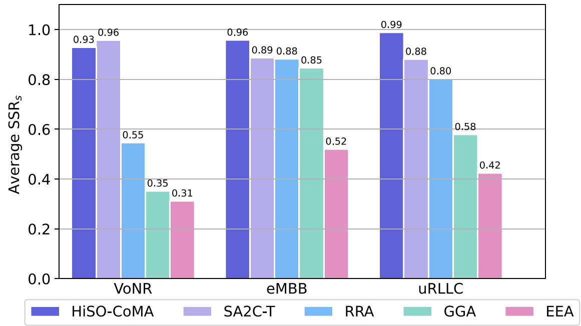


FIGURE 12. Average SSR for heterogeneous services under HWIs.

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.

G. 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. 14 and 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.

H. 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. 16, both the utility function and the spectral efficiency degrade as severity of HWIs increases. This highlights the effect 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

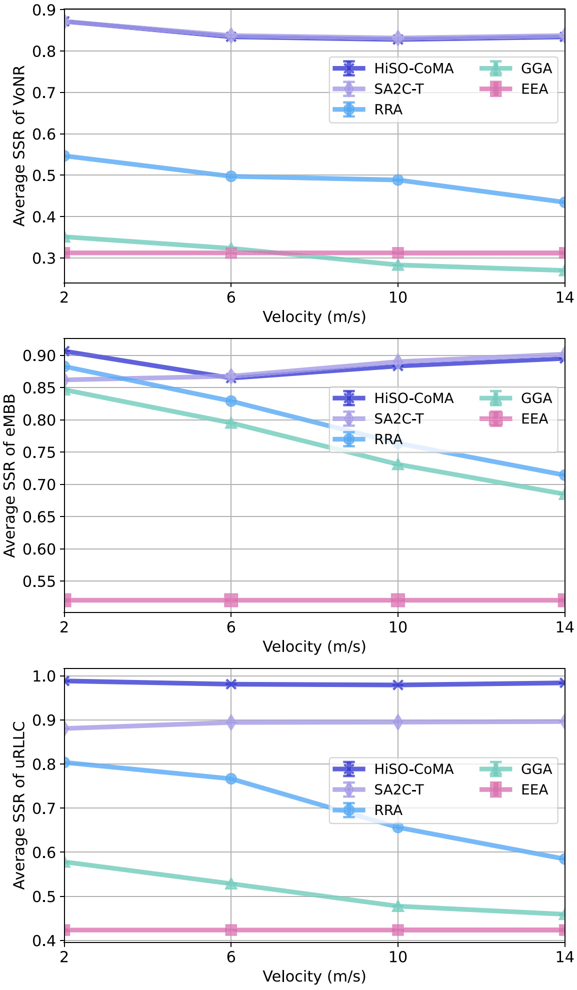


FIGURE 13. Impact of mobility on average QoS under various allocation schedulers.

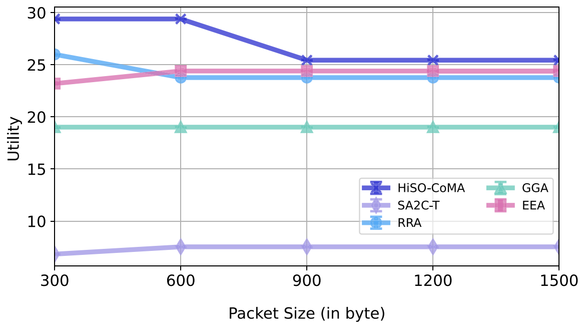


FIGURE 14. Utility of the proposed HiSO-CoMA framework vs. baselines strategies under HWIs and varying packet sizes for the eMBB slice.

