Adaptive Multi-Dimensional Resource Slicing in Cognitive Satellite-Terrestrial Vehicular Networks

Mingcheng He, *Member, IEEE*, Huaqing Wu, *Member, IEEE*, Conghao Zhou, *Member, IEEE*, Shisheng Hu, *Member, IEEE*, Xuemin Shen, *Fellow, IEEE*, and Weihua Zhuang, *Fellow, IEEE*

***Abstract*— Satellite-terrestrial vehicular networks (STVN) are envisioned as a promising architecture to provide ubiquitous connectivity for network-reliant vehicular services. In this paper, we investigate resource slicing in cognitive STVN by adaptively managing both communication and caching resources in low Earth orbit (LEO) satellites and terrestrial base stations to support High-Definition (HD) map distribution. Leveraging distinct multicast and unicast features of satellite and terrestrial networks, a novel resource slicing architecture is proposed for cognitive STVN. Two kinds of slices, one for multicast and another for unicast transmissions, are created to tailor the different resource usage and transmission characteristics. To address the challenges posed by spatiotemporal dynamics in service demands and satellite availability, we formulate a long-term resource slicing optimization problem. A two-layer resource slicing (TLRS) scheme is proposed for adaptive multi-dimensional resource management employing a hybrid data-model co-driven approach. In the inner layer, a swap-based matching algorithm is developed to determine the multicast and caching decisions within each slicing window. In the outer layer, a hybrid proximal policy optimization (HPPO)-based reinforcement learning algorithm is designed to adaptively adjust the slicing window length and communication resources in each slice. Simulation results demonstrate that the proposed TLRS scheme in cognitive STVN can effectively guarantee service requirements with efficient resource usage and lower delay performance.**

***Index Terms*— Cognitive satellite-terrestrial vehicular networks, adaptive resource slicing, many-to-many matching, hybrid proximal policy optimization.**

I. Introduction

With the rapid development of intelligent transportation systems, autonomous and connected vehicles (CAVs) have attracted increasing attention to enhance transportation efficiency and safety by enabling communication between vehicles and infrastructures [1], [2]. This advancement necessitates the development and implementation of the High-Definition (HD) map, which provides detailed and precise geographical information, including lane models, traffic signs, and road geometry for accurate localization and navigation [3]. However, the massive data volume of HD maps compared to traditional maps makes onboard storage impractical, necessitating robust and real-time delivery via vehicular networks, unlocking the full potential of HD maps [4], [5].

Currently, CAVs primarily rely on terrestrial vehicular networks, which can provide a high data rate and low delay for HD map distribution through base stations (BSs). While effective in urban areas, terrestrial vehicular networks may fail to provide adequate coverage in sub-urban or remote areas due to high deployment costs or geographical constraints. Low Earth orbit (LEO) satellites, with their wide coverage and dense deployment, show great potential in complementing terrestrial networks to form satellite–terrestrial vehicular networks (STVN), thereby enabling CAVs to access necessary data anytime and anywhere from LEO satellites or BSs [6], [7], [8]. Meanwhile, edge servers colocated with satellites and BSs enable selected HD map files to be cached closer to CAVs, thereby reducing transmission delays by avoiding long-distance data retrieval from centralized content servers. To efficiently manage such heterogeneous resources, STVN must be cognitive, adaptively managing communication and caching resources from both satellite and terrestrial networks in response to dynamic conditions, thereby optimizing resource provisioning.

To guarantee the service requirement of HD map distribution, resource slicing, a key technique for service-oriented resource management, will continue to play a pivotal role in cognitive STVN [9]. Specifically, resource slicing enables the construction of multiple virtual networks on top of the shared physical network infrastructure, referred to as network slices, to realize resource isolation for satisfying different service requirements. An essential stage of resource slicing is resource reservation, where network slices are configured by reserving resources over large timescales to simplify management and prevent competition between slices. To adapt to dynamic service demands while maintaining efficient resource utilization, slices must be adjusted in time, with the adjustment intervals

Received 6 March 2025; revised 1 September 2025 and 8 October 2025; accepted 13 October 2025. Date of publication 28 October 2025; date of current version 31 December 2025. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN-2023-03759. The associate editor coordinating the review of this article and approving it for publication was Y. Ji. *(Corresponding author: Conghao Zhou.)*

Mingcheng He, Shisheng Hu, Xuemin Shen, and Weihua Zhuang are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: mingcheng.he@uwaterloo.ca; s97hu@uwaterloo.ca; sshen@uwaterloo.ca; wzhuang@uwaterloo.ca).

Huaqing Wu is with the Department of Electrical and Software Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: huaqing.wu1@ucalgary.ca).

Conghao Zhou was with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada. He is now with the School of Telecommunications Engineering, Xidian University, Xi'an 710071, China (e-mail: zhouconghao@xidian.edu.cn).

Digital Object Identifier 10.1109/TCCN.2025.3626342

referred to as slicing windows. Within each slicing window, satellites and BSs then allocate the reserved resources in real time for efficient utilization and guaranteed service quality.

There are several research works studying multi-dimensional resource management that enables flexible resource orchestration in satellite-terrestrial integrated networks (STIN). To reduce content delivery delay in caching-enabled STINs, prior studies have explored caching node selection and content placement across satellites and terrestrial nodes, leveraging the wide coverage of satellites and the high transmission rate of terrestrial nodes to minimize backhaul usage [10] and enhance quality of experience [11]. In parallel, research on resource slicing in STVN has focused on service-oriented resource management, including a proposed slicing architecture [12], a data-driven framework to support heterogeneous services [13], and timely slice adjustment [14].

Despite the aforementioned research endeavors, adaptive multi-dimensional resource slicing in cognitive STVN remains an open problem with the following challenges. First, satellite networks differ from terrestrial networks in both transmission mode and resource usage, complementing each other in HD map distribution. The top-down multicast capability of LEO satellites enables efficient distribution of popular HD maps by serving multiple CAVs with similar channel conditions, thereby reducing resource consumption compared to the request–response mode. However, the periodic nature of multicast necessitates stricter delay requirements due to the waiting time associated with multicast intervals, making unicast still necessary for individual demands. By contrast, terrestrial networks typically rely on unicast to flexibly adapt to heterogeneous channel conditions of CAVs within a BS's coverage. Consequently, efficiently managing communication and caching resources for multicast and unicast across satellite and terrestrial networks is complicated. Second, the dynamic nature of service demand from vehicles and the high mobility of satellite links lead to rapidly varying network conditions and resource availability, making slice management more difficult than in terrestrial networks with relatively stable resources. For example, a fixed slicing window length is inadequate for STVN. Third, optimizing the slicing window length under spatiotemporal service dynamics and time-varying available resources increases the complexity of resource slicing. A long window risks resource underutilization during off-peak periods, whereas a short window improves adaptability but incurs higher reconfiguration costs. Although exploiting time-varying resources can help accommodate fluctuating traffic with lower costs and better resource utilization, striking the right balance between reserved resources and slicing window length remains a key challenge for efficient slicing in cognitive STVN.

To address the aforementioned challenges, we investigate the multi-dimensional resource slicing problem for HD map distribution in cognitive STVN, aiming to reserve resources across satellite and terrestrial networks to minimize delay and resource usage while maximizing service fulfillment. Motivated by the distinct characteristics of multicast and unicast transmissions, we design a novel resource slicing architecture that creates two collaborative slices to support HD map distribution. To tackle the complexity arising from inter-dependent multi-dimensional resources and slicing window optimization, we propose a two-layer resource slicing (TLRS) scheme as a data–model co-driven approach to adaptively find optimal solutions. Specifically, in the outer layer, we design an RL-based algorithm that accounts for time-varying resources and spatiotemporal service demand dynamics to determine the communication resources reserved in each slice and the slicing window length for long-term performance optimization. In the inner layer, given the reserved resources and window length, a swap-based matching algorithm is developed to optimize multicast and caching decisions within each slicing window. The main contributions of this paper are summarized as follows:

- We investigate resource slicing in cognitive STVN, where satellites and BSs collaborate to support HD map distribution. Leveraging the distinct characteristics of multicast and unicast transmissions, we design a resource slicing architecture for STVN, establishing two slices to cooperatively support HD map distribution. Based on this architecture, we formulate a multi-dimensional resource slicing problem to minimize overall cost by jointly optimizing communication and caching resources, slicing window length, and multicast decisions.
- We propose the TLRS scheme to enable adaptive multi-dimensional resource slicing in cognitive STVN. To reduce the complexity of sequential decision-making, the original problem is decoupled into two subproblems. In the outer layer, the slicing window length and communication resource reservation are jointly determined to capture long-term effects, while in the inner layer, multicast and caching decisions are optimized within each slicing window.
- We develop a hybrid proximal policy optimization (HPPO)-based RL algorithm to jointly determine both the discrete action of slicing window length and the continuous action of communication resource reservation at the beginning of each slicing window, aiming to minimize the long-term system cost. Based on these decisions, a swap-based matching algorithm is designed to optimize multicast and caching within each window.

The remainder of this paper is organized as follows. In Section II, we provide a comprehensive review of multi-dimensional resource management studies in STVN. The considered network scenario, designed resource slicing architecture, and service models are introduced in Sec. III. The problem is formulated in Sec. IV, followed by the proposed two-layer resource slicing scheme. Then, a swap-based matching algorithm is designed for determining multicast and caching files in Sec. V, and an HPPO-based RL algorithm for resource reservation and slicing window length is provided in Sec. VI. Performance evaluation is given in Sec. VII. Finally, in Sec. VIII, we conclude this paper and discuss the future work.

II. Related Work

Satellite–terrestrial integrated networks (STIN) have been envisioned as a promising architecture to complement

terrestrial networks and provide seamless, ubiquitous coverage. To reduce content delivery delay, a number of studies have explored caching-enabled STIN, focusing on caching node selection and content placement across both satellites and terrestrial nodes [15]. The central idea is to leverage the wide coverage of satellites and the high transmission rate of terrestrial nodes, such that popular content can be strategically cached across heterogeneous nodes to minimize energy consumption and delivery latency. A caching mechanism is proposed that allows users to retrieve data from nearby nodes, thereby reducing reliance on satellite links and effectively decreasing latency [10]. Moreover, a QoE-aware content distribution scheme is proposed in [11] that integrates caching node selection with content placement to enhance user experience Beyond caching, recent works extend the focus to joint orchestration of communication, computing, and caching (3C) resources. A software-defined networking (SDN)-based STIN architecture has been proposed to coordinate 3C resources with the aid of learning algorithms [16]. On top of this, different optimization frameworks have been developed, including matching algorithms to minimize delay and improve resource utilization [17], as well as Nash bargaining games to enhance throughput with fairness among users [18]. These studies highlight the potential of coordinated 3C management in STIN, providing important insights into how multi-dimensional resources can be jointly optimized to enhance service performance.

