Received 25 March 2026; accepted 12 April 2026. Date of publication 20 April 2026;
date of current version 5 May 2026. The review of this article was coordinated by Editor Zhuo Sun.

Digital Object Identifier 10.1109/OJVT.2026.3685549

DRL-Based RAN Slicing With Efficient Inter-Slice Isolation in Tactical Wireless Networks

ABDERRAHIME FILALI, DIALA NABOULSI (Member, IEEE), AND GEORGES KADDOUM (Senior Member, IEEE)

École de Technologie Supérieure, Montreal, QC H3C 1K3, Canada

CORRESPONDING AUTHOR: ABDERRAHIME FILALI (e-mail: filali.abderrahime@gmail.com).

This work was supported by Mitacs/Ultra Intelligence and Communications under Project IT25839.

ABSTRACT The next generation of tactical networks (TNs) is poised to further leverage the key enablers of 5G and beyond 5G (B5G) technology, such as radio access network (RAN) slicing and the open RAN (O-RAN) paradigm, to unlock new architectural options for innovative applications. RAN slicing and the O-RAN paradigm are considered game changers in TNs, where the former enables tailoring services to users' requirements, and the latter brings openness and intelligence to RAN management. In TNs, bandwidth scarcity requires dynamic bandwidth slicing strategies. Although such strategies ensure efficient bandwidth utilization, they may compromise RAN slicing isolation in terms of quality of service (QoS). To deal with this challenge, we propose a deep reinforcement learning (DRL)-based RAN slicing mechanism, termed here as sharing and inter- and intra-slice isolation (BS-IISI), which achieves a trade-off between efficient bandwidth sharing and robust inter- and intra-slice isolation. BS-IISI performs bandwidth allocation in two stages. In the first stage, bandwidth is allocated to RAN slices, while in the second stage, each slice distributes its allocated bandwidth among associated users. In both stages, the slicing operation is constrained by QoS-related considerations that enhance inter- and intra-slice isolation. BS-IISI relies on DRL algorithms to perform bandwidth sharing at each allocation stage. We propose deploying the mechanism within an O-RAN architecture and describe the relevant O-RAN functional blocks and the main phases of the DRL model lifecycle management. Finally, we develop three implementations of the proposed mechanism, each based on a different DRL algorithm, and evaluate their performance against several baselines under diverse network configurations.

INDEX TERMS Deep reinforcement learning, open RAN, RAN slicing, tactical network, 5G.

I. INTRODUCTION

The next generation of tactical networks (TNs) is expected to have advanced capabilities to support a plethora of applications related to command, control, communication, intelligence, and surveillance that will enhance the operation of the command-and-control structure. TNs have seized the opportunity to embrace 5G and beyond 5G (B5G) commercial technology to achieve this objective. Indeed, 5G and B5G networks are paving the way towards a robust, resilient, and secure communication infrastructure for TNs [1]. Embracing 5G and B5G technology enables TNs to leverage their key enablers, such as radio access network (RAN) slicing and the open RAN (O-RAN) paradigm.

RAN slicing technology makes it possible to virtually divide a shared physical RAN into several logical networks called RAN slices [2]. Each RAN slice can have its own network performance indicators in accordance with the quality of service (QoS) requirements that apply to the service it provides. In TNs, RAN slicing makes it possible to handle a variety of tactical applications by devoting to each RAN slice the RAN resources it requires. Therefore, the integration of RAN slicing in TNs marks a radical shift towards flexible, scalable, and resource-optimized networks.

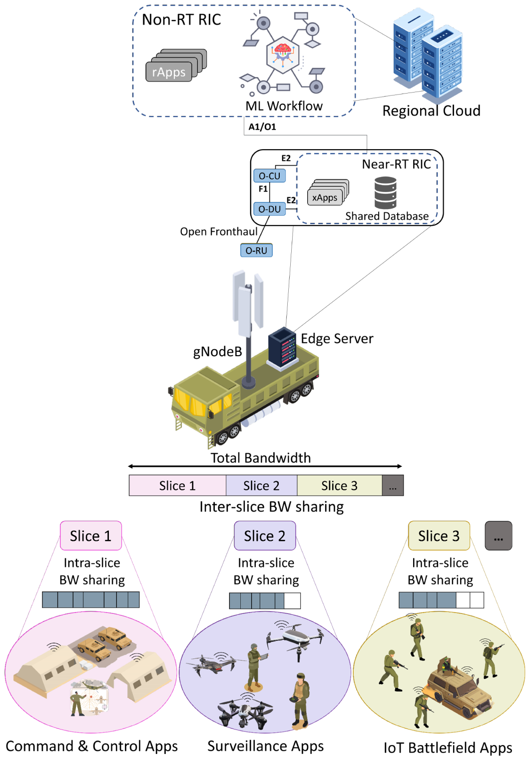


FIGURE 1. **O-RAN system model for a TN environment.**

To ensure the sustainable development of their RAN communication systems, TNs must become more receptive to existing RAN software and hardware products that have been developed by commercial technology suppliers. The flexibility to adopt third-party RAN solutions can be achieved through the O-RAN paradigm [3]. The main idea behind the O-RAN paradigm is to support RAN component intercompatibility and interoperability by making RAN hardware and software elements not locked to a specific supplier. The O-RAN paradigm enables TNs to take advantage of innovations developed by a wide range of suppliers to adapt their RAN infrastructure to cutting-edge technologies [4]. Another game-changing capability empowered by the O-RAN paradigm is the automation of RAN operations management, such as RAN slicing, using machine learning (ML) [5]. The desire for a more programmable RAN is driven by a pivotal component of the O-RAN architecture called the RAN intelligent controller (RIC) [6], [7], which is shown in Fig. 1. TNs can use the RIC to gather and analyze large volumes of data from the RAN. These data are leveraged to train and refine ML models embedded in the RIC, enabling the optimization of network operations and enhancing decision-making processes for command units. RAN slicing and the O-RAN paradigm hold significant promises for enabling the next generation of TNs to deliver innovative services. However, implementing RAN slicing in TNs is complicated by several inherent challenges. First, TNs operate in a highly dynamic RAN environment where wireless channel transmission conditions are subject to constant and unpredictable changes. In addition, the scarcity of radio resources, such as bandwidth, represents a significant limitation for these networks. Adapting to this fast-changing and resource-constrained environment requires dynamic bandwidth sharing. However, this approach introduces challenges in maintaining slice isolation. In this study, we define RAN slicing isolation from the perspective of Quality of Service (QoS) performance. Specifically, isolating a slice refers to ensuring it is allocated the resources needed to meet its QoS requirements. This reduces the impact of fluctuations in the resource requirements of one slice on the QoS performance of the other slices. These issues highlight the significant challenges involved in implementing effective and reliable RAN slicing in TNs.

RAN slicing in TNs has been investigated in the literature [8], [9], [10], but with many limitations, mainly the ignoring of isolation, user mobility and user QoS diversity in the RAN slicing schemes proposed. In non-TNs, RAN slicing isolation has been explored at different levels of granularity, namely the slice level and the user level. In addition, several resource- and performance-based metrics have been used to define isolation in a RAN [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26]. Nevertheles, the schemes proposed focus mainly on satisfying isolation constraints, without regard to resource wastage. In other words, these schemes tolerate a slice or a user for being allocated more than the minimum amount of resources needed to meet their QoS requirements. This can be acceptable in non-TNs, where the bandwidth is more easily accessible than in TNs. In TNs, however, bandwidth is a very precious resource and should be meticulously allocated to avoid wastage. Moreover, wasting it can be detrimental to RAN slicing isolation. In fact, wasting the bandwidth in RAN slicing leads to frequent reconfiguration of the resources allocated to slices or users, which has an impact on isolation.

The novelty of this work is to propose a RAN slicing mechanism for O-RAN TNs. The objective of the proposed mechanism is to achieve a trade-off between efficient RAN bandwidth sharing and appropriate inter- and intra-slice isolation. The mechanism is called the bandwidth sharing and inter- and intra-slice isolation (BS-IISI) mechanism.

The BS-IISI mechanism hierarchically performs bandwidth allocation in two stages, namely inter-slice bandwidth sharing and intra-slice bandwidth sharing. It ensures adequate isolation both at the inter- and intra-slice levels of granularity and in terms of resource and QoS performance. We define a utility function to calculate the degree of user satisfaction. It determines whether the bandwidth allocated to a user fulfills its QoS requirements and captures any resource wastage during bandwidth allocation. User satisfaction increases when a user's achieved data rate gets closer to the required data rate threshold and decreases when it gets further from the required threshold (above or below). This allows bandwidth to be allocated efficiently and prevents resources from being wasted, which, in turn, contributes to isolation. We define slice resource reconfiguration constraints to enhance inter-slice isolation. These constraints ensure that resources can be removed from a slice that has excess resources and added to a slice that requires additional resources. This avoids unnecessary slice resource reconfiguration, which improves inter-slice isolation.

In both stages of RAN slicing, i.e., inter- and intra-slice bandwidth sharing, the BS-IISI mechanism utilizes deep reinforcement learning (DRL) algorithms to perform bandwidth allocation. We propose to deploy the BS-IISI mechanism in an O-RAN architecture, which follows the specifications issued by the O-RAN Alliance Working Group (WG) 2. Integrating the BS-IISI mechanism in the O-RAN architecture allows for a high degree of automation and flexibility in RAN slicing operations to be able to instantly adapt to fluctuations in user traffic and the environmental conditions of TNs.

This paper's main contributions are summarized as follows:

- We model the bandwidth allocation problem as a continuous optimization problem to maximize the user degree of satisfaction and ensure inter- and intra-slice isolation.
- We propose a DRL-based RAN slicing mechanism to solve the bandwidth allocation problem in each stage of RAN slicing.
- We propose to deploy the BS-IISI mechanism in an O-RAN architecture, and describe the O-RAN functional blocks involved and the key DRL model lifecycle management phases involved.
- We develop three different implementations of the BS-IISI mechanism, each based on a different DRL algorithm, and subsequently evaluate their performance against multiple baselines across various parameters.

The rest of the paper is structured as follows: Section II reviews recent related work, Section III outlines the system model, Section IV formulates the problem, Section V details the proposed RAN slicing mechanism, Section VI evaluates its performance, and Section VII concludes the paper.

II. RELATED WORK

In this section, we first review recent research pertaining to RAN slicing in TNs. Then, we present an overview of relevant works related to RAN slicing isolation and classify these works based on their level of isolation granularity level and the isolation metrics they consider.

A. RAN SLICING IN TACTICAL NETWORKS

In [8], the authors use game theory to model resource slicing in ad-hoc TNs as a non-cooperative game where slices compete for shared bandwidth to maximize throughput-based utility. They propose an iterative algorithm that converges to a Nash equilibrium, enabling slices to alternately select bandwidth to optimize their throughput. In [9], the authors propose an algorithm to maximize device associations with tactical RAN slices using a priority mechanism based on the ratio of QoS to required bandwidth. In limited bandwidth scenarios, the algorithm reallocates resources from lower-priority slices to higher-priority ones to optimize device connectivity. In [10], the authors design a RAN slicing framework for managing the lifecycle of RAN slices in battlefield networks, with a slice controller that monitors QoS requirements and allocates necessary resources. These proposed approaches overlook inter-slice isolation, user mobility, and the clear definition of individual QoS requirements for slices.