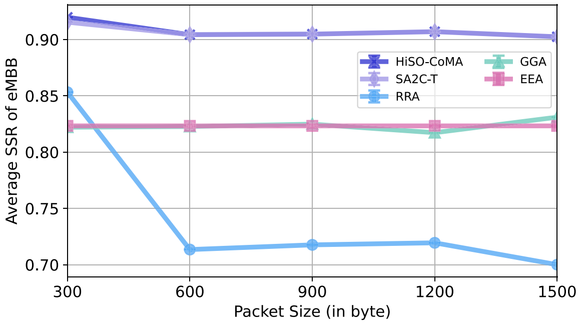
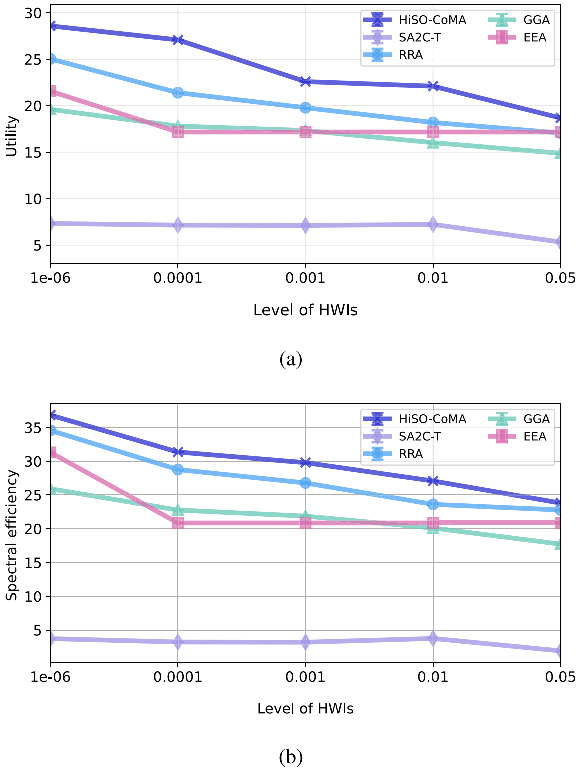


FIGURE 15. Average QoS of the eMBB slice for various packet sizes under HWIs.



(a)

(b)

FIGURE 16. Utility and spectral efficiency of the proposed framework vs. baselines under various levels of HWIs.

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 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. 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.

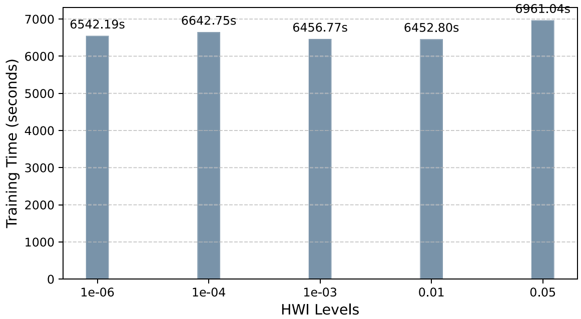


FIGURE 17. Training time of proposed HiSO-CoMA framework under various levels of HWIs.

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 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.

V. CONCLUSION AND FUTURE WORKS

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.

REFERENCES

[1] L. Yang, M. E. Rajab, A. Shami, and S. Muhaidat, "Enabling AutoML for zero-touch network security: Use-case driven analysis," *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 3, pp. 3555–3582, Jun. 2024.

[2] F. Rezazadeh, H. Chergui, and C. Verikoukis, "Zero-touch continuous network slicing control via scalable actor-critic learning," 2021, *arXiv:2101.06654*.

[3] O. Sabr, G. Kaddoum, and K. Kaur, "PABSO-DRL: Power and beam self-optimization scheme for multiple slices in MU-MISO systems," *IEEE Trans. Consum. Electron.*, vol. 71, no. 2, pp. 4343–4358, May 2025.

[4] M. Setayesh, S. Bahrami, and V. W. Wong, "Resource slicing for eMBB and uRLLC services in radio access network using hierarchical deep learning," *IEEE Trans. Wireless Commun.*, vol. 21, no. 11, pp. 8950–8966, Nov. 2022.

[5] F. Debbabi, R. Jmal, L. C. Fourati, and R. L. Aguiar, "An overview of interslice and intraslice resource allocation in B5G telecommunication networks," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 4, pp. 5120–5132, Dec. 2022.

[6] Y. Shao, R. Li, B. Hu, Y. Wu, Z. Zhao, and H. Zhang, "Graph attention network-based multi-agent reinforcement learning for slicing resource management in dense cellular network," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10792–10803, Oct. 2021.

[7] 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," in *Proc. IEEE Int. Conf. Commun.*, 2025, pp. 1414–1419.

[8] Y. Zhao, X. Chi, L. Qian, Y. Zhu, and F. Hou, "Resource allocation and slicing puncture in cellular networks with eMBB and URLLC terminals coexistence," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18431–18444, Oct. 2022.

[9] J. Tang, B. Shim, T.-H. Chang, and T. Q. S. Quek, "Incorporating URLLC and multicast eMBB in sliced cloud radio access network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–7.

[10] N. U. Ginige, K. B. S. Manosha, N. Rajatheva, and M. Latva-aho, "Admission control in 5G networks for the coexistence of eMBB-URLLC users," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, 2020, pp. 1–6.

