Reconfigurable RAN Slicing for Ultra-Dense LEO Satellite Networks via DRL

Yuru Liu, *Student Member, IEEE*, Ting Ma, *Member, IEEE*, Xiaohan Qin, *Student Member, IEEE*, Haibo Zhou, *Senior Member, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—Ultra-dense low earth orbit (LEO) satellite network (UD-LSN) is an emerging architecture in the sixth-generation communication system. Network slicing technology can build multiple virtual logical networks for services provided by UD-LSNs on the common physical network. The spatiotemporal variabilities of service requirements and available satellite resources make it necessary to perform reconfigurable resource slicing in UD-LSNs. In this paper, we present a reconfigurable radio access network (RAN) slicing architecture based on grouping and clustering in UD-LSNs. Time is separated into several slicing windows, each further separated into multiple time slots. We take into account the features of the rate-constrained and delay-constrained slices and formulate an optimization problem aiming at maximizing the long-term slicing revenue that involves resource utilization, the service level agreement satisfaction ratio (SSR), and reconfiguration revenues. The problem is tackled by a two-tier deep reinforcement learning (DRL)-based reconfigurable satellite RAN resource slicing and user access (TDRL-RSUA) algorithm. We decouple the original problem into the RAN resource slicing subproblem in slicing windows and user access subproblem at time slots. Specifically, the resource slicing subproblem is solved with the multi-discrete mask Proximal Policy Optimization (MDMPPO) algorithm, while the user access subproblem is solved with the many-to-one matching algorithm. Simulation results demonstrate that our TDRL-RSUA algorithm can improve resource utilization by more than 30% in comparison to the non-reconfigurable resource slicing strategy and achieves higher slicing revenue and SSR.**

***Index Terms*—Reconfigurable RAN slicing, ultra-dense LEO satellite networks, deep reinforcement learning, proximal policy optimization, many-to-one matching.**

I. Introduction

THE SIXTH-GENERATION (6G) communication networks aim to provide faster, more reliable, and lower-latency communication services to meet future demands for high bandwidth, high capacity, and multiple connections [1]. However, traditional terrestrial networks have limitations, like limited coverage and network congestion, making it difficult to adequately meet the requirements for high-performance communications. To compensate for these shortcomings, satellite networks have become a key technology option. Low earth orbit (LEO) satellites operate in lower orbits and have the advantages of lower signal delays, higher data transmission rates, and lower deployment costs in comparison to geosynchronous earth orbit (GEO) and medium earth orbit (MEO) satellites [2]. Consequently, companies like SpaceX [3] and OneWeb [4] program to build mega-satellite constellations by launching thousands of LEO satellites. These ultra-dense LEO satellite networks (UD-LSNs) can provide seamless global coverage and high-performance service, effectively making up for the shortcomings of terrestrial networks [5].

The services provided by UD-LSNs have different service level agreements (SLAs) regarding network bandwidth, latency, and reliability. For instance, rate-constrained services, such as enhanced mobile broadband (eMBB) and large-capacity transmission services, usually need to transmit numerous images or videos and require large transmission rates. Delay-constrained services, such as ultra-reliable low-latency communication (uRLLC) and emergency communication services, focus on low communication latencies [6]. Radio access network (RAN) slicing technology has arisen to meet various SLAs, constructing several virtual logical networks on a common physical network. Leveraging software-defined networking (SDN) technology, slices are created and further flexibly reconfigured according to the spatial and temporal requirements of services [7], [8]. Network resources are managed by different levels of SDN controllers at different temporal granularities. Specifically, in large time-scale windows, the global SDN controller first collects information on resource requirements and allocates resources to network slices to guarantee the SLAs for services. Subsequently, the local SDN controller schedules resources for individual users within the slices at small time-scale slots. Finally, the slicing decisions are adapted to the dynamic changes of spatiotemporal data traffic to improve the resource utilization of networks [9].

In the literature, several works on dynamic RAN slicing for terrestrial networks consider the problems of maximizing the long-term operators' revenues. Some of these adopt model-based optimization methods to solve the issues [10], [11]. However, with the increase in network scales and the lack of

Manuscript received 17 March 2024; revised 17 July 2024; accepted 12 August 2024. Date of publication 26 August 2024; date of current version 7 February 2025. This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1806104, in part by Natural Sciences and Engineering Research Council of Canada (NSERC). The associate editor coordinating the review of this article and approving it for publication was Z. Xiao. *(Corresponding author: Haibo Zhou.)*

Yuru Liu, Xiaohan Qin, and Haibo Zhou are with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China (e-mail: yuruliu@smail.nju.edu.cn; xhderemail@smail.nju.edu.cn; haibozhou@nju.edu.cn).

Ting Ma is with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: tingma@njust.edu.cn).

Xuemin (Sherman) Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/TCCN.2024.3449643

prior information about services, classical optimization techniques can hardly cope with the reconfigurable RAN slicing problem. As artificial intelligence (AI) technology evolves, several papers transform the original problems into Markov Decision Processes (MDPs) and use deep reinforcement learning (DRL) algorithms to address these problems [12], [13]. In addition to fluctuations in demand for services, satellite network resources are dynamically available due to the mobility of satellites. Thus, the dynamic RAN slicing strategies for terrestrial networks cannot be directly used for satellite networks. There are a few works about satellite RAN slicing [14], [15], where resources are sliced to reduce requirement violation cost or increase throughput. However, the reconfigurable RAN slicing issues in UD-LSNs have yet to be extensively studied. Resource utilization and the SLA satisfaction ratio (SSR), the essential performances of RAN slicing, have yet to be considered when making slicing decisions. Meanwhile, as the scale of the satellite network increases, resource management becomes more complex, so it is necessary to consider a suitable reconfigurable resource slicing scheme for UD-LSNs.

Motivated by the above, in this article, we present a reconfigurable RAN resource slicing framework for UD-LSNs, where network resource slicing and user admission decisions are made for the rate-constrained and delay-constrained services. To reduce the complexity of large-scale satellite network resource management, we follow a network management architecture based on grouping and clustering in UD-LSNs [16], [17]. In the architecture, LEO satellites are categorized into several groups and each LEO group is further categorized into several clusters, with each MEO satellite managing an LEO group and each cluster head (CH) LEO satellite managing an LEO cluster. One cluster of satellites in a group owns the same slicing decision and different clusters of satellites have independent slicing decisions. Considering spatial-temporal variations of the service demands and available satellite resources, reconfigurable RAN resource slicing is expressed as an optimization problem aiming at maximizing long-term system revenue. To tackle the formulated problem, we design a two-tier DRL-based reconfigurable satellite RAN resource slicing and user access (TDRL-RSUA) algorithm. The major contributions of our article are as follows:

- We present a reconfigurable RAN resource slicing architecture for UD-LSNs. In each large slice window, the MEO satellite allocates resources for the services and in each small time slot, the CH LEO satellite further dispatches the assigned resources to slice users.
- By analyzing the characteristics of the rate-constrained and the delay-constrained services, we formulate an optimization problem aiming at maximizing the long-term slicing system revenue, including resource utilization, SSR, as well as reconfiguration revenues.
- We propose a TDRL-RSUA algorithm to address the formulated optimization problem. The original problem is decoupled into RAN resource slicing and user access subproblems. Specifically, the resource slicing subproblem is transformed into an MDP and solved with the multi-discrete mask Proximal Policy Optimization (MDMPPO) algorithm, while the many-to-one matching algorithm is applied to solve the user access subproblem.
- Simulation results demonstrate that our designed TDRL-RSUA algorithm can improve resource utilization by more than 30% in comparison to the non-reconfigurable resource slicing strategy and achieves higher slicing revenue and SSR.

The remainder of our paper is arranged as follows. We introduce some related works in Section II. Then we model the system and formulate the reconfigurable slicing problem in Section III. Section IV shows our proposed TDRL-RSUA algorithm to address the formulated problem. The results of the simulations are given in Section V, with the conclusions finally summarized in Section VI.

II. Related Works

A. RAN Slicing in Terrestrial Networks

The RAN slicing technique has been initially studied and developed in terrestrial networks. In industry, the standardization of RAN slicing is investigated by the third-generation partnership project (3GPP) [18]. In academia, RAN slicing has also drawn much attention. Some related works adopt model-based optimization methods to make RAN slicing decisions. In [10], Tang et al. consider the uRLLC and eMBB services and formulate the network slicing problem, which is further solved by the semi-definite relaxation and Successive Convex Approximation (SCA) approach. In [11], Feng et al. introduces an innovative network slicing framework in Mobile Edge Computing (MEC) systems, encompassing slice request admission and an operator revenue model. They further use a Lyapunov-based optimization algorithm to solve the problem.

As AI technology evolves, several papers employ DRL algorithms in solving RAN slicing problems. The paper [12] proposes a DRL-based two-layer uplink and downlink decoupled RAN slicing approach for cellular vehicle-to-everything communications, which can improve network throughput while guaranteeing service requirements. In [13], a strategy that aims to maximize both the long-term Quality of Service (QoS) and the Spectrum Efficiency (SE) of slices is introduced. To realize the strategy, Mei et al. propose a model-free DRL framework, which collaboratively integrates the modified deep deterministic policy gradient (DDPG) and double deep-Q-network algorithm. In [19], Hua et al. investigate the integration of distributional DRL and generative adversarial network (GAN) to obtain the optimization problem for demand-aware resource management to improve slice success rate and SE. Different from the fixed resources in terrestrial networks, the movement of satellites leads to dynamic changes in available satellite resources. Therefore, the dynamic RAN slicing strategy in terrestrial networks cannot be directly used for satellite networks.