B. RAN SLICING ISOLATION

Various works in the literature consider intra-cell isolation in their proposed RAN slicing schemes. These schemes can be classified by their isolation granularity and the isolation metrics they consider. Isolation granularity consists of two main levels, namely inter-slice isolation and intra-slice isolation. Several metrics have been used to gauge isolation, such as minimum and maximum thresholds related to allocated resources or QoS performance, as well as load fluctuation. These isolation metrics can be grouped into two classes: resource-based metrics and performance-based metrics.

In [11], the authors focus on ensuring inter-slice isolation by minimizing violations of service level agreements (SLAs). SLA violations are found to occur when the data rate or latency requirements of a slice's users are not met. To further enhance inter-slice isolation, resource sharing among slices is prohibited, even when a slice cannot fully use the resources allocated to it within a time window. In [12], a two-level scheme for RAN resource sharing is proposed to ensure slice isolation. At a high level, the system is found to guarantee the availability of RAN resources by monitoring the network and resource availability, and then assigning each user's request to the appropriate slice type. At the low level, based on the high-level decisions, the necessary resources were allocated to each user, which maintained isolation between slices. In [13], RAN slice isolation is ensured by allocating to each slice a number of resource blocks (RBs) that is greater than the number requested and less than the maximum threshold value. In [14], isolation is considered at the slice level and the user level subject to the constraints that a RB can be allocated to at most one slice at a time and each user must have a minimum number of RBs. In [15], the objective is to maximize the number of RBs allocated to a slice, which is dependent on the slice's traffic variation. This slice isolation metric is related to how much consideration is given to slice traffic variation in the RBs allocation process.

In [16], [17], isolation is defined as a ratio index of a user's QoS requirement to the QoS they actually achieve. This index is limited by a minimum value. Furthermore, each RAN slice's throughput must not exceed a maximum threshold. In [18], RAN slice isolation is ensured by keeping the resources allocated to a given slice unchanged for a specific period of time. In other words, no RAN slice resource reconfiguration is performed during the time interval. In [19], the authors adopted a similar approach to sustain RAN slice isolation at a higher level, involving the prediction of slice resource requirements on a large time scale to guarantee stable resource allocation. The work in [20] ensures RAN slice isolation by setting minimum and maximum thresholds for the number of RBs allocated to enhanced mobile broadband (eMBB) users and capping allocations for ultra-reliable low-latency communication (URLLC) and massive machine-type

TABLE 1. Classification of RAN Slice Isolation Related Work

Ref.	Isolation granularity		Isolation metric	
	Inter-slice	Intra-slice	Resource-based	Performance-based
[13], [15], [26]	✓		✓	
[14], [18]	✓	✓	✓	
[16]	✓	✓		✓
[12], [17], [19], [21], [25]	✓			✓
[20]	✓	✓	✓	✓
[22], [23]		✓	✓	
[11], [24]	✓		✓	✓

communication (mMTC) users. Additionally, user-level isolation is maintained by enforcing a minimum data rate threshold for each user. In [21], the infrastructure provider ensures the isolation of tenants' RAN slices in terms of data rate as long as the number of users associated with each slice does not exceed the contracted threshold. In [22], the number of RBs a user requires is estimated from the throughput needed in the physical layer. Then, a slice receives the amount of resources that corresponds to the aggregation of the RBs estimated to schedule all its users.

In [24], the authors perform inter-slice resource allocation by reserving RBs for slices on long time scales based on predicted resource demands. To ensure a high level of resource isolation, the reservation process involves lending or borrowing resources between slices to maintain the necessary number of RBs. Next, a RB association operation is conducted to allocate the RBs to slices to uphold a high level of interference isolation. In [25], slice isolation is enhanced between interfering slices that operate on overlapping channels. This isolation is defined by the variation in user delay and throughput when traffic from interfering slices increases. An interference-aware channel allocation policy is proposed to reduce overlapping frequency channels. In [26], eMBB and URLLC users are grouped based on their channel quality. Isolation is achieved by allocating RBs to each group in a way that maximizes the throughput for eMBB users while satisfying the delay requirements for URLLC users. Table 1 summarizes a classification of related work that investigated RAN slicing isolation.Although RAN slicing isolation has been studied in previous works, there are still isolation-related research challenges that require more in-depth investigation. In this work, we propose a RAN slicing mechanism that ensures isolation both at the inter- and intra-slice levels of granularity and in terms of resource and QoS performance. Moreover, in previous research works, isolation has been achieved by allocating resources without any concern for their wastage. However, wasting resources can lead to poor isolation, since frequent resource reconfiguration is necessary when a large amount of resources are unused. The BS-IISI mechanism improves the efficiency of RAN slicing isolation while avoiding resource wastage.

III. SYSTEM MODEL

We consider a 5G O-RAN system composed of a single base station, i.e., a 5G next-generation node B (gNodeB), (see Fig. 1). The set of users associated with the gNodeB is denoted by $\mathcal{U} = \{1, \ldots, U\}$, where $|\mathcal{U}| = U$ .The gNodeB can support the deployment of a set of RAN slices denoted by $\mathcal{S} = \{1, \ldots, S\}$, where $|\mathcal{S}| = S$. Each RAN slice $s \in \mathcal{S}$ is designed to serve a subset of users denoted by $\mathcal{U}_s = \{1, \ldots, U_s\}$. Note that all users that are served by the same slice have the same QoS requirements. We assume that a user can be served by only one RAN slice, such that $\mathcal{U}_s \cap \mathcal{U}_{s'} = \emptyset, \forall s \neq s'$ and $\cup_{s\in\mathcal{S}}\mathcal{U}_s = \mathcal{U}$. We consider that the time dimension is divided into several time slots indexed by t. We assume that RAN slicing is performed at the beginning of each time slot t.

We consider that users move within the gNodeB's coverage area following the random waypoint (RWP) mobility model [27]. Each user $u \in \mathcal{U}$ selects a random destination and moves to it at a constant speed v_u. Upon arrival, the user pauses for a random period τ_u before moving on. The user's velocity and pause time are assumed to follow the uniform distribution $\mathcal{U}(v_{\min}, v_{\max})$ and $\mathcal{U}(0, \tau_{\max})$, respectively. The variables $v_{\min}$, $v_{\max}$, and $\tau_{\max}$ denote the user's minimum velocity, maximum velocity, and maximum pause time, respectively.

A. O-RAN ARCHITECTURE

Fig. 1 presents the considered O-RAN architecture that includes the following main components: the near-real-time (near-RT) RIC, the non-real-time (non-RT) RIC, the O-RAN radio unit (O-RU), the O-RAN distributed unit (O-DU), and the O-RAN central unit (O-CU). The near-RT RIC manages near-real-time control and optimization of O-RAN components and resources. By contrast, the non-RT RIC performs non-real-time management and optimization operations, handling tasks with latencies exceeding 1 s. In their turn, the O-RU, O-DU, and O-CU handle the gNodeB functions, with each unit responsible for specific tasks depending on the functional split option chosen, as defined by 3GPP [28]. The O-RAN Alliance has selected the 7.2x split option, with the O-RU handling lower-layer physical functions and the O-DU managing higher-layer tasks. This split ensures flexibility, with the precoding function located either in the O-DU (Category A) or in the O-RU (Category B) [29]. In this study, we selected Category A to keep the O-RU as simple as possible, as a less complex O-RU reduces size and power consumption, which are key factors in TNs. This architecture is ideal for TNs because it minimizes fronthaul bandwidth requirements, thus enabling low-latency and high-reliability communication in a resource-constrained environment.

To implement the O-RAN components in a network, the O-RAN Alliance defined six deployment scenarios [30]. The choice of a deployment scenario depends on factors such as network resource constraints and the services the RAN is designed to deliver. In the present study, we selected "Scenario A" where (i) the O-RU is located at the cell site; (ii) the O-DU, O-CU, and near-RT RIC are hosted on an edge server;

and (iii) the non-RT RIC is located in a regional cloud. The centralized control of the O-CU and O-DU enables efficient resource management, such as bandwidth allocation, which is critical for tactical operations requiring coordinated and real-time decision-making. Scenario A also reduces the need for extensive on-site equipment, aligning with the mobility and portability requirements of TNs. Furthermore, it supports scalability by allowing multiple O-RUs to connect to a centralized O-DU, which facilitates expanding the network as tactical situations evolve. This flexibility makes Scenario A ideal for large-scale deployments, such as covering multiple operational areas or supporting multiple tactical teams.

B. RAN SLICING MODEL

The gNodeB's total radio bandwidth, which is denoted by $\mathcal{W}$, is shared among the gNodeB's users. The slicing of $\mathcal{W}$ is performed hierarchically in two stages, namely inter-slice bandwidth sharing and intra-slice bandwidth sharing.

1) INTER-SLICE BANDWIDTH SHARING

In this slicing stage, a resource scheduler assigns a portion $f_s^t \in [0, 1]$ of the total bandwidth $\mathcal{W}$ to each slice $s \in \mathcal{S}$ at time slot t. The bandwidth allocated to slice s at time t is denoted by $\mathcal{W}_s^t$ and calculated as follows:

$$\mathcal{W}_s^t = f_s^t \mathcal{W}. \tag{1}$$

To ensure that the gNodeB's total bandwidth is not exceeded and to prevent it from being exploited by only a small subset of slices, we define the following two constraints:

$$\sum_{s=1}^{|\mathcal{S}|} f_s^t \leq 1 \tag{2}$$

and

$$f^{\min} \leq f_s^t \leq f^{\max}, \forall s \in \mathcal{S}. \tag{3}$$

2) INTRA-SLICE BANDWIDTH SHARING

In this slicing stage, each slice $s \in \mathcal{S}$ allocates to each of its users at time t a portion $f_{u,s}^t$ of the bandwidth $\mathcal{W}_s^t$ it was previously allocated in the inter-slice bandwidth sharing stage. We define the following two constraints for slice s:

$$\sum_{u=1}^{|\mathcal{U}_s|} f_{u,s}^t \leq 1 \tag{4}$$

and

$$f_s^{\min} \leq f_{u,s}^t \leq f_s^{\max}, \forall u \in \mathcal{U}_s, \forall s \in \mathcal{S}. \tag{5}$$

Constraint (4) guarantees that the sum of bandwidth portions allocated to users of slice s does not exceed the slice's total bandwidth $\mathcal{W}_s^t$. Constraint (5) prevents the bandwidth $\mathcal{W}_s^t$ from being used by only a small subset of users.

C. CHANNEL MODEL

We assume that the gNodeB knows the channel state information of its associated users. The downlink channel gain between a user $u \in \mathcal{U}_s, \forall s \in \mathcal{S}$ and the gNodeB at time slot t is denoted by $g_{u,s}^t$ and defined as follows:

$$g_{u,s}^t = 10^{-\frac{PL(d_{u,s})}{10}} |h_{u,s}^t|^2, \tag{6}$$

where $h_{u,s}^t$ is the small-scale fading coefficient, which is modeled as a complex Gaussian distribution. $PL(d_{u,s})$ is the path loss between user u and the gNodeB. The path loss model considered in this work corresponds to the 3GPP specifications [31] and can be calculated as follows:

$$PL(d_{u,s}) = 28 + 22 \log_{10}(d_{u,s}) + 20 \log_{10}(f_c) + \sigma_{SF}, \tag{7}$$

where $d_{u,s}$ is the distance between user u and the gNodeB, f_c is the central frequency of the 5G band, and σ_{SF} is the shadow fading that follows a normal distribution.

The downlink data rate achieved by user u at time t is calculated as follows:

$$r_{u,s}^t = f_{u,s}^t \mathcal{W}_s^t \log_2\left(1 + \frac{P g_{u,s}^t}{f_{u,s}^t \mathcal{W}_s^t N_0}\right), \tag{8}$$

where P is the downlink transmission power of the gNodeB and N_0 is the power of the additive white Gaussian noise (AWGN). We assume that P is the same for all users.