[11] A. Slalmi, H. Chaibi, R. Saadane, and A. Chehri, "Call admission control optimization in 5G in downlink single-cell MISO system," *Procedia Comput. Sci.*, vol. 192, pp. 2502–2511, Oct. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921017567

[12] F. Tan, S. Si, H. Chen, S. Li, and T. Lv, "Rate splitting multiple access assisted cell-free massive MIMO for URLLC services in 5G and beyond networks," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 6018–6032, 2024.

[13] S. K. Taskou and M. Rasti, "Resource allocation for FeMBB and eURLLC coexistence in RSMA-based cellular networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2024, pp. 1–6.

[14] R. Huang, V. W. Wong, and R. Schober, "Rate-splitting for intelligent reflecting surface-aided multiuser VR streaming," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 5, pp. 1516–1535, May 2023.

[15] O. Dizdar, Y. Mao, Y. Xu, P. Zhu, and B. Clerckx, "Rate-splitting multiple access for enhanced URLLC and eMBB in 6G: Invited paper," in *Proc. 17th Int. Symp. Wireless Commun. Syst. (ISWCS)*, 2021, pp. 1–6.

[16] T. Li, H. Zhang, S. Guo, and D. Yuan, "Robust rate-splitting and beamforming for ultra-reliable and low-latency communications," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 15571–15585, Oct. 2024.

[17] A. M. Nagib, H. Abou-Zeid, and H. S. Hassanein, "Safe and accelerated deep reinforcement learning-based O-RAN slicing: A hybrid transfer learning approach," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 2, pp. 310–325, Feb. 2024.

[18] C. Qi, Y. Hua, R. Li, Z. Zhao, and H. Zhang, "Deep reinforcement learning with discrete normalized advantage functions for resource management in network slicing," *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1337–1341, Aug. 2019.

[19] R. Li, C. Wang, Z. Zhao, R. Guo, and H. Zhang, "The LSTM-based advantage actor-critic learning for resource management in network slicing with user mobility," *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 2005–2009, Sep. 2020.

[20] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "GAN-powered deep distributional reinforcement learning for resource management in network slicing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 334–349, Feb. 2020.

[21] D. Yan, B. K. NG, W. Ke, and C.-T. Lam, "Multi-agent deep reinforcement learning joint beamforming for slicing resource allocation," *IEEE Wireless Commun. Lett.*, vol. 13, no. 5, pp. 1220–1224, May 2024.

[22] X. Chang, T. Ji, R. Zhu, Z. Wu, C. Li, and Y. Jiang, "Toward an efficient and dynamic allocation of radio access network slicing resources for 5G era," *IEEE Access*, vol. 11, pp. 95037–95050, 2023.

[23] D. Yan, B. K. Ng, W. Ke, and C.-T. Lam, "Deep reinforcement learning based resource allocation for network slicing with massive MIMO," *IEEE Access*, vol. 11, pp. 75899–75911, 2023.

[24] A. Filali, Z. Mlika, S. Cherkaoui, and A. Kobbane, "Dynamic SDN-based radio access network slicing with deep reinforcement learning for URLLC and eMBB services," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 4, pp. 2174–2187, Jul./Aug. 2022.

[25] J. Mei, X. Wang, K. Zheng, G. Boudreau, A. B. Sediq, and H. Abou-Zeid, "Intelligent radio access network slicing for service provisioning in 6G: A hierarchical deep reinforcement learning approach," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6063–6078, Sep. 2021.

[26] E. Björnson, J. Hoydis, M. Kountouris, and M. Debbah, "Hardware impairments in large-scale MISO systems: Energy efficiency, estimation, and capacity limits," in *Proc. 18th Int. Conf. Digit. Signal Process. (DSP)*, 2013, pp. 1–6.

[27] F. Zhang, S. Sun, B. Rong, F. R. Yu, and K. Lu, "A novel massive MIMO precoding scheme for next generation heterogeneous networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2015, pp. 1–6.

[28] P.-H. Liao, L.-H. Shen, P.-C. Wu, and K.-T. Feng, "Multi-agent deep reinforcement learning for energy efficient multi-hop STAR-RIS-assisted transmissions," in *Proc. IEEE 100th Veh. Technol. Conf. (VTC-Fall)*, 2024, pp. 1–5.

[29] M. Dubey, A. K. Singh, and R. Mishra, "AI based resource management for 5G network slicing: History, use cases, and research directions," *Concurr. Comput. Pract. Exp.*, vol. 37, no. 2, 2025, Art. no. e8327.

[30] Y. Mao, O. Dizdar, B. Clerckx, R. Schober, P. Popovski, and H. V. Poor, "Rate-splitting multiple access: Fundamentals, survey, and future research trends," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 4, pp. 2073–2126, 4th Quart., 2022.