Beyond the multi-dimensional resource orchestration, there are also some research efforts focusing on resource slicing in STIN, aiming at service-oriented resource management. An STIN architecture is proposed in [12], where a global controller has been designed to make resource slicing decisions. In [19], a slicing solution has been presented to satisfy different grades of QoS in dynamic scenarios for minimizing the latency for resource allocation. A slicing window adaptation method is proposed in [14] to adapt to time-varying service demands and dynamic network environments for timely slice adjustments. While conventional model-based approaches struggle to handle the complexity and uncertainty of dynamic environments, reinforcement learning (RL) offers a data-driven framework that adapts to resource slicing under fluctuating conditions and high-dimensional state spaces, effectively adapting to resource slicing in fluctuating environments and high-dimensional state spaces. A hybrid data-model co-driven resource slicing mechanism has been proposed in [20] to cope with spatiotemporal dynamics in service demands, considering multiple beam-coverage areas. In [13], a reconfigurable radio access network slicing scheme is proposed to support different services for maximum revenue.

In addition to STIN-specific studies, resource slicing and two-timescale management have also been investigated in other network scenarios. In [21], a two-timescale scheme is proposed in smart-grid powered heterogeneous cellular networks, where long-term energy scheduling is coordinated with short-term radio resource allocation to enhance energy efficiency and system performance. In vehicular networks, a hierarchical resource management framework is designed in [22] for ultra-reliable and low-latency communications, jointly optimizing resource allocation across different timescales to guarantee stringent delay and reliability requirements. Furthermore, a self-adjusting slicing scheme is proposed in [23] for UAV-enabled mobile edge computing, where slice adaptation is designed to support heterogeneous task offloading and improve resource utilization.

While several research efforts have investigated multi-dimensional resource management, supporting service-oriented slicing across heterogeneous satellite and terrestrial resources remains an open problem, especially considering the coexistence of multicast and unicast transmission across different access nodes, the dynamic nature of satellite and service demands, and the inherent tradeoff in selecting the slicing window length to balance adaptability and overhead. Therefore, in this paper, we designed a novel resource slicing architecture to support HD map distribution and propose a two-layer approach that jointly optimizes slicing window length, communication and caching resource reservation, and multicast decisions for efficient resource usage and lower delay performance with satisfied service requirements.

TABLE I
SUMMARY OF NOTATIONS

Notation	Definition
$\mathcal{BS}$, $\mathcal{S}$, $\mathcal{AP}$, $\mathcal{F}$	Sets of terrestrial base stations, LEO satellites, all access points, and HD map files
B_{ap}^{r}	Available resource r at AP ap
$\lambda_{b,f}^{k,t}$	Service demand intensity for file f within the range of BS b at time slot (k,t)
ω_k	Length of slicing window k
$v_{ap,x}^{k}$	Reserved resource ratio of resource r in AP ap to slice x in slicing window k
$u_{ap,f}^{k}$, $m_{ap,f}^{k}$	Caching and multicast indicator for file f in AP ap in slicing window k
$\tilde{\lambda}_{ap,x,f}^{k,t}$	Accommodated service demand intensity of file f from slice x of AP ap at time slot (k,t)
$D_{\mathrm{C},ap,x}^{\mathrm{Q},k,t}$, $D_{\mathrm{NC},ap,x}^{\mathrm{Q},k,t}$	Bounded queueing and transmission delay for cached and non-cached files
$D_{\mathrm{C},ap,x}^{\mathrm{Q_M},k,t}$, $D_{\mathrm{NC},ap,x}^{\mathrm{Q_M},k,t}$	Maximum required queueing and transmission delay for cached and non-cached files
$D_{ap}^{\mathrm{P},k,t}$, D^{BH}, $D_{ap,x,f}^{k,t}$	Propagation delay, backhaul delay, and overall delay
$D^{\max}$, ε	Delay requirement and maximum service violation probability of HD map distribution service
β_1, β_2, β_3, β_4	Weight of resource cost, delay cost, slice reconfiguration cost, and demand penalty

III. NETWORK SCENARIO AND SYSTEM MODEL

In this section, we introduce our considered cognitive STVN scenario with the resource slicing architecture for efficient resource management, followed by the service models. Key notations are listed in Table I for easy reference.

A. Cognitive Satellite-Terrestrial Vehicular Network Scenario

In our considered cognitive STVN scenario, CAVs in the target area, primarily including urban and suburban areas, can be served by either terrestrial BSs or satellites to receive HD map files. Each CAV requests location-specific HD map

files, with varying demands among them. The overall delay of receiving an HD map file should be less than $D^{\max}$ with a maximum service violation probability ε. The set of all HD map files for the target area is denoted by $\mathcal{F} = \{1, \ldots, F\}$, with each file having a data size of κ. A content server, located remotely, stores all HD maps for the area.

Denote the set of BSs in the area and LEO satellites in a constellation by $\mathcal{BS}$ and $\mathcal{S}$, respectively. Specifically, LEO satellites are equipped with steerable antennas that can direct a beam to the target area when the elevation angle constraint is met. Let $\mathcal{AP} = \mathcal{BS} \cup \mathcal{S}$ represent the set of all access points (APs), where each AP is equipped with caching capabilities for data storage. The total available resource r at AP ap are denoted by B^r_{ap}, where $r \in \mathbf{r} = \{\mathrm{cm}, \mathrm{ch}\}$ indicating communication spectrum bandwidth and caching capacity, respectively. An SDN controller is deployed to adaptively manage the resources of all APs serving the area.

B. Resource Slicing Architecture for Cognitive STVN

To efficiently distribute HD map files while satisfying service requirements, resource slicing is adopted for cognitive STVN by establishing separate slices for multicast and unicast transmission modes, leveraging the complementary features of satellite and terrestrial networks. Since multicast and unicast modes have distinct features and requirements, a single slice with shared resources between them may compromise performance. Therefore, as illustrated in Fig. 1, we create two isolated slices: a multicast slice (MS), which utilizes only satellite resources, and a unicast slice (US), which involves both satellite and BS resources.

Resource slicing in cognitive STVN operates in a time-slotted system. To ensure high slicing performance, slicing decisions are adjusted over slicing windows, denoted by $k \in \{1, 2, \ldots\}$. Each slicing window consists of ω_k time slots, where each time slot $t \in \{1, 2, \ldots, \omega_k\}$ has a fixed duration τ. The maximum slicing window length is $\omega_{\max}$ time slots. For simplicity, let (k, t) represent the t-th time slot in slicing window k. Due to the time-varying available resources resulting from the mobility of satellites and the spatiotemporal dynamic of service demand in STVN, the slicing window length should be adaptively adjusted to adapt to these network variations.

Let $P^{k,t}_{ap}$ denote the coordinate of AP ap at time (k, t) in an Earth-centered Earth-fixed (ECEF) coordinate system, with the center of the Earth at its origin. The coordinates of each LEO satellite remain the same in each time slot and change over time slots due to their mobility. Let $a^{k,t}_{ap}$ denote the accessibility indicator of AP ap for the target area at time slot (k, t), where $a^{k,t}_{ap} = 1$ indicates that AP ap can serve the area, and $a^{k,t}_{ap} = 0$ otherwise. In this case, let $\mathcal{S}^k = \{ap \mid \sum_{t=1}^{\omega_k} a^{k,t}_{ap} > 0, \forall ap \in \mathcal{S}\}$ and $\mathcal{AP}^k = \mathcal{BS} \cup \mathcal{S}^k$ denote the set of satellites being able to cover the target area and all APs being able to serve CAVs in the area in slicing window k, respectively. Note that the slicing decisions for the k-th slicing window are made by the ground controller for all satellites in $\mathcal{S}^k$ at the beginning of the window. However, due to satellite mobility, the corresponding slices will only be deployed in the specific time slots when the satellites can actually cover the target area.

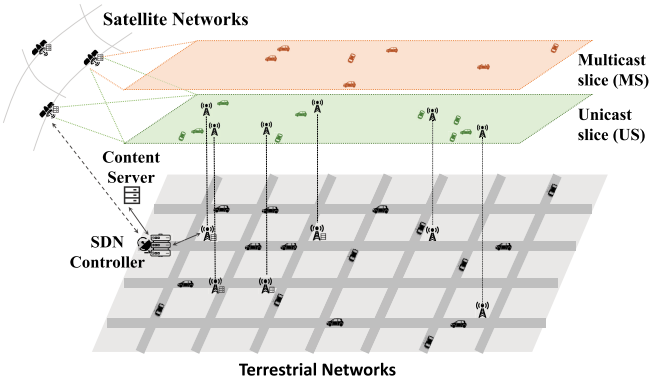


Fig. 1. Resource slicing architecture for cognitive STVN.

We model the service demand for HD maps as a non-homogeneous Poisson process (NHPP) [24], where the request intensity (in packets per second) varies over time to depict the busy and non-rush periods but remains constant within each time slot. Let $\lambda^{k,t}_f$ denote the total service demand intensity of file f in the target area at time slot (k, t), represented by $\lambda^{k,t}_f = \sum_{b \in \mathcal{BS} \cup \{0\}} \lambda^{k,t}_{b,f}$. Here, $\lambda^{k,t}_{b,f}$ is the service demand intensity for file f within the coverage of BS b at time slot (k, t), and $\lambda^{k,t}_{0,f}$ indicates the request intensity for file f that cannot be served by any BSs.

In this paper, we mainly focus on the resource reservation in each slice to satisfy service requirements in long-term. At the beginning of each slicing window, the SDN controller can predict the future service demand[1] and make the communication and caching resource reservation decisions for each AP involved in the slicing window. These decisions remain unchanged within the slicing window. Let $v^k_{ap,x} \in [0, 1]$ denote the communication resource ratio reserved from AP ap to slice $x \in \mathbf{x} = \{0, 1\}$ in slicing window k, where $x = 0$ represents MS and $x = 1$ refers to US, respectively. Let $u^k_{ap,f} \in \{0, 1\}$ denote the caching decision, where $u^k_{ap,f} = 1$ if file f is cached in AP ap and $u^k_{ap,f} = 0$ otherwise.

Given the communication resource reservation decision $v^k_{ap,x}$, the transmission rate for delivering packets (in packets per second) from AP ap to CAV g within its coverage can be expressed as

$$\xi^{k,t}_{ap,x,g} = \frac{a^{k,t}_{ap} v^k_{ap,x} B^{\mathrm{cm}}_{ap}}{\kappa} \log_2\left(1 + \frac{p_{ap} G_{ap,g} (l^{k,t}_{ap,g})^{-\rho} |h^{k,t}_{ap,g}|^2}{\sigma^2}\right), \tag{1}$$

where p_{ap} denotes the transmission power of ap, σ^2 is the noise power, $G_{ap,g}$ is the antenna gain, $l^{k,t}_{ap,g}$ is the distance between AP ap to CAV g with the path loss exponent ρ. In addition, by primarily serving CAVs urban and sub-urban areas, where the possibility of direct Line-of-Sight (LoS) signals from satellites is low due to obstructions from buildings, trees, and other infrastructure, we adopted the

[1]Note that the service demand prediction is not the primary focus of this paper and we assume that the service demand in the following $\omega_{\max}$ time slots can be predicted by adopting existing algorithms, such as RNN [25] and LSTM [26] algorithms.

Rayleigh fading model for both satellite and ground communications as a simplified approximation for small-scale fading, where $h_{ap,g}^{k,t} \sim \mathcal{CN}(0,1)$ represents the Rayleigh fading coefficient.