D. DEGREE OF SATISFACTION

A user is satisfied if the allocated bandwidth is sufficient to achieve a data rate that meets its QoS requirements. Since bandwidth is a limited resource, its efficient allocation is crucial to satisfying users' diverse QoS requirements. Inefficient bandwidth allocation can result in data rates below or above the required threshold. A data rate below the required threshold diminishes the QoS provided by the RAN slices, directly affecting users' degree of satisfaction. While exceeding the required data rate does not affect the QoS, it implies a waste of resources that could be allocated more effectively to other users facing bandwidth shortages.

To accurately gauge users' degree of satisfaction, we need to define a utility function that effectively evaluates how well the bandwidth allocated meets user's QoS requirements while capturing any resource wastage during the allocation process. For this reason, we model the degree of satisfaction of user $u \in \mathcal{U}_s, \forall s \in \mathcal{S}$ at time t using (9) [32]:

$$\Gamma_{u,s}^t = \frac{1 - e^{\left(-\frac{\gamma_{u,s}^t}{\rho \frac{r_{u,s}^t}{R_s^{req}}}\right)}}{\varphi}, \tag{9}$$

where

$$\gamma_{u,s}^t = \frac{\left(\rho \frac{r_{u,s}^t}{R_s^{req}}\right)^{\xi}}{1 + \left(\rho \frac{r_{u,s}^t}{R_s^{req}}\right)^{\xi}}, \tag{10}$$

ρ and ξ adjust the elasticity of the function. R_s^{req} denotes the minimum data rate required by each user served by slice $s \in \mathcal{S}$. Note that all users served by the same slice have the same minimum data rate. φ is a normalisation parameter that ensures that $\Gamma^t_{u,s} \in [0, 1]$ and is given by (11):

$$\varphi = 1 - e^{-\frac{1}{(\xi-1)^{\frac{1}{\xi}} + (\xi-1)^{\frac{1-\xi}{\xi}}}}. \tag{11}$$

According to (9), as users' satisfaction increases, the achieved data rate approaches the required data rate, and decreases as it deviates from it, either above or below. This helps to limit wasting resources by allocating users more than they require.

The normalized degree of satisfaction of a slice $s \in \mathcal{S}$ at time t is defined as follows:

$$\Gamma^t_s = \frac{1}{|\mathcal{U}_s|} \sum_{u=1}^{|\mathcal{U}_s|} \Gamma^t_{u,s}. \tag{12}$$

The total normalized degree of satisfaction of the entire system at time t is defined as follows:

$$\Gamma^t = \frac{1}{|\mathcal{S}|} \sum_{s=1}^{|\mathcal{S}|} \Gamma^t_s. \tag{13}$$

E. INTER-SLICE ISOLATION

To deal with the fast-changing and resource-constrained tactical RAN environment, the resources allocated to RAN slices during the inter-slice bandwidth sharing stage need to be dynamically reconfigured. Indeed, some slices may require additional resources to maintain their users' degree of satisfaction at a high level. However, adjusting the bandwidth allocated to the RAN slices can adversely impact inter-slice isolation. Adding resources to or removing resources from a slice can reduce the degree of satisfaction of the slice's associated users, which affects isolation among slices.

We define the reconfiguration cost of a slice $s \in \mathcal{S}$ at time t, which is denoted by $\mathcal{C}^t_s$, as follows:

$$\mathcal{C}^t_s = \begin{cases} \Gamma^{t-1}_s - \Gamma^t_s & \text{if } \mathcal{W}^{t-1}_s \neq \mathcal{W}^t_s \text{ and } \Gamma^t_s < \Gamma^{t-1}_s \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

In (14), the reconfiguration cost of slice s is considered if the slice underwent bandwidth reconfiguration at time t, i.e., $\mathcal{W}^{t-1}_s \neq \mathcal{W}^t_s$, and the slice's degree of satisfaction is lower at time t than it was at time $t-1$, i.e., $\Gamma^t_s < \Gamma^{t-1}_s$. The value of the reconfiguration cost, $\Gamma^{t-1}_s - \Gamma^t_s$, is proportional to the severity of the QoS deterioration. In other words, the more the slice's degree of satisfaction has deteriorated, the higher the reconfiguration cost, and vice versa.

The total normalized reconfiguration cost at time t is defined as follows:

$$\mathcal{C}^t = \frac{1}{|\mathcal{S}|} \sum_{s=1}^{|\mathcal{S}|} \mathcal{C}^t_s. \tag{15}$$

In this study, we consider that isolating a slice refers to ensuring it is allocated the bandwidth it requires to meet its QoS performance. This implies reducing the impact of fluctuations in the resource demands of one slice on the QoS performance of the other slices. In other words, increasing or decreasing the portion of bandwidth allocated to a slice should not impact the other slices' degree of satisfaction. To this end, we define slice resource reconfiguration constraints to efficiently reconfigure the bandwidth allocated to the slicesand, in turn, ensure inter-slice isolation.

We consider slice s to require additional resources at time t if one the following constraints is satisfied:

Constraint 1:

- almost all its allocated bandwidth $\mathcal{W}^{t-1}_s$ was used at time $t-1$, i.e., $(1 - \sum_{u=1}^{|\mathcal{U}_s|} f^{t-1}_{u,s}) \leq f^{\min}$, and
- there is at least one unsatisfied user at time $t-1$, i.e., $\mathcal{U}^{t-1}_{s,uns} \neq \emptyset$, where $\mathcal{U}^{t-1}_{s,uns} = \{u \in \mathcal{U}_s, r^{t-1}_{u,s} \leq R^{req}_s\}$.

Constraint 2:

- its allocated bandwidth $\mathcal{W}^{t-1}_s$ was not fully used at time $t-1$, i.e., $(1 - \sum_{u=1}^{|\mathcal{U}_s|} f^{t-1}_{u,s}) \geq f^{\min}$,
- $\mathcal{U}^{t-1}_{s,uns} \neq \emptyset$, and
- its remaining resources are not sufficient to meet the requirements of the unsatisfied users, i.e., $f^{t-1}_{u^*,s}|\mathcal{U}^{t-1}_{s,uns}| \geq (1 - \sum_{u=1}^{|\mathcal{U}_s|} f^{t-1}_{u,s})$, where $f^{t-1}_{u^*,s}$ is the portion of bandwidth allocated to the user u^* with the lowest channel gain. We consider the amount of resources allocated to the user with the lowest channel gain to account for the worst-case scenario.

Constraint 1 indicates that although slice s used almost all of its allocated bandwidth $\mathcal{W}^{t-1}_s$ at time $t-1$, it did not satisfy all its users. *Constraint 2*states that even if slice s partitioned its bandwidth poorly among its users, its unused resources are not sufficient to accommodate all its users.

If either *Constraint 1*or *Constraint 2*is satisfied for a slice $s \in \mathcal{S}$, we consider that this slice requires additional bandwidth at time t and define this resource demand using the following binary parameter (see (16)):

$$\vartheta^t_{s,1} = \begin{cases} 1 & \text{if } s \text{ requires additional resources} \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

We model whether a slice $s \in \mathcal{S}$ has experienced an increase in its allocated bandwidth portion f^t_s at time t using the following binary parameter (see (17)):

$$\varkappa^t_{s,1} = \begin{cases} 1 & \text{if } f^t_s \geq f^{t-1}_s \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

The relationship between $\vartheta^t_{s,1}$ and $\varkappa^t_{s,1}$ can be expressed as a logical implication, i.e., $\vartheta^t_{s,1} \Rightarrow \varkappa^t_{s,1} \equiv \bar{\vartheta}^t_{s,1} \vee \varkappa^t_{s,1}$.

We consider slice s to have available resources at time t if one of the following constraints is satisfied:

Constraint 3:

- all users associated with slice s are satisfied, i.e., $\mathcal{U}^{t-1}_{s,uns} = \emptyset$, and
- $\Gamma^{t-1}_s < \Gamma_{th}$, where Γ_{th} is a slice satisfaction threshold.

Constraint 4:

- $\mathcal{U}^{t-1}_{s,uns} = \emptyset$,
- $\Gamma^{t-1}_s \geq \Gamma_{th}$, and
- there are available resources $(1 - \sum_{u=1}^{|\mathcal{U}_s|} f^{t-1}_{u,s}) \geq f^{\min}_s$.

Constraint 3 indicates that all users associated with slice s meet the minimum required data rate, yet exhibit a low average degree of satisfaction, implying inefficient intra-slice bandwidth allocation of $\mathcal{W}_s^{t-1}$ at time $t-1$. In other words, some users associated with slice s are allocated more bandwidth than they require. *Constraint 4*ensures that there are unused resources in the slice.

If either *Constraint 3*or *Constraint 4*is satisfied for a slice $s \in \mathcal{S}$, we consider that this slice to have unused bandwidth at time t and define this resource excess using the following binary parameter (see (18)).

$$\vartheta_{s,2}^t = \begin{cases} 1 & \text{if } s \text{ has available resources} \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

We model whether a slice $s \in \mathcal{S}$ has experienced a decrease in its allocated bandwidth portion f_s^t at time t using the binary parameter shown in (19):

$$\varkappa_{s,2}^t = \begin{cases} 1 & \text{if } f_s^t \leq f_s^{t-1} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

The relationship between $\vartheta_{s,2}^t$ and $\varkappa_{s,2}^t$ can be defined as a logical implication, i.e., $\vartheta_{s,2}^t \Rightarrow \varkappa_{s,2}^t \equiv \bar{\vartheta}_{s,2}^t \vee \varkappa_{s,2}^t$.

IV. PROBLEM FORMULATION

In order to satisfy the QoS of users, we need to achieve a trade-off between efficient RAN bandwidth sharing among slices and appropriate inter-slice isolation. Therefore, we should maximize each slice's degree of satisfaction while minimizing its reconfiguration cost. We transform the two objectives into a weighted sum by introducing a weight factor $\alpha \in [0, 1]$. The global bandwidth allocation optimization problem is formulated as follows:

$$\underset{f_s^t, f_{u,s}^t}{\text{maximize}} \quad \alpha\, \Gamma^t - (1-\alpha)\, \mathcal{C}^t \tag{20a}$$

$$\text{subject to} \quad \sum_{s=1}^{|\mathcal{S}|} f_s^t \leq 1 \tag{20b}$$

$$f^{\min} \leq f_s^t \leq f^{\max}, \forall s \in \mathcal{S} \tag{20c}$$

$$\sum_{u=1}^{|\mathcal{U}_s|} f_{u,s}^t \leq 1, \forall s \in \mathcal{S} \tag{20d}$$

$$f_s^{\min} \leq f_{u,s}^t \leq f_s^{\max}, \forall u \in \mathcal{U}_s, \forall s \in \mathcal{S} \tag{20e}$$

$$r_{u,s}^t \geq R_s^{req}, \forall u \in \mathcal{U}_s, \forall s \in \mathcal{S} \tag{20f}$$

$$\bar{\vartheta}_{s,1}^t \vee \varkappa_{s,1}^t = 1, \forall s \in \mathcal{S} \tag{20g}$$

$$\bar{\vartheta}_{s,2}^t \vee \varkappa_{s,2}^t = 1, \forall s \in \mathcal{S} \tag{20h}$$

$$\varkappa_{s,1}^t \oplus \varkappa_{s,2}^t = 1, \forall s \in \mathcal{S} \tag{20i}$$

$$f_s^t, f_{u,s}^t \in [0, 1], \forall u \in \mathcal{U}_s, \forall s \in \mathcal{S} \tag{20j}$$

$$\varkappa_{s,1}^t, \varkappa_{s,2}^t, \vartheta_{s,1}^t, \vartheta_{s,2}^t \in \{0, 1\}, \forall s \in \mathcal{S}. \tag{20k}$$

