

Multilayer Satellite Network Collaborative Mobile Edge Caching: A GCN-Based Multi-Agent Approach

Yang Jie¹, He Jingchao¹, Cheng Nan^{1,*}, Yin Zhisheng¹, Han Dairu², Zhou Conghao³, Sun Ruijin¹

¹ School of Telecommunications Engineering, Xidian University, Xi'an 710071, China

² School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210000, China

³ Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON N2L 3G1, Canada

* The corresponding author, email: dr.nan.cheng@ieee.org

Cite as: J. Yang, J. He, *et al.*, "Multilayer satellite network collaborative mobile edge caching: A gcn-based multi-agent approach," *China Communications*, 2024, vol. 21, no. 11, pp. 56-74. DOI: 10.23919/JCC.fa.2024-0216.202411

Abstract: With the explosive growth of high-definition video streaming data, a substantial increase in network traffic has ensued. The emergency of mobile edge caching (MEC) can not only alleviate the burden on core network, but also significantly improve user experience. Integrating with the MEC and satellite networks, the network is empowered popular content ubiquitously and seamlessly. Addressing the research gap between multilayer satellite networks and MEC, we study the caching placement problem in this paper. Initially, we introduce a three-layer distributed network caching management architecture designed for efficient and flexible handling of large-scale networks. Considering the constraint on satellite capacity and content propagation delay, the cache placement problem is then formulated and transformed into a markov decision process (MDP), where the content coded caching mechanism is utilized to promote the efficiency of content delivery. Furthermore, a new generic metric, content delivery cost, is proposed to elaborate the performance of caching decision in large-scale networks. Then, we introduce a graph convolutional network (GCN)-based multi-agent advantage actor-critic (A2C) algorithm to optimize the caching decision. Finally, extensive simulations are

conducted to evaluate the proposed algorithm in terms of content delivery cost and transferability.

Keywords: cache placement; coded caching; graph convolutional network (GCN); mobile edge caching (MEC); multilayer satellite network

I. INTRODUCTION

With the surge in Internet of Things (IoT) devices and the evolution of 6G technology, the demand for network traffic has experienced unprecedented growth. Ericsson's forecast indicates that by 2030, global network traffic could increase by 5 to 10 times the current volume. This surge not only imposes higher requirements on bandwidth, but also exacerbates issues related to network latency and energy consumption. To mitigate these issues, mobile edge computing (MEC) technology has been proposed to enhance the service capabilities of cellular networks. By caching popular content at the network edge, MEC alleviates the load on the core network and reduces content retrieval latency, significantly improving the quality of experience (QoE) for users [1]. However, in geographically complex or remote areas, such as oceans, deserts, rural regions, and mountainous zones [2], the placement of traditional edge base stations (BSs) is limited and struggles to meet the expectations of 6G networks for massive connectivity and low-latency communications [3].

Received: Apr. 29, 2024

Revised: Jul. 05, 2024

Editor: Feng Wei

Multilayer satellite networks, particularly those in low Earth orbit (LEO), offer a viable solution for network coverage in remote regions [4, 5]. LEO satellite networks have garnered attention due to their extensive coverage and reduced communication latency. These attributes not only enhance global connectivity, but also support a diverse range of applications, from IoT placements to emergency disaster response. Despite the considerable potential of LEO satellite networks, their utilization in remote regions remains low, attributed to the sparse population and limited demand for caching services. Therefore, it is imperative to fully leverage the caching resources of the global LEO satellite network to balance network load and alleviate traffic congestion. When integrated with MEC technology, LEO satellite networks can ensure continuous and efficient network services even in the most isolated regions [6].

Furthermore, the global demand for data traffic is highly uneven, predominantly concentrated in densely populated hotspot areas. In these zones, traditional base stations are often required to cache popular content at multiple locations due to frequent user requests for the same material, leading to suboptimal utilization of caching resources. LEO satellite networks, by deploying MEC servers on satellites, can cover a much broader area than terrestrial BSs [7], with a single satellite like Starlink providing coverage ranging from 350 to 1100 kilometers, far exceeding the 1 to 3 kilometers range of 4G BSs. Consequently, positioning MEC servers on LEO satellites can offer caching services to a wider user base, significantly enhancing cache efficiency. Additionally, the content demands of users within different LEO satellite coverage areas can be relayed through inter-satellite links (ISLs), facilitating multi-hop transmission. This not only optimizes the use of caching and transmission resources but also reduces the latency of content delivery, fulfilling the requirements for instantaneous communication.

Although LEO satellite networks equipped with MEC servers can expand the range of caching services and enhance resource utilization, there are several new challenges associated with cache placement in such networks [8]. Firstly, the highly dynamic topology of LEO satellite networks results in frequent handovers between users and LEO satellites, leading to the redundant transmission of the same content and consequent consumption of link resources.

Secondly, the limited cache capacity at LEO satellite nodes cannot satisfy all user demands. Thirdly, existing satellite network cache placement optimization algorithms do not consider the unique structural characteristics of satellite networks. Addressing these issues, in our work [9], we propose a centralized caching management architecture centered around geostationary Earth orbit (GEO) satellites. This architecture optimizes the utilization of the entire cache resources of the LEO satellite network by transmitting cached content to other satellites in the network via ISL. Subsequently, we integrated graph convolutional networks (GCN) and reinforcement learning to extract network features and optimize cache placement. While this approach mitigates the caching management challenges in LEO satellite networks to a certain extent and enhances cache efficiency, centralized management still presents limitations for the needs of future large-scale satellite network management. Furthermore, there is a need to further reduce content retransmissions caused by dynamic topology changes.

Consequently, the aim of our paper is to investigate the cache placement issue in dynamic LEO satellite networks. We propose a multilayer distributed cache management architecture to enable autonomy within subnetworks and effective collaboration across them, ensuring the network can adapt to the demands of various application scenarios. Considering the limitations of cache capacity, we further introduce an optimization problem aimed at adjusting the placement of cached content, while also incorporating a content encoding mechanism to enhance transmission efficiency. To overcome the shortcomings of existing algorithms that inadequately account for the dynamic topology of satellite networks, we also propose an efficient optimization algorithm. This algorithm is capable of adapting to the network's dynamic changes, achieving optimal cache decisions, thereby reducing unnecessary data transmission and resource wastage, and accelerating content delivery to users. The main contributions of this paper are summarized as follows:

- We propose a three-layer distributed network cache management architecture that supports software defined network (SDN)/network functions virtualization (NFV). LEO satellites equipped with MEC servers constitute multiple LEO satellite distributed autonomous domains (DADs). Medium Earth orbit (MEO) satellites manage and

coordinate these LEO satellite DADs in a distributed manner, while GEO satellites are aware of the network state and user demands, issuing control information to MEO satellites. This results in an intelligent service model that combines centralized management within DADs and distributed management across them.

- To mitigate the transmission costs under the constraints of limited LEO satellite cache capacity and dynamic network topology, we have formulated an optimization problem for cache placement. This problem is further transformed into a markov decision process (MDP), wherein a content encoding caching mechanism is employed to enhance the efficiency of content transmission. Building upon this, we propose a GCN-based multi-agent advantage actor-critic (A2C) cache placement (GMACP) algorithm, which makes decisions on the placement of popular content under non-standardized and non-uniform distribution of content requests.
- We propose a novel breadth-first search (BFS)-based approach for evaluating cache placement strategies in LEO satellite networks, termed content delivery cost. Extensive simulation results demonstrate that the proposed GMACP algorithm effectively reduces network caching resource consumption compared to other benchmark algorithms. Furthermore, we have investigated the impact of varying numbers of LEO DADs on performance, as well as the transferability of the proposed algorithm, thereby laying a foundation for future research on distributed caching.

The remainder of this paper is organized as follows. Section II provides an overview of related works. Section III describes the system model for the distributed caching network architecture, the content model, and elaborates on the optimization problem of cache placement strategies. In Section IV, we detail the process of modeling the optimization problem as an MDP and the optimization of cache placement strategies using the GMACP algorithm. Numerical results that demonstrate the efficacy of the proposed algorithm are presented in Section V. Finally, the paper is concluded in Section VI.

II. RELATED WORKS

In the context of network management architectures, existing studies have favored the use of hierarchical network structures to cope with the management of complex, dynamic, and large-scale networks. In [10], authors introduce a two-layer spatial information network architecture, where control functions are distributed across various nodes, including GEO satellites, data centers, and satellite gateways, with other network nodes forming the data plane. Similarly, in [11], LEO satellites are grouped and managed in clusters, adopting a distributed network architecture within each cluster. A centralized controller, deployed at the ground station data center, manages the LEO clusters. In [12, 13], authors propose a multi-domain SDN integrated space-terrestrial network architecture based on multi-layer networks, where the management plane consists of GEO satellites equipped with SDN super-controllers, and MEO satellites with standard SDN controllers centrally manage the LEO satellites that form the data plane. In [14], GEO satellites are designated as domain controllers, and extended OpenFlow-enabled LEO (SD-LEO) satellites are divided into different domains according to the coverage area of the GEO satellites. GEO satellites can manage these SD-LEO satellites. In [15], authors provide a novel perspective, suggesting that satellite-unmanned aerial vehicle (UAV) networks can offer on-demand communication services by integrating task-level dependencies and robotic collaborative operations, thereby allocating resources in a task-oriented manner. However, these network management architectures have not fully considered the diverse content requirements and the coordination mechanisms between DADs, leading to inefficient content management schemes. There is an urgent need for a lightweight and efficient intelligent collaborative system architecture.