[31] M. M. Usha, G. Sadashivappa, and R. Palepu, "Integration of RIC and xApps for open-radio access network for performance optimization," in *Proc. 7th Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. (CSITSS)*, 2023, pp. 1–4.

[32] M. V. Ngo et al., “RAN intelligent controller (RIC): From open-source implementation to real-world validation,” *ICT Exp.*, vol. 10, no. 3, pp. 680–691, 2024.

[33] H. Zhang, G. Pan, S. Xu, S. Zhang, and Z. Jiang, “A hard and soft hybrid slicing framework for service level agreement guarantee via deep reinforcement learning,” in *Proc. IEEE 95th Veh. Technol. Conf.*, 2022, pp. 1–5.

[34] A. Anand and G. de Veciana, “Resource allocation and HARQ optimization for URLLC traffic in 5G wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2411–2421, Nov. 2018.

[35] J. Ge, Y.-C. Liang, J. Joung, and S. Sun, “Deep reinforcement learning for distributed dynamic MISO downlink-beamforming coordination,” *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 6070–6085, Oct. 2020.

[36] X. Wang, G. Sun, Y. Xin, T. Liu, and Y. Xu, “Deep transfer reinforcement learning for beamforming and resource allocation in multi-cell MISO-OFDMA systems,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 815–829, 2022.

[37] J. Tang, B. Shim, and T. Q. S. Quek, “Service multiplexing and revenue maximization in sliced C-RAN incorporated with URLLC and multicast eMBB,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 881–895, Apr. 2019.

[38] M. Zangooei, N. Saha, M. Golkarifard, and R. Boutaba, “Reinforcement learning for radio resource management in RAN slicing: A survey,” *IEEE Wireless Commun. Mag.*, vol. 61, no. 2, pp. 118–124, Feb. 2023.

[39] S. Pala, M. Katwe, K. Singh, B. Clerckx, and C.-P. Li, “Spectral-efficient RIS-aided RSMA URLLC: Toward mobile broadband reliable low latency communication (MBRLLC) system,” *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 3507–3524, Apr. 2024.

[40] N. Saha, M. Zangooei, M. Golkarifard, and R. Boutaba, “Deep reinforcement learning approaches to network slice scaling and placement: A survey,” *IEEE Commun. Mag.*, vol. 61, no. 2, pp. 82–87, Feb. 2023.

[41] W. Liu, Y. Fu, Y. Guo, F. Lee Wang, W. Sun, and Y. Zhang, “Two-timescale synchronization and migration for digital twin networks: A multi-agent deep reinforcement learning approach,” *IEEE Trans. Wireless Commun.*, vol. 23, no. 11, pp. 17294–17309, Nov. 2024.

[42] W. Wang, L. Tang, T. Liu, X. He, C. Liang, and Q. Chen, “Toward reliability-enhanced, delay-guaranteed dynamic network slicing: A multiagent DQN approach with an action space reduction strategy,” *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9282–9297, Mar. 2024.

[43] D.-D. Tran, S. K. Sharma, V. N. Ha, S. Chatzinotas, and I. Woungang, “Multi-agent DRL approach for energy-efficient resource allocation in URLLC-enabled grant-free NOMA systems,” *IEEE Open J. Commun. Soc.*, vol. 4, pp. 1470–1486, 2023.

[44] O. Dizdar, Y. Mao, and B. Clerckx, “Rate-splitting multiple access to mitigate the curse of mobility in (massive) MIMO networks,” *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 6765–6780, Oct. 2021.

[45] Y. Sun and X. Zhang, “A2C learning for tasks segmentation with cooperative computing in edge computing networks,” in *Proc. IEEE Global Commun. Conf.*, 2022, pp. 2236–2241.

[46] L. Zhu and L. Tan, “Task offloading scheme of vehicular cloud edge computing based on digital twin and improved A3C,” *Internet of Things*, vol. 26, Jul. 2024, Art. no. 101192.

OHOOD SABR (Graduate Student Member, IEEE) received the B.Eng. (First Class Hons.) and M.Sc. (Distinction) degrees in network engineering from the University of Northampton, U.K. She is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, École de Technologie Supérieure, Université du Québec, Montreal, Canada. Her research interests include networks, mobile applications, machine learning, the Internet of Things, the Internet of Everything, network security, Web security, indoor localization, identification systems, and IBM Bluemix applications. Throughout her academic journey, she received numerous prestigious awards, including the Prestigious HMGCC (Her Majesty’s Government Communications Centre) Award for the Best Project in 2015 and 2016. In 2017, she was the winner of the Northampton Branch Prize, awarded by the British Computer Society for the Best Final Year Student on the B.Sc. Computing Engineering, as well as the BCS Membership Prize. Also, she won the Babylon M.Sc. Scholarship and the HMGCC Prize for the Best Final Student 2017, recognizing her exceptional academic achievements. Moreover, she was honored the 2nd prize with her team in the Internet of Things Workshop’s competition in U.K.