Constraint (20b) ensures that the sum of the bandwidth portions allocated to the slices does not exceed the total bandwidth of the system. Constraint (20d) ensures that each slice does not receive more bandwidth that it has been allocated. Constraints (20c) and (20e) set minimum and maximum thresholds for the bandwidth portions allocated to slices and users, respectively. Constraint (20f) ensures that the data rate achieved by each user is above the minimum threshold. Constraints (20g) and (20h) ensure that the logical implication relationships between $\varkappa_{s,1}^t$ and $\vartheta_{s,1}^t$ and between $\varkappa_{s,2}^t$ and $\vartheta_{s,2}^t$ are maintained, respectively. Constraint (20h) implicitly includes Γ_{th}, making the solution space constrained by the degree of slice satisfaction. Constraint (20i) prevents the unfeasible slice resource allocation scenario in which a slice requires resources and at the same time has available resources. Constraint (20j) presents the optimization variables. Constraints (20c), (20g), and (20h) preserve inter-slice isolation. Constraint (20c) ensures each slice receives a minimum amount of resources and prevents resources from being monopolized by a subset of slices. Constraint (20g) ensures that added resources are allocated to slices that need them, while Constraint (20h) ensures that removed resources are taken from slices with unused resources." As a result, unnecessary slice resource reconfiguration is avoided, which in turn promotes inter-slice isolation. Constraints (20e) and (20f) preserve intra-slice isolation during dynamic resource sharing among users within a slice. Isolation is guaranteed in terms of resources by (20e), and performance by (20f).

Due to the nonlinear and time-coupled structure of the formulated problem, as well as the logical reconfiguration constraints, it is challenging to derive a tractable solution using conventional optimization methods, particularly in a dynamic RAN environment. Therefore, we adopt a DRL-based framework to learn an effective bandwidth allocation policy that approximates the solution of the formulated optimization problem, while adapting to time-varying system conditions.

V. BANDWIDTH SHARING INTER-SLICE AND INTRA-SLICE ISOLATION MECHANISM

We leveraged DRL techniques to address the RAN slicing problem at each bandwidth allocation stage. DRL algorithms can be classified into model-based and model-free approaches. Model-based algorithms learn an explicit environment model, including the state transitions, to guide action selection. In contrast, model-free algorithms learn optimal state–action associations directly from experience without explicit environment knowledge.

The choice between model-based and model-free methods depends on the problem being investigated. Model-based methods are suited for fixed environments, (e.g., factory robots or chess games), while model-free methods are preferable for unknown or complex dynamics. In this work, we adopt model-free algorithms due to: (i) the highly dynamic and continuously varying RAN environment and (ii) the lack of a reliable environment model during RAN slicing. Each bandwidth allocation problem at each RAN slicing stage is formulated as a Markov decision process (MDP).

A. MDP FORMULATION OF INTER-SLICE BANDWIDTH ALLOCATION

In this slicing stage, we consider that a global agent is responsible for allocating portions of the entire system's bandwidth to the slices.

1) THE STATE SPACE

To choose an action at time t, the global agent observes the system performance resulting from its decision at time $t-1$. The observation at time t is defined as follows:

$$\mathcal{O}_g^t = \langle \boldsymbol{\Upsilon}^{t-1}, \boldsymbol{\mathcal{V}}_1^{t-1}, \boldsymbol{\mathcal{V}}_2^{t-1}, \boldsymbol{\mathcal{F}}^{t-1} \rangle, \tag{21}$$

where $\boldsymbol{\Upsilon}^{t-1} = (\Gamma_s^{t-1} : s \in \mathcal{S})$ represents the slices' degree of satisfaction at time $t-1$. $\boldsymbol{\mathcal{V}}_1^{t-1} = (\vartheta_{s,1}^{t-1} : s \in \mathcal{S})$ and $\boldsymbol{\mathcal{V}}_2^{t-1} = (\vartheta_{s,2}^{t-1} : s \in \mathcal{S})$ denote the resource reconfiguration constraints at time $t-1$ related to adding resources to and removing resources from the slices, respectively. $\boldsymbol{\mathcal{F}}^{t-1} = (f_s^{t-1} : s \in \mathcal{S})$ represents the portions of bandwidth $\mathcal{W}$ that were allocated to the slices at time $t-1$.

2) THE ACTION SPACE

To perform slice bandwidth allocation, the global agent must decide what portion of the entire system's bandwidth $\mathcal{W}$ to allocate to each slice. The global agent's action space is defined as follows:

$$\mathcal{A}_g^t = [f^{\min}, f^{\max}]^{|\mathcal{S}|}. \tag{22}$$

An action $a_g^t \in \mathcal{A}_g^t$ is a row vector, where $a_g^t = [f_1^t, f_2^t, \ldots, f_{|\mathcal{S}|}^t]$.

3) THE REWARD FUNCTION

The main objective of inter-slice bandwidth allocation is to maximize the total degree of satisfaction of the entire system, but not at the expense of inter-slice isolation. Accordingly, the reward the global agent receives at time t when it chooses an action $a_g^t \in \mathcal{A}_g^t$ is related to: (i) upholding the system's total bandwidth constraint defined in (20b) as well as the minimum and maximum slice bandwidth portion thresholds defined in (20c), (ii) satisfying the slice reconfiguration constraints defined in (20g) and (20h), and (iii) Γ^t, and $\mathcal{C}^t$. When the global agent selects an action that violates at least one of constraints (20b), (20c), (20g) and (20h), the action is considered as an invalid action. The reward the global agent receives at time t is defined as follows:

$$\mathcal{R}_g^t = \begin{cases} \alpha\,\Gamma^t - (1-\alpha)\,\mathcal{C}^t & \text{if } a_g^t \text{ is valid} \\ -1 & \text{if } a_g^t \text{ is invalid.} \end{cases} \tag{23}$$

When the global agent chooses a valid action, it receives the value of the objective function defined in (20a). On the other hand, if it chooses an invalid action, it is penalized with a negative reward to discourage it from choosing the same action again in the future.

B. MDP FORMULATION OF INTRA-SLICE BANDWIDTH ALLOCATION

In this RAN slicing stage, we consider that each slice is assigned an agent that allocates portions of the slice's bandwidth to the slice's associated users.

1) THE STATE SPACE

The state of the environment that is observed by slice s's agent at time t is defined as follows:

$$\mathcal{O}_s^t = \langle \boldsymbol{\mathcal{G}}_s^t \rangle, \tag{24}$$

where $\boldsymbol{\mathcal{G}}_s^t = (g_{u,s}^t : u \in \mathcal{U}_s)$ represents the channel gain between the gNodeB and the users associated with slice s at time t. Agent s's observation at time t is given by the row vector $[g_{1,s}^t, g_{2,s}^t, \ldots, g_{|\mathcal{U}_s|,s}^t]$. At each time t, the positions of the users associated with slice s vary. This leads to variation in the users' channel gain values in accordance with (6) and (7). Hence, agent s's observation varies at each time t depending on the dynamics of the wireless channel.

2) THE ACTION SPACE

To perform intra-slice bandwidth allocation, slice s's agent must observe its environment and decide what portion of slice s's allocated bandwidth $\mathcal{W}_s^t$ to assign to its users. Slice s agent's action space at time t is defined as follows:

$$\mathcal{A}_s^t = [f_s^{\min}, f_s^{\max}]^{|\mathcal{U}_s|}. \tag{25}$$

An action $a_s^t \in \mathcal{A}_s^t$ is a row vector, where $a_s^t = [f_{1,s}^t, f_{1,s}^t, \ldots, f_{|\mathcal{U}_s|,s}^t]$.

3) THE REWARD FUNCTION

The reward function represents the agent's objective of allocating sufficient resources to meet users' QoS requirements. The reward value depends on the degree of satisfaction of the associated users. Intra-slice bandwidth allocation is deemed successful if it adheres to constraints (20d) and (20e). Actions violating these constraints are considered invalid. The reward for slice s's agent at time t is defined as:

$$\mathcal{R}_s^t = \begin{cases} \Gamma_s^t & \text{if } a_s^t \text{ is valid} \\ -1 & \text{if } a_s^t \text{ is invalid.} \end{cases} \tag{26}$$

C. ON-POLICY AND OFF-POLICY DRL

Model-free DRL algorithms are categorized as on-policy or off-policy. During training, the agent uses a behavior policy π_B to select actions based on observed environmental states and collect data. It then employs a target policy π_T to update Q-values by evaluating actions based on rewards. In on-policy algorithms, the behavior and target policies are the same, whereas in off-policy algorithms, they differ.

There are advantages and drawbacks to each class of algorithms. For instance, off-policy algorithms are more likely to find a global optimum solution, while on-policy algorithms may get trapped in a local optimum. On the other hand, on-policy algorithms are less computationally complex than off-policy algorithms since π_B and π_T are the same.

On-policy and off-policy algorithms have been widely exploited to solve RAN slicing problems [33], [34]. Both classes of algorithms have proven effective in RAN resource allocation problems. Therefore, determining the most appropriate algorithm from the two classes depends on the use case. In this work, we have chosen algorithms from both classes - the off-policy algorithms TD3 [35] and DDPG [36], and the on-policy algorithm PPO [37], to evaluate how they perform with the proposed RAN slicing mechanism. We designed three implementations of the proposed BS-IISI mechanism, each of which is based on a different DRL algorithm. For instance, in the TD3-based BS-IISI implementation, TD3 is used in both the inter-slice bandwidth sharing stage and the intra-slice bandwidth sharing stage.

DDPG and TD3 are actor-critic algorithms with deterministic policies. In DDPG, the actor chooses actions based on the current policy, while the critic estimates Q-values to evaluate these actions. TD3 follows the same strategy, but uses two critic networks to compute the minimum predicted value for target updates. Both DDPG and TD3 use experience replay and target networks to enhance stability. By contrast, PPO is an actor-critic algorithm with stochastic policies, where the actor outputs a probability distribution over actions and the critic evaluates the expected return of the chosen policy. PPO optimizes a surrogate objective with a clipping mechanism to constrain policy updates, ensuring stable and reliable learning.

D. BS-IISI DEPLOYEMENT IN AN O-RAN ARCHITECTURE

The pseudo-code of the proposed BS-IISI mechanism is presented in Algorithm 1. To ensure temporal consistency between control decisions and wireless channel evolution, the channel model follows a slot-based block fading assumption. Specifically, the channel gains used in agents' observations are updated at the beginning of each slot according to the updated user positions and fading realizations, and remain constant during the slot. In practical deployments, the channel coherence time in mobile wireless environments is typically on the order of milliseconds, while the DRL inference interval associated with near-RT RIC control loops operates on a slightly larger timescale, typically tens of milliseconds (e.g., ≥ 20 ms). This allows the slicing policy to be updated frequently enough to capture mobility-driven channel variations while maintaining a feasible processing load for the near-RT RIC. Inter-slice bandwidth allocation precedes intra-slice bandwidth allocation, and both processes are performed at the beginning of each time slot t. Prior to theses processes, we verify whether any RAN slices need additional resources. This can be done by observing the parameter $\vartheta^{t-1}_{s,1}\ \forall s \in \mathcal{S}$ in line 1. If $\vee_{s\in\mathcal{S}}\vartheta^{t-1}_{s,1} = 1$, the global agent utilizes its trained DRL model to perform RAN slicing by allocating a portion of the system's bandwidth to each RAN slice, (see line 2). When none of the RAN slices requires additional resources, each slice maintains the bandwidth portion it obtained at time $t-1$, such that $f^{t-1}_s = f^t_s$, (see line 6). Verifying condition

Algorithm 1: BS-IISI Mechanism.