Due to the limited caching capacity, caching all content on edge nodes is impractical. Hence, the caching strategy under limited resources is indispensable. To satisfy users' QoE requirements, current studies primarily employ conventional learning methods to solve optimization problems. The authors in [16] propose a cooperative transmission scheme in integrated satellite-terrestrial networks (ISTNs) to minimize the average file transmission delay by jointly optimizing

cache placement and content scheduling. In [17], caching placement and power allocation are optimized jointly in LEO satellite-terrestrial networks. In [18], authors investigate the problem of joint cache placement and cooperative multicast beamforming to provide content-centric data services for mobile users in the ISTN. Addressing the constraint of limited cache capacities, [8] formulates the content placement problem to minimize the average content retrieving delay for users. In [19], the authors developed a cooperative caching placement scheme based on a satellite weighted time-varying graph, which caches content proportionally along the data transmission path, effectively enhancing the efficiency of content distribution.

As artificial intelligence (AI) technology matures and becomes more widespread, its potential in solving optimization problems is increasingly evident. The array of advantages that AI methods offer in addressing optimization challenges, such as adaptability to uncertainty, efficient parallel computation, and the capability to handle large-scale datasets, has introduced new perspectives to the research field. Particularly in the context of content caching optimization, these advantages offer researchers novel tools and methods to explore more efficient and intelligent solutions. In [20], a hierarchical deep Q learning (HDQL) algorithm is introduced to minimize long-term overall content delivery delay. In [21], a SDN-based ISTN architecture is proposed by to jointly manage the routing, caching, and computing resources in networks. To address the challenges of low cache efficiency in satellite network nodes, a deep learning-enabled file popularity-aware caching replacement mechanism is proposed in [22]. This mechanism aims to enhance file distribution efficiency within the satellite-integrated content-centric network (SCCN). In [23, 24], the authors employ reinforcement learning (RL) techniques for cache placement to achieve reduced placement latency and maximize the energy efficiency of terrestrial satellite networks. Taking into account the current state of research, the utilization of a learning-based edge caching scheme is viewed as a promising strategy for improving network efficiency [25].

However, given the complexity of large-scale LEO satellite networks, traditional algorithms face limitations in their applicability and computational complexity when dealing with cache placement issues. Additionally, while RL techniques have made cer-

tain advancements in optimizing cache resources, they have not fully accounted for the unique structural information of LEO satellite networks, resulting in diminished performance in dynamic network topologies [26]. For example, conventional network architectures such as convolutional neural networks (CNNs) are primarily tailored for two-dimensional data structures and fail to adequately capture the features of the complex, multi-dimensional structures inherent in LEO satellite networks. By contrast, the advent of graph neural networks (GNNs) provides a novel approach for the feature extraction of multi-dimensional data, which is conducive to adapting to the dynamic variations in satellite network conditions [27–29]. Consequently, this study introduces a multi-agent RL algorithm based on GCNs designed to optimize cache placement within a hybrid centralized and distributed cache management architecture. The objective is to harmonize cache locations with the satellite network topology in order to boost the overall caching efficiency of MEC.

III. SYSTEM MODEL

Figure 1 illustrates the proposed three-layer distributed network caching management architecture that supports SDN/NFV, comprising a cache control plane & intelligent plane, a data plane, and a user plane. MEO and GEO satellites, due to their higher orbital positions, are capable of covering more extensive areas. However, the significant data propagation delay renders them unsuitable for latency-sensitive task transmission, making them more apt to serve as control nodes for regional or global network management. Thus, within our architecture, the cache control & intelligent plane include GEO satellites, MEO satellites, and ground stations, all equipped with SDN controllers [30]. These components utilize network functions provided by SDN to achieve unified management of DADs. In this plane, the GEO satellites are responsible for collecting global state information and controlling the transmission of data within the satellite network. MEO satellites, serving as DAD management nodes, are tasked with forming DAD within the LEO satellite network. These MEO satellites exert centralized control over the LEO satellites within these DADs. Furthermore, for distributed control, the inter-DADs must collaborative interactions

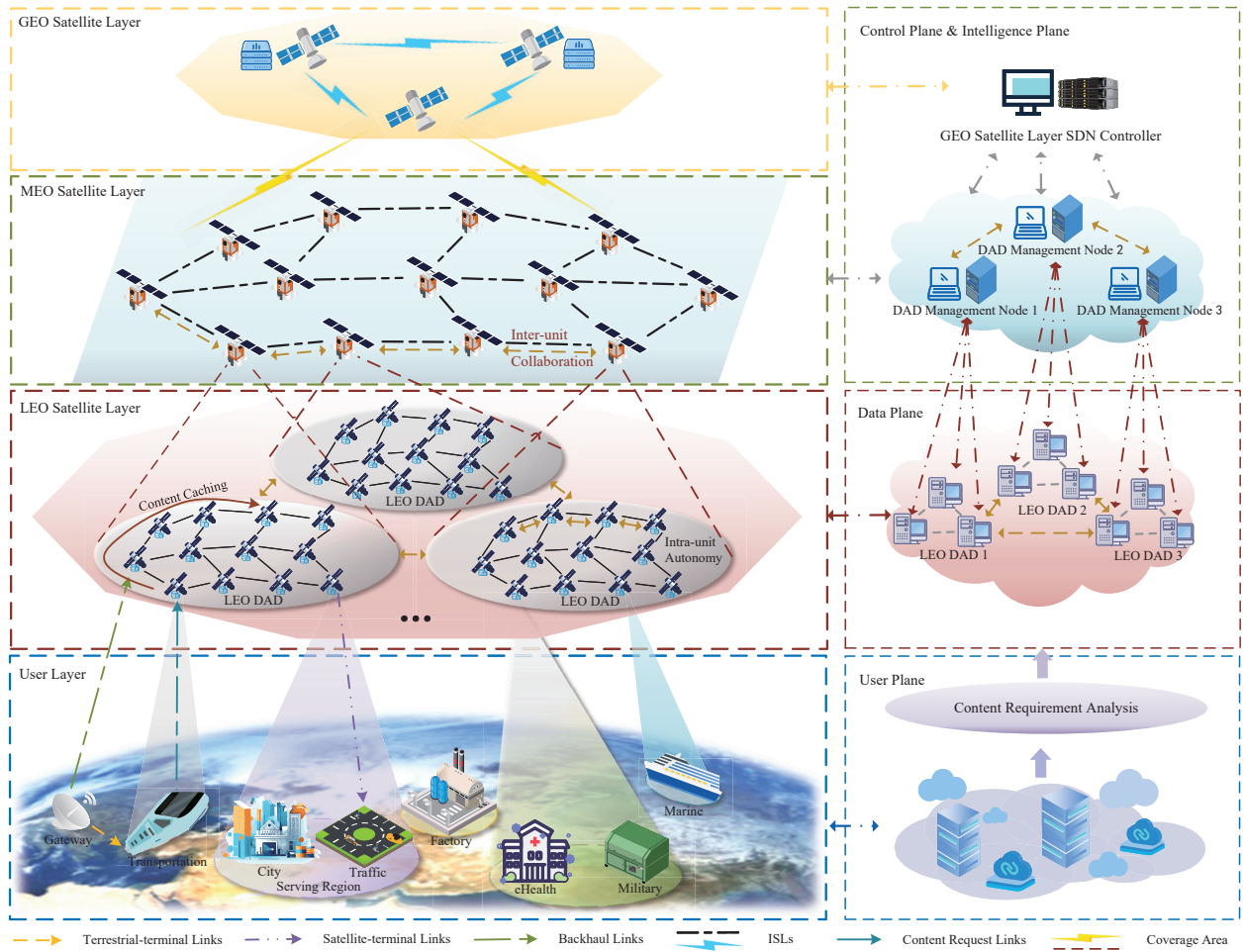


Figure 1. Three-layer distributed network caching management architecture.

through the MEO satellites. The data plane primarily comprises LEO satellites equipped with caching units, which, due to their lower orbits, benefit from low latency and high signal strength, thus enabling them to swiftly respond to user content requests and provide faster and more reliable content distribution. The user plane analyzes the diverse content demands of users. During the entire system operation, the cache control plane & intelligent plane gather network status from LEO satellites, obtain global user demands, and make caching decisions for each satellite. These decisions then guide the placement of content, ensuring it is proactively cached within the LEO satellites. Consequently, when users initiate requests, they can access the required content from the nearest satellite. Some key notations and related descriptions are listed in Table 1.

3.1 Network Model

In our network model, the constellation of satellites consists of $Q = M \times K$ homogeneous satellites, uniformly distributed across M circular LEO orbits [31]. Delving into the specifics, the satellite network topology can be represented by a graph $G = (Q, \mathcal{E})$, where $Q = \{1, 2, \dots, Q\}$ denotes the set of LEO satellites, with Q being the total number of LEO satellites, and $\mathcal{E} = \{e_{(q,q')}^t \mid q \neq q'\}$ is the set of inter-satellite links (ISLs) connecting these satellites. Each LEO satellite DAD is denoted as $\mathcal{D} \subseteq Q$, with the number of DADs set to Y . The number of elements in set $\mathcal{D}$ is equal to the number of LEO satellites in DAD, denoted by D . In terms of satellite configuration, each LEO satellite is equipped with four transceivers, and the ISLs utilize laser links. Of these, two transceivers are dedicated to establishing connections with intra-orbit satellites in the same orbital plane, while the re-

Table 1. Summary of symbols.