B. RAN Slicing in Satellite Networks

As LEO satellite networks evolve, RAN slicing in satellite networks is also drawing attention. In industry, the 3GPP is devoted to standardizing RAN slicing related to satellite networks [20]. In academia, the necessity and challenges

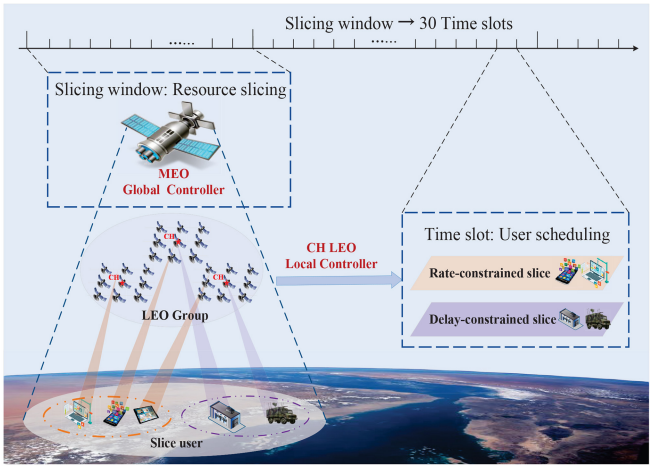


Fig. 1. The reconfigurable RAN slicing framework based on grouping and clustering in UD-LSNs.

of designing resilient dynamic network slicing are described in [21]. To realize flexible, reliable, and scalable network resource management, a software-defined framework for Space-Air-Ground Integrated Vehicular Networks (SAGVN) is designed in [22]. This framework explores AI-based engineering solutions to facilitate efficient network slicing. In space-terrestrial integrated networks, Wu et al. study the RAN resource slicing and scheduling issues in [14]. They introduce a two-tier RL-based Joint Resource Slicing and Scheduling scheme to address these challenges. Focusing on resource slicing in SAGVN, Lyu et al. present a real-time control framework in [15]. This framework enables online decision-making for admission requests, unmanned autonomous vehicle (UAV) placement, and RAN resource slicing. In [23], Zhou et al. establish the low-delay, high-throughput, and wide-coverage RAN slices in SAGIN. They propose two central and distributed multiagent DDPG algorithms to jointly optimize the service delay, throughput and coverage area. Although there are a few related works, the RAN slicing problem in UD-LSNs has not been thoroughly investigated. As the satellite network scales up, slicing resources individually for each LEO satellite results in significant computation complexity. While considering the variability among LEO satellites, applying the same slicing decision to all satellites leads to suboptimal performance. Meanwhile, when making slicing decisions, resource utilization and SSR need to be taken into account, which are the essential performances of RAN slicing. Therefore, the reconfigurable RAN slicing problem in UD-LSNs to improve resource utilization while guaranteeing SSR is worth investigating.

III. System Model and Problem Formulation

In the section, the network model based on grouping and clustering in UD-LSNs is first described and after that we propose a dynamic reconfigurable RAN resource slicing framework. Subsequently, we analyze the performances of the rate-constrained slice and the delay-constrained slice according to their service features. Lastly, we describe the slicing system revenue and pose a long-term optimization problem.

TABLE I
Description of Major Used Symbols

Symbol	Description
$\mathcal{L}^w, \mathcal{U}_r^w, \mathcal{U}_d^w$	Set of available LEO satellites, the rate-constrained slice users, and the delay-constrained slice users in slicing window w.
N	The capacity of LEO satellite channel resources.
N_r^w, N_d^w	The RAN channel resources allocated to the rate-constrained slice and delay-constrained slice in slicing window w.
$a_{l,u_r}^w(t), a_{l,u_d}^w(t)$	The access decision of the rate-constrained slice user and delay-constrained slice user at time (w,t).
$r_{l,u}^w(t)$	The transmission rate between the LEO satellite l and the slice user u.
$d_{l,u}^w(t)$	The distance between the LEO satellite l and the slice user u.
R_r	The minimum transmission rate requirement of the rate-constrained slice.
D_d	The maximum communication delay requirement of the delay-constrained slice.
RU^w, SSR^w, RC^w	The resources utilization, SSR, and slice reconfiguration revenues in slicing window w.

A. Network Model

The requests of slice users are served by UD-LSNs in our considered scenario, as shown in Fig. 1. To reduce the complexity of large-scale satellite network management, LEO satellites are divided into several groups, each managed by an MEO satellite. A group of LEO satellites are further divided into several clusters, each managed by a CH LEO satellite. Specifically, the LEO satellites covered by the same MEO satellite are grouped. If an LEO satellite is covered by two MEO satellites simultaneously, it is assigned to the group associated with the closer MEO satellite. These groups are initially divided into clusters based on maximum connectivity. Clusters with too many LEO satellites are further subdivided, while clusters with too few are merged with others. In addition, the number of clusters in one group is determined by the number of LEO satellites in the group and the maximum number of satellites allowed in a cluster [16]. We consider two slices to support two types of services in UD-LSNs: the rate-constrained slice as large-capacity transmission service and the delay-constrained slice as ultra-remote real-time service. In our scenario, the slice users are distributed over a terrestrial area. LEO satellite resources are available when the angle of elevation between a user and an LEO satellite is greater than the minimum angle of elevation. Because of the fluctuations in slice requests and the movement of LEO satellites, the slice resource requirements and the availability of spectrum resources in the target area change with time.

B. Dynamic Reconfigurable RAN Slicing Framework

The reconfigurable RAN slicing framework operates in a time-slot mode. The time is separated into a number of slicing windows $w \in \mathcal{W} = \{1, 2, \ldots, W\}$, and the satellite networks are grouped and clustered in each slicing window, assuming that the topology of the satellites remains unchanged during this period [16]. The global controller MEO satellite takes resource slicing decisions according to the current service requirements and available LEO satellite resources at the start of the slicing window. Every slicing window is further

separated into multiple time slots $t \in \mathcal{T} = \{1, 2, \ldots, T\}$, during which the local controller CH LEO satellite schedules network resources to users. The MEO global controller assesses the slicing performance according to the responses from the CH LEO satellite at the end of each slicing window and makes adjustments to the resource slicing decisions for the subsequent slicing window. For simplicity in notation, we represent the *t*-th time slot within the slicing window *w* as (*w*, *t*). The major symbols used in our paper are summarized in **Table I**.

The spectrum resources of the LEO satellite are allocated in a unit of channels. Thus, when the user *u* access to one channel of the LEO satellite *l* at time (*w*, *t*), the transmission rate $r_{l,u}^w(t)$ is given by

$$r_{l,u}^w(t) = W_c \log_2\left(1 + \frac{PG d_{l,u}^{w\ -\alpha} h_{l,u}^w(t)}{\sigma^2}\right), \tag{1}$$

where W_c is the bandwith of one channel, σ^2 denotes the Gaussian noise power, *P* denotes the transmit power and *G* denotes the gain factor of power. Assuming that perfect CSI is available, the channel transmission rate $r_{l,u}^w(t)$ is affected by the distance $d_{l,u}^w$ from a user to an LEO satellite and the path loss exponent α due to the large-scale fading. Meanwhile, the small-scale channel fading $h_{l,u}^w(t)$ also affects the transmission rate. Since the transmission between user *u* and LEO satellite *l* is a typical line-of-sight (LoS) communication, the channels are modeled as Rician fading channels [14]. The probability density function of the Rician fading channel is expressed as

$$f(x) = \frac{K+1}{\Omega} exp\left\{-K - \frac{(K+1)x}{\Omega}\right\} I_0\left(2\sqrt{\frac{K(K+1)x}{\Omega}}\right). \tag{2}$$

where Ω denotes the overall received signal power, *K* represents the ratio of the power in the LoS path to the scattering paths, and $I_0(\cdot)$ stands for the modified Bessel function of the first kind with zero order.

In the slicing window *w*, we let N_r^w and N_d^w indicate the number of channels assigned to the rate-constrained slice and delay-constrained slice. Since the channel resources assigned to these services should not be beyond the LEO satellite's total channel resources, we have

$$N_r^w + N_d^w \le N, \ \forall w, \tag{3}$$

where *N* is the capacity of LEO satellite channel resources. It is noted that one cluster of satellites in a group owns the same slicing decision and different clusters of satellites have independent slicing decisions.

At the time (*w*, *t*), if the LEO satellite is communicated with the slice user, we have $a_{l,u}^w(t) = 1$, meanwhile the LEO satellite *l* allocate a channel spectrum resource to the user to satisfy the service demand. Otherwise, we let $a_{l,u}^w(t) = 0$. A user could be served by no more than an LEO satellite, which could be expressed as

$$\begin{aligned} \sum_{l \in \mathcal{L}^w} a_{l,u}^w(t) \le 1, \ \forall t, \ u, \ w, \\ a_{l,u}^w(t) = \{0, 1\}, \ \forall t, \ u, \ l, \ w. \end{aligned} \tag{4}$$

where $l \in \mathcal{L}^w = \{1, 2, \ldots, L^w\}$ and $\mathcal{L}^w$ denote the set of available satellites in a cluster in slicing window *w*.

C. Performance Analyses for Two Types of Slices

In this subsection, we consider the user arrival model and analyze the SSR based on the service characteristics of two slices.

1) The Rate-Constrained Slice: There are U_r^w rate-constrained slice users in slicing window *w*. The users associated with the rate-constrained service typically generate continuous traffic with an infinite packet size (i.e., full-buffer traffic). SSR measures the satisfaction ratio of slice users with the quality of service and has a value between 0 and 1. Inspired by [24], when the transmission rate of the rate-constrained slice user $u_r \in \mathcal{U}_r^w = \{1, 2, \ldots, U_r^w\}$ is $r_{l,u_r}^w(t)$, we could express the SSR of this user as

$$SSR_{l,u_r}^w(t) = \begin{cases} \frac{1}{1+e^{-b_r\left(r_{l,u_r}^w(t) - R_r\right)}}, & a_{l,u_r}^w(t) = 1, \\ 0, & a_{l,u_r}^w(t) = 0, \end{cases} \tag{5}$$

where R_r is the minimum required transmission rate of rate-constrained slice. The b_r is a positive constant so that the $SSR_{l,u_r}^w(t)$ increases with the transmission rate $r_{l,u_r}^w(t)$. The b_r also affects the shape of the SSR curve, and by adjusting this value, it could realize that $SSR_{l,u_r}^w(t)$ tends to 0 when the transmission rate $r_{l,u_r}^w(t)$ is less than the threshold R_r and tends to 1 when the transmission rate is quite large.