C. Performance Analysis for Different Slices

To characterize the HD map file delivery procedure, we model the service in each slice in each AP as a queueing system. Specifically, if the required file is cached in an AP, a transmission queue handles the file transmission to CAVs. For non-cached files, a backhaul transmission from the content server to the AP is required, followed by a transmission queue for file delivery to CAVs.

Let $m_{ap,f}^{k}$ denote the indicator of multicast service, where $m_{ap,f}^{k}=1$ represents that file f is multicast by satellite ap in slicing window k, and $m_{ap,f}^{k}=0$, otherwise. Considering the periodical multicast mechanism, let T^{M} denote the time interval between two multicast events. When a CAV requests a file, the request will be processed at the next multicast starting time, therefore introducing a waiting time. In addition, as MSs are only established for satellites, the propagation delay cannot be ignored. Therefore, the average overall delay of receiving file f from satellite ap through MS at time slot (k,t) can be represented by

$$D_{ap,0,f}^{k,t}=\begin{cases} D_{\mathrm{C},ap,0}^{\mathrm{Q},k,t}+T^{\mathrm{M}}/2+D_{ap}^{\mathrm{P},k,t}, & \text{if } u_{ap,f}^{k}=1,\\ D_{\mathrm{NC},ap,0}^{\mathrm{Q},k,t}+T^{\mathrm{M}}/2+D_{ap}^{\mathrm{P},k,t}+D^{\mathrm{BH}}, & \text{otherwise,}\end{cases} \tag{2}$$

where $D_{\mathrm{C},ap,x}^{\mathrm{Q},k,t}$ and $D_{\mathrm{NC},ap,x}^{\mathrm{Q},k,t}$ denote the queueing and transmission delays for cached and non-cached files from AP ap to CAVs at time slot (k,t). Here, we consider an average waiting time of $T^{\mathrm{M}}/2$ for processing requests between two multicast events according to the Poisson feature of requests. Let $D_{ap}^{\mathrm{P},k,t}$ indicate the propagation delay from satellite ap, and D^{BH} represent the backhaul delay between the content server and the satellite.

Meanwhile, the overall delay from AP ap to CAVs through US at time slot (k,t) is denoted by

$$D_{ap,1,f}^{k,t}=\begin{cases} D_{\mathrm{C},ap,1}^{\mathrm{Q},k,t}+\mathbb{1}_{ap\in\mathcal{S}}D_{ap}^{\mathrm{P},k,t}, & \text{if } u_{ap,f}^{k}=1,\\ D_{\mathrm{NC},ap,1}^{\mathrm{Q},k,t}+\mathbb{1}_{ap\in\mathcal{S}}D_{ap}^{\mathrm{P},k,t}+D^{\mathrm{BH}}, & \text{otherwise,}\end{cases} \tag{3}$$

where $\mathbb{1}_{cond}=1$ if $cond$ is true and $\mathbb{1}_{cond}=0$ otherwise.

Due to the time-varying nature of wireless channels and stochastic service demand, it is difficult to guarantee the deterministic queueing and transmission delay performance of queues in each slice. On the other hand, the average delay performance alone is insufficient for depicting the service requirement satisfaction within each slice. To model the statistical delay performance, the large deviation theory can be employed to evaluate the delay-bounded violation probability [27]. Specifically, the violation probability of delay D exceeding the maximum allowable queueing and transmission delay should be less than ε, approximated as

$$\begin{cases} P(D\geq D_{\mathrm{C},ap,x}^{\mathrm{Q_M},k,t})\approx e^{-\theta A_{\mathrm{C},ap,x}^{k,t}D_{\mathrm{C},ap,x}^{\mathrm{Q_M},k,t}}\leq\varepsilon,\\ P(D\geq D_{\mathrm{NC},ap,x}^{\mathrm{Q_M},k,t})\approx e^{-\theta A_{\mathrm{NC},ap,x}^{k,t}D_{\mathrm{NC},ap,x}^{\mathrm{Q_M},k,t}}\leq\varepsilon.\end{cases} \tag{4}$$

Here, $D_{\mathrm{C},ap,x}^{\mathrm{Q_M},k,t}$ and $D_{\mathrm{NC},ap,x}^{\mathrm{Q_M},k,t}$ indicates the maximum allowable queueing and transmission delay for cached and non-cached files, $A_{\mathrm{C},ap,x}^{k,t}$ and $A_{\mathrm{NC},ap,x}^{k,t}$ are the service rate for serving cached and non-cached files, and θ is the exponential decay rate of delay violation probability. In our model, we assign the same value of θ to queues in all APs. In addition, the maximum allowable queueing and transmission delay for cached and non-cached files can be given by

$$D_{\mathrm{C},ap,x}^{\mathrm{Q_M},k,t}=\begin{cases} D^{\max}-T^{\mathrm{M}}-D_{ap}^{\mathrm{P},k,t}, & \text{if } x=0,\\ D^{\max}-\mathbb{1}_{ap\in\mathcal{S}}D_{ap}^{\mathrm{P},k,t}, & \text{if } x=1,\end{cases} \tag{5}$$

$$D_{\mathrm{NC},ap,x}^{\mathrm{Q_M},k,t}=\begin{cases} D^{\max}-T^{\mathrm{M}}-D_{ap}^{\mathrm{P},k,t}-D^{\mathrm{BH}}, & \text{if } x=0,\\ D^{\max}-\mathbb{1}_{ap\in\mathcal{S}}D_{ap}^{\mathrm{P},k,t}-D^{\mathrm{BH}}, & \text{if } x=1,\end{cases} \tag{6}$$

where the maximum allowable queueing and transmission delay for MS should account for all CAV requests within the multicast interval T^{M}. To satisfy Eq. (4), the minimum required service rate can be given by

$$A_{\mathrm{C},ap,x}^{\min,k,t}=-\frac{\log(\varepsilon)}{\theta D_{\mathrm{C},ap,x}^{\mathrm{Q_M},k,t}} \text{ and } A_{\mathrm{NC},ap,x}^{\min,k,t}=-\frac{\log(\varepsilon)}{\theta D_{\mathrm{NC},ap,x}^{\mathrm{Q_M},k,t}}. \tag{7}$$

To evaluate the performance of reserved resources in a slice, the effective capacity $C_{ap,x}^{k,t}$, which quantifies the maximum accommodated service rate, is defined as

$$C_{ap,x}^{k,t}=-\frac{1}{\theta}\log\left(\mathbb{E}\left\{e^{-\theta\xi_{ap,x,g}^{k,t}}\right\}\right). \tag{8}$$

With a given $C_{ap,x}^{k,t}$, we define the bounded queueing and transmission delay for both cached and non-cached files as the maximum delay satisfying the violation probability ε, given by

$$D_{\mathrm{C},ap,x}^{\mathrm{Q},k,t}=D_{\mathrm{NC},ap,x}^{\mathrm{Q},k,t}=-\frac{\log(\varepsilon)}{\theta C_{ap,x}^{k,t}}. \tag{9}$$

According to the effective capacity theory [27], the effective capacity should be larger or equal to the minimum required service rate to satisfy Eq. (4), indicating

$$C_{ap,x}^{k,t}\geq A_{\mathrm{C},ap,x}^{k,t}\Leftrightarrow D_{\mathrm{C},ap,x}^{\mathrm{Q},k,t}\leq D_{\mathrm{C},ap,x}^{\mathrm{Q_M},k,t}, \tag{10}$$

$$C_{ap,x}^{k,t}\geq A_{\mathrm{NC},ap,x}^{k,t}\Leftrightarrow D_{\mathrm{NC},ap,x}^{\mathrm{Q},k,t}\leq D_{\mathrm{NC},ap,x}^{\mathrm{Q_M},k,t}, \tag{11}$$

for supporting cached files and non-cached files, respectively.

When reserved communication resources are insufficient to accommodate all required service demands or to satisfy Eq. (10) and Eq. (11), each AP can only fulfill a portion of the service demand. Let $\tilde{\lambda}_{ap,x,f}^{k,t}$ denote the accommodated service demand intensity of file f from AP ap in slice x, where the details will be presented in Sec. V-A. Note that BSs only establish US, i.e., $\tilde{\lambda}_{ap,0,f}^{k,t}=0,\forall ap\in\mathcal{BS}$.

IV. Problem Formulation and TLRS Scheme Design

A. Problem Formulation

Considering the time-varying service demand and available resources, we aim to minimize the long-term overall system cost while satisfying the service requirement of HD map distribution. The system cost includes the resource reservation cost from both terrestrial BSs and LEO satellites, the delay cost, the slice reconfiguration cost, and penalties for unmet service demand.

1) Resource Cost: The resource reservation cost for each slicing window is the time-averaged cost of all resources in different slices from BSs and LEO satellites serving the target area. Specifically, the resource reservation cost in slicing window k is defined in Eq. (12), as shown at the bottom of the page, where α_{ap}^{r} is the unit cost of resource r for AP ap.

2) Delay Cost: Although the delay requirement for HD map distribution is not stringent, CAVs generally seek to minimize delays when requesting content. Thus, we aim to reduce the delay for HD map distribution by effectively managing communication and caching resources and developing multicast policies to enhance the user experience. The delay cost in slicing window w is defined in Eq. (13), as shown at the bottom of the page, which represents the average HD map distribution delay across all slices supported by APs.

3) Slice Reconfiguration Cost: At the beginning of each slicing window, the controller can adjust the communication resource reservation decisions to adapt to dynamic service demand and satellite mobility. However, this resource adjustment process incurs slice reconfiguration costs. Since releasing resources can be easily accomplished, the resource releasing cost can be neglected. In slicing window k, let H_k^{slice} represent the reconfiguration cost, which quantifies the discrepancy in resource reservation decisions between adjacent slicing windows, given by Eq. (14), as shown at the bottom of the next page, where function $[\cdot]^+$ represents $\max\{\cdot, 0\}$.

4) Service Demand Penalty: The slices for HD map distribution aim to accommodate maximum service demand intensity with satisfied service requirements in the target area. To this end, we introduce a service demand penalty for each slicing window, which indicates the time-averaged service demand intensity not accommodated in each slicing window, given by Eq. (15), as shown at the bottom of the next page. Note that BSs can only serve the service demand within their coverage while satellites can support all service demand in the target area.

Combining all terms, the overall system cost in slicing window k is defined in Eq. (16), as shown at the bottom of the next page, where β_1, β_2, β_3, and β_4 indicate the relative importance of different costs and penalty.