Notation	Explanation
M	The number of LEO orbits
K	The number of LEOs per orbit
$\mathcal{Q}$	The set of LEO satellites
$\mathcal{U}$	The set of users
Y	The number of DADs
$\mathcal{D}$	The set of LEO satellite DAD
L_q	The caching capacity of LEO satellites
Γ	The set of SC
T	The total number of SCs
δ	The duration of each SC
$\mathcal{F}$	The catalog of popular content
$\mathcal{M}$	The set comprising F content items
s	The size of original block
C_f	The number of original blocks for content f
$\mathcal{Z}$	The set of coded blocks
Z_f	The total number of coded blocks for content f
$\mathbf{x}$	The caching strategy matrix for satellite q
$x_{q,f,z}^t$	The caching strategy for satellite q
$\mathcal{R}$	The set of content request SR
β	The zipf parameter of content popularity distribution for SR r
$\alpha_{r,f}$	The popularity rank of content f within SR r
$\theta_{r,f}^t$	The probability of content requests from users in SRs at SC t
N_r^t	The number of users in each SR
$\lambda_{\text{total}}^t(f,r)$	The average number of requests per SC t for content f
$\psi_{r,f}^k$	The probability of k requests
$h_{q,f}$	The number of ISL hops
E_{cache}	The total content delivery cost of system
ω	The delay penalty coefficient
$r(t)$	The immediate global gains
$r_{\mathcal{D}y}^t$	The immediate DAD gains
$a_{\mathcal{D}y}^t$	The DAD action space
$s(t)$	The state space
ε	The probability of selecting an action randomly
τ	The discount coefficient for the reward
$V(s(t))$	The average value of the selected action
$Q(s(t), a(t))$	The Q-value corresponding to action

maining pair are used for forming links with the nearest neighboring inter-orbit satellites in adjacent orbital planes. To retain a broad perspective, we proceed under the presumption that users display a preference for accessing the LEO satellite characterized by the shortest line of sight distance. The set of users is denoted as $\mathcal{U} = \{1, 2, \dots, U\}$. Given the extensive coverage area of LEO satellites, we assume that there is an overlap in the regions covered by LEO satellites. Consequently, users in each region have access to more than one satellite. The caching capacity of LEO satellites is limited and is represented by L_q , where L_q denotes the number of cacheable content blocks per satellite.

To optimize the content distribution efficiency in LEO satellite networks and address the issue of fre-

quent service handovers caused by the rapid movement of satellites, we assume that the content requested by users is not only cached on the LEO satellite providing the service but can also be stored on any satellite node connected through ISLs. In our system, we maintain a fixed caching placement within each service cycle (SC). We denote the set of SCs as $\Gamma = \{1, 2, \dots, T\}$, where T represents the total number of SCs, and δ denotes the duration of each SC. At the beginning of the SC, we construct the network topology G based on the current ISL status $e_{(q,q')}^t$ at SC t , and accordingly make decisions about cache placement and deploy the cached content. This content remains static throughout the SC. With the end of the SC, we construct a new network topology G_- based on the ISL status $e_{(q,q')}^{t-}$ at the next SC t_- , to simulate the changes in network topology induced by satellite movement.

3.2 Content Coded Caching Model

Coded caching technology is crucial in ensuring the reliability and efficiency of content storage. By introducing coding redundancy, it allows the storage system to guarantee the recovery of original content, even amidst the adversity of partial node failures. In our framework, we assume the implementation of coded caching in satellite is facilitated through the use of fountain codes. We define the catalog of popular content as $\mathcal{F} = \{1, 2, \dots, F\}$, where F is the total number of popular content. We denote $\mathcal{M} = \{M_1, M_2, \dots, M_F\}$ as a set comprising F content items, each of varying size, quantified in Mbits. Each content item within the set is segmented into equal-sized original blocks s . Consequently, each content item can be partitioned into $C_f = \left\lceil \frac{M_f}{s} \right\rceil$ original blocks, where if the size of the final block is less than s , padding is applied to extend the block to the length of s . The original blocks are encoded into coded blocks for caching [32]. We define the set of coded blocks for each content f as $\mathcal{Z} = \{Z_1, Z_2, \dots, Z_f\}$, where Z_f represents the total number of coded blocks for content f . For each content f , the number of coded blocks must satisfy $Z_f \geq C_f$. Additionally, it is possible to configure an increased number of coded blocks to provide supplementary redundancy and fault tolerance. In accordance with coded caching technology, these coded blocks are distributed and cached across

multiple nodes within the satellite network. When users require access to the original content f , it can be retrieved by decoding operations using any combination of $\geq C_f$ coded blocks from the nodes within the satellite network [33]. We denote the caching strategy matrix for LEO satellite q as $\mathbf{x} = [x_{q,f,z}^t]^{Q \times F \times Z}$, where $x_{q,f,z}^t \in \{0, 1\}$ indicates whether LEO satellite q has chosen to cache coded block z of content f , and $x_{q,f,z}^t = 1$ means block z of content f is cached on satellite n ; otherwise $x_{q,f,z}^t = 0$. It is required that each LEO satellite adheres to the following constraint

$$0 \leq \sum_{z \in \mathcal{Z}} x_{q,f,z}^t \leq Z_f, \quad \forall q \in \mathcal{Q}, \forall f \in \mathcal{F}, \quad (1)$$

which ensures that on LEO satellite q , caching resources are not expended to cache more encoded blocks than the total number Z_f of content f . Additionally, the number of cache coded blocks stored in a LEO satellite's cache does not exceed the corresponding cache capacity, which is expressed as

$$\sum_{f \in \mathcal{F}} \sum_{z \in \mathcal{Z}} x_{q,f,z}^t \leq L_q, \quad \forall q \in \mathcal{Q}. \quad (2)$$

3.3 Content Request Popularity Model

Video and multimedia content, as the primary constituents of cached content, now account for a major portion of network traffic. Given their relatively static nature and infrequent updates, it is feasible to presume that the popularity distribution of the cached content is time-invariant over a SC t . However, the dynamism of LEO satellites alters the correspondence between users and the satellites they access. Consequently, for LEO satellites, the cache content changes over the course of the SC. Some new popular content may need to be cached, while some content that is no longer popular may need to be replaced. We are able to effectively predict the popularity distribution of the cached content. In this paper, we represent the set of content request service regions (SRs) with $r \in \mathcal{R} = \{1, 2, \dots, R\}$, where r denotes the SR identifier. Acknowledging the variance in content preferences across different SRs, we introduce a generic content request popularity model that emulates the popularity of video content within actual network

contexts. This model employs the zipf distribution to approximate the probability of content requests from users in SRs at SC t [34], which is denoted as

$$\theta_{r,f}^t = \frac{(\alpha_{r,f})^{-\beta}}{\sum_{i \in \mathcal{F}} (\alpha_{r,i})^{-\beta}}, \quad \forall f \in \mathcal{F}, \quad \forall r \in \mathcal{R}, \quad (3)$$

where β is the zipf parameter of content popularity distribution for SR r , typically ranging between 0.5 and 2, with larger values indicating a higher concentration of requests. The term $\alpha_{r,f}$ denotes the popularity rank of content f within SR r . For instance, $\alpha_{r,f} = 1$ represents the content f is the most popular content in SR r . It is important to note that the sum of the probabilities of requesting contents within each SR equals one [35], which is denoted as

$$\sum_{f \in \mathcal{F}} \theta_{r,f}^t = 1, \quad \forall r \in \mathcal{R}. \quad (4)$$

In this paper, we assume that users within the same SR exhibit identical content request probabilities for content f , which can be represented as $\theta_f^t = \theta_{r,f}^t$.

We employ a poisson process to model content request events by users within the LEO satellite network. Specifically, each user's content requests are considered mutually independent and follow a zipf distribution, while the request events themselves are characterized by a poisson distribution. Taking into account the varying population densities across different SR, we assume that there are N_r^t users u , within the satellite coverage SR at time t . Consequently, the average number of requests per SC for a specific content f by all users within the coverage SR can be expressed as

$$\lambda_{\text{total}(f,r)}^t = N_r^t \cdot \theta_{r,f}^t. \quad (5)$$

Subsequently, for all users within the coverage SR, we can employ the probability mass function of the Poisson distribution to calculate the probability of k requests being made for a specific content f at time t , denoted as

$$\psi_{r,f}^k = \frac{\left(\lambda_{\text{total}(f,r)}^t\right)^k e^{-\lambda_{\text{total}(f,r)}^t}}{k!}. \quad (6)$$

Therefore, within a specific SR r , the probability that a LEO satellite receives at least one request for

content f during a given SC t can be represented as $1 - \psi_{r,f}^0$.

3.4 Delay Model

Given that satellites are equipped with high-speed laser links, the transmission delay for content can be considered negligible. When content is cached on non-access satellite, the forwarding link includes multiple hops through ISLs [36]. Hence, in this context, we only take into account the propagation delay incurred by content forwarding via ISLs [37]. The formula is as

$$T_p = x_{q,f,z}^t \cdot h_{q,f} \cdot \frac{d_e}{c}, \quad \forall q \in \mathcal{Q}, f \in \mathcal{F}, r \in \mathcal{R}, z \in \mathcal{Z}, \quad (7)$$

where, d_e denotes the distance of the ISL, c represents the speed of light, which is valued at 3×10^5 km/s, and $h_{q,f}$ is the number of ISL hops required for the caching content.