Input: $\mathcal{W}, \mathcal{S}, \mathcal{U}$
Output: $f^t_s, f^t_{u,s}, \forall u \in \mathcal{U}_s, \forall s \in \mathcal{S}$
foreach *time* t **do**
1 **if** $\vee_{s\in\mathcal{S}}\vartheta^{t-1}_{s,1} = 1$ **then**
2 The global agent uses its trained DRL model to choose $f^t_s, \forall s \in \mathcal{S}$;
3 Each agent $s \in \mathcal{S}$:
4 uses its trained model to select $f^t_{u,s}, \forall u \in \mathcal{U}_s$;
5 publishes $\Gamma^t_s, \vartheta^t_{s,1}, \vartheta^t_{s,2}$ in the shared database;
else
6 $f^{t-1}_s = f^t_s, \forall s \in \mathcal{S}$;
7 Each agent $s \in \mathcal{S}$:
8 uses its trained model to select $f^t_{u,s}, \forall u \in \mathcal{U}_s$;
9 publishes $\Gamma^t_s, \vartheta^t_{s,1}, \vartheta^t_{s,2}$ in the shared database;

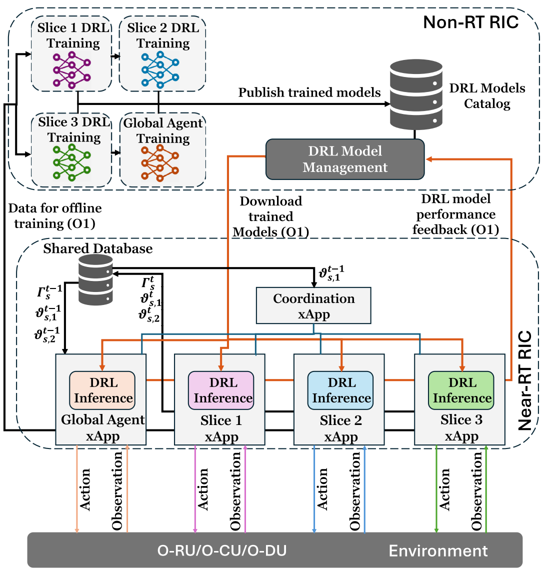


FIGURE 2. BS-IISI mechanism in a O-RAN framework.

$\vee_{s\in\mathcal{S}}\vartheta^{t-1}_{s,1} = 1$ makes it possible to efficiently perform inter-slice bandwidth allocation, since it is triggered only when there is a lack of resources in slices. This reduces the cost of reconfiguring slice resources, which in turn enhances inter-slice isolation.

Once the system's bandwidth has been allocated to the slices, it's time for intra-slice bandwidth allocation. Each slice s's agent uses its DRL inference model to allocate to each of the slice's associated users a portion $f^t_{u,s}$ of the bandwidth $\mathcal{W}^t_s$ it was previously allocated by the global agent, (see lines 4 and 8). Then, in lines 5 and 9, each slice agent publishes Γ^t_s, $\vartheta^t_{s,1}$, $\vartheta^t_{s,2}$ in the shared database.

Fig. 2 illustrates the use of the BS-IISI mechanism in a high-level O-RAN architecture. Following the O-RAN

Alliance WG2 deployment scenarios for RL-based mechanisms [38], we adopt "Scenario 1.4" with offline learning, which is well aligned with the stringent requirements of TNs. In this scenario, training is performed in the non-RT RIC, while inference is executed in the near-RT RIC. This setup addresses key constraints of tactical environments, including latency and limited computing and storage resources. Since both inter-slice and intra-slice bandwidth allocation operate on short time scales, inference must run in the near-RT RIC, whereas the non-RT RIC uses its abundant resources to handle training without burdening field-deployed infrastructure. In the O-RAN architecture, xApps running in the near-RT RIC support closed-loop control operations on time scales typically ranging from 10 ms to 1 s, which defines the practical latency budget for inference and decision-making processes [39]. Recent studies on O-RAN intelligence orchestration and AI-driven RAN control demonstrated that machine-learning-based xApps can retrieve network measurements and compute control actions within these time constraints, thereby enabling latency-sensitive network optimization tasks [40], [41]. Overall, this architecture enables efficient resource use and low-latency operation, meeting the dynamic and mission-critical demands of TNs.

The inference models of the global agent and each RAN slice agent are each deployed in an xApp. Each agent receives observation data and executes bandwidth slicing actions through the E2 interface. Following intra-slice bandwidth allocation, each slice s's agent publishes its satisfaction degree Γ_s^t, and parameters $\vartheta_{s,1}^t$ and $\vartheta_{s,2}^t$ to a shared database accessible by all xApps. The global agent uses information shared by slice agents for inter-slice bandwidth allocation. A coordinating xApp ensures that the condition $\vee_{s\in\mathcal{S}}\vartheta_{s,1}^{t-1}=1$ is satisfied before each allocation. At the beginning of each time slot t, the coordinating xApp: (i) reads the $\vartheta_{s,1}^{t-1}$ values from the shared database, (ii) checks $\vee_{s\in\mathcal{S}}\vartheta_{s,1}^{t-1}=1$, and (iii) decides whether the global agent proceeds with inter-slice bandwidth allocation. The Datasets collected during inference are sent to DRL training hosts for retraining through the O1 interface. Once a model is trained, tested, and validated, it is published in the model catalog. During inference, slice agents permanently report model performance to the model management entity, which decides whether an inference model needs to be updated. When an inference model suffers from a serious degradation in performance, the model management entity: (i) terminates the model, (ii) selects a new trained model from the catalog, and (iii) deploys it to the near-RT RIC.

E. COMPUTATIONAL COMPLEXITY OF THE BS-IISI MECHANISM

At each time slot t, the online execution cost of the proposed BS-IISI mechanism is dominated by the following three components: (i) slice-level monitoring and trigger evaluation; (ii) neural-network forward inference for inter-slice and intra-slice decisions; and (iii) computation of the performance indicators published for closed-loop control. The coordinating logic first evaluates the trigger condition $\vee_{s\in\mathcal{S}}\vartheta_{s,1}^{t-1}=1$ (see line 1), which requires scanning the $|\mathcal{S}|=S$ slices and, therefore, incurs a computational cost of $\mathcal{O}(S)$. When the trigger condition is satisfied, the inter-slice policy performs one forward pass to produce the S-dimensional bandwidth allocation vector $\{f_s^t\}$ (see line 2). The computational cost of this inference step scales as $\mathcal{O}(\mathcal{P}_g)$, where $\mathcal{P}_g$ denote the number of parameters of the global inter-slice neural network and reflects the number of multiply–accumulate operations required by the network. The value of $\mathcal{P}_g$ depends on the architecture of the neural network, including the number of layers and the size of each layer. Independently of the trigger condition, intra-slice inference is executed for every slice $s\in\mathcal{S}$, where each slice-level agent performs one forward pass to determine the user-level allocation vector $\{f_{u,s}^t\}_{u\in\mathcal{U}_s}$. Let $\mathcal{P}_s$ denote the number of parameters of the neural network associated with slice s. Therefore, the total computational cost of intra-slice inference per time slot scales as $\mathcal{O}(\sum_{s\in\mathcal{S}}\mathcal{P}_s)$. In addition, the system computes and publishes slice performance indicators, including per-user rate and degree-of-satisfaction metrics and their aggregated slice-level statistics used to generate Γ_s^t, $\vartheta_{s,1}^t$, $\vartheta_{s,2}^t$. Assuming that the underlying rate and utility expressions can be evaluated in constant time per user, this step scales linearly with the number of users and incurs a cost of $\mathcal{O}(U)$. Accordingly, the worst-case computational complexity per time slot can thus be expressed as follows:

$$T_{slot}=\mathcal{O}(S+U+\mathcal{P}_g+\sum_{s\in\mathcal{S}}\mathcal{P}_s). \tag{27}$$

Since the inter-slice inference is executed only when the trigger condition is satisfied, the expected per-slot complexity becomes:

$$\mathcal{O}(S+U+p\mathcal{P}_g+\sum_{s\in\mathcal{S}}\mathcal{P}_s), \tag{28}$$

where $p=Pr(\vee_{s\in\mathcal{S}}\vartheta_{s,1}^{t-1}=1)$ denotes the probability that the inter-slice policy is activated.

In addition to the above theoretical analysis, we evaluated the inference execution time of the proposed DRL models to assess their practical feasibility. The corresponding results are presented in Section VI.

VI. SIMULATION RESULTS

A. SIMULATION SETTINGS

The proposed BS-IISI mechanism is designed to meet the QoS requirements of any tactical RAN slice. Due to the lack of standardized QoS specifications for TN applications, we adopted a classification aligned with 5G service categories. Accordingly, we consider a RAN configuration comprising $S=3$ slices corresponding to eMBB, URLLC and mMTC services, deployed within a single gNodeB. The gNodeB covers an area of (500 m × 500 m), where users move according to the random waypoint (RWP) model. Each slice serves its own set of users, with maximum numbers of 20, 70, and 210 users for the eMBB, URLLC, and mMTC slices, respectively.

TABLE 2. RAN Parameters

Parameter	Value
$\mathcal{W}, P, N_0, f_c$	20 MHz, 30 dBm, -174 dBm/Hz, 3 GHz
σ_{SF}, $h_{u,s}^t$	$\mathcal{N}(0, 4^2)$, $\mathcal{CN}(0,1)$
$[v_{min}, v_{max}], \tau_{max}$	[1 m/s,4 m/s], 300 seconds
R_s^{req}	eMBB: 10 Mbps; URLLC: 250 kbps; mMTC: 12 kbps
$[f_s^{min}, f_s^{max}]$	eMBB: [0.005,0.5]; URLLC: [0.0014,0.14]; mMTC: [0.00047,0.047].
$[f^{min}, f^{max}]$	[0.01, 0.95]
α, ρ, ζ, Γ_{th}	0.5, 1.3, 5, 0.8

TABLE 3. Training Parameters

Hyperparameter	TD3	DDPG	PPO
No. of hidden layers	2	2	2
No. of hidden units/layer	Global agent: 300/200 Slice agent: 500/400	Global agent: 300/200 Slice agent: 500/400	Global agent: 300/200 Slice agent: 500/400
Actor learning rate	0.0001	0.0001	0.0001
Critic learning rate	0.001	0.001	0.001
Replay buffer size	Global agent: 10^5 Slice agent: 10^6	Global agent: 10^5 Slice agent: 10^6	None
Batch size	128	128	500
Optimizer	Adam	Adam	Adam
Activation function	ReLU	ReLU	ReLU
Exploration noise distribution	Global agent: $\mathcal{N}(0, 0.2)$ eMBB agent: $\mathcal{N}(0, 0.1)$ URLLC agent: $\mathcal{N}(0, 0.1)$ mMTC agent: $\mathcal{N}(0, 0.1)$	Global agent: $OU(0, 0.2)$ eMBB agent: $OU(0, 0.5)$ URLLC agent: $OU(0, 0.1)$ mMTC agent: $OU(0, 0.1)$	None
No. of epochs	None	None	10
Target net. update rate / freq.	0.001 / 10	0.001 / 10	None
Discount factor	0.99	0.99	0.99
No. of steps/episode	50	50	50
Clipping parameter	None	None	0.2

The remaining RAN simulation parameters are summarized in Table 2. We implement the proposed BS-IISI mechanism using DRL algorithms and develop three implementations based on TD3, DDPG, and PPO. For instance, in the TD3-based implementation, TD3 is used in both the inter-slice and intra-slice bandwidth sharing stages. All algorithms were implemented in Python using the Pytorch framework. For each algorithm, performance is evaluated over 1000 independent random realizations using the corresponding inference models, and the results are averaged. Table 3 summarizes the hyperparameter settings for each DRL algorithm and bandwidth slicing stage.