To this end, we formulate the problem of resource slicing in cognitive STVN with two slices to support HD map distribution. Our objective is to minimize the long-term overall system cost, formulated as

$$\mathbf{P0} \quad \min_{\substack{\omega_k, m_{ap,f}^k, \\ v_{ap,x}^k, u_{ap,f}^k}} \mathbb{E}\left[\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} H_k^{\text{sys}}(\omega_k, m_{ap,f}^k, v_{ap,x}^k, u_{ap,f}^k)\right] \tag{17}$$

$$\text{s.t.} \sum_{f\in\mathcal{F}} m_{ap,f}^k \le d_{ap}^k, \tag{17a}$$

$$\sum_{x\in\mathbf{x}} v_{ap,x}^k \le \mathbb{1}_{ap\in\mathcal{AP}^k}, \tag{17b}$$

$$\sum_{f\in\mathcal{F}} u_{ap,f}^k \kappa \le \mathbb{1}_{ap\in\mathcal{AP}^k} B_{ap}^{\text{ch}}, \tag{17c}$$

$$\omega_k \in \{1, \ldots, \omega_{\max}\}, v_{ap,x}^k \in [0,1], \tag{17d}$$

$$m_{ap,f}^k \in \{0,1\}, u_{ap,f}^k \in \{0,1\}. \tag{17e}$$

In problem **P0**, constraint (17a) constrains that the number of files served by the MS of each AP should be less than the maximum number of multicast files d_{ap}^k with the detailed expression given in Sec. V. Constraints (17b) and (17c) ensure that the overall ratio of reserved communication resources cannot exceed 1 and the size of cached files in each AP remains within the caching capacity if the AP is accessible within the slicing window. Constraint (17d) limits the maximum length of each slicing window and the continuous decisions between 0 and 1 for communication resource reservation. Constraints (17e) ensure the binary decisions for multicast and caching.

B. Design of Two-Layer Resource Slicing Scheme in Cognitive STVN

The formulated problem **P0** is a stochastic optimization problem due to the spatiotemporal dynamic service demand and time-varying available resources in STVN. The limited information on future service demands, as well as the inclusion of both continuous and discrete decisions, make it highly complex and hard to find a globally optimal solution with traditional optimization approaches. A potential solution is to apply an RL-based approach, which operates without requiring knowledge of future information. However, the coupling decisions and the huge action dimension make a fully data-driven design inefficient and challenging to converge.

Observe that the long-term impact within the objective function exists only in the slice reconfiguration cost according to Eq. (14). This cost depends on slicing window length ω_k and reserved communication resources $v_{ap,x}^k$ for each slice. By isolating these variables, we can equivalently transform the original objective function (17) into

$$\begin{aligned}&\min_{\substack{\omega_k, m_{ap,f}^k, \\ v_{ap,x}^k, u_{ap,f}^k}} \mathbb{E}\left[\lim_{K\to\infty} \frac{1}{K} \sum\nolimits_{k=1}^{K} H_k^{\text{sys}}(\omega_k, v_{ap,x}^k, u_{ap,f}^k)\right] \\ &= \min_{\omega_k, v_{ap,x}^k} \mathbb{E}\left[\lim_{K\to\infty} \frac{1}{K} \sum\nolimits_{k=1}^{K} O_k(\omega_k, v_{ap,x}^k)\right],\end{aligned} \tag{18}$$

$$H_k^{\text{res}}(\omega_k, v_{ap,x}^k, u_{ap,f}^k) = \frac{1}{\omega_k} \sum_{t=1}^{\omega_k} \sum_{ap\in\mathcal{AP}^k} a_{ap}^{k,t} \left(\sum_{x\in\mathbf{x}} \alpha_{ap}^{\text{cm}} v_{ap,x}^k B_{ap}^{\text{cm}} + \alpha_{ap}^{\text{ch}} \sum_{f\in\mathcal{F}} u_{ap,f}^k \kappa\right). \tag{12}$$

$$H_k^{\text{d}}(\omega_k, m_{ap,f}^k, v_{ap,x}^k, u_{ap,f}^k) = \frac{1}{\omega_k} \sum_{t=1}^{\omega_k} \frac{\sum_{f\in\mathcal{F}} \sum_{ap\in\mathcal{AP}^k} a_{ap}^{k,t} \sum_{x\in\mathbf{x}} D_{ap,x,f}^{k,t} \tilde{\lambda}_{ap,x,f}^{k,t}}{\sum_{f\in\mathcal{F}} \sum_{ap\in\mathcal{AP}^k} a_{ap}^{k,t} \sum_{x\in\mathbf{x}} \tilde{\lambda}_{ap,x,f}^{k,t}}. \tag{13}$$

where

$$O_k(\omega_k, v^k_{ap,x}) = \min_{m^k_{ap,f}, u^k_{ap,f}} H^{\text{sys}}_k(\omega_k, m^k_{ap,f}, v^k_{ap,x}, u^k_{ap,f}). \tag{19}$$

In this case, we propose a two-layer resource slicing (TLRS) scheme and decouple problem **P0** into two subproblems to address the multi-dimensional resource slicing problem in cognitive STVN, with the working diagram shown in Fig. 2:

1) Caching and multicast file selection subproblem in the inner layer: In this subproblem, the long-term impact from slice reconfiguration is isolated. We aim to find minimize system cost within each slicing window, denoted by $O_k(\omega_k, v^k_{ap,x})$, by optimizing caching $u^k_{ap,f}$ and multicast $m^k_{ap,f}$ decisions for each slice of APs, with given slicing window length ω_k and reserved communication resources $v^k_{ap,x}$. The subproblem can be formulated as

$$\mathbf{P1} \quad \min_{m^k_{ap,f}, u^k_{ap,f}} \quad H^{\text{sys}}_k(\omega_k, m^k_{ap,f}, v^k_{ap,x}, u^k_{ap,f}) \tag{20}$$

$$\text{s.t. } \omega_k = \hat{\omega}_k, \tag{20a}$$

$$v^k_{ap,x} = \hat{v}^k_{ap,x}, \forall x \in \mathbf{x}, \forall ap \in \mathcal{AP}^k, \quad (17a), (17c), (17e), \tag{20b}$$

where $\hat{\omega}_k$ and $\hat{v}^k_{ap,x}$ are the given slicing window length and communication resource reservation decisions in slicing window k, respectively. A matching-based algorithm is developed to address this subproblem.

2) Slicing window length and communication resource reservation determination subproblem in the outer layer: With the policy derived from the inner layer, we optimize the slicing window length and the reserved communication resources in both MS and US of satellites and BSs to obtain the long-term system cost minimization, formulated by

$$\mathbf{P2} \min_{\omega_k, v^k_{ap,x}} \mathbb{E}\left[\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} H^{\text{sys}}_k(\omega_k, m^k_{ap,f}, v^k_{ap,x}, u^k_{ap,f})\right] \quad \text{s.t. } (17b), (17d). \tag{21}$$

Taking the long-term impact into account with dynamic environments, an HPPO-based RL algorithm is designed to solve this subproblem.

By combining RL for adaptive slice management and model-based optimization for decision-making within slices, our TLRS scheme effectively balances flexibility and complexity, making it well-suited for the dynamic and heterogeneous nature of STVN. In the following, we present our designed algorithms for solving subproblems **P1** and **P2** in Sections V and VI, respectively.

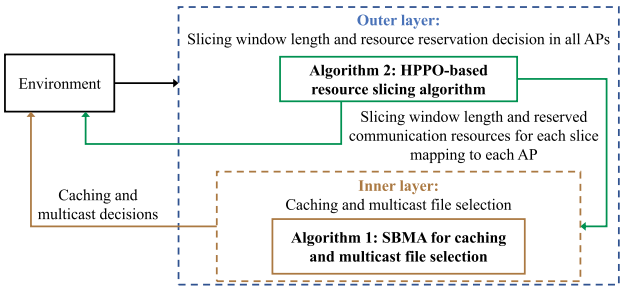


Fig. 2. Working diagram of the proposed TLRS scheme in cognitive STVN.

V. Matching-Based Caching and Multicast File Selection

In this section, we first derive the accommodated service demand intensity in different slices for calculating the system cost. Then, we reformulate the caching and multicast file selection subproblem as a many-to-many matching problem and develop a swap-based matching algorithm to solve this problem.

A. Accommodated Service Demand Intensity of Each Slice

For each slice, the accommodated service demand intensity can be determined according to the reserved communication resources, as well as caching and multicast decisions. For the MS in satellite ap, if file f is selected to be multicast, the accommodated service demand intensity in time slot (k,t) can be regarded as the total service demand intensity for file f in the area considering the large coverage of satellite, given by

$$\tilde{\lambda}^{k,t}_{ap,0,f} = \begin{cases} m^k_{ap,f}\lambda^{k,t}_f, & \text{if } D^{\text{Q},k,t}_{\text{C},ap,0} \le D^{\text{Q}_\text{M},k,t}_{\text{C},ap,0}, u^k_{ap,f} = 1, \\ & \text{or } D^{\text{Q},k,t}_{\text{NC},ap,0} \le D^{\text{Q}_\text{M},k,t}_{\text{NC},ap,0}, u^k_{ap,f} = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

Based on the effective capacity theory, with given reserved resources for US in AP ap, the maximum accommodated service demand intensity following the Poisson process can be given by [28]

$$\hat{\lambda}^{k,t}_{ap} = \frac{\theta C^{k,t}_{ap,1}}{e^\theta - 1}. \tag{23}$$

$$H^{\text{slice}}_k(\omega_k, v^k_{ap,x} | v^{k-1}_{ap,x}) = \frac{1}{\omega_k} \sum_{ap \in \mathcal{AP}^{k-1} \cup \mathcal{AP}^k} \sum_{x \in \mathbf{x}} \left[v^k_{ap,x} - v^{k-1}_{ap,x}\right]^+. \tag{14}$$

$$H^{\text{pen}}_k(\omega_k, m^k_{ap,f}, v^k_{ap,x}, u^k_{ap,f}) = \frac{1}{\omega_k} \sum_{t=1}^{\omega_k} \left[\lambda^{k,t}_f - \left(\sum_{ap \in \mathcal{BS}} \min\left\{\tilde{\lambda}^{k,t}_{ap,1,f}, \lambda^{k,t}_{ap,f}\right\} + \sum_{ap \in \mathcal{S}_k} \left(\tilde{\lambda}^{k,t}_{ap,0,f} + \tilde{\lambda}^{k,t}_{ap,1,f}\right)\right)\right]^+. \tag{15}$$

$$\begin{aligned} H^{\text{sys}}_k(\omega_k, m^k_{ap,f}, v^k_{ap,x}, u^k_{ap,f}) = {} & \beta_1 H^{\text{res}}_k(\omega_k, v^k_{ap,x}, u^k_{ap,f}) + \beta_2 H^{\text{d}}_k(\omega_k, m^k_{ap,f}, v^k_{ap,x}, u^k_{ap,f}) \\ & + \beta_3 H^{\text{slice}}_k(\omega_k, v^k_{ap,x} | v^{k-1}_{ap,x}) + \beta_4 H^{\text{pen}}_k(\omega_k, m^k_{ap,f}, v^k_{ap,x}, u^k_{ap,f}). \end{aligned} \tag{16}$$

Let $\hat{\lambda}^{k,t}_{\mathrm{C},ap}$ and $\hat{\lambda}^{k,t}_{\mathrm{NC},ap}$ denote the maximum accommodated service demand intensity in US for cached and non-cached files to satisfy Eq. (10) and Eq. (11), respectively, given in Eq. (24), as shown at the bottom of the next page.