3.5 Problem Formulation

In this paper, we consider a proactive caching strategy, which involves caching content during off-peak periods based on content popularity. Considering the limitations of satellite caching abilities, it is impractical to store all contents within each individual satellite. To fulfill users' content requests, contents must be transferred via ISL, thereby generating additional resource consumption.

To minimize the total consumption of content transmission, we have optimized our caching strategy. However, content delivery remains a significant and complex research area. Hence, we introduce a mechanism based on BFS to estimate the expected transmission outcomes, hereafter referred to as content delivery cost. The hierarchical search characteristic of BFS is particularly well-suited for traversing the graph structure of satellite networks. Within such networks, BFS can initiate from a starting node, collating information from all reachable satellite nodes, thereby forming a connectivity path graph that determines the shortest path between any two satellite nodes. Content delivery cost is subsequently evaluated based on the shortest paths determined by BFS, in conjunction with the probability of user requests and the caching

strategy $\mathbf{x}$. The formula for content delivery cost can be expressed as

$$E_{\text{cache}} = \sum_{q \in \mathcal{Q}} \sum_{f \in \mathcal{F}} \sum_{r \in \mathcal{R}} \sum_{z \in \mathcal{Z}} (1 - \psi_{r,f}^0) \cdot x_{q,f,z}^t \cdot h_{q,f}. \quad (8)$$

Taking into account constraints such as the capacity of LEO satellites and the delay in content propagation, we propose an optimization problem for content cache placement. The problem is formulated as

$$\begin{aligned} & \min_{\mathbf{x}} E_{\text{cache}}, \\ \text{s.t. } & C1 : T_p < T_{\max}, \quad \forall q \in \mathcal{Q}, f \in \mathcal{F}, r \in \mathcal{R}, z \in \mathcal{Z}, \\ & C2 : 0 \leq \sum_{z \in \mathcal{Z}} x_{q,f,z}^t \leq Z_f, \quad \forall q \in \mathcal{Q}, \forall f \in \mathcal{F}, \\ & C3 : \sum_{f \in \mathcal{F}} \sum_{z \in \mathcal{Z}} x_{q,f,z}^t \leq L_q, \quad \forall q \in \mathcal{Q}, \\ & C4 : Z_f \geq C_f, \quad \forall f \in \mathcal{F}, \\ & C5 : x_{q,f,z}^t \in \{0, 1\}, \quad \forall q \in \mathcal{Q}, \forall f \in \mathcal{F}, \forall z \in \mathcal{Z}, \\ & C6 : \sum_{f \in \mathcal{F}} \theta_{r,f}^t = 1, \quad \forall r \in \mathcal{R}, \end{aligned} \quad (9)$$

where $C1$ denotes the constraint that the time required to cache each encoded block should not exceed the maximum delay threshold, $C2$ stipulates that the number of blocks of the same content cached by a satellite cannot exceed the total number of encoding blocks of that content, $C3$ dictates that the cache capacity used by a satellite for storing contents must not surpass the upper limit of the satellite's cache capacity, $C4$ asserts that the number of encoding blocks of a content must be greater than or equal to the number of the content's original blocks, $C5$ ensures that the caching strategy for satellite q , and $C6$ indicates that the sum of content request probabilities from users in each SR equals one. In order to minimize the consumption of system caching resources, the system must, while meeting the specified constraints, endeavor to cache content on MEC servers aboard LEO satellites that are in closer proximity to the users, in accordance with the probability of content demand.

IV. CACHING PLACEMENT ALGORITHM

In this section, we introduce a GCN-based multi-agent A2C caching placement algorithm, referred to as the

GMACP algorithm, designed to address the optimization problem Eq. (9). Initially, to capture the dynamic transitions between network states during each operation period, We have transformed the optimization problem into a MDP [38]. Subsequently, we provide a detailed description of our CMACP algorithm.

4.1 Modeling Caching Strategy Problem as the MDP

In the dynamically varying heterogeneous satellite networks, real-time optimization of MEC strategies is crucial for maintaining network performance. We formalize the caching decision problem as a MDP to better adapt to the dynamic evolution of network states. Within the proposed MDP framework, the state space S encompasses the real-time status information of the LEO satellite network and the instantaneous demands of users for contents. The action space A delineates a set of potential caching strategies that the network might currently undertake. The reward function R , centered on content delivery cost, provides quantified feedback for each state transition and its corresponding action, assessing their alignment with the established optimization objectives [39]. To address the

dynamic nature of satellite networks, we further integrate an analysis of heterogeneous network topologies across different SCs, aiming to enhance the foresight and adaptability of caching decisions. The details are as follows.

4.1.1 State Definition

The state space represents the various states within the caching decision process. The state space $s(t)$ encompasses the ISL connectivity status under the current network topology as well as the characteristics of the satellite nodes. The ISL connectivity status can be derived from a satellite constellation simulator, denoted as a set $\{e(q, q')\}^{Q \times 2}$, which is used to construct the topology graph; hence, it is not further described within the state space matrix. The node characteristics of each LEO satellite are defined by the probability of user requests and the cache strategy of the satellite nodes. Therefore, the entire system's state space $s(t)$ at SC t is represented as

$$s(t) = [V_1 \ V_2 \ \cdots \ V_Z], \quad (10)$$

where the matrix V_z is defined as

$$V_z = \begin{bmatrix} 1 - \psi_{1,1}^0 & \cdots & 1 - \psi_{1,F}^0 & x_{1,1,z}^t & \cdots & x_{1,F,z}^t \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 - \psi_{R,1}^0 & \cdots & 1 - \psi_{R,f}^0 & x_{Q,1,z}^t & \cdots & x_{Q,F,z}^t \end{bmatrix}. \quad (11)$$

The state of the satellite network environment $s(t)$ is input into a GCN to extract features, yielding $\bar{s}(t)$. Subsequently, each MEO satellite management node makes action selections based on the features $\bar{s}_{\mathcal{D}_y}^t$ extracted from the LEO satellite's DAD.

4.1.2 Action Definition:

The action space consists of the caching decisions for all LEO satellites within the caching management framework. In this architecture, the MEO satellites within the caching control & intelligence plane are required to make content caching placement decisions based on the current SC t state $\bar{s}_{\mathcal{D}_y}^t$, while also adhering to the capacity constraints of the LEO satellites. Therefore, different LEO satellite DADs correspond to different sizes of action space. After each MEO satel-

lite makes a caching decision, it issues control commands to the LEO satellite DADs in the data plane to execute content forwarding. Therefore, the action space for each DAD at SC t is represented as

$$a_{\mathcal{D}_y}^t = [A_1 \ A_2 \ \cdots \ A_D], \quad (12)$$

where the matrix A_d is defined as

$$A_d = \begin{bmatrix} x_{d,1,1}^t & \cdots & x_{d,1,Z}^t \\ \vdots & \ddots & \vdots \\ x_{d,F,1}^t & \cdots & x_{d,F,Z}^t \end{bmatrix}. \quad (13)$$

The aggregation of the action spaces $a_{\mathcal{D}_y}^t$ for each DAD yields the action space for the entire system, which is denoted as

$$a(t) = [a_{\mathcal{D}_1}^t \ a_{\mathcal{D}_2}^t \ \cdots \ a_{\mathcal{D}_Y}^t]. \quad (14)$$

To prevent the algorithm from converging to a local optimum, we employ an ε -greedy strategy to balance exploration and exploitation. With a probability of ε , an action at SC t is selected randomly, whereas with a probability of $1 - \varepsilon$, the action corresponding to the maximum value from the probability distribution output by the actor network is chosen. This can be formally expressed as

$$a_{\mathcal{D}^y}^t = \begin{cases} \argmax P_{\mathcal{D}^y}^t, & 1 - \varepsilon, \\ \text{random action}, & \varepsilon, \end{cases} \quad (15)$$

where ε represents the probability of selecting an action randomly, and it is a dynamic decay value.

$$r(t) = \begin{cases} -\sum_{q \in \mathcal{Q}} \sum_{f \in \mathcal{F}} \sum_{r \in \mathcal{R}} \sum_{z \in \mathcal{Z}} (1 - \psi_{r,f}^0) \cdot x_{q,f,z}^t \cdot h_{q,f}, & T_p < T_{max} \\ -\omega \sum_{q \in \mathcal{Q}} \sum_{f \in \mathcal{F}} \sum_{r \in \mathcal{R}} \sum_{z \in \mathcal{Z}} (1 - \psi_{r,f}^0) \cdot x_{q,f,z}^t \cdot h_{q,f}, & T_p \geq T_{max} \end{cases} \quad (16)$$

4.2 GCN-Based Multi-Agent A2C Caching Placement Algorithm

In this subsection, we propose a GMACP algorithm aimed at optimizing the caching placement decisions for the LEO satellite network. The satellite network graph possesses not only node features but also complex structural characteristics. Traditional neural networks such as convolutional neural networks (CNNs) excel at extracting features from two-dimensional images, yet they struggle to capture the structural features of the LEO satellite network graph. In contrast, GCNs can directly handle graph-structured data and effectively learn node features and inter-node relationships, thereby enhancing the algorithm's transferability. With the aid of the deep graph library (DGL) framework, GCNs can automatically handle the normalization process for adjacency matrices and features. We utilize the output from the GCN for A2C neural network training. The A2C network introduces an advantage function to address the high variance issue encountered when calculating gradients in value-based methods, leading to more stable learning and improved convergence characteristics. In our model, the satellite network comprises multiple LEO DADs, and each MEO satellite node, acting as an agent, makes decisions without relying on a central controller for centralized control. Therefore, we em-

4.1.3 Reward Definition

At SC t , each LEO satellite caches content based on the collective caching decision $a(t)$ made by all MEO satellites. The feedback received by each LEO satellite upon the execution of this decision is defined as the reward. Given the limited caching capabilities of LEO satellites, it is impractical to cache all contents on every satellite. Thus, LEO satellites need to collaborate on caching content to optimize content delivery cost. However, during the content delivery process, the propagation delay T_p incurred by ISL transmission must not exceed the maximum allowable propagation delay T_{max} . Therefore, the reward for the entire system at SC t can be formalized as

ploy the GMACP algorithm to optimize caching decisions.