2) The Delay-Constrained Slice: In slicing window *w*, the number of delay-constrained slice users is U_d^w, and at time slot (*w,t*) is $U_d^w(t)$. The users associated with the delay-constrained service typically generate bursts of packets following the Poisson Point Process (PPP) with the arrival rate of λ^w. The communication delay of user $u_d \in \mathcal{U}_d^w(t) = \{1, 2, \ldots, U_d^w(t)\}$ consists of transmission and propagation delay. The transmission rate $r_{l,u_d}^w(t)$ affect the transmission delay $D_{l,u_d}^{w,tran}(t)$, and the distance $d_{l,u_d}^w(t)$ between the user and the associated LEO satellite influence the propagation delay $D_{l,u_d}^{w,prop}(t)$. Therefore, the communication delay can be given by

$$\begin{aligned} D_{l,u_d}^w(t) &= D_{l,u_d}^{w,tran}(t) + D_{l,u_d}^{w,prop}(t) \\ &= \frac{p_{u_d}^w(t)}{r_{l,u_d}^w(t)} + \frac{d_{l,u_d}^w(t)}{c}, \end{aligned} \tag{6}$$

where $p_{u_d}^w(t)$ is the packet size of the arrival user and *c* is the speed of light, with a value of $3 \times 10^8\ m/s$. Similarly, we could express the SSR of the delay-constrained slice user as

$$SSR_{l,u_d}^w(t) = \begin{cases} \frac{1}{1+e^{-b_d\left(D_d - D_{l,u_d}^w(t)\right)}}, & a_{l,u_d}^w(t) = 1, \\ 0, & a_{l,u_d}^w(t) = 0, \end{cases} \tag{7}$$

where D_d is the maximum communication delay requirement of delay-constrained slice. The b_d is a negative constant so that the $SSR_{l,u_d}^w(t)$ decreases with the communication delay $D_{l,u_d}^w(t)$. By adjusting b_d to realize that $SSR_{l,u_d}^w(t)$ tends to 0 when the communication delay $D_{l,u_d}^w(t)$ is more than the threshold D_d and tends to 1 when the communication delay is quite small.

D. Problem Formulation

In our paper, we consider system revenue from the perspective of resource utilization, SSR, and slice reconfiguration revenues.

1) The Resources Utilization Revenue: Resource utilization is an essential performance metric of network slicing, which measures whether network resources are effectively allocated and fully utilized. According to [24], resource utilization is defined as the ratio of used slice resources to the configured slice resources. Specifically, one LEO channel resource is used when it is accessed by a slice user. The resource utilization within a slicing window is the average resource utilization across all time slots. Thus, the resource utilization RU^w is obtained by

$$RU^w = \frac{\sum_{t\in\mathcal{T}}\sum_{l\in\mathcal{L}^w}}{L^w T}\left(\frac{\sum_{u_r\in\mathcal{U}_r^w} a_{l,u_r}^w(t) + \sum_{u_d\in\mathcal{U}_d^w(t)} a_{l,u_d}^w(t)}{N_r^w + N_d^w}\right). \tag{8}$$

2) The SSR Revenue: SSR is another performance metric of network slicing that measures the service experiences of sliced users. While not wasting resources, satellite network operators want to provide high-quality services to as many slice users as possible. Due to (5) and (7), a slice user's SSR can be represented as

$$SSR_u^w(t) = \sum_{l\in\mathcal{L}^w} SSR_{l,u}^w(t). \tag{9}$$

Similarly, the SSR within a slicing window is the average of all slice users' SSR across all time slots. Thus, in slicing window w, we have

$$SSR^w = \frac{\sum_{t\in\mathcal{T}}}{2T}\left(\frac{\sum_{u_r\in\mathcal{U}_r^w} SSR_{u_r}^w(t)}{U_r^w} + \frac{\sum_{u_d\in\mathcal{U}_{u_d}^w(t)} SSR_{u_d}^w(t)}{U_d^w(t)}\right). \tag{10}$$

3) Slice Reconfiguration Revenue: The controller might modify the channel resources assigned to slices in different slicing windows according to the current available LEO resources and the demand for slice services. However, the process of slice resource adjustment incurs slice reconfiguration costs. In slicing window w, let RC^w represent the reconfiguration cost, which quantifies the discrepancy in resource assigned decisions between adjacent slicing windows, i.e.,

$$RC^w = \left[N_r^w - N_r^{w-1}\right]^+ + \left[N_d^w - N_d^{w-1}\right]^+, \tag{11}$$

where function $[x^w - x^{w-1}]^+$ equal to $\max\{x^w - x^{w-1}, 0\}$, since the cost of releasing resources is negligible [14].

With (8)-(11), the overall system revenue in slicing window w can be modeled as

$$U^w = w_{ru} RU^w + w_{ssr} SSR^w - w_{rc} RC^w, \tag{12}$$

where parameters w_{ru}, w_{ssr}, and w_{rc} affect the relative significance of the three types of revenues. Specifically, when configuring slice resources, there is a trade-off between resource utilization and SSR by adjusting the weights w_{ru} and w_{ssr}. Meanwhile, the reconfiguration weight w_{rc} is set according to the cost of reconfiguring resources on satellites.

Taking into account the spatial-temporal variability of the available LEO satellite network resources and slice service requirements, it is of significant importance to perform reconfigurable resource slicing to maximize the long-term system revenue. The reconfigurable satellite RAN resource slicing and user access (RSUA) problem can be formulated as

$$P_0: \max_{\{N_r^w, N_d^w, a_{l,u}^w(t)\}} \sum_{w\in\mathcal{W}} U^w \tag{13}$$

$$\text{s.t.} \quad N_r^w + N_d^w \le N, \ \forall w, \tag{13a}$$

$$\sum_{l\in\mathcal{L}^w} a_{l,u}^w(t) \le 1, \ \forall t, \ u, \ w, \tag{13b}$$

$$\sum_{u_r\in\mathcal{U}_r^w} a_{l,u_r}^w(t) \le N_r^w, \ \forall t, \ l, \ w, \tag{13c}$$

$$\sum_{u_d\in\mathcal{U}_d^w(t)} a_{l,u_d}^w(t) \le N_d^w, \ \forall t, \ l, \ w, \tag{13d}$$

$$a_{l,u}^w(t) \in \{0,1\}, \ \forall t, \ u, \ l, \ w, \tag{13e}$$

$$N_r^w, N_d^w \in \mathbb{Z}^+, \ \forall w, \tag{13f}$$

where constraint (13a) guarantees the total channel resource allocated to the rate-constrained slice and delay-constrained slice should not be beyond the LEO satellite's total channel resources. Constraint (13b) ensures that one user can access no more than an LEO satellite. Constraints (13c) and (13d) guarantee that the number of channels allocated to rate-constrained slice and delay-constrained slice users cannot exceed the intra-slice channel resources. Constraints (13e) and (13f) indicate the access decisions are 0-1 variables and the resource allocation decisions are int variables.

Problem P_0 is a long-term integer nonlinear programming problem and belongs to stochastic optimization because of spatiotemporal variations of the slice service requirements and available LEO satellite resources. The slicing revenue is jointly decided by the coupled slicing resource configuration and user access decisions. Meanwhile, classical optimization techniques can hardly cope with the reconfigurable slicing problem due to a lack of priori information about the services. With the development of AI, this problem can be effectively tackled by DRL. Hence, we design a two-tier DRL-based reconfigurable satellite RAN resource slicing and user access (TDRL-RSUA) algorithm to resolve this problem.

IV. Design of the TDRL-RSUA Algorithm

In the section, the original problem is decoupled into RAN resource slicing and user access subproblems. The resource slicing subproblem is solved with the multi-discrete mask Proximal Policy Optimization (MDMPPO) algorithm in slicing windows, while the user access subproblem is solved with the many-to-one matching algorithm in time slots.

A. Matching-Based User Access Subproblem

According to (11), user access decisions $a_{l,u}^w(t)$ have no effect on slice reconfiguration revenue. Therefore, maximizing the sum of resource utilization and SSR revenues is the goal of the user access subproblem. With slice isolation, the process

of user access to the LEO satellite channel in both slices is independent of each other. To simplify the notation, we omit the (w, t). Thus, user access in the rate-constrained slice can be formulated as follows:

$$P_1: \max_{a_{l,u_r}} \sum_{l\in\mathcal{L}} \sum_{u_r\in\mathcal{U}_r} \left(\frac{w_{ru} a_{l,u_r}}{LN_r} + \frac{w_{ssr} SSR_{l,u_r}}{U_r} \right) \quad (14)$$

$$\text{s.t.} \quad a_{l,u_r} \in \{0,1\}, \ \forall \ l, \ u_r, \quad (14a)$$

$$\sum_{l\in\mathcal{L}} a_{l,u_r} \le 1, \ \forall \ u_r, \quad (14b)$$

$$\sum_{u_r\in\mathcal{U}_r} a_{l,u_r}(t) \le N_r, \ \forall \ l. \quad (14c)$$

Similarly, user access in the delay-constrained slice can be formulated as follows:

$$P_2: \max_{a_{l,u_d}} \sum_{l\in\mathcal{L}} \sum_{u_d\in\mathcal{U}_d} \left(\frac{w_{ru} a_{l,u_d}}{LN_d} + \frac{w_{ssr} SSR_{l,u_d}}{U_d} \right) \quad (15)$$

$$\text{s.t.} \quad a_{l,u_d} \in \{0,1\}, \ \forall \ l, \ u_d, \quad (15a)$$

$$\sum_{l\in\mathcal{L}} a_{l,u_d} \le 1, \ \forall \ u_d, \quad (15b)$$

$$\sum_{u_d\in\mathcal{U}_d} a_{l,u_d}(t) \le N_d, \ \forall \ l. \quad (15c)$$

These two problems are integer nonlinear programming problems and aim to maximize the number and SSR of successful access slice users. To solve these problems, we formulate them into a generic matching problem [25]. We construct a bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{L}, \xi)$, in which $\mathcal{U}$ is the group of the rate-constrained slice users or the delay-constrained lice users, $\mathcal{L}$ is the group of available LEO satellites, and ξ is the group of edges connecting users in $\mathcal{U}$ and LEO satellite in $\mathcal{L}$. For an edge (u,l) connecting slice user u and LEO satellite l ($u \in \mathcal{U}, l \in \mathcal{L}$), we have $(u, l) \in \xi$. Therefore, the slice user access problem aims to discover the matching $\mathcal{M} \in \mathcal{G}$ that maximizes the number and SSR of successful access slice users [26]. To satisfy constraints (14b) and (15b), each user could be paired with no more than one LEO satellite, and each LEO satellite could be paired with several users according to its channel resources. Therefore, the form of a matching is described as:

Definition 1: Suppose there are two disjoint sets of the slice users $\mathcal{U}$ and the available LEO satellites $\mathcal{L}$. Define a many-to-one matching $\mathcal{M}$ as a mapping from the set consisting of the subsets of $\mathcal{U} \cup \mathcal{L}$ into the set $\mathcal{U} \cup \mathcal{L}$, such that for every $u \in \mathcal{U}$ and $l \in \mathcal{L}$, we have: 1) $\mathcal{M}(u) \subseteq \mathcal{L}$ and $\mathcal{M}(l) \subseteq \mathcal{U}$; 2) $M(u) = l \Leftrightarrow u \in \mathcal{M}(l)$; 3) $|\mathcal{M}(u)| \le 1$, $|\mathcal{M}(l)| \le N$. Specifically, when u belongs to the rate-constrained slice u_r, N equals N_r. While u belongs to the delay-constrained slice u_d, N is equal to N_d [27]. When $M(u) = l$, it implies that $a_{l,u} = 1$; otherwise $a_{l,u} = 0$. To reach a stable result matching, the slice user's preference list $\mathcal{P}L_u$ is sorted in a descending order based on the slice user's SSR.