Since CAVs within an AP's coverage area request different HD map files, each AP can either deliver files from its local cache or retrieve them from the content server. If a requested file is cached, the AP can serve more CAVs requesting the cached file with reduced delay, while CAVs requesting other files might be supported by other APs. However, accurately determining the accommodated service demand intensity for each file at an AP is challenging due to the impact of real-time resource scheduling. Therefore, for large timescale resource reservation, we focus on maximizing the total accommodated service demand intensity given the reserved resources and caching decisions among all APs, given by

$$\mathbf{P3} \max_{\tilde{\lambda}^{k,t}_{ap,1,f}} \sum_{ap\in\mathcal{AP}^k}\sum_{f\in\mathcal{F}} \tilde{\lambda}^{k,t}_{ap,1,f} \tag{25}$$

$$\text{s.t.} \quad \sum_{f\in\mathcal{F}} \tilde{\lambda}^{k,t}_{ap,1,f} \le a^{k,t}_{ap}\hat{\lambda}^{k,t}_{ap}, \tag{25a}$$

$$\sum_{ap\in\mathcal{AP}^k} (\tilde{\lambda}^{k,t}_{ap,0,f} + \tilde{\lambda}^{k,t}_{ap,1,f}) \le \lambda^{k,t}_{f}, \tag{25b}$$

$$\tilde{\lambda}^{k,t}_{ap,1,f} \le u^{k}_{ap,f}\hat{\lambda}^{k,t}_{\mathrm{C},ap} + (1-u^{k}_{ap,f})\hat{\lambda}^{k,t}_{\mathrm{NC},ap}. \tag{25c}$$

Here, constraint (25a) limits the available accommodated service demand intensity in each AP; (25b) constrains the service demand intensity request of each file; and (25c) regulates the maximum accommodated service demand intensity in each AP for a file according to the delay performances given by Eq. (24). Since **P3** is a linear programming program, it can be directly solved by some convex optimization solvers, such as CVX [29], to achieve the accommodated service demand intensity from the US in all APs under the given resource reservation and caching decisions.

B. Many-to-Many Matching Game Formulation for Caching and Multicast File Selection

To solve subproblem **P1**, we optimize the multicast and caching decisions, $m^{k}_{ap,f}$ and $u^{k}_{ap,f}$, in each AP within each slicing window. For simplicity, we omit the notation k in this section, as the decisions are made for each slicing window with a known slicing window length. Considering that each BS has sufficient storage capacity to store all HD map files related to its coverage area, we mainly investigate caching and multicast file decisions from LEO satellites for both MS and US. Since each file can be cached and multicast through multiple satellites and each satellite can cache and multicast multiple files, subproblem **P1** can be re-formulated into a graph-based many-to-many matching problem by constructing a bipartite graph denoted by $\mathcal{G} = (\Phi, \mathcal{F}, \mathcal{E})$. Here, $\Phi = \{(ap, m) \mid \forall ap \in \mathcal{S}^k, m = 1, 2, 3, 4\}$ represents the satellite service mode set where (ap, m) indicates service mode m of satellite ap. Specifically, $m = 1$ is multicast with caching, $m = 2$ indicates multicast without caching, $m = 3$ represents unicast with caching, and $m = 4$ represents unicast without caching. $\mathcal{E}$ is the set of edges connecting vertices in HD map file set $f \in \mathcal{F}$ and satellite service mode set $(ap, m) \in \Phi$, where each connected edge forms a matching pair $(f, (ap, m))$, representing that file f is served by satellite ap with service mode m. The objective is to find the optimal matching to minimize the overall system cost within the slicing window. The matching can be formally presented as follows.

Definition 1: Given a bipartite graph $\mathcal{G} = (\Phi, \mathcal{F}, \mathcal{E})$, a matching Ψ is a subgraph of $\mathcal{G}$ such that for every $f \in \mathcal{F}$ and $(ap, m) \in \Phi$, we have:

1) $\Psi((ap, m)) \subseteq \mathcal{F}$ and $\Psi(f) \subseteq \Phi$;
2) $|\Psi((ap, 1))| \le d_{ap,1}$ and $|\Psi((ap, 2))| \le d_{ap,2}$;
3) $|\Psi((ap, 1))| + |\Psi((ap, 3))| \le \lfloor B^{\mathrm{ch}}_{ap}/\kappa \rfloor$;
4) $|\Psi(f)| = |\mathcal{S}^k|$;
5) $(ap, m) \in \Psi(f)$ if and only if $f \in \Psi((ap, m))$;
6) $\Psi((ap, m)) \cap \Psi((ap, m')) = \emptyset, m \ne m'$.

Condition 1) states that each service mode is matched with a subset of HD map files, and each file is matched with a subset of satellite service modes. In condition 2), each satellite can only support a limited number of files through MS. Specifically, for satellite ap, $d_{ap,1}$ denotes the maximum number of cached multicast files, and $d_{ap,2}$ represents the maximum number of non-cached multicast files. The value of $d_{ap,1}$ is calculated using equation (26), as shown at the bottom of the next page, determining the maximum number of cached multicast files that can be periodically delivered within the multicast interval T^{M} while satisfying the delay requirement. Considering the satellite's mobility, the number of cached multicast files that can be supported under given reserved communication resources varies over time. To account for the worst-case scenario, we set $d_{ap,1}$ as the minimum number of cached multicast files that can be supported when the satellite is accessible within the slicing window. Meanwhile, the maximum number of files that can be served by MS is equal to $d_{ap,1}$, because it is easier to meet the delay requirements to support cached files than non-cached files with the given communication resources. Therefore, in slicing window k, the maximum number of files d^{k}_{ap} served by MS in constraint (17a) is the same as $d_{ap,1}$ in this section. On the other hand, the value of $d_{ap,2}$ is obtained by subtracting the number of selected files in mode 1 from $d_{ap,1}$, given by Eq. (27), as shown at the bottom of the next page. For condition 3), since the MS and US share the same storage in each satellite, the cached files cannot exceed the satellite's maximum available caching resources. Condition 4) states that the size of $\Psi(f)$ should equal the satellite number $|\mathcal{S}^k|$ in the slicing window, indicating that each satellite must select one service mode for each file. Note that the selection of the service mode does not directly correspond to the accommodated service demand intensity. In other words, even if file f is selected to be served with mode m in satellite ap, the accommodated service demand could be 0, as introduced in Sec. IV-A. Condition 5) states that if satellite service mode (ap, m) is matched to file f, file f is also matched to satellite service mode (ap, m), and vice versa. Condition 6) states that a file can only choose one service mode in each satellite.

Based on the many-to-many matching Ψ, we can obtain the caching and multicast file decisions for each satellite

ap respectively. Specifically, matching pair $(f,(ap,1))$ corresponds to decisions $m^k_{ap,f}=1$ and $u^k_{ap,f}=1$; $(f,(ap,2))$ corresponds to $m^k_{ap,f}=1$ and $u^k_{ap,f}=0$; $(f,(ap,3))$ corresponds to $m^k_{ap,f}=0$ and $u^k_{ap,f}=1$; and $(f,(ap,4))$ corresponds to $m^k_{ap,f}=0$ and $u^k_{ap,f}=0$. Based on the caching and multicast file decisions, we can derive the accommodated service demand intensities for each slice by solving **P3** and then obtain the utility $H(\Psi)$, which is the value of $-H^{\text{sys}}_k$ for the slicing window k.

C. Swapping-Based Matching Algorithm for Caching and Multicast File Selection

The formulated matching game is a many-to-many matching game with externalities [30]. Note that if the performance of a matching pair $(f,(ap,m))$ is better than that of $(f',(ap,m))$, then satellite service mode (ap,m) prefers HD map file f over f'. In the formulated matching game, the preferences of satellite service mode depend not only on the HD map files that they are matched with but also on the modes associated with the same files. As a result, static preference lists are no longer appropriate in this matching game.

To solve this matching problem, we propose a swap-based matching algorithm (SBMA) for caching and multicast file selection to maximize the overall utility. Specifically, the swap operation in a matching allows two players in the same set to exchange their matches while keeping other players' matching [31], which can be presented as

Definition 2 (Swap Operation): Given a matching Ψ and two pairs $(f,(ap,m)),(f',(ap',m'))\in\Psi$, a swap matching is

$$\Psi^{ap,m,f}_{ap',m',f'}=\Psi\setminus\{(f,(ap,m)),(f',(ap',m'))\}\cup\{(f',(ap,m)),(f,(ap',m'))\}. \quad (28)$$

The proposed SBMA for caching and multicast file selection, as shown in Algorithm 1, operates based on the swap operation. In the initialization phase, each file is matched with the unicast and non-caching mode of each satellite. Then, in the swap matching phase, each file evaluates whether a swap operation can be found to increase the overall utility in each iteration. The iterations stop when no swapping can be found to further increase the utility value.

Property 1 (Finite Convergence of SBMA): The designed SBMA always converges within a finite number of swaps. At each step, an accepted swap strictly increases the system utility $H(\Psi)$, while the utility function is upper-bounded by the available caching and multicast resources. Therefore, the algorithm must terminate after a finite number of iterations.

In this case, the final matching Ψ is obtained with the given slicing window length and reserved communication resources.

Algorithm 1 SBMA for Caching and Multicast File Selection

1: **Initialization Phase**
2: Let $\Psi=\emptyset$;
3: **for** $f\in\mathcal{F}$ **do**
4: Match f with the service mode $(ap,4)$ of each satellite, $\Psi=\Psi\cup(f,(ap,4))$;
5: **Swap Matching Phase**
6: **for** $ap\in\mathcal{S}^k, m\in\{1,2,3,4\}$ **do**
7: Calculate $H(\Psi)$, set $\Psi_{\text{temp}}=\Psi$;
8: For a file $f\in\Psi((ap,m))$, the controller searches the satellite service mode $(ap',m')\in\Phi\setminus\{(ap,m)\}$ with $f'\in\Psi((ap',m'))$ for a swap operation;
9: **if** $H(\Psi^{ap,m,f}_{ap',m',f'})>H(\Psi_{\text{temp}})$ **then**
10: Swap the pair according to **Definition** 2 and obtain Ψ' as new matching policy;
11: $\Psi_{\text{temp}}=\Psi'$, and go back to Line 8;
12: Update the matching with $\Psi=\Psi_{\text{temp}}$;
13: Go back to Line 6 and repeat until no swapping pairs can be found;
14: **Output:** Caching indicator $u^k_{ap,f}$ and multicast decision $m^k_{ap,f}$ in the slicing window according to matching policy Ψ;