Our algorithm employs a centralized training but decentralized execution strategy. During the training phase, MEO satellite agents have access to global information and leverage GCNs to process this information, thereby learning how to make caching placement decisions based on the global state of the network. During the execution phase, each MEO satellite agent makes decisions independently based on its local observations [40]. This approach enables our algorithm to utilize global information to optimize caching decisions while maintaining efficiency and transferability during execution.

4.2.1 GCN-Based Network Feature Extraction

In our architecture, the initial state features of the system are denoted as $s(t)$. To extract high-level representations of these features, we have constructed a GCN composed of two layers. The objective of the first layer of the GCN is to map the raw features into a higher-dimensional feature space, revealing the local structural properties of the nodes. Subsequently, the second layer of the GCN refines these features to capture more extensive neighborhood information, leveraging data from multi-hop neighboring nodes.

On the implementation front, we employ the DGL

framework, which automates the normalization process for adjacency matrices and features, significantly reducing programming complexity [41]. With this approach, our model can effectively learn useful information from the initial state features and apply it to guide the caching placement strategies of the MEO satellite agents. The entire process of feature extraction and transformation can be summarized by the following formulation

$$\bar{s}(t) = \text{GCNConv}(\sigma(\text{GCNConv}(s(t), F_1)), F_2), \quad (17)$$

where $\text{GCNConv}(\cdot)$ represents the graph convolutional layer implemented in the DGL, and $\sigma(\cdot)$ denotes a nonlinear activation function. In this paper, the ReLU function is employed, defined as $\sigma(x) = \max(0, x)$. An activation function is applied following the first GCN layer to introduce nonlinearity, while the output of the second GCN layer is used directly as the extracted feature $\bar{s}(t)$. Within the GCN framework, state features of a batch size number of LEO satellite network graphs are extracted concurrently to match the training with multiple synchronous workers constructed by the A2C network.

4.2.2 Multi-Agent A2C Caching Placement Algorithm

In our multi-agent A2C framework, each MEO agent is equipped with an actor network responsible for generating caching decision actions based on the local state $\bar{s}_{\mathcal{D}_y}^t$ extracted by the GCN. The actor network is implemented using a multi-layer perceptron (MLP), which analyzes the input state and outputs a probability distribution representing the likelihood of different caching decision actions [40]. During this process, the objective of the actor network is to learn a policy that, given the current local state $\bar{s}_{\mathcal{D}_y}^t$, can guide the agent to make caching decisions $a_{\mathcal{D}_y}^t$ that are most beneficial for the overall network performance.

The architecture of the actor network includes multiple hidden layers, which are interconnected through nonlinear activation functions to enhance the network's capacity to capture complex relationships. The structure of the actor network is as

$$P_{\mathcal{D}_y}^{(t)} = \text{Softmax}(\text{MLP}(\bar{s}_{\mathcal{D}_y}^t)), \quad (18)$$

where, Softmax ensures that the sum of the output action probability distribution equals one. Based on the probability distribution $P_{\mathcal{D}_y}^t$ output by the actor network of each MEO agent, we employ an ε -greedy strategy to determine the global action $a(t)$ for each satellite network graph. Subsequently, for every satellite network graph, the global reward $r(t)$ is computed according to the global decision action $a(t)$.

In such a centralized control and distributed execution framework of a multi-agent A2C network, our critic network employs a global evaluation, taking into account the joint state and actions of all agents, with the network architecture consisting of three layers. The output of the critic network is no longer the state value of a single agent but rather an estimate of the value of the entire system's state. Consequently, the output $V(\bar{s}(t))$ from the critic network, given the current state $\bar{s}(t)$, is utilized as a substitute for the baseline. The Q-value, $Q(\bar{s}(t), a(t))$, can be represented using the value function $V(\bar{s}(t))$ output by a target network with an identical architecture for the subsequent state $\bar{s}(t_-)$ [21]. Thus, the advantage function can be articulated as

$$\delta(\bar{s}(t), a(t), \bar{s}(t_-)) = V(\bar{s}(t)) - (\tau \cdot V(\bar{s}(t_-)) + r(t)), \quad (19)$$

where τ denotes the discount factor for future rewards. Eq. (19) also delineates the TD-error within the A2C network framework. Under the global evaluation mechanism, the TD-error of the critic network is based on the global reward $r(t)$, which reflects the decision-making performance of the entire system. Consequently, the critic network is updated through the value of the TD-error [42]. The loss function for the Critic network is expressed as

$$L_{\text{critic}} = \frac{1}{B} \sum_{i=1}^B (\delta(\bar{s}(t), a(t), \bar{s}(t_-))_i)^2, \quad (20)$$

where B denotes the batch size. Within our multi-agent A2C network framework, the policy update for the actor network takes into account the state $\bar{s}(t)$ and action $a(t)$ information from all MEO agents. During the training phase, the policy update of the actor network utilizes global information to better learn collaborative behaviors in a multi-agent setting. This implies that the policy gradient update is based on a global

policy estimate as well as the approximation of the advantage function [43]. Consequently, the loss function for the actor network is expressed as

$$L_{\text{actor}}^y = \frac{1}{B} \sum_{i=1}^B \log(\pi(a(t)|\bar{s}(t))_i) \cdot \delta(\bar{s}(t), a(t), \bar{s}(t_-))_i. \quad (21)$$

During the backpropagation phase of model training, gradients are computed via loss functions, and the Adam optimization algorithm is employed to iteratively update the parameters of the actor and critic networks. The objective is to solve for the optimal caching placement strategy, thereby reducing the content delivery costs. Upon entering the execution phase, the trained parameters of the GCN and the actor network are replicated across each MEO agent. These MEO agents operate independently, making caching decisions based solely on the local state information from their respective LEO satellite DADs, without the need to access the state data of other DADs. This approach not only enhances decision-making efficiency but also reduces the communication load on the system, representing an effective strategy for efficient cache management in complex dynamic network environments.

Algorithm 1 delineates the detailed training procedure of our proposed GMACP algorithm. Specifically, for each LEO satellite network graph, we commence by initializing the parameters of the A2C networks, θ_{actor} , θ_{critic} , θ_{target} , as well as the parameters of the GCN, θ_w . Subsequently, based on the probability of user demand and the characteristic values of cached encoding blocks, we acquire the LEO satellite node information, $s(t)$. Thereafter, $s(t)$ is input into the GCN to extract state features, yielding global features, $\bar{s}(t)$, and local features for each DAD, $\bar{s}_{\mathcal{D}_y}^t$. For each LEO satellite DAD, the extracted state features $\bar{s}_{\mathcal{D}_y}^t$ are input into the actor network to determine the action $a_{\mathcal{D}_y}^{(t)}$. The actions from each agent, $a_{\mathcal{D}_y}^t$, are amalgamated to form the global action $a(t)$. This global action $a(t)$ interacts with the environment to obtain the next state information of the LEO satellite node $s(t_-)$ and the corresponding global reward $r(t)$. Subsequently, the global loss function is computed, and the network parameters are updated accordingly. This process is iterated for each epoch, culminating in the attainment of the globally optimal cache placement for the LEO

Algorithm 1. GMACP algorithm training procedure.

Input: Initialize $\alpha_{r,f}$, β , Q , learning rate, discount factor γ , GCN network parameter θ_w , critic network parameter θ_{critic} , actor network parameter θ_{actor} , target network parameter θ_{target} , target replace iter.

Output: The globally optimal cache placement for the LEO satellite network.

```

1: for  $i$  in  $epoch$  do
2:    $time \leftarrow 0$ .
3:   while True do
4:     for  $b$  in batch size  $B$  do
5:       Reset the LEO satellite environment to get  $s(t)$ .
6:       The state is input into the GCN network to obtain the global state feature  $\bar{s}(t)$  and DAD state feature  $\bar{s}_{\mathcal{D}_y}^t$ .
7:       for  $y$  in  $Y$  do
8:         each DAD state features  $\bar{s}_{\mathcal{D}_y}^t$  are input into the actor network and exploration to obtain the action  $a_{\mathcal{D}_y}^t$ .
9:       end for
10:      Obtain immediate global reward  $r(t)$  and next state  $s(t_-)$ .
11:      The critic network gets  $V(\bar{s}(t))$  through  $\bar{s}(t)$ .
12:      The target network gets  $V(\bar{s}(t_-))$  through  $\bar{s}(t_-)$ .
13:      Calculate the global TD-error by (19).
14:    end for
15:    Update critic network by minimizing the loss (20).
16:    Update actor network by minimizing the loss (21).
17:     $time \leftarrow time + 1$ .
18:    if  $k \% target\_replace\_iter == 0$  then
19:      Update target network parameters  $\theta_{\text{target}} \leftarrow \theta_{\text{critic}}$ .
20:    end if
21:    if  $time == max\_time$  then
22:      break
23:    end if
24:  end while
25: end for

```

satellite network.