Definition 2: Denote $\succ$ as the preferences of slice users. For the rate-constrained slice users, we have

$$l_i \succ_{u_r} l_j \Leftrightarrow r_{l_i,u_r} > r_{l_j,u_r}, \quad (16)$$

Algorithm 1: Many-to-One Matching User Access Algorithm

Input: Slice users $u \in \mathcal{U}$, available LEO satellites $l \in \mathcal{L}$, slice channel resource N.

Output: Slice user access decisions $a_{l,u}$.

Initialization: Construct many-to-one mapping $\mathcal{M}$ to record slice users and available LEO satellites which are matched.
For each slice user u, construct a preference list $\mathcal{P}L_u$ for LEO satellites according to (16) or (17). All users are not associated with the satellite $a_{l,u} = 0$.
repeat
for *all* $u \in \mathcal{U}$ **do**
  if $|M(u)| = 1$ **then**
    The slice user u has been accessed by one of the LEO satellites.
  else
    The slice user u sends an access request to its preferred LEO satellite l and deletes this LEO satellite from its $\mathcal{P}L_u$.
  end if
end for
for *all* $l \in \mathcal{L}$ **do**
  if $|M(l)| = N$ **then**
    The channel resource of the LEO satellite l has been consumed.
  else
    The LEO satellite l checks the received proposals and determines which proposals to accept based on the remaining channel resources such that maximize the total value of (14) or (15). The access decisions of accepted users are set to $a_{l,u} = 1$.
  end if
end for
until All slice users are matched, or there are no available LEO satellite resources.

which implies that the rate-constrained slice user u_r prefers LEO satellite l_i to l_j only when user u_r can get higher transmission rate at LEO satellite l_i. Similarly, the preference of the delay-constrained slice user is described as

$$l_i \succ_{u_d} l_j \Leftrightarrow D_{l_i,u_d} < D_{l_j,u_d}, \quad (17)$$

which implies that the delay-constrained slice user u_d prefers the LEO satellite that offers low communication delay D_{l_i,u_d}.

The overall process of slice user access is described in **Algorithm 1** and it is implemented on the local controller CH LEO satellite. In UD-LSNs, the matching scale is reduced through grouping and clustering architecture. By the way, the local controller CH LEO satellite can tell the satellite about the remaining resources of other satellites. When the slice user is rejected by the current satellite and removes it from its preference list, it can also synchronously remove the satellites with no remaining resources, thus avoiding unnecessary signaling overhead caused by sending access

requests to these satellites. The rate-constrained slice users and the delay-constrained slice users implement the many-to-one matching algorithm independently, and user access issues are resolved in a low complexity.

B. DRL-Based Reconfigurable Resource Slicing Subproblem

The reconfigurable resource slicing optimization subproblem is formulated as follows:

$$P_3: \max_{\{N_r^w, N_d^w\}} \sum_{w \in \mathcal{W}} U^w \quad \text{s.t. (13a) and (13f).} \tag{18}$$

The subproblem is optimized to find resource slicing decisions for each slicing window to maximize the long-term system revenue, and it belongs to the class of MDP. Therefore, we redescribe this optimization subproblem into an MDP. In particular, the agent is the MEO satellite. The MEO satellite agent collects state data s^w and assigns channel resources a^w. The environment then returns the reward r^w and goes to the next state s^{w+1}. We then provide a detailed description of the action, state, and reward in this MDP.

Action: In slicing window w, the action a^w is the assignment of channel resources of LEO satellites to the rate-constrained slice and the delay-constrained slice, which can be described as

$$a^w = \{N_r^w, N_d^w\}. \tag{19}$$

State: The MEO satellite obtains the environmental status data, including the number of the rate-constrained and the delay-constrained slices arriving users U_r^w and U_d^w, the available LEO satellites number L^w, the packet poisson arrival rate of the delay-constrained slice user λ^w as well as the resource slicing action in the preceding window a^{w-1}. Therefore, we can define the state as

$$s^w = \left\{U_r^w, U_d^w, L^w, \lambda^w, a^{w-1}\right\}. \tag{20}$$

Reward: The reward is designed to be the slicing revenue in terms of resource utilization and SSR obtained from the many-to-one matching access algorithm, as well as the reconfiguration revenue. Thus, the reward is expressed as

$$r^w(s^w, a^w) = U^w. \tag{21}$$

A resource assignment policy defines how the MEO satellite assigns LEO channel resources according to the current network state at the start of slicing windows. The set of all probable resource assignment policies is denoted by Π. Thus, our objective is to identify the resource assignment strategy $\pi^\star \in \Pi$ that could maximize the cumulative discounted reward over a few slicing windows, which is described as

$$P_3': \max\nolimits_{\pi \in \Pi} \left[\sum_{w=1}^{W} \gamma^w r^w(s^w, a^w)|\pi\right] \quad \text{s.t. (13a) and (13f).} \tag{22}$$

where the discount factor $\gamma \in (0, 1)$ represents the agent's trade-off from immediate to future rewards. As the discount factor γ approaches 1, the agent prioritizes future rewards, while approaching 0 emphasizes immediate rewards. When the discount factor is close to one, P_3' can effectively approximate the problem P_3 [14].

Algorithm 2: MDMPPO-Based Reconfigurable Satellite RAN Slicing Algorithm

Initialization: Initialize buffer $\mathcal{D}$. Initialize the actor and critic networks with parameters θ_A and θ_C.
for *episode* $= 1 : E_p$ **do**
  Reset the state about the satellites and users and get the initial environment state s^0.
  for *slicing window* $w = 1 : W$ **do**
    The MEO satellite controller observes state s^w and configures slice channel resources a^w based on policy $\pi^k = \pi(\theta_A^k)$ and multi-discrete mask action layer.
    The MEO satellite controller gets resource utilization and SSR via **Algorithm 1** and in turn calculates reconfiguration costs to get the overall reward r^w by (21).
    The environment goes to the next state s^{w+1}. Meanwhile, the MEO satellite controller records the experience $\left\{s^w, a^w, r^w, s^{w+1}\right\}$ in the replay buffer $\mathcal{D}$.
  end for
  ▷ When M episodes have been conducted, update the neural networks parameters:
  if episode % M == 0 **then**
    Compute GAE $A^{\pi_{\theta_{old}}}(w)$ based on the current critic networks parameters according to (24).
    for *epoch* $= 1 : K$ **do**
      Randomly select a minibatch of experiences from the replay buffer D.
      Update the actor networks according to (27) via the policy gradient method. Update the critic networks by (28) via the gradient descent method.
    end for
    Clean buffer $\mathcal{D}$.
  end if
end for

The satellites in the next generation networks have additional onboard processing capabilities. Meanwhile, the advancement in computing resources has greatly facilitated the deployment of AI algorithms on satellites [28]. Due to the absence of future information regarding time-varying service requirements in UD-LSNs, there exists an unknown state transition probability. Consequently, the optimal resource allocation policy could be acquired through a model-free policy gradient (PG)-based RL algorithm, that requires no state transition probabilities. The algorithm design is elaborated as follows.

1) Proximal Policy Optimization (PPO): The PG-based DRL algorithm PPO falls under random RL strategies. Unlike the deterministic strategies, its strategy function is related to probability distribution. A stochastic policy predicts the

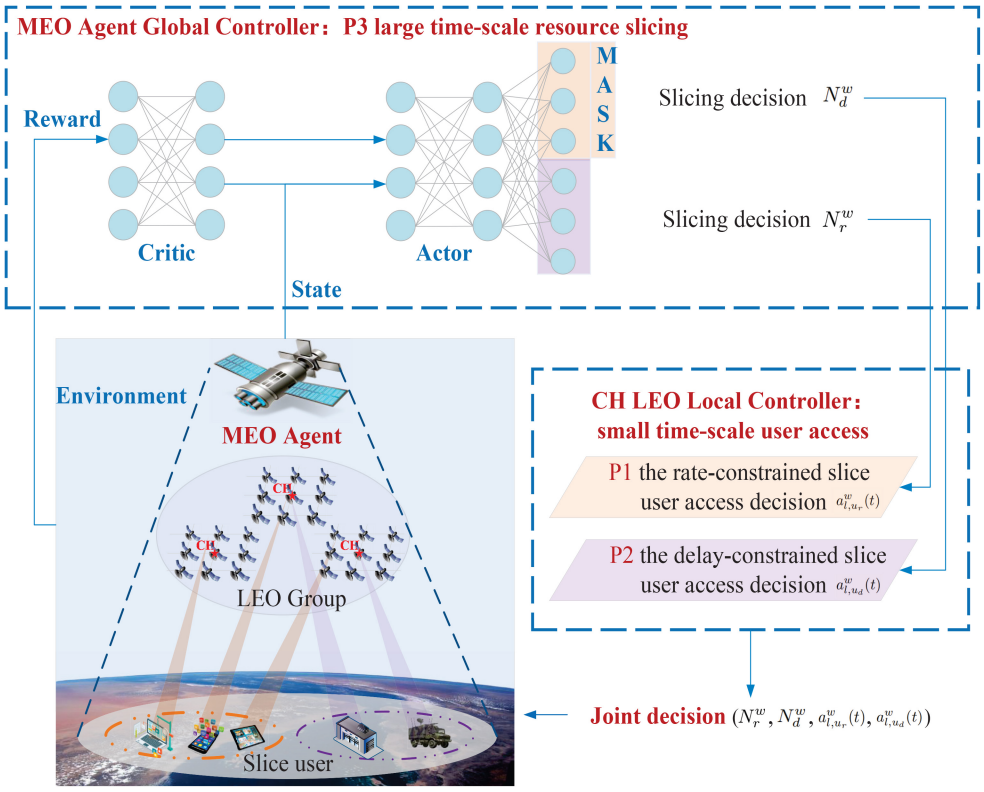


Fig. 2. Overview of the proposed TDRL-RSUA algorithm.

probability distribution of actions at the state s^w and samples actions according to the distribution. Consequently, the strategy $\pi(a^w|s^w)$ will output different actions even in the same state. The fundamental structure of the PPO algorithm comprises several deep neural networks, i.e., the actor and the critic networks. The probability distribution of actions is mapped from state s^w by the actor networks. The critic network functions as a state value that assesses the degree of goodness or badness of current states. It leverages reward to reinforce or diminish the probability of chosen behaviors directly, enhancing the probability of selecting favorable actions [29].