D. Computational Complexity Analysis

The swap-based matching algorithm iteratively searches for beneficial swap pairs to improve the caching and multicast assignment. To verify whether a candidate swap increases the system utility $H(\Psi)$, the algorithm must recompute the accommodated service demand intensities by solving subproblem **P3**. The subproblem is a structured linear programming problem, which can be solved in polynomial time by optimization algorithms such as Dinic's method, with the complexity of $O(T_\omega|\mathcal{AP}^k|F\cdot(|\mathcal{AP}^k|+F)^2)$. Then, for the worst case, the number of candidate swap pairs in each iteration is $SL_{\max}L_{\text{tot}}$, where $L_{\max}=\min\{\max_{ap\in\mathcal{S}^k}(\lfloor B^{\text{ch}}_{ap}/\kappa\rfloor+d_{ap,1}),(F-1)/2\}$ is the maximum number of files eligible for swapping in each satellite, and $L_{\text{tot}}=\sum_{ap\in\mathcal{S}^k}(\lfloor B^{\text{ch}}_{ap}/\kappa\rfloor+d_{ap,1})$ indicates total number of candidate swaps across all satellites, which serves as an upper bound for the complexity analysis. Therefore, the overall computational complexity for

$$\hat{\lambda}^{k,t}_{\text{C},ap}=\begin{cases}\hat{\lambda}^{k,t}_{ap}, & \text{if } D^{\text{Q},k,t}_{\text{C},ap,1}\le D^{\text{Q}_\text{M},k,t}_{\text{C},ap,1},\\ 0, & \text{otherwise,}\end{cases}\quad\text{and}\quad\hat{\lambda}^{k,t}_{\text{NC},ap}=\begin{cases}\hat{\lambda}^{k,t}_{ap}, & \text{if } D^{\text{Q},k,t}_{\text{NC},ap,1}\le D^{\text{Q}_\text{M},k,t}_{\text{NC},ap,1},\\ 0, & \text{otherwise.}\end{cases} \quad (24)$$

$$d_{ap,1}=\begin{cases}\min\limits_t\left\{\left\lfloor C^{k,t}_{ap,0}T^{\text{M}}\right\rfloor\;\middle|\;a^{k,t}_{ap}=1\right\}, & \text{if } \exists t, D^{\text{Q},k,t}_{\text{C},ap,0}\le D^{\text{Q}_\text{M},k,t}_{\text{C},ap,0},\\ 0, & \text{otherwise,}\end{cases} \quad (26)$$

$$d_{ap,2}=\begin{cases}d_{ap,1}-|\Psi((ap,1))|, & \text{if } \exists t, D^{\text{Q},k,t}_{\text{NC},ap,0}\le D^{\text{Q}_\text{M},k,t}_{\text{NC},ap,0},\\ 0, & \text{otherwise,}\end{cases} \quad (27)$$

determining caching and multicast files for each slicing window is $O(ISL_{\max}L_{\text{tot}}\cdot T_\omega|\mathcal{AP}^k|F\cdot(|\mathcal{AP}^k|+F)^2)$, where I is the finite number of iterations required for SBMA algorithm.

In practice, although the theoretical bound appears high, the actual runtime is much smaller. For a target area, the number of files F is small, and the number of visible satellites and access points in $\mathcal{AP}^k$ per slicing window is moderate. Moreover, the swap-based matching converges within tens of iterations, and the effective number of **P3** evaluations is far smaller than the worst-case number of candidate swaps. Since all computations are executed by the ground controller with powerful computational capability, the runtime for the designed SBMA algorithm is significantly shorter than the slicing window duration, which can be in the order of tens or hundreds of milliseconds. Thus, the SBMA algorithm is computationally feasible and practically acceptable for determining caching and multicast decisions for satellites.

VI. HPPO-Based Slicing Window Length Determination and Communication Resource Reservation

In this section, we investigate the optimal slicing window length and communication resource reservation decisions to solve subproblem **P2** in the outer layer. Considering the long-term impacts on sequential decision-making from the time-varying service demand and the adjustment of reserved resources, we reformulate the subproblem by leveraging the Markov decision process (MDP) and propose an RL algorithm to solve it.

A. Markov Decision Process Formulation

Due to the dynamic nature of service demand and the impact of slicing reconfiguration cost from the previous decisions, optimizing the slicing window length and communication resource reservation requires considering sequential decisions. Deterministic optimization algorithms are inadequate for solving such problems. Thus, we reformulate subproblem **P2** by leveraging MDP to model the network dynamics. Let $\zeta^k = [\Lambda^k, P^k, \mathbf{v}^{k-1}] \in \mathfrak{S}$ denote the state of cognitive STVN in slicing window k with dimension $1 \times (|\mathcal{F}| * (|\mathcal{BS}| + 1) * \omega_{\max} + 3M * \omega_{\max} + 3M)$. Here, Λ^k is the predicted service demand intensity set for future $\omega_{\max}$ time slots; P^k is the set of positions of APs, denoted by $\mathcal{AP}^k_{\max}$, in the next $\omega_{\max}$ time slots, where $\mathcal{AP}^k_{\max} = \mathcal{BS} \cup \{ap \mid \sum_{t=1}^{\omega_{\max}} a^{k,t}_{ap} > 0, \forall ap \in \mathcal{S}\}$; and $\mathbf{v}^{k-1}$ is the set of resource reservation decisions in the previous slicing window, where $\mathbf{v}^k = \{v^k_{ap,x} \mid \forall ap \in \mathcal{AP}^k_{\max}, \forall x \in \mathbf{x}\}$.

Let $\mathfrak{A} = \mathfrak{A}_1 \otimes \mathfrak{A}_2$ denote the action space, which combines the discrete action space of slicing window length and continuous action space for communication resource reservation, where $\otimes$ is the Cartesian product. Thus, the action in slicing window k is denoted by $\mathcal{A}^k = (\mathcal{A}^k_1, \mathcal{A}^k_2) = (\omega^k, \mathbf{v}^k) \in \mathfrak{A}$, representing the slicing window length and reserved communication resources in each slice, with a dimension of $\mathcal{A}^k \in \{1, \ldots, \omega_{\max}\} \times [0,1]^{|\mathcal{AP}^k_{\max}|}$.

To evaluate an action in a state, the set of rewards is denoted by $\mathfrak{R} := \mathfrak{S} \times \mathfrak{A} \to \mathbb{R}$. We define the reward obtained from taking action $\mathcal{A}^k$ in state ζ^k as

$$\mathcal{R}(\zeta^k, \mathcal{A}^k) = -H^{\text{sys}}_k. \tag{29}$$

Let $\mathbf{\Pi}$ denote the set of all policies, which is the function mapping from state space $\mathfrak{S}$ to the action space $\mathfrak{A}$. Specifically, an accumulated discounted reward function is considered to estimate the expected return of the current decision-making. Therefore, the problem of minimizing the long-term overall system cost in subproblem **P2** can be approximated by minimizing the long-term accumulated discounted reward, given by

$$\begin{aligned} \mathbf{P2}' \max_{\pi \in \mathbf{\Pi}} \ & \mathbb{E}\left[\lim_{K\to\infty} \sum_{k=1}^{K} \gamma^k \mathcal{R}(\zeta^k, \mathcal{A}^k) \middle| \pi\right] \\ \text{s.t.} \ & \text{(17b), (17d).} \end{aligned} \tag{30}$$

B. HPPO-Based Algorithm for Resource Reservation

Considering the lack of state transition probability information in the MDP-based problem, an RL-based algorithm offers a promising solution, as it does not require explicit state transition probabilities. However, it is challenging to determine the non-homogeneous actions in subproblem **P2′**, involving interdependent discrete and continuous actions. Some related studies investigate methods to output both kinds of actions using a single neural network, such as discretizing the continuous action or converting the discrete action into a continuous one [32]. However, these approaches cause scalability issues or complicate the approximation and generalization of continuous action in piecewise-function action subspaces.

Inspired by studies [33], [34], [35] on parameterized action spaces in RL with both discrete and continuous actions and the hybrid actor-critic RL framework, we hereby developed an HPPO-based algorithm to manage high-dimensional state space and hybrid action nature of this problem, as shown in Fig. 3. The HPPO-based algorithm is designed based on the proximal policy optimization (PPO) approach [36], which can effectively handle large-scale problems while ensuring a stable and reliable training process with its clipped objective function to prevent large policy variations. It employs two actor networks to tackle the hybrid action issue: a discrete actor network for deciding the slicing window length and a continuous actor network for resource reservation. Specifically, in the proposed HPPO model, the discrete actor network first generates the slicing window length based on the input state. Subsequently, the continuous actor network determines the resource reservation for each slice by taking both the state and the slicing window length as input, thereby ensuring that the optimal decisions align with the slicing window. Owing to their distinct roles and input–output relationships, the two actor networks operate sequentially without parameter sharing, and each is optimized with task-specific gradients. This design avoids unnecessary enlargement of the action dimensionality and simplifies the optimization process, while enhancing flexibility and specialization in decision-making. The overall action is obtained by combining the outputs of the two actor networks, and the joint policy is formulated as the product of the two policies, with the continuous policy conditioned

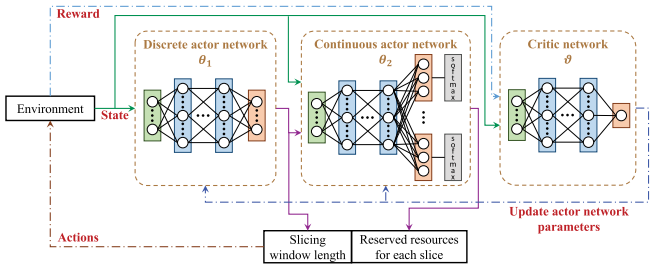


Fig. 3. Illustration of proposed HPPO-based algorithm.

on the discrete actor's decision. Additionally, a critic network as a state-value function is leveraged to generate state values for updating both discrete and continuous actor networks. This design effectively captures the interplay between slicing window length and resource reservation decisions to reach a better performance.

Considering the varying AP numbers serving the target area due to satellite mobility, the variability in state and action spaces poses a challenge in solving the problem using an HPPO-based algorithm. To address this, we use the maximum number of M APs to regulate the overall state and action spaces. Let θ_1 and θ_2 be the parameters for the discrete and continuous actor networks, respectively. Note that, directly outputting resource reservation decisions for each slice cannot guarantee constraints (17b) and (17d). To rectify this, we design a softmax-based continuous actor network to regulate the output decisions. This design employs a softmax activation function in the output layer to normalize each 3-dimension vector representing the ratios of reserved resources in MS and US, and unreserved resources. Thus, the overall output vector from the continuous actor network has a dimension of $3M$. This yields a normalized output vector for each AP given by $\{v^k_{ap,0}, v^k_{ap,1}, v^k_{ap,2}\}$, which satisfies $\sum_{x=0}^{2} v^k_{ap,x} = \mathbb{1}_{ap\in\mathcal{AP}_k}$. Here, $v^k_{ap,2}$ is the ratio of unreserved resources in AP ap. Therefore, the sum ratio of reserved resources in MS and US is $\sum_{x=0}^{1} v^x_{ap,k} \leq \mathbb{1}_{ap\in\mathcal{AP}_k}$, satisfying the constraints.

To train the HPPO-based algorithm, we need to find the optimal parameters θ_1 and θ_2 for the two actor networks. The overall policy, parameterized by $\Theta = [\theta_1, \theta_2]$, combines these two policies and is given by $\pi_\Theta(\mathcal{A}|\zeta) = \pi_{\theta_2}(\mathcal{A}_2|\mathcal{A}_1, \zeta)\pi_{\theta_1}(\mathcal{A}_1|\zeta)$.