We compared the proposed BS-IISI mechanism with the following four baselines: TD3 without isolation constraints (TD3-WIC), DDPG-WIC, PPO-WIC, and the resource sharing and slice isolation based on an iterative process (RSSI-IP) [21].

In the TD3-WIC, DDPG-WIC, and PPO-WIC implementations, the BS-IISI mechanism operates without inter- and intra-slice isolation constraints defined in (20c), (20e), (20g), and (20h). The RSSI-IP mechanism addresses the RAN slicing problem by formulating it as a convex optimization problem to maximize user data rates. This mechanism uses an iterative process to achieve flexible performance isolation among slices while ensuring fair resource allocation. Slice performance isolation is maintained as long as the number of users does not exceed a contracted threshold between the slice tenant and the infrastructure provider. When some slices do not fully use their contracted resources, the excess resources are allocated to other slices whose demands exceed their thresholds.

TABLE 4. Inference Execution Time of the BS-IISI Mechanism

Algorithm	Inference time (ms)
TD3	3.29
DDPG	4.60
PPO	7.86

We opted for RSSI-IP as a benchmark since its objectives closely align with those defined in this work, notably in achieving efficient dynamic RAN resource sharing and robust slice isolation. In addition, in [21], slice isolation is defined in accordance with our perspective, focusing on QoS performance, where isolating a slice refers to reducing the impact that fluctuations in the resource requirements of one slice have on the QoS performance of other slices. Furthermore, the RSSI-IP mechanism operates under assumptions that match our scenario, namely: i) a single cell RAN environment is considered; ii) each user is served by only one RAN slice; iii) each RAN slice supports a single service type with identical QoS requirements for its users; and iv) the data rate achieved by the slice's users should meet a minimum data rate threshold.

To evaluate the practical feasibility of the proposed BS-IISI mechanism, we measured the inference execution time of the DRL models used for inter-slice and intra-slice decisions. The experiments were conducted on a workstation equipped with an Intel Core i9-9920X CPU, using CPU-based inference. The results in Table 4 show that the inference latency of the considered DRL implementations (TD3, DDPG, and PPO) is on the order of a few milliseconds per decision step. This confirms that the proposed mechanism can operate within the near-real-time RIC control loop timescale (10 ms–1 s), making it suitable for latency-sensitive RAN slicing applications.

B. TRAINING PERFORMANCE

To evaluate the BS-IISI mechanism, we analyze the training performance of the DRL algorithms considered in this study and compare it with the selected baselines. Specifically, we examine, in Figs. 3 and 4(a)–(c), the reward behavior of each algorithm at each RAN slicing stage, i.e., inter-slice bandwidth sharing and intra-slice bandwidth sharing. This analysis provides insights into the adaptability and optimization capabilities of the DRL algorithms in response to dynamic environmental conditions. The training performance results of the two RAN slicing stages mutually impact each other since

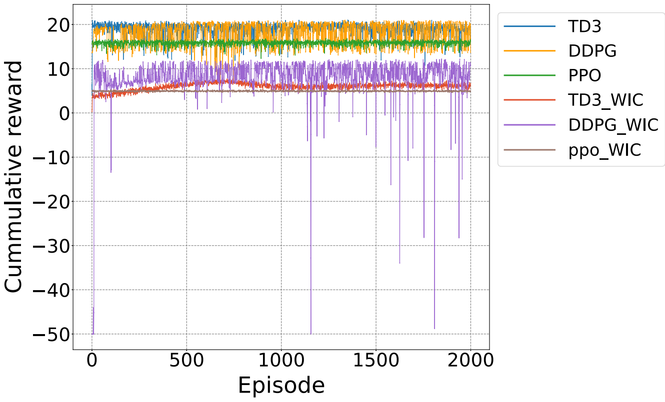


FIGURE 3. **Inter-slice training performance.**

the learning process of one is dependent on the bandwidth slicing results provided by the other.

Importantly, the plotted metric in Figs. 3 and 4(a)–(c) corresponds to the cumulative reward per episode, obtained by summing the rewards over multiple interaction steps. This aggregation produces a smoothed episode-level curve. Consequently, in some curves, relatively high reward values observed in early episodes do not indicate immediate optimal behavior, but reflect instead feasible initial allocations under bounded action spaces. As training progresses, improvements are mainly reflected in more consistent allocation decisions and a reduced number of constraint violations.

Fig. 3 illustrates the learning performance of the DRL algorithms during the inter-slice bandwidth sharing stage over 2000 training episodes. TD3 and DDPG achieve similar cumulative rewards (around 20), although DDPG exhibits higher learning variance. PPO also achieves a relatively stable performance, while the reward is slightly lower than that of TD3 and DDPG, which can be attributed to its reliance on recent data and limited use of past experience compared to off-policy methods. TD3-WIC, DDPG-WIC, and PPO-WIC present a more volatile behavior, with DDPG-WIC showing pronounced reward fluctuations and occasional negative spikes.The absence of isolation constraints leads to broader action exploration, resulting in unstable rewards and reduced bandwidth allocation efficiency.

Figs. 4(a), (b), and (c) depict the reward values of the six DRL algorithms for the eMBB slice agent, the URLLC slice agent, and the mMTC slice agent, respectively. The TD3 algorithm performs well across all slices, consistently achieving high cumulative rewards with less fluctuation. The DDPG algorithm shows a lower performance as compared to TD3, especially for the eMBB slice where the user requirements in terms of data rate are very high. In addition, DDPG exhibits high instability, with large fluctuations during the early training episodes before stabilizing, especially for the mMTC slice. This outcome is due to the high variance of DDPG algorithm policy updates. However, the TD3 algorithm uses a more stable actor-critic approach with delayed policy updates and target networks. PPO demonstrates a relatively high reward during early training episodes, which is however followed by a decline to stabilize at a lower cumulative reward than TD3 and DDPG. This can occur because the PPO algorithm samples actions from a stochastic policy, which frequently leads to violations of intra-slice isolation constraints. TD3-WIC, DDPG-WIC, and PPO-WIC underperform their constrained counterparts across all slices, exhibiting slower early training and rapid convergence to low cumulative rewards, particularly for PPO-WIC. This indicates that, without isolation constraints, learning efficiency is significantly degraded.

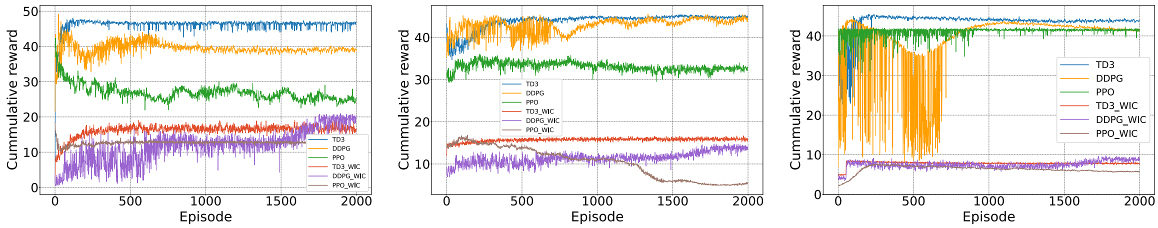
C. INTER-SLICE BANDWIDTH SHARING PERFORMANCE

To evaluate inter-slice bandwidth allocation, we compare the performance of all algorithms using the objective function and the reconfiguration cost defined in (20a) and (15), respectively. For the RSSI-IP mechanism, the objective function value excludes the reconfiguration cost, as it is not considered in [21]. Each algorithm is evaluated under various network configurations, particularly varying the number of users. The DRL algorithm's inference model was trained on a fixed number of users per slice: 20 for eMBB, 70 for URLLC, and 210 for mMTC. This enables an evaluation of the algorithms' robustness in maintaining effective performance despite changes in environmental conditions.

Fig. 5 illustrates the behavior of the objective function under varying system configurations, represented by different numbers of users. Among the evaluated algorithms, TD3 algorithm consistently achieves the highest objective function values across all configurations, confirming its effectiveness in optimizing the objective function. Furthermore, its stable performance as the number of users changes demonstrates its robustness to scaling. DDPG performs slightly below TD3, but still remains consistent across all configurations. In its turn, PPO shows moderate performance, with some fluctuation in its behavior when the number of users varies, indicating less stability than to TD3 and DDPG.

RSSI-IP outperforms TD3-WIC, DDPG-WIC, and PPO-WIC under most network configurations, but remains inferior to TD3, DDPG, and PPO. RSSI-IP prioritizes generous resource allocation to satisfy minimum data-rate requirements, disregarding potential resource wastage. Consequently, its performance improves with the number of users: with fewer users, abundant resources lead to over-allocation and lower satisfaction due to penalization for resource wastage in the utility function defined in (9). This explains why the system's total degree of satisfaction is significantly lower for 108 users than for 300 users. The consistently poor performance of TD3-WIC, DDPG-WIC, and PPO-WIC further highlights the critical role of isolation constraints.

Fig. 6 compares the reconfiguration cost defined in (15) for TD3, DDPG, PPO, TD3-WIC, DDPG-WIC, and PPO-WIC across different numbers of users. The results reveal that the isolation constraints considered in TD3, DDPG, and PPO lead to reduced reconfiguration costs as compared to



(a) eMBB agent (b) URLLC agent (c) mMTC agent

FIGURE 4. **Intra-slice training performance.**

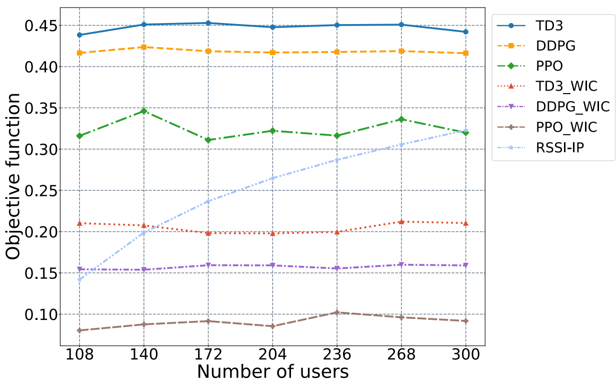


FIGURE 5. **Objective function value (20a) vs. the total number of users for the proposed BS-IISI implementations and baseline schemes.**