V. SIMULATION RESULTS

In this section, we first introduce our simulation environment. We then evaluate the effectiveness of the GMACP algorithm's caching decisions under various environmental parameters by comparing it with several benchmark algorithms. Finally, we illustrate the performance fluctuations of the GMACP algorithm with respect to the scaling of LEO satellite DAD and also examine the transferability of the CMACP and GCP algorithms.

5.1 Simulation Setup

The simulations were conducted on a computer equipped with an NVIDIA GeForce RTX 3060 Ti GPU and an Intel Core i5-12400F CPU operating at 2.5 GHz.

Our algorithm is applicable to large-scale satellite networks. However, to reduce the complexity of the simulation, we referenced [8] and considered a Walker Star LEO satellite constellation with a satellite count of $Q = 100$. The LEO satellites have an orbital altitude of 2000 km to support a broad coverage area. We selected six ground SRs and simulated the interaction of 100 satellites with these SRs using a satellite constellation simulator. The actual location data of these SRs were set as the positions of each user [39]. Each SR is served by two access LEO satellites, and the number of users N_r^t in each SR ranges from 2 to 6. Users in SRs at SC t exhibit unique probabilities of content requests. Specifically, for each SR r , the Zipf distribution parameter β for content popularity and the popularity rank $\alpha_{r,f}$ for content f differ [34]. In our simulation, the total number of cache coded blocks for all content is at least 48, while the maximum number of cache coded blocks for a single LEO satellite is 11. Consequently, the content demanded by some users needs to be transmitted through the network via ISLs.

Additionally, in Algorithm 1, the feature dimensions of the two-layer GCN are $F_1 = 64$ and $F_2 = 32$, which can capture the characteristics of the nodes themselves as well as the information of their first and second-order neighbors. The actor network includes a three-layer MLP, with hidden layer feature dimensions of (128, 64). Each layer undergoes a non-linear transformation via the ReLU activation function. We

define a training iteration with $episodes = 600$. The environment is randomly reset at the beginning of each iteration.

Due to high dynamism, the maximum service time between a user and a LEO satellite at an altitude of 2000 km is approximately 17 minutes. Moreover, different types of services have varying impacts on the frequency of content popularity updates. A higher update frequency can increase network load. Conversely, a lower update frequency, such as updating the cache once per second, may prevent users from accessing the latest and most popular content and significantly increase the complexity of the simulation. Therefore, to simplify the simulation and take into account the aforementioned factors, we set the duration of a SC to $\delta = 100s$. Each SC corresponds to a different network topology. Consequently, the training of our GMACP algorithm considers changes in network topology across multiple SCs, thereby more effectively adapting to network dynamics. During execution, for long-term cached content, we can also configure cache placement to occur once over several SCs. Table 2 displays the main simulation parameters.

We evaluate the performance and compare it with two benchmarks as follows:

- The first benchmark we utilized is the MLP. MLP is a type of feedforward artificial neural network that, through its multiple fully connected layers and nonlinear activation functions, is capable of approximating arbitrarily complex functions. In the paper, We use an MLP with a three-layer neural network to learn the intricate nonlinear relationships between network states and user requests, thereby outputting caching strategies for various contents.
- The second benchmark we employed the GCN-based caching placement (GCP) algorithm previously proposed in prior work [9], which adopts a centralized training and execution framework in contrast to the GMACP algorithm proposed in this paper. Specifically, within the application scenario of the GCP algorithm, it is assumed that the satellite network caching architecture does not include LEO satellite DADs, but rather achieves centralized training and control through a GEO satellite equipped with an SDN controller. The GCP algorithm begins by inputting the global

Table 2. *Simulation parameters.*

Parameter	Value
The number of orbital planes M	10
The number of satellites per orbit K	10
The height of LEO satellite orbit R_e	2000 km
The caching capacity of LEO satellites L_q	6-11 coded blocks
The size of original block s	[40, 60] Mbits
The size of popular content M_f	100 Mbits
Zipf parameter β	0.2-2.0
The number of users in each SR N_r^t	2-6
The total number of popular content F	24-42
The speed of light c	3×10^5 km/s
The probability of initial selecting an action ε	0.7
The discount coefficient for the reward τ	0.9
Learning rate of actor network	0.01
Learning rate of critic network	0.01
The delay penalty coefficient ω	2

satellite state information $s(t)$ into a GCN to perform feature extraction, resulting in the state $\bar{s}(t)$. Subsequently, an A2C network utilizes the global state $\bar{s}(t)$ to make cache placement decisions $a(t)$, obtaining a global reward $r(t)$ and the state of the next time step $s(t+)$. Through iterative training, the algorithm ultimately achieves globally optimal cache placement decisions. To ensure fairness in comparison, the GCP algorithm parameters used in this study are consistent with those of the GMACP algorithm, with the only difference being the structure of the neural network.

- The third benchmark we utilized is the random distribution algorithm (RDA). In the simulations, we utilize the RDA to evaluate the practicality of our study.

5.2 Performance of GMACP Algorithm

Figure 2 compares the convergence performance of the GMACP algorithm under different numbers of users per SR, denoted by N_r^t . Here, we set the cache capacity of the LEO satellite, $L_q = 9$, the zipf parameter for the content popularity distribution in SR r , denoted by β , ranging from 0.2 to 2.0, and the number of contents, $F = 30$. The figure reveals that the objective value of our algorithm steadily increases and converges after slightly more than 300 iterations, thereafter maintaining stability. This indicates that the proposed GMACP algorithm is capable of rapid convergence.

In Figure 3, we demonstrate the effectiveness of the proposed GMACP algorithm by comparing its content delivery cost with other benchmark algorithms while

varying the cache capacity of LEO satellites. In this scenario, the zipf parameter for the content popularity distribution in SR r , β , ranging from 0.2 to 2.0, the number of contents, $F = 30$, and the numbers of users per SR r , $N_r^t = 6$. It is evident that our GMACP algorithm outperforms both the MLP algorithm and the RDA. Moreover, the content delivery cost achieved by our proposed GMACP algorithm is approximately 10% lower than that of the MLP and 23% lower than that of the RDA when the cache capacity of the LEO satellite, L_q , is set to 11. Compared to the GCP algorithm, our proposed GMACP algorithm experiences a 10% increase in content delivery cost when the cache capacity is set to 11. This disparity is primarily attributed to the centralized nature of the GCP algorithm, which leverages global state information to make caching decisions, thereby achieving a globally optimal caching strategy. In contrast, the GMACP algorithm employs distributed training and tends to pursue locally optimal solutions in terms of LEO satellite DAD, rather than a global optimum.

In Figure 4, we validate the variation in content delivery cost as a function of the zipf parameter for the content popularity distribution in SR r . Our GMACP algorithm continues to demonstrate superior performance compared to the benchmark algorithms. It can be observed that as β increases, the content delivery cost of all four algorithms gradually decreases. This trend is attributable to the fact that a larger β implies a higher request probability for each SR r . Consequently, the satellite network tends to cache contents closer to the requesting users' access satellites, thereby reducing the number of hops $h_{q,f}$ required for accessing the cached content.

In Figure 5, we investigate the system's content delivery cost against varying numbers of users in each SR r . In Figure 6, we further corroborate the efficacy of our proposed algorithms by comparing content delivery cost while varying the total number of content items. We set the cache capacity of the satellite, $L_q = 11$, and the size of the original encoding block, $s = 40$. With the increase in both the number of users per SR r and the total number of content items, the GMACP and GCP algorithms exhibit significantly superior performance. This observation underscores the effectiveness of both algorithms in managing complex scenarios.

In Figure 7, we verify the impact of the number of

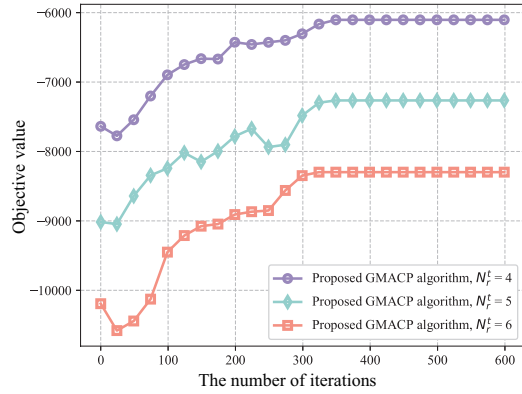


Figure 2. The convergence performance of GMACP algorithm versus different number of user in each SR.

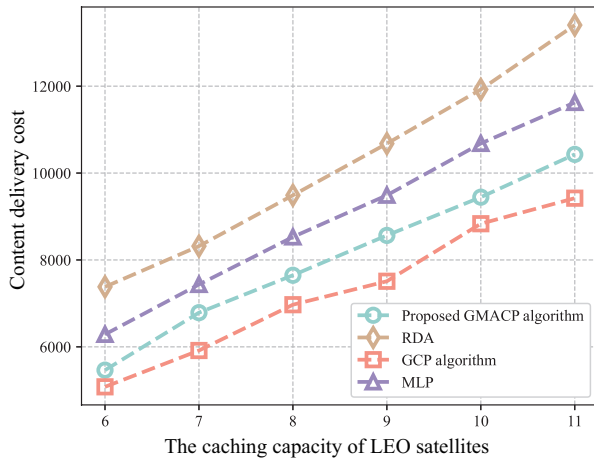


Figure 3. The content delivery cost of system versus different cache capacity of LEO satellites.