PPO utilizes the policy gradient algorithm for updates, and the manner is usually defined as

$$g = E[\nabla_\theta log\pi_\theta(a^w|s^w)A(s^w, a^w)]. \tag{23}$$

where ∇_θ is a policy that is parameterized by θ, $A(s^w, a^w)$ is the estimated advantage function at slicing window w. PPO considers the effect of bias and variance on the model, incorporating generalized advantage estimation (GAE) to compute the advantage function [30], expressed as

$$\begin{aligned} A(s^w, a^w) = \delta(w) + (\gamma\eta)\delta(w+1) \\ + \cdots + (\gamma\eta)^{W-w-1}\delta(W-1). \end{aligned} \tag{24}$$

where $\delta(w) = r^w(s^w, a^w) + \gamma V_\theta(s^{w+1}) - V_\theta(s^w)$, $V_\theta(s^w)$ is the state-value function at slicing window w approximated by the critic network V_θ and η is a discount hyperparameter.

To avoid significant changes in the policy, PPO introduces a clipped function to restrict deviations from the previous policy. The ratio φ_θ of the probability between the current and old policies is

$$\varphi(\theta) = \frac{\pi_\theta(a^w|s^w)}{\pi_{\theta_{old}}(a^w|s^w)}. \tag{25}$$

The goal of the PPO algorithm is to discover the optimal policy that enables the following objective function to be maximized

$$\begin{aligned} L^{clip}(\theta) = E_{s^w, a^w}[min(\varphi(\theta)A^{\pi_{\theta_{old}}}(s^w, a^w), \\ clip(\varphi_\theta, 1-\varepsilon, 1+\varepsilon)A^{\pi_{\theta_{old}}}(s^w, a^w))], \end{aligned} \tag{26}$$

where $A^{\pi_{\theta_{old}}}(s^w, a^w)$ is the GAE for policy θ_{old} and can be calculated by (24). The loss function of the actor networks could be described as

$$L^{actor}(\theta) = L^{clip}(\theta) + cH(\pi_\theta, s^w), \tag{27}$$

where $H(\pi_\theta, s^w)$ denotes the entropy of policy π_θ at state s^w with c being a hyperparameter, which is usually equal to 0.01. The parameters of the critic network are updated by minimizing the loss function, i.e.,

$$L^{critic}(\theta) = E\big[V^{tar}(s^w, a^w) - V_\theta(s^w)\big], \tag{28}$$

where $V^{tar}(s^w, a^w) = A^{\pi_{\theta_{old}}}(s^w, a^w) + V_{\theta_{old}}(s^w)$.

2) Multi-Discrete Mask Action Layer Design: The action network of PPO traditionally outputs the probability distribution of a one-dimensional decision. However, resource slicing involves a multi-dimensional coupled decision process. If we map the resource slicing decision into one dimension, the mapping method and the increase in dimensionality can

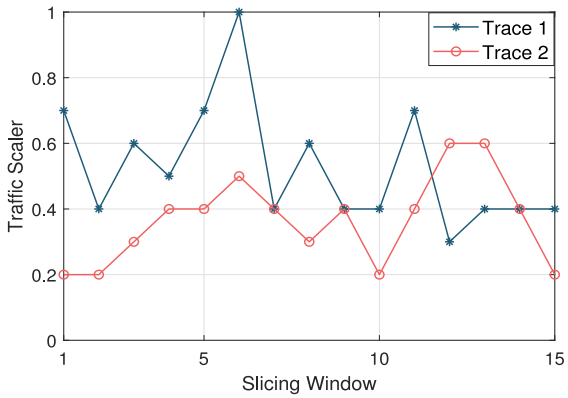


Fig. 3. The traffic scalers of slice users.

impact training performance. Therefore, we utilize multi-discrete actor networks [31]. Specifically, the state $s(w)$ is mapped into two heads via several shared middle layers. One head is responsible for allocating channel resources to the rate-constrained slice, while the other head determines resource allocation for the delay-constrained slice. Every head generates N digits, which are subsequently processed by the softmax function to produce an N-dimensional vector representing the probability of allocating $1\sim N$ channel resources to the corresponding slice. We sample the probability distribution to obtain the resource allocation action. Nonetheless, the total channel resources allocated to both slices should not be beyond the LEO satellite's capacity, so it is necessary to mask the actions violating this constraint [32]. Expressly, when the rate-constrained slice is assigned N_r channel resource, the resource of the delay-constrained slice is limited to $N - N_r$, and actions beyond this number are invalid actions. To achieve this, we design an action mask layer between the output and softmax layers for the delay-constrained slice head. The action mask layer helps to avoid sampling invalid actions by setting virtually zero to the outputted probabilities of the invalid actions [33]. The entire training process is presented in **Algorithm 2**, and it is implemented on the global controller MEO satellite. The notation E_p denotes the total number of training episodes, the notation M denotes the neural network parameters update period, and the notation K denotes the number of minibatch updates.

V. Simulation Results

In the section, we run several simulations to assess the TDRL-RSUA algorithm's performances.

A. Simulation Setup

We first model the UD-LSNs topology of Starlink with 11927 LEO satellites [34]. For satellite networks, satellite movements are divided into topology initialization and topology updating stages. In the initialization stage, the mobility module generates initial coordinates based on constellation parameters. In the updating stage, the SGP4 orbit prediction model calculates current coordinates, considering space dynamics. The target area has a latitude of 31.5°N to 33°N and a longitude of 79.5°W to 81°W, which belongs to the Atlantic Ocean region near the east coast of the United States of America. The LEO satellites in our target area belong to the same group and are managed by an MEO satellite. Within the group, the LEO satellites are further divided into seven clusters, and we perform simulations in one cluster as an example. The slice users are distributed randomly in our target region. The minimal access elevation angle is 40° for the first phase of Starlink and 35° for the second phase.

TABLE II
Main Simulation Parameters

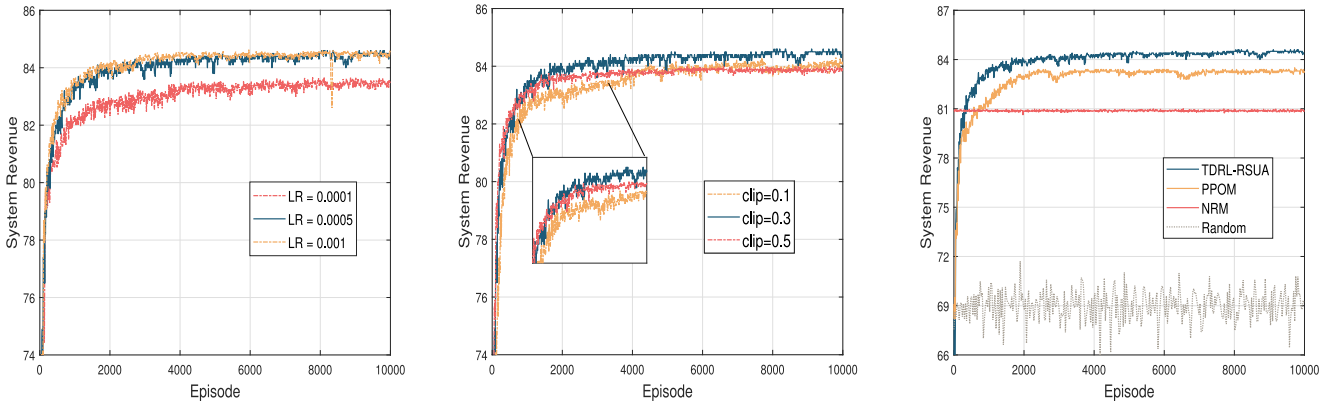
Parameter	Value	Parameter	Value
G	30dbi [32]	σ^2	-204dbm/hz [40]
α	3 [41]	P	2W [42]
W_c	10MHz [32]	P_{u_d}	5Kb [19]
R_r	10Mbps [43]	D_d	10ms [44]
K	10 [40]	N	10 [45]

Since it is currently difficult to obtain traffic information about satellite networks, we utilize ground-based traffic to stand for it. Like [35], there are nearly 220 users in our target area according to GeoLite2 IP geolocation database [36]. The length of one slicing window is set to be 1 minute [37], while the time slot length is 2 seconds [38]. To obtain the characteristics of traffic over time, two different traffic scalers from 14:00:00 9/21/2023 to 14:15:00 9/21/2023 are extracted from WIDE Project [39]. We show the traffic scalers in Fig. 3, which represent the active probability of total users. Hence, by multiplying the traffic scalers by the number of total users, we could get the time-varying slice users in each slicing window.

The number of LEO satellite channels is set to 10, with a channel bandwidth of 10 MHz. The packet arrival rate of the delay-constrained slice users in different slicing windows is stochastically selected from [1,4] packets/s. In our simulation scenario, it is easy to provide high SSR due to the relatively sufficient satellite resources, but the problem of low resource utilization exists. Our goal is to improve resource utilization while guaranteeing high SSR, thus the weights of resource utilization and SSR are set as $w_{ru} = 1$, $w_{ssr} = 5$. For flexible slicing decisions, the weight of the slice reconfiguration is set to $w_{rc} = 0.01$. The other experimental parameters are summarized in **Table II**.

For both the actor and critic networks, we employ a two-tier fully connected neural network with [512, 512] neurons, which use the Tanh activation function. Furthermore, the Adam optimizer is adopted to train networks parameters. Each training episode comprises 15 slicing windows and 450 time slots. To show the validity of our proposed TDRL-RSUA algorithm, we use the following benchmark algorithms for comparison:

- PPO with matching strategy (PPOM): The traditional PPO algorithm is used to make resource slicing decisions in each large slicing window. Different from our proposed TDRL-RSUA algorithm, which generates two-dimensional masked resource slicing decisions, the traditional PPO algorithm gives a one-dimensional slicing decision. This decision can be mapped to the channel resources allocated in the rate-constrained and delay-constrained slices that satisfy the resource constraints.



(a) Training records under several learning rates. (b) Training records under several clipping ratios. (c) Training records of different algorithms.

Fig. 4. Convergence property of the system revenue.

Meanwhile, the proposed matching algorithm is still used at small time slots.

- Non-reconfigurable RAN slicing with matching strategy (NRM): The slice resources are not reconfigurable and maintain a fixed value in all large slicing windows. In order to make reasonable non-reconfigurable slicing decisions, we compute the average of the channel resources that slice users require overall slicing windows. Then, we perform non-reconfigurable slice resource allocation decisions based on the ratio of two slices' required resources. Meanwhile, the proposed matching algorithm is also used at small time slots.
- Random strategy (Random): The resource slicing decisions are made randomly at each slicing window, and the user access decisions are also made randomly at each time slot.