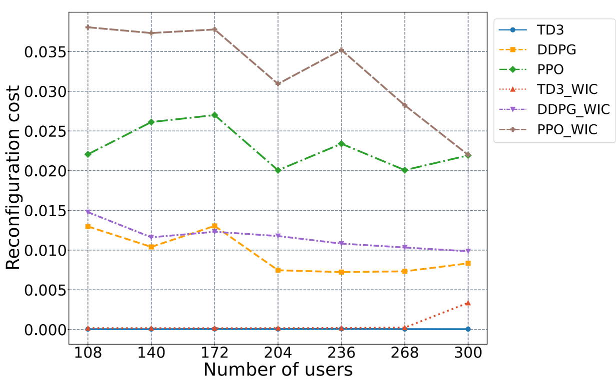


FIGURE 6. **Total reconfiguration cost (15) vs. the total number of users for the proposed BS-IISI implementations and baseline schemes.**

their respective unconstrained algorithms. This demonstrates the importance of isolation constraints in enhancing inter-slice isolation. TD3, DDPG, and PPO exhibit relatively stable performance regardless of the number of users, demonstrating robustness to different network scales. To evaluate the impact of the mobility model on the performance of the proposed BS-IISI mechanism, we conducted an additional sensitivity analysis using the Reference Point Group Mobility (RPGM) model. The RPGM model is parameterized by the number of groups, the group sizes, the velocity range, and an aggregation factor controlling the spatial dispersion of users. In our simulations, the number of groups is scaled with the total number of users to maintain an average group size of approximately 30 users (e.g., 10 groups for 300 users, 7 groups

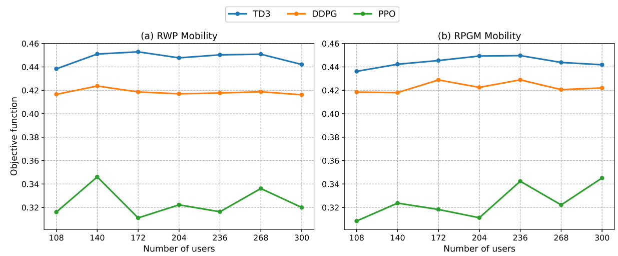


FIGURE 7. **Impact of the mobility model on performance: comparison between RWP and RPGM in terms of the objective function defined in (20a).**

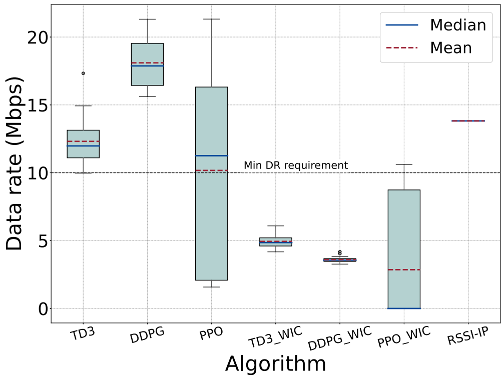
for 204 users, and 4 groups for 108 users). The user velocity is uniformly distributed in the range [1, 4] m/s, identical to the RWP setting, ensuring a fair comparison between mobility models. The aggregation factor is set to 0.5, corresponding to moderately compact user groups.

Fig. 7 compares the performance of the considered DRL algorithms under both RWP and RPGM models in terms of the objective function defined in (20a). The results show that the relative performance ranking of TD3, DDPG, and PPO remains unchanged across all tested user configurations, and that the variations in performance are limited. This indicates that the proposed BS-IISI mechanism is robust to the choice of mobility model and that the conclusions drawn from the RWP-based evaluation remain valid under more realistic group mobility conditions.

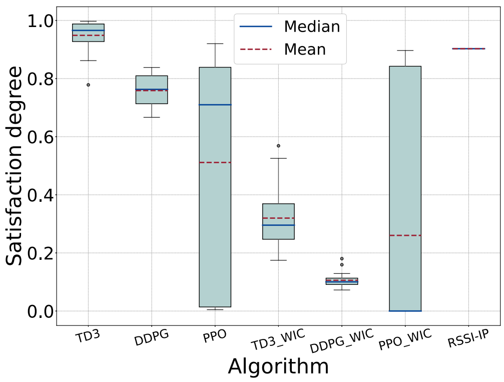
D. INTRA-SLICE BANDWIDTH SHARING PERFORMANCE

To evaluate intra-slice bandwidth allocation, we analyze the performance of the selected algorithms for the eMBB, URLLC, and mMTC slices in terms of users' achieved data rate, degree of satisfaction, and resource wastage. This analysis provides important insights into how effectively each algorithm allocates bandwidth among different groups of users, each with distinct requirements. In addition, the resource wastage metric evaluates the efficiency of bandwidth sharing across users.

Figs. 8, 9, and 10 show the box plots of the data rate and degree of satisfaction for 20 eMBB users, 70 URLLC users and 210 mMTC users, respectively. The following observations are drawn from these results. For the eMBB slice, TD3 achieves the highest median degree of satisfaction, close to 1, with a small interquartile range (IQR) (see Fig. 8(b))indicating

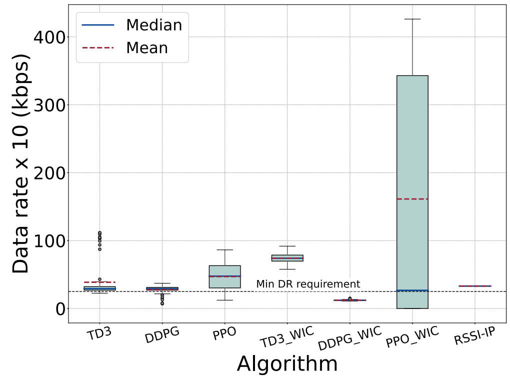


(a) Data rate of users, Eq. (8).

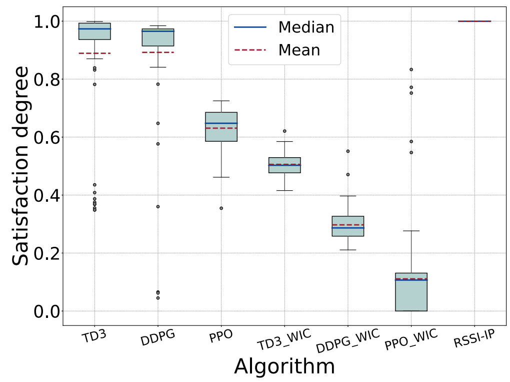


(b) Satisfaction degree of users, Eq. (9).

FIGURE 8. QoS achieved by eMBB users.



(a) Data rate of users, Eq. (8).



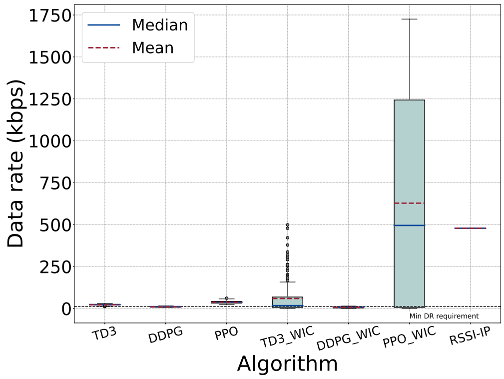
(b) Satisfaction degree of users, Eq. (9).

FIGURE 9. QoS achieved by URLLC users.

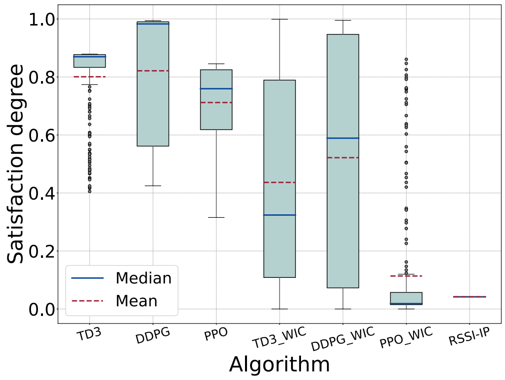
efficient bandwidth allocation that keeps users' data rates near the required threshold of 10 Mbps. RSSI-IP attains a slightly lower median satisfaction with a flat distribution due to uniform resource allocation among users. DDPG performs marginally worse than TD3 and RSSI-IP, as it allocates excessive bandwidth, yielding the highest median data rate (see Fig. 8(a)), which is penalized by the utility function in (9). Its larger IQR reflects higher variability in user satisfaction, though without significant impact on overall performance. In contrast, PPO exhibits a wide dispersion in users' data rates, leading to frequent over- or under-allocation of bandwidth and a corresponding drop in users' satisfaction. For TD3-WIC, DDPG-WIC, and PPO-WIC, the absence of isolation constraints causes a significant drop in the users' satisfaction.

For the URLLC slice, TD3 achieves significantly higher data rates for some users than RSSI-IP (see Fig. 9(a)), resulting in a higher mean data rate above the minimum threshold. Consequently, RSSI-IP outperforms TD3 in user satisfaction. Despite this, TD3 guarantees near-perfect user satisfaction, with an IQR between 0.9 and 1 (see Fig. 9(b)), demonstrating consistent satisfaction for most users. DDPG has several outliers below the required minimum data rate, revealing fairness and consistency issues despite comparable median and mean data rates to TD3. The PPO algorithm's data rate box plot shows a larger IQR above the required minimum threshold compared to RSSI-IP, TD3, and DDPG, indicating a greater dispersion in the data rates provided to URLLC users. This variability leads to reduced user satisfaction. TD3-WIC, DDPG-WIC, and PPO-WIC perform poorly overall, with frequent violations of the minimum data-rate requirement (e.g., DDPG-WIC) or or large variability in data rates (e.g., PPO-WIC). For the mMTC slice, DDPG outperforms TD3, achieving a median degree of satisfaction close to 1 (see Fig. 10(b)), but with a larger IQR, indicating higher variability in user satisfaction. By contrast, TD3's satisfaction distribution includes lower quartiles and outliers, reflecting more fluctuating performance for some users. PPO achieves a lower median satisfaction of approximately 0.7, and despite a narrower IQR than DDPG, its lower median indicates reduced satisfaction for most users. RSSI-IP allocates resources generously, as shown in Fig. 10(a), but this over-allocation leads to low user satisfaction. Although DDPG-WIC achieves data rates comparable to DDPG and TD3, most values remain below the required threshold, resulting in a lower median satisfaction (∼0.6) compared to DDPG (∼1.0) and TD3 (∼0.85). PPO-WIC performs worst, suffering from severe bandwidth over-allocation and yielding the lowest median satisfaction of approximately 0.05.

Fig. 11 illustrates the average rate of resource wastage in each RAN slice. Resource wastage is defined based on the extent to which a user's achieved data rate exceeds the minimum required rate. For a user $u \in \mathcal{U}_s, \forall s \in \mathcal{S}$, if $r^t_{u,s} > R^{req}_s$, the resource wastage is calculated as: $wst^t_{u,s} = e^{-\frac{R^{req}_s}{r^t_{u,s}}}$.



(a) Data rate of users, Eq. (8).



(b) Satisfaction degree of users, Eq. (9).

FIGURE 10. **QoS achieved by mMTC users.**

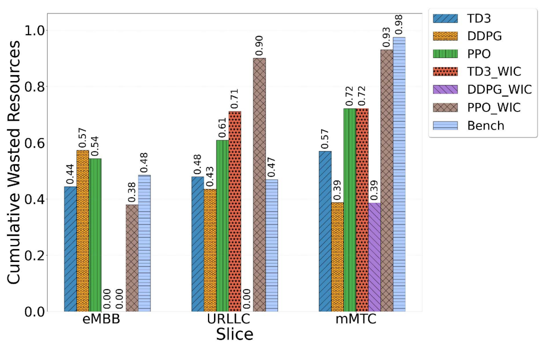


FIGURE 11. **Resource wastage by slice.**

The average resource wastage in slice s is given by: $wst_s^t = \frac{1}{|\mathcal{U}_s|} \sum_{u \in \mathcal{U}_s} wst_{u,s}^t$.