DAD in LEO satellite networks on the performance of the caching algorithms. The results indicate that when the number of DADs is one, the distributed GMACP algorithm exhibits a content delivery cost comparable to that of the centralized GCP algorithm. As the number of DADs increases, we observe an overall trend of rising content delivery costs. However, it is noteworthy that when the number of DADs is 3 and 5, the content delivery costs are substantially similar. This finding provides an empirical basis for future research to determine the number of DADs in satellite networks of varying scales.

Finally, we compare the transferability of several algorithms in Table 3. These three algorithms were all trained on the same satellite network topology dataset. In the executing phase, we evaluated the content de-

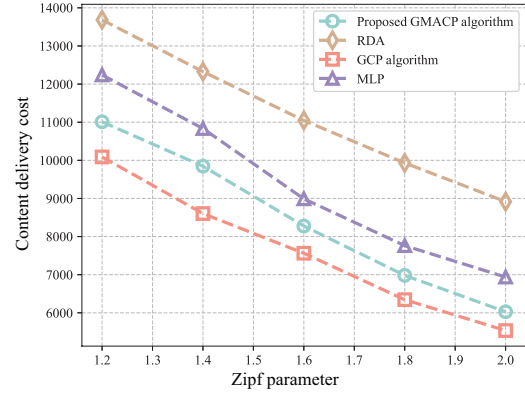


Figure 4. The content delivery cost of system versus different zipf parameter.

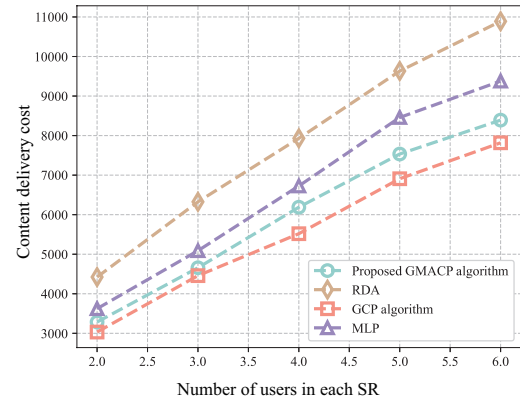


Figure 5. The content delivery cost of system versus different number of user in each SR.

Table 3. The transferability of different algorithms.

Algorithm	Satellite network topology datasets			
	datasets 1	datasets 2	datasets 3	datasets 4
GMACP Test	8300	8228	8419	8087
GCP Test	7665	7616	7768	7406
MLP Test	10208	10132	10244	9787

livery cost of these algorithms on satellite network topologies under different SCs, distinct from the training dataset. The results indicate that compared to MLP, our proposed RL algorithm based on GCNs can effectively adapt to dynamic network topology changes and make caching decisions. Even on datasets not encountered during training, it can effectively reduce content delivery costs, demonstrating good transferability.

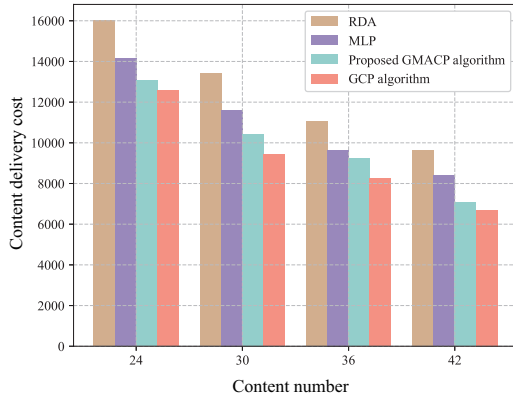


Figure 6. The content delivery cost of system versus different content number.

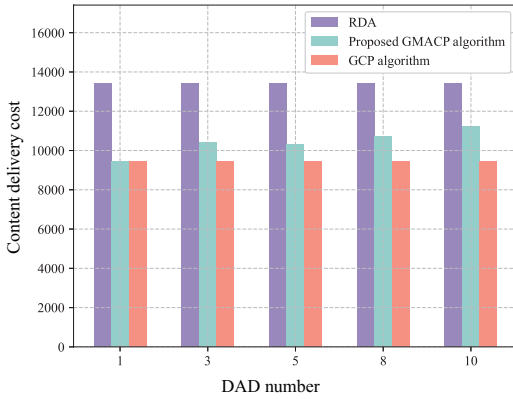


Figure 7. The content delivery cost of system versus different DAD number.

VI. CONCLUSION

In this paper, we studied the MEC problem in SDN-enabled multilayer satellite networks. We first present a three-layer distributed network caching management architecture and then, the caching placement problem in LEO satellite networks under limited caching capacity and propagation delay is established by using cache coding and expressed as MDP. To solve the proposed problem, a GMACP algorithm was proposed to optimize the caching decision. Extensive simulation results illustrate the performance of proposed algorithm in convergence and content delivery cost. Furthermore, Furthermore, we have validated the transferability of our GCN-based algorithm and demonstrated the performance variations of the algorithm due to different number of LEO satellite DADs.

Due to current economic, technological, and hardware constraints, including the incomplete deployment of both satellite ISLs and SDN controllers, there exist challenges in applying our caching placement method to practical satellite scenarios. For large-scale satellite networks, the dimensions of the state space and action space rapidly increase. This could potentially result in the RL algorithm requiring a longer convergence time and substantial computational resources. However, our GMACP algorithm employs MEO satellites for distributed decision-making, which reduces the computational demands on the satellites and enhances the transferability and robustness of the network. As for the real-time data required for training, we can obtain this through simulation software or predictive models. In the future, we aim to further investigate distributed caching strategies, explore the optimization of neural network structures, and develop lightweight models to reduce computational needs and content delivery costs.

ACKNOWLEDGEMENT

This work was supported by the National Key Research and Development Program of China under Grant 2020YFB1807700, the National Natural Science Foundation of China (NSFC) under Grant (No.62201414, 62201432), the Qinchuangyuan Project (OCYRCXM-2022-362), the Fundamental Research Funds for the Central Universities and the Innovation Fund of Xidian University under Grant YJSJ24017, and the Guangzhou Science and Technology Program under Grant 202201011732.

References

- [1] Z. Zhang and M. Tao, "Deep learning for wireless coded caching with unknown and time-variant content popularity," *IEEE Transactions on Wireless Communications*, 2020, vol. 20, no. 2, pp. 1152–1163.
- [2] S. Gu, Y. Wang, *et al.*, "Intelligent optimization of availability and communication cost in satellite-uav mobile edge caching system with fault-tolerant codes," *IEEE Transactions on Cognitive Communications and Networking*, 2020, vol. 6, no. 4, pp. 1230–1241.
- [3] C. Wang, X. You, *et al.*, "On the road to 6g: Visions, requirements, key technologies, and testbeds," *IEEE Communications Surveys & Tutorials*, 2023, vol. 25, no. 2, pp. 905–974.