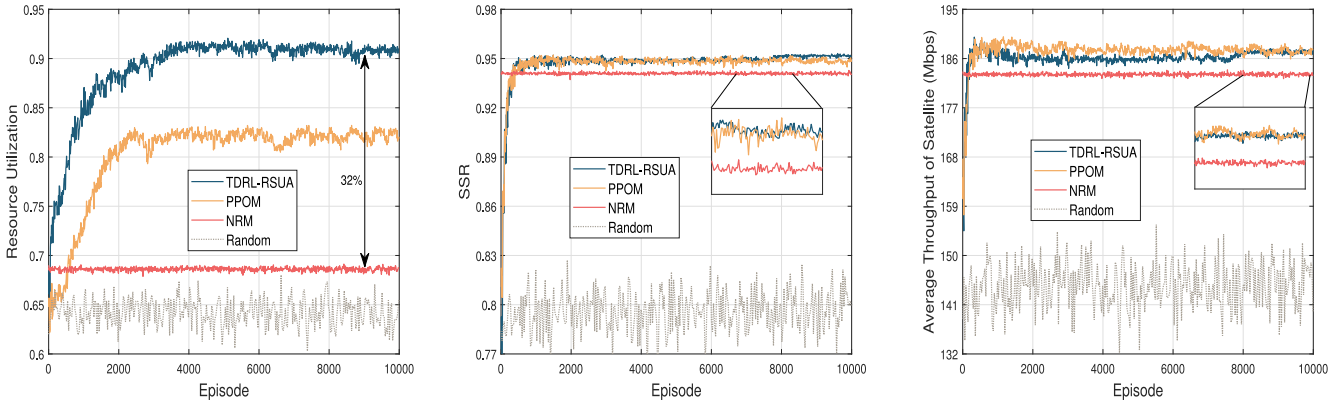
B. Performance Analysis

The performance analysis of our TDRL-RSUA algorithm comprises two parts. Initially, we present the training processes of all the algorithms. Subsequently, the well-trained network models are conducted under several scenarios to verify the effectiveness of our proposed algorithm.

1) Convergence Analysis: Fig. 4 presents the convergence performance of our proposed TDRL-RSUA algorithm. A 10-point moving average of the system revenue is used to accentuate the convergence trend. Fig. 4(a) illustrates the system revenue with different learning rates, where *LR* represents the learning rate for the networks. It is observed that all curves increase with the increase in training episodes and eventually stabilize at respective optimal system revenues, verifying the convergence performance of our TDRL-RSUA algorithm. Additionally, the convergence speed of system revenue is quick when the learning rate is large. However, a learning rate that is too large can lead to unstable convergence performance. Fig. 4(b) presents the system revenue with different clipping ratios. It is evident that when the clipping ratio is large, the system revenue converges quickly because of drastic policy updates in the iteration. Nonetheless, a system revenue with an excessively big clipping ratio tends to converge prematurely and may fall into a local optimal value. Thus, we opt for a clipping ratio of 0.3 in the remaining simulations.

To study the convergence performance of our proposed TDRL-RSUA algorithm more comprehensively, we make a comparison of its convergence with other benchmark algorithms, as shown in Fig. 4(c). The system revenue of the Random strategy fluctuates over a wide range during the training episodes, while the system revenue of the NRM strategy is maintained near a relatively stable value. The system revenues of two AI-based strategies, the TDRL-RSUA and the PPOM algorithms, gradually rise and converge with the increase in training episodes. It is observed that the convergence value of the proposed TDRL-RSUA algorithm is above the other benchmark strategies due to considering branching architectures and action masks. We next compare the variation in slicing resource utilization and SSR of the proposed TDRL-RSUA algorithm with the other benchmark algorithms during the training episodes in Fig. 5. It is seen that the curve of system revenue is influenced by a combination of resource utilization and SSR curves. Due to the greater weighting of SSR, in the early training phase, the MEO satellite controller first trains the network to provide high SSR. Once the SSR is maintained at a high level, the agent starts training for resource utilization improvement. Thus, system revenue is further enhanced in the later stage of training.

Specifically, for the convergence performance of the resource utilization in the different algorithms in Fig. 5(a), the Random algorithm has the lowest and most volatile resource utilization due to the overly random decision making. Although the NRM resource slicing policy allocates resources in proportion to the overall slice resource requirements, the NRM resource allocation does not apply to the real-time scenario as the service requirements and network resources change, resulting in low resource utilization. The resource utilizations of two AI-based strategies gradually rise and converge as the training episodes increase. Meanwhile, the resource utilization of the proposed TDRL-RSUA algorithm is able to converge to a higher value, which can be improved by 32% compared to the resource utilization of the NRM algorithm. For the convergence performance of the SSR

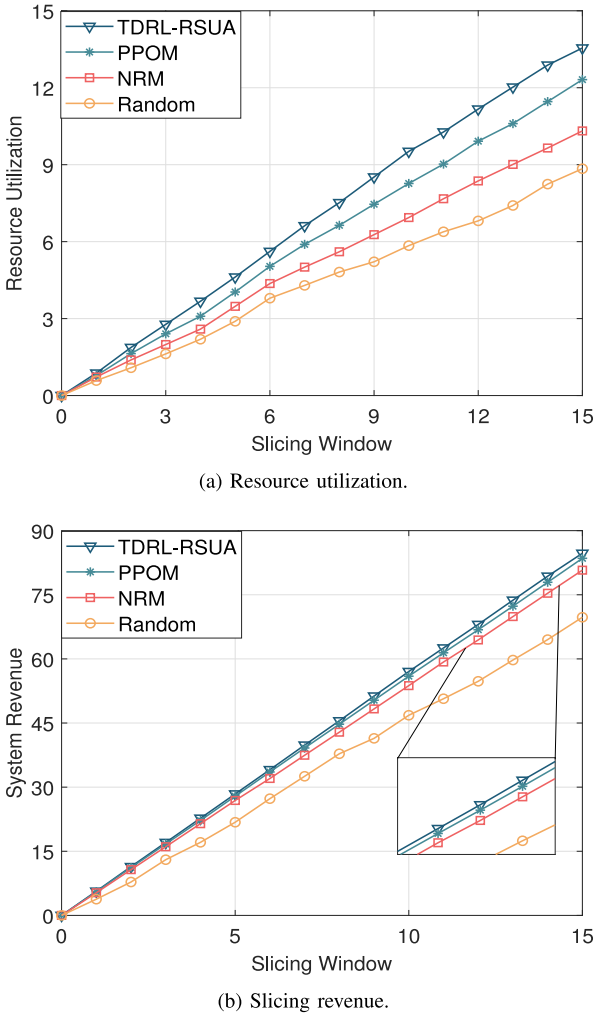


(a) Resource utilization. (b) SSR. (c) Average throughput of satellite.

Fig. 5. Convergence property of the slicing performance in different algorithms.

in the different algorithms in Fig. 5(b), the Random algorithm has the lowest and most volatile SSR as well. The NRM resource slicing strategy achieves high user satisfaction because the resource division is proportional to the demand for the resources. For AI-based algorithms, the SSRs of our proposed algorithm and the PPOM strategy converge to higher values than that of the NRM strategy. This shows that the algorithms guarantee SSR while greatly improving resource utilization. When resource allocation is insufficient, resource utilization and SSR can increase simultaneously with the increase of resource allocation. However, resources may become redundant as resource allocation increases, resulting in decreased resource utilization. Therefore, when configuring slice resources, there needs to be a trade-off between resource utilization and SSR. Our proposed TDRL-RSUA algorithm has a smaller action space than the PPOM algorithm, so it can converge to the resource allocation decision with higher overall system utility. This decision can achieve a favorable SSR close to the PPOM and higher resource utilization, resulting in a better trade-off between resource utilization and SSR. Meanwhile, we also analyze the average throughput of the satellite provided by different algorithms in Fig. 5(c), which is decided by the transmission traffic of all slice users successfully accessed by satellites. Because throughput strongly correlates with SSR, our proposed algorithm can provide higher throughput than the NRM strategy.

2) Slicing Performance Analysis: When the DRL algorithm is well-trained, we assess the cumulative resource utilization over fifteen slicing windows in Fig. 6(a). As expected, the TDRL-RSUA algorithm incurs the highest resource utilization among all the algorithms. Since our proposed algorithm maintains high resource utilization over all slicing windows, the resource utilization accumulates smoothly. The other algorithms have low resource utilization and fluctuate within different windows. We then evaluate the overall system revenue accumulated over fifteen slicing windows in Fig. 6(b). As expected, our TDRL-RSUA algorithm incurs the highest system revenue cost among all the algorithms. As previously analyzed, the TDRL-RSUA, PPOM, and NRM algorithms maintain a high level of SSR throughout the slicing windows. At the same time, the SSR holds more significant weight in

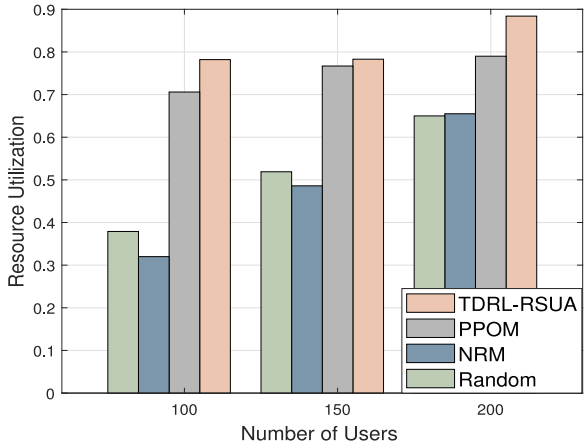


(a) Resource utilization.

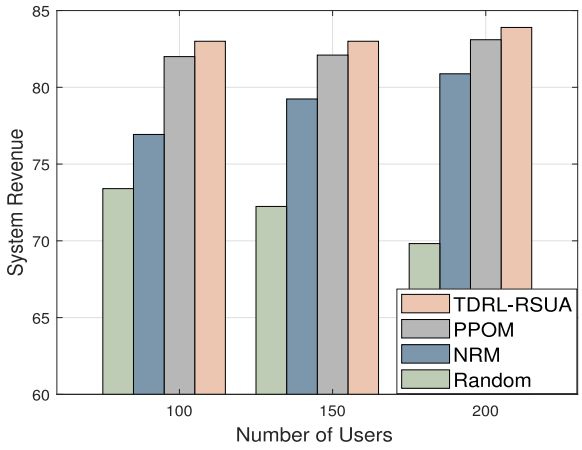
(b) Slicing revenue.