For the eMBB slice, TD3-WIC and DDPG-WIC exhibit no resource wastage, as none of the users reach the minimum required data rate. TD3 incurs some resource wastage but achieves the highest user's degree of satisfaction (see Fig. 8(b)), balancing resource efficiency and user satisfaction. The resource wastage results for the URLLC and mMTC slices are consistent with the data-rate results in Figs. 9(a) and 10(a), respectively. Although PPO, DDPG, and RSSI-IP show variability in resource wastage, TD3 maintains a relatively stable resource wastage rate across all slices, highlighting robustness and adaptability.

VII. CONCLUSION AND FUTURE WORK

This paper presents the BS-IISI mechanism, a RAN slicing solution for TNs designed to balance dynamic bandwidth sharing with robust inter- and intra-slice isolation. The mechanism operates in two stages: inter-slice bandwidth sharing, where slices are allocated portions of the total system bandwidth, and intra-slice bandwidth sharing, where each slice distributes its allocated bandwidth among its users. The bandwidth allocation problem is formulated as an optimization task and solved using DRL algorithms. By leveraging DRL at both stages and enforcing QoS-driven resource reconfiguration constraints, BS-IISI dynamically allocates bandwidth while preserving isolation. We proposed to deploy the BS-IISI mechanism within an O-RAN architecture, detailing the functional blocks supporting its operation and the lifecycle management of the DRL models used. Performance evaluation was conducted using three DRL-based implementations—TD3, DDPG, and PPO—and compared against multiple baselines. The results demonstrate that the TD3-based implementation consistently outperforms both the baselines and the other DRL algorithms.

In future work, we will consider that a user can be associated with multiple slices In addition, we will investigate techniques to accurately monitor DRL inference models used in RAN slicing and detect when they suffer from a severe degradation in performance so they can be proactively replaced by appropriate new models.

REFERENCES

[1] V. M. Baeza and L. C. Salor, "New horizons in tactical communications: An overview of emerging technologies possibilities," *IEEE Potentials*, vol. 43, no. 1, pp. 12–19, Jan./Feb. 2024.

[2] S. E. Elayoubi, S. B. Jemaa, Z. Altman, and A. Galindo-Serrano, "5G RAN slicing for verticals: Enablers and challenges," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 28–34, Jan. 2019.

[3] O-RAN Alliance, "O-RAN: Towards an open and smart RAN," White Paper, Oct. 2018.

[4] D. C. Marques et al., "5G O-RAN potential for military communications," in *Proc. Int. Conf. Mil. Commun. Inf. Syst.*, 2023, pp. 1–8.

[5] Y. Azimi, S. Yousefi, H. Kalbkhani, and T. Kunz, "Applications of machine learning in resource management for RAN-slicing in 5G and beyond networks: A survey," *IEEE Access*, vol. 10, pp. 106581–106612, 2022.

[6] O-RAN Working Group 2, "O-RAN Non-RT RIC architecture," Tech. Rep. O-RAN.WG2.TS.Non-RT-RIC-ARCH-R004-v07.00, Jun. 2025.

[7] O-RAN Working Group 3, "Near-RT RIC architecture," Tech. Rep. O-RAN.WG3.TS.RICARCH-R005-v08.00, Feb. 2026.

[8] F. Yang et al., "A game theoretic framework for distributed mission slice allocation and management for tactical networks," *J. Netw. Syst. Manage.*, vol. 31, no. 1, 2023, Art. no. 23.

[9] A. Castañares et al., "Tactical topology optimization methodology for slice aware and reconfigurable battlefield networks," in *Proc. IEEE Mil. Commun. Conf.*, Rockville, MD, USA, 2022, pp. 273–278.

[10] A. Castañares, D. K. Tosh, and C. A. Kamhoua, "Slice aware framework for intelligent and reconfigurable battlefield networks," in *Proc. IEEE Mil. Commun. Conf.*, San Diego, CA, USA, 2021, pp. 489–494.

[11] J. Dai et al., "O-RAN-enabled intelligent network slicing to meet service-level agreement (SLA)," *IEEE Trans. Mobile Comput.*, vol. 24, no. 2, pp. 890–906, Feb. 2025.

[12] N. Ghafouri, J. S. Vardakas, K. Ramantas, and C. Verikoukis, "A multi-level deep RL-Based network slicing and resource management for O-RAN-Based 6G cell-free networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 11, pp. 17472–17484, Nov. 2024.

[13] A. Gholami, N. Torkzaban, and J. S. Baras, "Mobile network slicing under demand uncertainty: A stochastic programming approach," in *Proc. IEEE NETSOFT*, Madrid, Spain, 2023, pp. 272–276.
[14] S. F. Abedin, A. Mahmood, N. H. Tran, Z. Han, and M. Gidlund, "Elastic O-RAN slicing for industrial monitoring and control: A distributed matching game and DRL approach," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10808–10822, Oct. 2022.
[15] D. Marabissi and R. Fantacci, "Highly flexible RAN slicing approach to manage isolation, priority, efficiency," *IEEE Access*, vol. 7, pp. 97130–97142, 2019.
[16] J. Zhang, Y. Zu, Y. Zhang, and B. Hou, "Dynamic RAN slicing with effective isolation under imperfect CSI," in *Proc. IEEE Int. Conf. Comput. Commun. Eng. Technol*, 2022, pp. 238–242.
[17] L. Tian, L. Dai, Q. Sun, Y. Zhou, Y. Wang, and C. Feng, "Wireless resource management in sliced networks based on isolation indexes," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2021, pp. 1–6.
[18] Y. Azimi, S. Yousefi, H. Kalbkhani, and T. Kunz, "Energy-efficient deep reinforcement learning assisted resource allocation for 5G-RAN slicing," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 856–871, Jan. 2022.
[19] Y. Azimi, S. Yousefi, H. Kalbkhani, and T. Kunz, "Mobility aware and energy-efficient federated deep reinforcement learning assisted resource allocation for 5G-RAN slicing," *Comput. Commun.*, vol. 217, pp. 166–182, 2024.
[20] M. K. Motalleb, V. Shah-Mansouri, S. Parsaeefard, and O. L. A. López, "Resource allocation in an open ran system using network slicing," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 1, pp. 471–485, Mar. 2023.
[21] N. Yarkina et al., "Multi-tenant resource sharing with equitable-priority-based performance isolation of slices for 5G cellular systems," *Comput. Commun.*, vol. 188, pp. 39–51, 2022.
[22] C. Gijón, M. Toril, and S. Luna-Ramıŕez, "Data-driven estimation of throughput performance in sliced radio access networks via supervised learning," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 1008–1023, Jun. 2023.
[23] H. Bai, Y. Zhang, Z. Zhang, and S. Yuan, "Latency equalization policy of end-to-end network slicing based on reinforcement learning," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 1, pp. 88–103, Mar. 2023.
[24] Y. Hou, K. Zhang, X. Liu, G. Chuai, W. Gao, and X. Chen, "An inter-slice RB leasing and association adjustment scheme in O-RAN," *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 1, pp. 402–417, Feb. 2024.
[25] S. A. Hashemian and F. Ashtiani, "Analytical modeling and improvement of interference-coupled RAN slicing," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 13472–13486, Dec. 2024.
[26] H. A. Akyıldız, Ö. F. Gemici, I. Hökelek, and H. A. Çırpan, "Hierarchical reinforcement learning based resource allocation for RAN slicing," *IEEE Access*, vol. 12, pp. 75818–75831, 2024.
[27] E. Hyytiä and J. Virtamo, "Random waypoint mobility model in cellular networks," *Wireless Netw.*, vol. 13, no. 2, pp. 177–188, 2007.
[28] 3GPP (3rd Generation Partnership Project), "Study on new radio access technology: Radio access architecture and interfaces (Release 14)," 3GPP, Sophia Antipolis, France, Tech. Rep. R 38.801 v14.0.0, Mar. 2017.
[29] O-RAN WG 4, "Control, user and synchronization plane specification," O-RAN Alliance, Alfter, Germany, Tech. Specification O-RAN.WG4.CUS.0-R004-v16.01, Oct. 2024.
[30] O-RAN Alliance, "O-RAN Use Cases and Deployment Scenarios," White Paper, Feb. 2020.
[31] 3GPP (3rd Generation Partnership Project), "5G study on channel model for frequencies from 0.5 to 100 GHz," 3GPP, Sophia Antipolis, France, Tech. Rep. 38.901 V15.0.0, Release 15, Jul. 2018.
[32] G. Zhao, S. Qin, G. Feng, and Y. Sun, "Network slice selection in softwarization-based mobile networks," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 1, 2020, Art. no. e3617.
[33] 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.
[34] A. Abouaomar, A. Taik, A. Filali, and S. Cherkaoui, "Federated deep reinforcement learning for open ran slicing in 6G networks," *IEEE Commun. Mag.*, vol. 61, no. 2, pp. 126–132, Feb. 2023.
[35] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1587–1596.
[36] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[37] J. Schulman et al., "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[38] O-RAN Working Group 2, "AI/ML workflow description and requirements," O-RAN Alliance, Alfter, Germany, Tech. Specification O-RAN.WG2.AIML-v01.03, Oct. 2021.
[39] 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.
[40] S. D'Oro, L. Bonati, M. Polese, and T. Melodia, "OrchestRAN: Orchestrating network intelligence in the open RAN," *IEEE Trans. Mobile Comput.*, vol. 23, no. 07, pp. 7952–7968, Jul. 2024.
[41] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "ColO-RAN: Developing machine learning-based xApps for open RAN closed-loop control on programmable experimental platforms," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 5787–5800, Oct. 2023.

ABDERRAHIME FILALI received the Ph.D. degree in electrical engineering from Université de Sherbrooke, Sherbrooke, QC, Canada, in 2021. He is currently a Postdoctoral Research Fellow with the Department of Software and Information Technology Engineering, École de Technologie Supérieure (ÉTS), Montreal, QC. His research interests include resource allocation in mobile wireless communications and edge computing.

DIALA NABOULSI (Member, IEEE) received the Ph.D. degree in computer science from INSA Lyon, Villeurbanne, France, in 2015. She is currently an Associate Professor with École de Technologie Supérieure, Montreal, QC, Canada. Her research interests include mobile networks, edge computing, and virtualized networks.

GEORGES KADDOUM (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from École Nationale Supérieure de Techniques Avancées, Palaiseau, France, the M.S. degree in telecommunications and signal processing from Télécom Bretagne, Brest, France, in 2005, and the Ph.D. (with high Hons.) degree in signal processing and telecommunications from INSA Toulouse, France, in 2009. He is currently a Professor École de Technologie Supérieure (ÉTS), Montreal, QC, Canada, where he also serves as Research Director of the Resilient Machine Learning Institute and Tier 2 Canada Research Chair. He has authored more than 400 journal and conference publications, two book chapters, and holds several patents. His research interests include wireless communications, tactical and resilient networks, radio resource management, and network security. He was the recipient of several Best Paper Awards, including at IEEE IWCMC 2023, IEEE PIMRC 2017, and IEEE WiMob 2014, for his work. He was awarded IEEE Transactions on Communications Exemplary Reviewer Award, major distinctions, including the Université du Québec Research Excellence Award (2018), ÉTS Research Excellence Awards (2019, 2025), IEEE TCSC Award for Excellence (2022), MITACS Award for Exceptional Leadership (2023), NSERC Synergy Award (2025), and Engineers Canada Gold Medal (2025). He serves as Area Editor of IEEE TRANSACTIONS ON MACHINE LEARNING IN COMMUNICATIONS AND NETWORKING and Editor of IEEE TRANSACTIONS ON COMMUNICATIONS.