The goal is to maximize the clipped surrogate objective

$$L^{\text{clip}}(\Theta) = \mathbb{E}\Big\{ \min(\xi_\Theta A^{\pi_{\Theta_{\text{old}}}}, \text{clip}(\xi_\Theta, 1-\delta, 1+\delta) A^{\pi_{\Theta_{\text{old}}}}) \Big\}. \tag{31}$$

The policy update ratio ξ_Θ based on the current policy for Θ is defined as

$$\xi_\Theta = \frac{\pi_\Theta(\mathcal{A}|\zeta)}{\pi_{\Theta_{\text{old}}}(\mathcal{A}|\zeta)} = \frac{\pi_{\theta_2}(\mathcal{A}_2|\mathcal{A}_1,\zeta)\pi_{\theta_1}(\mathcal{A}_1|\zeta)}{\pi_{\theta_{\text{old},2}}(\mathcal{A}_2|\mathcal{A}_1,\zeta)\pi_{\theta_{\text{old},1}}(\mathcal{A}_1|\zeta)}. \tag{32}$$

Here, $\pi_{\theta_{\text{old},i}}$ is the old policy of actor network i. Moreover, $A^{\pi_{\Theta_{\text{old}}}}$ is the estimated advantage from policy $\pi_{\Theta_{\text{old}}}$ as the combination of two policies, which can be expressed as

$$A^{\pi_{\Theta_{\text{old}}}}(\zeta^k, \mathcal{A}^k) = \mathcal{R}(\zeta^k, \mathcal{A}^k) + \gamma V_\vartheta(\zeta^{k+1}) - V_\vartheta(\zeta^k), \tag{33}$$

Algorithm 2 HPPO-Based Slicing Window Length and Communication Resource Reservation

1: **Initialization:** Parameters $\Theta = [\theta_1, \theta_2]$ for policy, ϑ for state-value function, and buffer $\mathcal{B}$;
2: **for** $j \in \{1, 2, \ldots, J\}$ **do**
3: Obtain initial state ζ^1;
4: $k = 1$;
5: **while** $\sum_{l=1}^{k} \omega_l < T_{\text{all}}$ **do**
6: Select and execute action $\mathcal{A}^k$ based on $\pi_\Theta(\mathcal{A}^k|\zeta^k)$;
7: Compute reward $\mathcal{R}(\zeta^k, \mathcal{A}^k)$;
8: Obtain new state ζ^{k+1};
9: Record transition $(\zeta^k, \mathcal{A}^k, \mathcal{R}^k, \zeta^{k+1})$ into $\mathcal{B}$;
10: $k = k + 1$;
11: **for** $e \in \{1, 2, \ldots, E\}$ **do**
12: Sample a minibatch of $\mathcal{B}_N$ transitions from buffer $\mathcal{B}$;
13: Update parameters $\Theta \leftarrow \arg\max_\Theta L^{\text{clip}}(\Theta)$ and $\vartheta \leftarrow \arg\min_\vartheta L^{\text{critic}}(\vartheta)$;
14: Empty buffer $\mathcal{B}$;
15: **Output:** Parameters θ_1, θ_2;

where $V_\vartheta(\zeta^k)$ is a state-value function of the critic network parameterized by ϑ, given by

$$V_\vartheta(\zeta^k) = \mathbb{E}\left\{ \sum_{l=0}^{\infty} \gamma^l \mathcal{R}(\zeta^{k+l+1}, \mathcal{A}^{k+l+1}) \,|\, \zeta^k \right\}. \tag{34}$$

The clipping function clips the ratio ξ_Θ between $1-\delta$ and $1+\delta$ using a hyperparameter δ, preventing the new policy from deviating far from the old policy to improve the learning stability. The measure of the accuracy of the critic network for the state-value function is based on the mean square error loss function, given by

$$L^{\text{critic}}(\vartheta) = \mathbb{E}\left\{ \left(\mathcal{R}^k + V_\vartheta(\zeta^{k+1}) - V_\vartheta(\zeta^k) \right)^2 \right\}. \tag{35}$$

The overall HPPO-based slicing window length and communication resource reservation determination algorithm is given in Algorithm 2, where J is the number of episodes. To set the number of time slots per training episode, we account for the variability in slicing window lengths. Since a fixed number of time slots per episode may not fully cover complete slicing windows, we set the total number of time slots such that when the cumulative slots in a slicing window reach or exceed T_{all}, the last slot of that window becomes the final slot of the episode. The model training starts by initializing the policy and the value functions with parameters θ_1, θ_2, and ϑ, respectively. A buffer $\mathcal{B}$ is used to store the transitions of interactions with the environment. During each slicing window, the controller collects the state from the environment, then selects and takes actions based on the current policy π_Θ, computes the reward, observes the new state, and records transitions into the buffer, as presented in Line 6 to 9. At the end of each episode, parameters θ_1 and θ_2 are updated by maximizing the clipped surrogate objective (31) via gradient ascent while parameter ϑ is updated by minimizing the loss function via gradient descent, as shown in Line 13. To ensure the stability of policy updates, the policy and value functions are updated E times in each episode to take full use of the samples generated by the current strategy.

The training complexity of the HPPO-based algorithm mainly depends on the size of the state and action spaces.

As the number of satellites increases, the state representation becomes larger by consisting of more satellite information. Similarly, the continuous action dimension, given by the resource reservation vector for all APs, linearly grows with the number of satellites. However, the actor and critic networks are implemented with a fixed number of hidden layers and neurons, which ensures that both training and inference scale approximately linearly with the number of satellites. In addition, HPPO updates are performed on a fixed mini-batch of transitions, so the number of training samples per epoch does not explode with the system size. Therefore, the overall training and inference complexity of the HPPO-based algorithm grows only linearly with the number of satellites, which ensures its scalability to large networks.

VII. Simulation Results

In this subsection, extensive simulations are carried out to evaluate the proposed TLRS scheme. Specifically, we first elaborate on the simulation settings. Afterward, the simulation results are presented to demonstrate the effectiveness of our proposed scheme.

A. Simulation Setting

In our simulation scenario, we consider the target area with a radius of 15 km, centered at 30°N latitude and 82.5°W longitude. A set of 8 HD map files is requested by different CAVs, each has a size of 10 Mb. We adopt the Starlink Phase 1 satellite constellation consisting of 72 orbits with 22 satellites per orbit, an inclination angle of 53 degrees, and an altitude of 550 km. By default, we set the minimal access elevation angle to be 35 degrees. For terrestrial networks, BSs are distributed in the area with a density of 0.1 /km^2, each with a coverage radius of 1 km. The unit costs for satellite communication and caching resources $\alpha_{ap}^{\mathrm{cm}}$ and $\alpha_{ap}^{\mathrm{ch}}$ are both 0.1, while the unit costs of BSs are 0.02. The total duration of the simulation comprises 30 min and the length of each time slot is set to be 20 s. In the simulation, the slicing window can range from 1 to 9 time slots. The delay requirement of the HD map distribution is 2.5 s with a reliability of 0.99. The default multicast interval is 1.5 s. In our HPPO-based RL model, both actor and critic networks consist of three fully connected hidden layers with [256, 256, 128] neurons. For the activation functions, we use the tanh function in the hidden layers of both actor networks to facilitate stable training. In the discrete actor network's output layer, we apply the softmax function to produce a valid probability distribution over the discrete actions. Furthermore, the Adam optimizer is adopted to train the network parameters. More simulation parameters related to the communication of satellites and BSs [37], [38], are listed in Table II.

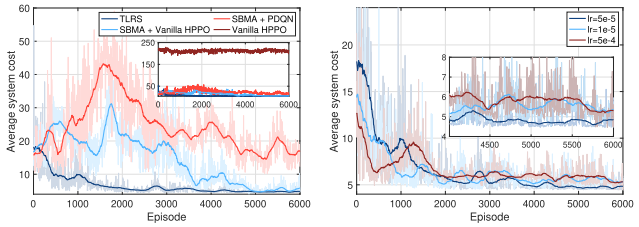
TABLE II
Simulation Parameters

LEO satellite transmission power	40 dBm
LEO satellite carrier frequency	40 GHz
LEO satellite antenna gains	32 dBi
LEO satellite bandwidth	250 MHz
LEO satellite path-loss exponent	2.5
Noise power	-174 dBm/Hz
BS transmission power	32 dBm
BS carrier frequency	28 GHz
BS antenna gains	10 dBi
BS bandwidth	100 MHz
BS path-loss exponent	3.5
Parameters β_1, β_2, β_3, and β_4	0.01, 1, 0.5, 1.5

B. Performance Evaluation

First, we evaluate the learning performance of our proposed TLRS scheme. For the comparison, we introduce the following three benchmark algorithms: 1) *SBMA+Vanilla HPPO*: Vanilla HPPO for the outer layer decisions, where the discrete and continuous actors are trained separately with only the state as input, instead of feeding discrete decisions to the continuous actor [39], while the SBMA for the inner layer remains the same as in the proposed scheme; 2) *SBMA+PDQN*: Parameterized DQN (PDQN) for the outer layer decisions, which extends DQN to parameterized actions by learning a joint Q-function where each discrete action is associated with a continuous parameter vector [34], while the SBMA for the inner layer remains the same as in the proposed scheme; 3) *Vanilla HPPO*: All decisions are directly made by a vanilla HPPO algorithm instead of a two-layer framework.

The convergence performance of the proposed TLRS scheme is shown in Fig. 4 to illustrate the average cost of each episode. First, Fig. 4(a) compares the convergence performance of our proposed TLRS scheme with other benchmark learning algorithms. To highlight the convergence trend, the dark line is the 50-point moving average of the average system cost, and the shadowed lines are the performances of each episode. We can observe that the proposed TLRS scheme converges within a relatively small number of episodes, demonstrating the convergence property of the proposed HPPO-based algorithm. In contrast, the *SBMA+Vanilla HPPO* approach shows a slower convergence rate and greater fluctuations, as its continuous actor only observes the state and thus learns a marginal policy averaged across different discrete choices. The *SBMA+PDQN* method performs even worse, showing slow and unstable convergence due to the difficulty of approximating the joint Q-function over discrete actions and continuous parameters. Meanwhile, *Vanilla HPPO*, which outputs all decisions directly, fails to converge and results in persistently high system cost, due to the enlarged and highly coupled action space. These results demonstrate that our proposed SBMA effectively decomposes caching and multicast into a structured matching process, which greatly simplifies the decision space and enables the HPPO to converge quickly. To this end, by separately addressing the distinct tasks of slicing window length and resource reservation with the assistance of the inner later SBMA algorithm, our TLRS enables more efficient exploration of the optimal decisions while ensuring a stable training process. Moreover, we compare the convergence performance of the HPPO-based algorithm with different learning rates in Fig. 4(b), where an identical learning rate is set for the discrete actor network, continuous actor network, and critic network. We can observe that all curves decrease with the increase of training episodes and tend to



(a) Convergence of different algorithms. (b) Different learning rates.

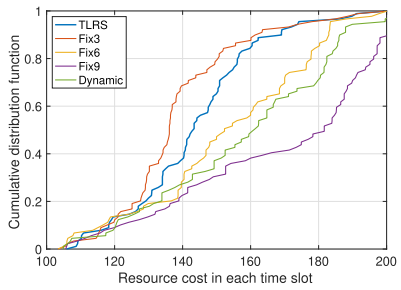
Fig. 4. Convergence performances.

converge with a large episode number. Notably, larger learning rates lead to faster convergence speed but result in performance instability. Therefore, in this remaining part of this section, the learning rate for the HPPO-based algorithm is set to be 5×10^{-5}.