Fig. 6. The cumulative slicing performance in all slicing windows.

the system revenue, so the system utilities steadily increase with the increase of the slicing windows. In contrast, the system revenue of the Random strategy increases instability. Furthermore, the results indicate that the performance improvement is greater as the number of slicing windows increases.

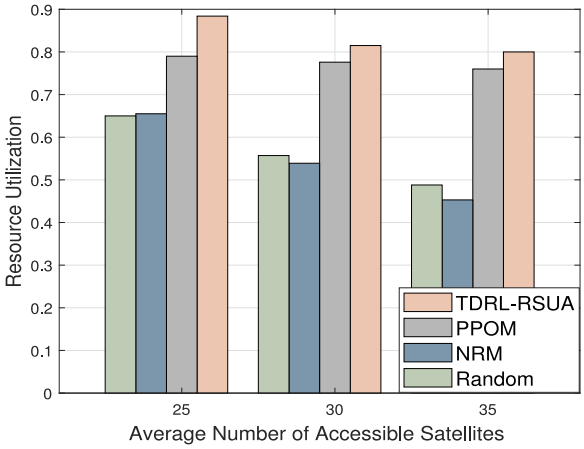


(a) Resource utilization.

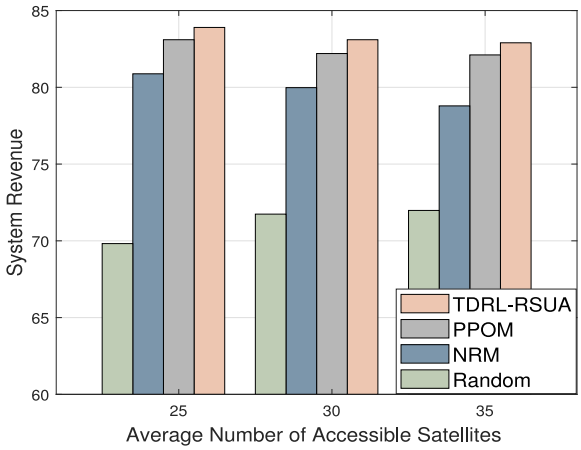


(b) Slicing revenue.

Fig. 7. Slicing performance in the different number of users scenarios.



(a) Resource utilization.



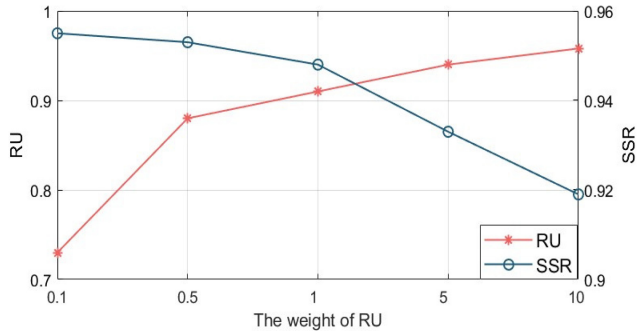
(b) Slicing revenue.

Fig. 8. Slicing performance in the different number of average available satellites scenarios.

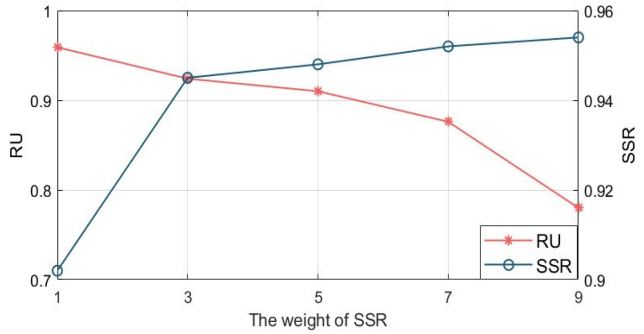
3) Influence of the Number of Users and Available Satellites: To evaluate the influence of the number of users, we simulate our proposed algorithm in several slice user number cases and compare it with the other benchmark algorithms. Fig. 7(a) presents the average resource utilization in all slicing windows. As user density increases, more resources are used, increasing resource utilization for all algorithms. The proposed TDRL-RSUA algorithm achieves the greatest resource utilization in all three scenarios and improves resource utilization by over 30% in comparison to the NRM strategy, especially in scenarios with a small number of users. Fig. 7(b) shows the system revenue of all strategies, and the proposed strategy attains the highest system revenue in all three user-density scenarios as well. Since resource utilization rises with increasing user density, the system revenue of AI-based and NRM strategies increases. However, as the number of users increases, the resource allocation decisions of the Random strategy make more users unsuccessfully served, leading to a decrease in SSR, which further results in a decrease in system revenue.

To evaluate the impact of average available satellites, we adjust the minimum elevation angle of LEO satellites and run all algorithms for several sets of average available satellite scenarios. Fig. 8(a) presents the average resource utilization in all slicing windows. Our proposed algorithm makes slicing decisions based on the currently available satellite resources and achieves the highest resource utilization in multiple resource scenarios. However, the Random and NRM strategies do not adapt well to the changes in satellite resources, resulting in low resource utilization in resource-sufficient scenarios. Fig. 8(b) shows the system revenue of all algorithms, and the proposed algorithm attains the highest system revenue in all three scenarios. As the number of average available satellites increases, the resource utilization decreases, so the system revenue of AI-based and NRM slicing strategies decreases. However, as the number of average available satellites increases, the Random strategy leads to an increase in SSR, which improves the system revenue.

4) Influence of Weights: First, we analyze the effect of resource utilization weight w_{ru} on slicing performance. Specifically, the SSR weight w_{ssr} is set to 5, the reconfiguration weight w_{rc} to 0.01, and the resource utilization weight w_{ru} to different values for multiple simulations. As the resource utilization weight w_{ru} increases, the resource utilization increases while the SSR decreases in Fig. 9(a). When the resource utilization weight is too low, the resource utilization performance decreases significantly. Meanwhile, when the resource utilization weight is too high, the SSR performance drops sharply. Next, we analyze the effect of SSR weight w_{rc} on slicing performance. The resource utilization w_{rc} is set to 1, the reconfiguration weight w_{rc} to 0.01, and the

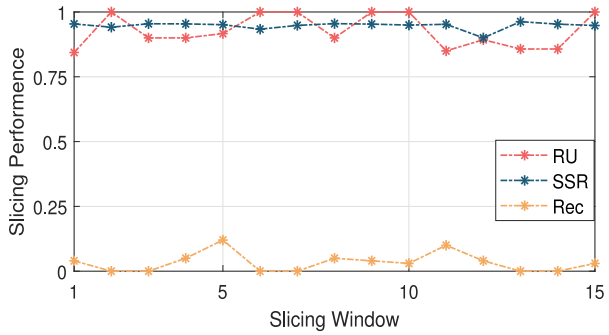


(a) The SSR weight w_{ssr} is 5.

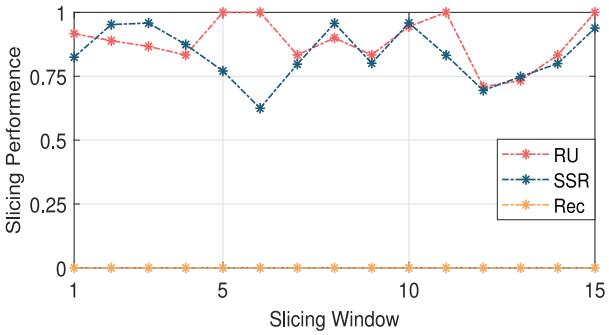


(b) The RU weight w_{ru} is 1.

Fig. 9. Slicing performance in different RU or SSR weights.



(a) The reconfiguration weight w_{rc} is 0.01.



(b) The reconfiguration weight w_{rc} is 1.

Fig. 10. Slicing performance in the different weight of the reconfiguration.

SSR weight w_{ssr} to different values for multiple simulations. Similarly, as the SSR weight w_{ssr} increases, the SSR increases while the resource utilization decreases in Fig. 9(b). When the SSR weight is too low, the SSR performance decreases significantly. Meanwhile, when the SSR weight is too high, the resource utilization performance decreases dramatically. There is a trade-off between resource utilization and SSR by adjusting the weights w_{ru} and w_{ssr}.

To evaluate the impact of the reconfiguration cost weight, we simulate the resource utilization, SSR, and reconfiguration cost in each slicing window under two sets of slice reconfiguration weights. In Fig. 10(a), the reconfiguration weight is set to 0.01. When the reconfiguration weight is small, the slicing resources could be reconfigured flexibly, and the resource utilization and SSR are maintained at a high value in all slicing windows. In Fig. 10(b), the reconfiguration weight is set to 1. Since no cost is incurred for releasing slicing resources, the reconfigurable slicing strategy can promptly release unnecessary resources. However, it fails to increase the required resources in time when the reconfiguration weight is large, which results in a low SSR and fluctuating resource utilization. By comparison, the reconfigurable slicing algorithm with a small reconfiguration weight is more suitable for UD-LSNs with fluctuating service requirements and available LEO satellite resources.

VI. Conclusion

In this paper, we have proposed a reconfigurable RAN slicing architecture for UD-LSNs. We have considered the characteristics of the rate-constrained slice and the delay-constrained slice, and formulated an optimization problem aiming at maximizing the long-term slicing system revenue, including resource utilization, SSR, as well as reconfiguration cost. The original problem has been decoupled into the resource slicing and user access subproblems and resolved by the proposed TDRL-RSUA algorithm. Simulation results have been provided to demonstrate the advantage of our proposed TDRL-RSUA algorithm. This work can provide useful insights for reconfigurable RAN slicing to improve resource utilization in UD-LSNs and other large-scale mobile networks. For the future work, we will consider high dimensional resources slicing in UD-LSNs.

References

[1] C.-X. Wang et al., "On the road to 6G: Visions, requirements, key technologies, and Testbeds," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 905–974, 2nd Quart., 2023.

[2] B. Di, L. Song, Y. Li, and H. V. Poor, "Ultra-dense LEO: Integration of satellite access networks into 5G and beyond," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 62–69, Apr. 2019.

[3] *Non-Geostationary Satellite System*, SpaceX, Washington, DC, USA, 2018.

[4] C. Zou, H. Wang, J. Chang, F. Shao, L. Shang, and G. Li, "Optimal progressive pitch for OneWeb constellation with seamless coverage," *Sensors*, vol. 22, no. 16, p. 6302, 2022.