GEORGES KADDOUM (Senior Member, IEEE) received the bachelor’s degree in electrical engineering from the École Nationale Supérieure de Techniques Avancées, Brest, France, the M.S. degree in telecommunications and signal processing (circuits, systems, and signal processing) from the Université de Bretagne Occidentale and Telecom Bretagne, Brest, in 2005, and the Ph.D. degree (High Hons.) in signal processing and telecommunications from the National Institute of Applied Sciences, University of Toulouse, Toulouse, France, in 2009. He is currently a Professor, the Research Director with the Resilient Machine Learning Institute, and the Tier 2 Canada Research Chair with the École de Technologie Supérieure (ÉTS), Université du Québec, Montreal, Canada. He has published over more than 300 journals, conference papers, two chapters in books, and has eight pending patents. His recent research interests include wireless communication networks, tactical communications, resource allocations, and network security. He received the Best Papers Awards at the 2014 IEEE International Conference on Wireless and Mobile Computing, Networking, Communications, the 2017 IEEE International Symposium on Personal Indoor and Mobile Radio Communications, and the 2023 IEEE International Wireless Communications and Mobile Computing Conference. Moreover, he received the IEEE Transactions on Communications Exemplary Reviewer Award in 2015, 2017, and 2019. In addition, he received the Research Excellence Award of the Université du Québec in 2018. In 2019, he received the Research Excellence Award from ÉTS in recognition of his outstanding research outcomes. He also won the 2022 IEEE Technical Committee on Scalable Computing Award for Excellence (Middle Career Researcher). Finally, he has received the prestigious 2023 MITACS Award for Exceptional Leadership. He served as an Associate Editor for IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and IEEE COMMUNICATIONS LETTERS. He is currently serving as an Area Editor for the IEEE TRANSACTIONS ON MACHINE LEARNING IN COMMUNICATIONS AND NETWORKING and an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS.

KULJEET KAUR (Member, IEEE) has been an Associate Professor with École de technologie supérieure (ÉTS), Montreal, since 2020. Her research interests are cybersecurity, cloud/edge computing, the Internet of Things, applied machine learning and artificial intelligence, communications, and smart grids. She published over 75 scientific/technical articles and 3 books. She has secured research funding from various sources, such as the Natural Sciences and Engineering Research Council of Canada, Fonds de Recherche du Québec Nature et technologies, Department of Science and Technology, and TCS Innovations Labs.

Dr. Kaur is the recipient of the 2023 N2Women Rising Stars in Networking and Communications and the 2021 IEEE Technical Committee on Scalable Computing (TCSC) Award for Excellence in Scalable Computing for Early Career Researchers. She was also awarded the 2021 IEEE System Journal and 2018 IEEE ICC best paper awards. She also received the Best Research Paper Awards from the Thapar Institute of Engineering & Technology in 2022 and 2019. She has been serving as an Editor/ a Guest Editor for various international journals of repute, such as *Computer Communications*, *Ad hoc Networks*, *Wiley Security and Privacy Journal*, *Journal of Information Processing Systems*, *Human-centric Computing and Information Sciences* (Springer), *Frontiers in Communications and Networking: Board of Smart Grid Communications*, *International Journal of Applied Engineering Research*, and *Human-centric Computing and Information Sciences* (Springer). She has organized different special issues at different venues, such as IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS OF INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, and IEEE Open Journal of the Computer Society. She has been organizing international symposiums and workshops for several flagship conferences, such as IEEE GLOBECOM, IEEE INFOCOM, IEEE ICC, and ACM MOBICOM. She served as a Technical Program Committee Member for several international conferences, including IEEE GLOBECOM and IEEE ICC. She serves as the Faculty Representative for the IEEE ÉTS Student Branch and a Secretary for the IEEE ComSoc Women in Communications Engineering. She served as the Deputy Secretary for the IEEE WICE from 2022 to 2024, the Vice-Chair from 2020 to 2022 of the IEEE Montreal Young Professionals Affinity Group and the Website Co-Chair from 2020 to 2022 for the Networking Women (N2Women), a Discipline-Specific Community for female researchers.