- [4] J. Liu, Y. Shi, *et al.*, "Space-air-ground integrated network: A survey," *IEEE Communications Surveys & Tutorials*, 2018, vol. 20, no. 4, pp. 2714–2741.
- [5] S. Gu, X. Sun, *et al.*, "Energy-aware coded caching strategy design with resource optimization for satellite-uav-vehicle-integrated networks," *IEEE Internet of Things Journal*, 2021, vol. 9, no. 8, pp. 5799–5811.
- [6] N. Cheng, W. Quan, *et al.*, "A comprehensive simulation platform for space-air-ground integrated network," *IEEE Wireless Communications*, 2020, vol. 27, no. 1, pp. 178–185.
- [7] C. Xu, J. Feng, *et al.*, "Cross-layer optimization for cooperative content distribution in multihop device-to-device networks," *IEEE Internet of Things Journal*, 2017, vol. 6, no. 1, pp. 278–287.
- [8] X. Zhu, C. Jiang, *et al.*, "Cooperative multilayer edge caching in integrated satellite-terrestrial networks," *IEEE Transactions on Wireless Communications*, 2021, vol. 21, no. 5, pp. 2924–2937.
- [9] J. Yang, J. He, *et al.*, "Mobile edge caching in leo satellite network: A graph convolution network-based approach," in *2023 IEEE 23rd International Conference on Communication Technology (ICCT)*, 2023, pp. 1129–1134.
- [10] J. Li and Y. Han, "Optimal resource allocation for packet delay minimization in multi-layer uav networks," *IEEE Communications Letters*, 2016, vol. 21, no. 3, pp. 580–583.
- [11] H. Wang, J. Wang, *et al.*, "Spectrum sharing planning for full-duplex uav relaying systems with underlaid d2d communications," *IEEE Journal on Selected Areas in Communications*, 2018, vol. 36, no. 9, pp. 1986–1999.
- [12] K. Shi, X. Zhang, *et al.*, "Time-expanded graph based energy-efficient delay-bounded multicast over satellite networks," *IEEE Transactions on Vehicular Technology*, 2020, vol. 69, no. 9, pp. 10380–10384.
- [13] P. Dong, M. Gao, *et al.*, "Multi-layer and heterogeneous resource management in sdn-based space-terrestrial integrated networks," in *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2020, pp. 377–384.
- [14] S. Xu, X. Wang, *et al.*, "Software-defined next-generation satellite networks: Architecture, challenges, and solutions," *IEEE Access*, 2018, vol. 6, pp. 4027–4041.
- [15] P. Wei, W. Feng, *et al.*, "Robot-oriented 6g satellite-uav networks: Requirements, paradigm shifts, and case studies," *China Communications*, 2024, vol. 21, no. 2, pp. 74–84.
- [16] Q. Liang, Y. Liu, *et al.*, "Joint cache placement and content scheduling in integrated leo satellite-terrestrial networks," in *2022 IEEE/CIC International Conference on Communications in China (ICCC)*, 2022, pp. 642–648.
- [17] D. Han, W. Liao, *et al.*, "Joint cache placement and cooperative multicast beamforming in integrated satellite-terrestrial networks," *IEEE Transactions on Vehicular Technology*, 2021, vol. 71, no. 3, pp. 3131–3143.
- [18] H. Zhang, J. Xu, *et al.*, "Joint optimization of caching placement and power allocation in virtualized satellite-terrestrial network," *IEEE Transactions on Wireless Communications*, 2023, vol. 22, no. 11, pp. 7932–7943.
- [19] J. Zhang, Y. Yang, *et al.*, "Content-aware proportional caching for efficient data delivery over satellite network," in *GLOBECOM 2023-2023 IEEE Global Communications Conference*, 2023, pp. 4890–4895.
- [20] M. He, C. Zhou, *et al.*, "Learning-based cache placement and content delivery for satellite-terrestrial integrated networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.
- [21] C. Qiu, H. Yao, *et al.*, "Deep q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Transactions on Vehicular Technology*, 2019, vol. 68, no. 6, pp. 5871–5883.
- [22] L. Liu, Y. Li, *et al.*, "Deep learning-enabled file popularity-aware caching replacement for satellite-integrated content-centric networks," *IEEE Transactions on Aerospace and Electronic Systems*, 2022, vol. 58, no. 5, pp. 4551–4565.
- [23] R. Hu, L. Ma, *et al.*, "Dueling-dqn-assisted beam hopping and power allocation with dynamic cache for leo satellite communications," in *2023 3rd International Conference on Electronic Information Engineering and Computer Communication (EIECC)*, 2023, pp. 377–382.
- [24] X. Li, H. Zhang, *et al.*, "Multi-agent drl for resource allocation and cache design in terrestrial-satellite networks," *IEEE Transactions on Wireless Communications*, 2022, vol. 22, no. 8, pp. 5031–5042.
- [25] Z. Chang, L. Lei, *et al.*, "Learn to cache: Machine learning for network edge caching in the big data era," *IEEE Wireless Communications*, 2018, vol. 25, no. 3, pp. 28–35.
- [26] T. Leng, Y. Xu, *et al.*, "Caching-aware intelligent handover strategy for leo satellite networks," *Remote Sensing*, 2021, vol. 13, no. 11, p. 2230.
- [27] F. Scarselli, M. Gori, *et al.*, "The graph neural network model," *IEEE Transactions on Neural Networks*, 2008, vol. 20, no. 1, pp. 61–80.
- [28] Z. Wu, S. Pan, *et al.*, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020, vol. 32, no. 1, pp. 4–24.
- [29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
- [30] Z. Lin, X. Chen, *et al.*, "Energy harvesting space-air-sea integrated networks for mec-enabled maritime internet of things," *China Communications*, 2022, vol. 19, no. 9, pp. 47–57.
- [31] X. Xu, J. Cai, *et al.*, "Local-state routing in satellite constellation networks from the perspective of complex network," *China Communications*, 2023, vol. 20, no. 7, pp. 72–88.
- [32] M. Dutta and A. Thomas, "Decentralized coded caching for shared caches," *IEEE Communications Letters*, 2021, vol. 25, no. 5, pp. 1458–1462.
- [33] A. M. Ibrahim, A. A. Zewail, *et al.*, "Benefits of edge caching with coded placement for asymmetric networks and shared caches," *IEEE Journal on Selected Areas in Information Theory*, 2021, vol. 2, no. 4, pp. 1240–1252.
- [34] C. Fang, H. Xu, *et al.*, "Cache-assisted content delivery for noma-based satellite-edge-terminal cooperation networks," in *GLOBECOM 2023-2023 IEEE Global Communications Conference*, 2023, pp. 4871–4876.
- [35] D. Tran, S. Chatzinotas, *et al.*, "Satellite-and cache-assisted uav: A joint cache placement, resource allocation, and trajectory optimization for 6g aerial networks," *IEEE Open*

- Journal of Vehicular Technology*, 2022, vol. 3, pp. 40–54.
- [36] Y. Zhan, G. Zeng, *et al.*, “Networked tt&c for mega satellite constellations: A security perspective,” *China Communications*, 2022, vol. 19, no. 9, pp. 58–76.
 - [37] H. Zhang, R. Liu, *et al.*, “Satellite edge computing with collaborative computation offloading: An intelligent deep deterministic policy gradient approach,” *IEEE Internet of Things Journal*, 2023, vol. 10, no. 10, pp. 9092–9107.
 - [38] X. Wang, L. Fu, *et al.*, “Joint flying relay location and routing optimization for 6g uav–iot networks: A graph neural network-based approach,” *Remote Sensing*, 2022, vol. 14, no. 17, p. 4377.
 - [39] J. He, N. Cheng, *et al.*, “Load-aware network resource orchestration in leo satellite network: A gat-based approach,” *IEEE Internet of Things Journal*, 2024.
 - [40] H. He, D. Zhou, *et al.*, “Hierarchical cross-domain satellite resource management: An intelligent collaboration perspective,” *IEEE Transactions on Communications*, 2023, vol. 71, no. 4, pp. 2201–2215.
 - [41] W. Li, Z. Luo, *et al.*, “A gcn-based method for extracting power lines and pylons from airborne lidar data,” *IEEE Transactions on Geoscience and Remote Sensing*, 2021, vol. 60, pp. 1–14.
 - [42] M. AlSaffar and M. Gul, “Data-efficient maddpg based on self-attention for iot energy management systems,” *IEEE Access*, 2023.
 - [43] J. Du, Z. Kong, *et al.*, “Maddpg-based joint service placement and task offloading in mec empowered air-ground integrated networks,” *IEEE Internet of Things Journal*, 2023.

Biographies



Yang Jie received the B.E. degree in Computer Science and Technology, Inner Mongolia University of Technology, Hohhot, China, in 2022. He is currently pursuing her master’s degree in Xidian University, Xi’an, China. His research interests include space–air–ground integrated networks and satellite network resource management.



He Jingchao received the B.E. degree in communication engineering from Xidian University, Xi’an, China, in 2020, where he is currently pursuing the Ph.D degree in communication and information systems. His research interests include system design and resource allocation in space-air-ground integrated networks.



Cheng Nan received the Ph.D degree from the Department of Electrical and Computer Engineering, University of Waterloo in 2016, and B.E. degree and the M.S. degree from the Department of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2009 and 2012, respectively. He worked as a Post-doctoral fellow with the Department of Electrical and Computer Engineering, University of Toronto, from 2017 to 2019. He is currently a professor with State Key Lab. of ISN and with School of Telecommunications Engineering, Xidian University, Shaanxi, China. He has published over 90 journal papers in *IEEE Transactions* and other top journals. He serves as associate editor for *IEEE Transactions on Vehicular Technology*, *IEEE Open Journal of the Communications Society*, and Peer-to-Peer Networking and Applications, and serves/served as guest editors for several journals. His current research focuses on B5G/6G, AI-driven future networks, and space-air-ground integrated network.



Yin Zhisheng received his Ph.D degree from the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China, in 2020, and the B.E. degree from the Wuhan Institute of Technology, the B.B.A. degree from the Zhongnan University of Economics and Law, Wuhan, China, in 2012, and the M.Sc. degree from the Civil Aviation University of China, Tianjin, China, in 2016. From Sept. 2018 to Sept. 2019, Dr. Yin visited in BCCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently an Assistant Professor with State Key Lab. of ISN and with School of Telecommunications Engineering, Xidian University, Xi’an, China. He is also an Associate Editor of *IEEE Internet of Things Journal*. His research interests include space-air-ground integrated networks, wireless communications, digital twin, and physical layer security.



Han Dairu received the B.S. and Ph.D degrees from Nanjing University of Science and Technology, Nanjing, China, respectively. He is currently a Postdoctoral Research Fellow with the Nanjing University of Science and Technology. From Nov. 2019 to May 2021, he was a visiting Ph.D student with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include satellite communication and cooperative transmission in integrated satellite-terrestrial networks.



Zhou Conghao received the Ph.D degree in Electrical and Computer Engineering from University of Waterloo, Waterloo, ON, Canada, in 2022. He received the B.Eng. degree from Northeastern University, Shenyang, China, in 2017 and the M.Sc. degree from University of Illinois at Chicago, Chicago, IL, USA, in 2018. He is currently a Postdoctoral Fellow at University of Waterloo, Waterloo, ON, Canada. His research interests include space-air-ground integrated networks, network slicing, and machine learning for wireless networks.



Sun Ruijin received the Ph.D degree from the Beijing University of Posts and Telecommunications, China, in 2019. From September 2017 to September 2018, she was a Visiting Student with the University of Waterloo, Canada. From 2019 to 2021, she was a joint Post-Doctoral Fellow with the Peng Cheng Laboratory and Tsinghua University. She is currently a Lecturer with the School of Telecommunications Engineering and the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China. Her research interests include knowledge-driven wireless resource allocation.