[5] M. Sheng, D. Zhou, W. Bai, J. Liu, and J. Li, "6G service coverage with mega satellite constellations," *China Commun.*, vol. 19, no. 1, pp. 64–76, 2022.

[6] O. Kodheli et al., "Satellite communications in the new space era: A survey and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 70–109, 1st Quart., 2021.

[7] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2429–2453, 3rd Quart., 2018.

[8] G. Dandachi, A. De Domenico, D. T. Hoang, and D. Niyato, "An artificial intelligence framework for slice deployment and orchestration in 5G networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 858–871, Jun. 2020.

[9] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y.-C. Liang, "Intelligent resource scheduling for 5G radio access network slicing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7691–7703, Aug. 2019.
[10] 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.
[11] J. Feng, Q. Pei, F. R. Yu, X. Chu, J. Du, and L. Zhu, "Dynamic network slicing and resource allocation in mobile edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7863–7878, Jul. 2020.
[12] K. Yu, H. Zhou, Z. Tang, X. Shen, and F. Hou, "Deep reinforcement learning-based RAN slicing for UL/DL decoupled cellular V2X," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3523–3535, May 2022.
[13] 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.
[14] H. Wu, J. Chen, C. Zhou, J. Li, and X. Shen, "Learning-based joint resource slicing and scheduling in space-terrestrial integrated vehicular networks," *J. Commun. Inf. Netw.*, vol. 6, no. 3, pp. 208–223, 2021.
[15] F. Lyu et al., "Service-oriented dynamic resource slicing and optimization for space-air-ground integrated vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7469–7483, Jul. 2022.
[16] T. Ma, B. Qian, X. Qin, X. Liu, H. Zhou, and L. Zhao, "Satellite-terrestrial integrated 6G: An ultra-dense LEO networking management architecture," *IEEE Wireless Commun.*, vol. 31, no. 1, pp. 62–69, Feb. 2024.
[17] X. Qin, T. Ma, Z. Tang, X. Zhang, H. Zhou, and L. Zhao, "Service-aware resource orchestration in ultra-dense LEO satellite-terrestrial integrated 6G: A service function chain approach," *IEEE Trans. Wireless Commun.*, vol. 22, no. 9, pp. 6003–6017, Sep. 2023.
[18] "Technical specification group services and system aspects; telecommunication management; study on management and orchestration of network slicing for next generation network; (Release 15), Version 15.0.0," 3GPP, Sophia Antipolis, France, Rep. TS 28.533, 2018.
[19] 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.
[20] "Technical specification group services and system aspects study on management and orchestration aspects of integrated satellite components in a 5G network; (Release 17)," 3GPP, Sophia Antipolis, France, Rep. 28.808, 2021.
[21] H. H. Esmat, B. Lorenzo, and W. Shi, "Toward resilient network slicing for satellite–terrestrial edge computing IoT," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14621–14645, Aug. 2023.
[22] H. Wu et al., "Resource management in space-air-ground integrated vehicular networks: SDN control and AI algorithm design," *IEEE Wireless Commun.*, vol. 27, no. 6, pp. 52–60, Dec. 2020.
[23] G. Zhou, L. Zhao, G. Zheng, S. Song, J. Zhang, and L. Hanzo, "Multiobjective optimization of space–air–ground-integrated network slicing relying on a pair of central and distributed learning algorithms," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 8327–8344, Mar. 2024.
[24] G. Sun, G. O. Boateng, D. Ayepah-Mensah, G. Liu, and J. Wei, "Autonomous resource slicing for virtualized vehicular networks with D2D communications based on deep reinforcement learning," *IEEE Syst. J.*, vol. 14, no. 4, pp. 4694–4705, Dec. 2020.
[25] B. Qian et al., "Enabling fully-decoupled radio access with elastic resource allocation," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 4, pp. 1025–1040, Aug. 2023.
[26] J. Zhang, H. Wu, X. Tao, and X. Zhang, "Adaptive bitrate video streaming in non-orthogonal multiple access networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 3980–3993, Apr. 2020.
[27] Q. Zhang, H. Wang, Z. Feng, and Z. Han, "Many-to-many matching-theory-based dynamic bandwidth allocation for UAVs," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9995–10009, Jun. 2021.
[28] S. Mahboob and L. Liu, "Revolutionizing future connectivity: A contemporary survey on AI-empowered satellite-based non-terrestrial networks in 6G," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 2, pp. 1279–1321, 2nd Quart., 2024.
[29] J. Xue, K. Yu, T. Zhang, H. Zhou, L. Zhao, and X. Shen, "Cooperative deep reinforcement learning enabled power allocation for packet duplication URLLC in multi-connectivity vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 8, pp. 8143–8157, Aug. 2024.
[30] M. Sherman, S. Shao, X. Sun, and J. Zheng, "Optimizing AoI in UAV-RIS-assisted IoT networks: Off policy versus on policy," *IEEE Internet Things J.*, vol. 10, no. 14, pp. 12401–12415, Jul. 2023.
[31] A. Kanervisto, C. Scheller, and V. Hautamäki, "Action space shaping in deep reinforcement learning," in *Proc. IEEE Conf. Games (CoG)*, 2020, pp. 479–486.
[32] Z. Shan, P. Liu, L. Wang, and Y. Liu, "A cognitive multi-carrier radar for communication interference avoidance via deep reinforcement learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 6, pp. 1561–1578, Dec. 2023.
[33] S. Huang and S. Ontañón, "A closer look at invalid action masking in policy gradient algorithms," 2020, *arXiv:2006.14171*.
[34] X. Liu, T. Ma, Z. Tang, X. Qin, H. Zhou, and X. Shen, "UltraStar: A lightweight simulator of ultra-dense LEO satellite constellation networking for 6G," *IEEE/CAA J. Automatica Sinica*, vol. 10, no. 3, pp. 632–645, Mar. 2023.
[35] H. Jia, C. Jiang, L. Kuang, and J. Lu, "Adaptive access control and resource allocation for random access in NGSO satellite networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 4, pp. 2721–2733, Aug. 2022.
[36] "Geolite2 free geolocation data." MaxMind. Nov. 2021. [Online]. Available: https://dev.maxmind.com/geoip/geoip2/geolite2
[37] B. Tao, M. Masood, I. Gupta, and D. Vasisht, "Transmitting, fast and slow: Scheduling satellite traffic through space and time," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, New York, NY, USA, 2023.
[38] H. Zhang and V. W. S. Wong, "A two-timescale approach for network slicing in C-RAN," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6656–6669, Jun. 2020.
[39] J. Tang, G. Li, D. Bian, and J. Hu, "Traffic prediction based capacity pre-assignment scheme for low latency in LEO satellite communication systems," in *Proc. Int. Conf. Wireless Satell. Syst.*, 2021, pp. 9–20.
[40] X. Zhang et al., "Cybertwin-assisted mode selection in ultra-dense LEO integrated satellite-terrestrial network," *J. Commun. Inf. Netw.*, vol. 7, no. 4, pp. 360–374, 2022.
[41] H. F. Ates, S. M. Hashir, T. Baykas, and B. K. Gunturk, "Path loss exponent and shadowing factor prediction from satellite images using deep learning," *IEEE Access*, vol. 7, pp. 101366–101375, 2019.
[42] R. Deng, B. Di, H. Zhang, and L. Song, "Ultra-dense LEO satellite constellation design for global coverage in terrestrial-satellite networks," in *Proc. IEEE Global Commun. Conf. GLOBECOM*, 2020, pp. 1–6.
[43] Y. Yang, K. Hiltunen, and F. Chernogorov, "On the performance of co-existence between public eMBB and non-public URLLC networks," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC)*, 2021, pp. 1–6.
[44] 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, Aug. 2022.
[45] D. Zhao, H. Qin, N. Xin, and B. Song, "Flexible resource management in high-throughput satellite communication systems: A two-stage machine learning framework," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2724–2739, May 2023.

Yuru Liu (Student Member, IEEE) received the B.S. degree in electronic information science and technology from Central South University, Changsha, China, in 2022. She is currently pursuing the Ph.D. degree in communications and information system with Nanjing University, Nanjing, China. Her research interests include ultra-dense LEO satellite networks, network resource management, network slicing, and deep reinforcement learning.

Ting Ma (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in statistics from Sichuan University, Chengdu, China, in 2013, 2016, and 2020, respectively. From 2020 to 2023, she was a Postdoctoral Fellow with the School of Electronic Science and Engineering, Nanjing University, Nanjing, China. She is currently an Associate Professor with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing. Her current research interests mainly include space-air-ground integrated network, convex optimization theory, robust hypothesis testing, and game theory.

Xiaohan Qin (Student Member, IEEE) received the B.S. degree in communication engineering from Central South University, Changsha, China, in 2020. She is currently pursuing the Ph.D. degree in communications and information system with Nanjing University, Nanjing, China. Her research interests include space-air-ground integrated networks, network resource management, and game theory.

Haibo Zhou (Senior Member, IEEE) received the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2014. From 2014 to 2017, he was a Postdoctoral Fellow with the Broadband Communications Research Group, Department of Electrical and Computer Engineering, University of Waterloo. He is currently a Full Professor with the School of Electronic Science and Engineering, Nanjing University, Nanjing, China. His research interests include resource management and protocol design in B5G/6G networks, vehicular ad hoc networks, and space-air-ground integrated networks. He was a recipient of the 2019 IEEE ComSoc Asia–Pacific Outstanding Young Researcher Award, the 2023 IEEE ComSoc WTC Outstanding Young Researcher Award, the IEEE ComSoc Distinguished Lecturer from 2023 to 2024, and the IEEE VTS Distinguished Lecturer from 2023 to 2025. He served as a Track/Symposium Co-Chair for IEEE/CIC ICCC 2019, IEEE VTC-Fall 2020, IEEE VTC-Fall 2021, WCSP 2022, IEEE GLOBECOM 2022, IEEE ICC 2024, and IEEE GLOBECOM 2024. He is currently an Associate Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK MAGAZINE, and *Journal of Communications and Information Networks*.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is a Registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular networks. He received the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals, ON, Canada, in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society (ComSoc), and the Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for IEEE GLOBECOM' 16, IEEE INFOCOM'14, IEEE VTC'10 Fall, IEEE GLOBECOM' 07, and the Chair for the IEEE ComSoc Technical Committee on Wireless Communications. He is the Past President of the IEEE ComSoc. He was the Vice President for Technical and Educational Activities, the Vice President for Publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and the Member of IEEE Fellow Selection Committee of the ComSoc. He served as an Editor-in-Chief for IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*.