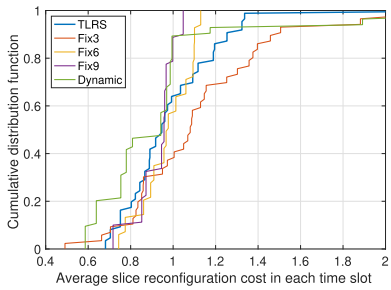
Next, we perform several experiments to validate the efficacy of different components in the TLRS scheme, where the TLRS scheme is fully trained for the comparisons. First, we evaluate the effectiveness of the slicing window length decision from the HPPO-based resource reservation algorithm. Specifically, we adopt two benchmark algorithms: 1) *Fix* n: where the slicing window length is fixed at n time slots, and 2) *Dynamic*: where the slicing window length is adjusted based on service demand fluctuations such that the variance of the accumulative service demand intensity within each slicing window [14]. For all benchmark algorithms, resource reservation decisions are made with a conventional PPO algorithm, and caching and multicast decisions are made via SBMA.

The performances of different slicing window length determination policies are shown in Fig. 5, evaluating the resource cost and slice reconfiguration cost in each time slot throughout the simulation, and showing the statistical performance of unsupported service demand and overall system cost. The cumulative distribution function (CDF) of the resource cost in each time slot is depicted in Fig. 5(a). The resource cost of *Fix3* is the lowest due to efficient resource reservation, which can quickly adapt to the dynamic service demand with minimum resource waste. The proposed TLRS scheme achieves performance comparable to *Fix3* due to its adaptive slicing window length, which effectively accommodates dynamic service demand with efficient resource usage. In contrast, *Fix6* and *Fix9* incur higher resource costs because their longer slicing window lengths make them less responsive to bursty demands. Additionally, while the *Dynamic* policy is tailored to service demand variations, it overlooks the time-varying resource resulting from satellite mobility. Consequently, a long slicing window can lead to a high resource cost when reserving resources on satellites with limited communication capabilities, such as those at low elevation angles. The CDF of time-averaged slice reconfiguration cost is shown in Fig. 5(b). From the figure, we can observe that *Fix3* has the largest slice reconfiguration cost due to frequent slice adjustments, while *Dynamic* and *Fix9* policies maintain relatively low costs due to large slicing durations. Then, Fig. 5(c) illustrates the slicing window adjustment during the simulation. We observe that the slicing window length under the proposed TLRS scheme typically ranges between 3 and 6 time slots (i.e., 1–2 minutes). This is because the TLRS scheme determines the slicing window length by jointly considering the impact of service demand variation, satellite mobility (which makes the accessible satellite set and link quality time-varying), and slice reconfiguration. Therefore, the TLRS scheme can adaptively determine the slicing window length to effectively balance resource cost and slice reconfiguration cost, yielding a lower overall system cost. In contrast, the *Dynamic* policy adjusts the slicing window simply according to service demand fluctuations, ignoring satellite mobility and slice reconfiguration, which leads to inefficient resource utilization, especially during periods of low service demand fluctuations. Moreover, Fig. 5(d) presents the unsupported service demand in the target area. In the per-slot results, the proposed TLRS scheme and *Fix3* exhibit lower unsupported demand across most time slots. Both methods achieve superior demand accommodation compared to the others. Although *Fix3* yields a slightly lower mean and variance of unsupported demand, it incurs a much higher slice reconfiguration cost as shown in Fig. 5(b). By contrast, TLRS adaptively adjusts the slicing windows according to dynamic demand and available resources, thereby minimizing unsupported demand while maintaining a balanced tradeoff with reconfiguration cost. Finally, Fig. 5(e) shows statistical results of the overall system cost, including the mean value, standard deviation, and the first and third quartile of the system cost for each policy within the entire simulation duration. The proposed TLRS has the lowest mean value and outperforms in the first and third quartiles, which indicates that TLRS can achieve better performance in most time slots from its adaptation, showing the effectiveness of the adaptive slicing window length for resource reservation in cognitive STVN, as it can jointly take the slice reconfiguration and the available resources into account for a long-term optimum. Although the *Dynamic* policy has a similar third quartile to the TLRS, it has a larger mean cost since the slicing window length is only determined by service demand variations. Since the *Dynamic* policy does not consider the available resources over time, it incurs occasional high costs due to unnecessary resource reservation and unsupported service demands. For the fixed slicing window policies, longer slicing windows result in smaller standard deviations, reflecting more consistent resource costs within each slicing window.

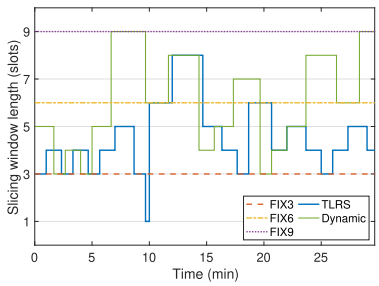
The average system cost performance of our proposed scheme under different minimum satellite elevation angles is shown in Fig. 6. The results show that when the minimum elevation angle is small, allowing more satellites to serve CAVs in the target area, the system cost becomes small due to efficient resource reservation decisions with adaptive slicing window lengths. Specifically, with more satellites available, *Fix3* has the highest cost compared with other slicing window length policies due to high costs from frequent slice reconfiguration. With the increment of the elevation angle, the overall cost increases correspondingly. Specifically, when the minimum elevation angle is large, the limited number of available satellites may not be able to meet all service demands in areas outside BSs' coverage, leading to a high



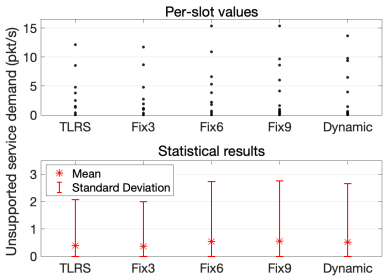
(a) CDF of resource cost.



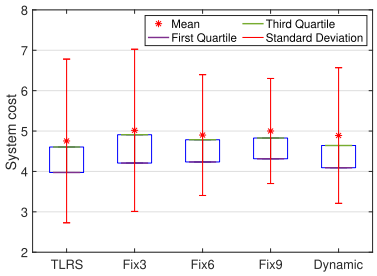
(b) CDF of slice reconfiguration cost.



(c) Slicing window length.



(d) Unsupported service demand.



(e) Overall system cost.

Fig. 5. Performance under different slicing window length determination policies.

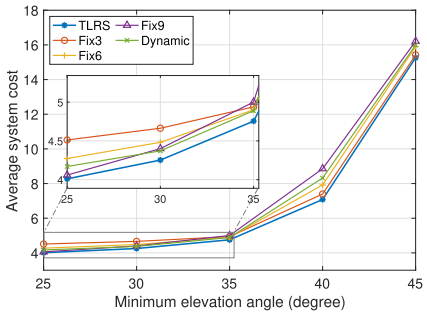


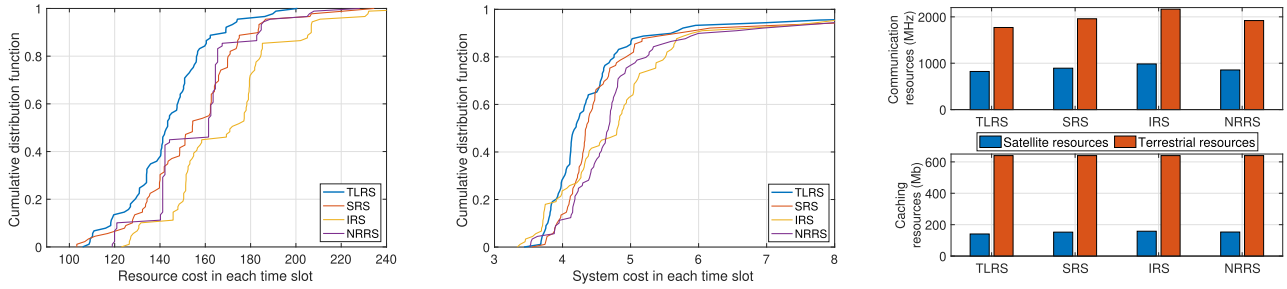
Fig. 6. Average system cost under different minimum elevation angles of LEO satellites.

cost from service demand penalty. In this case, *Fix3* exhibits a better performance compared to *Fix9* due to its frequent slice reconfiguration, reflecting a shift in the dominant cost factor from resource reconfiguration cost to service demand penalties. While the proposed TLRS scheme occasionally performs similarly to one of the benchmark algorithms under specific conditions, it consistently maintains the lowest cost, which shows its robustness and adaptability across varying network conditions.

Next, we investigate the effectiveness of the HPPO-based resource reservation strategy in our proposed TLRS scheme. For comparison, we introduce the following benchmark algorithms: 1) *SRS*: Resource reservation is determined separately for satellite and terrestrial networks [13], i.e., all involved satellites and BSs have consistent resource reservation decisions within the slicing window, although these decisions may differ between satellites and BSs; 2) *IRS*: All involved APs within the slicing window have identical resource reservation decisions within each slicing window; and 3) *NRRS*: Resource reservation decisions for satellite and terrestrial networks are fixed across all slicing windows, which are made according to the average available resources throughout the simulation to satisfy the average service demand from CAVs. Both *SRS* and *IRS* policies adopt an HPPO-based algorithm, with *SRS* employing two actions and *IRS* one action for resource reservation.

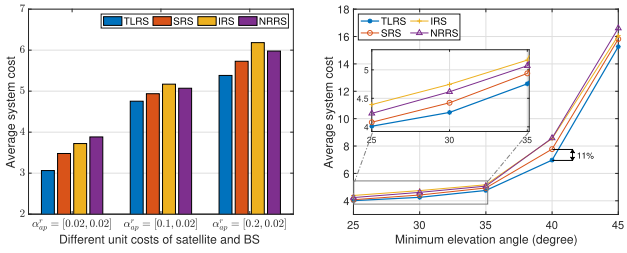
The comparisons of the performance under different resource reservation strategies are given in Fig. 7. First, Figs. 7(a) and 7(b) show the CDFs of resource cost and system cost, respectively. The proposed TLRS scheme reserves the least resources and achieves the lowest system cost, owing to its flexibility in adaptively reserving resources for each AP in each slicing window. In contrast, *SRS* and *IRS* are limited by their reduced action dimensions, which prevent them from capturing spatial demand dynamics and satellite mobility, leading to less efficient decisions. *NRRS* also incurs high system cost since it overlooks the changes in satellite resources and only considers the average service demand across all time slots. From Fig. 7(c), we can also observe the slot-averaged consumption of communication and caching resources from both satellite and terrestrial networks. As satellite resources are more costly, terrestrial resources dominate the total usage, while the consumption of satellite resources varies across schemes. The proposed TLRS scheme achieves the lowest satellite communication and caching consumption, since adaptive slicing enables efficient and coordinated resource usage. In contrast, benchmark strategies with less flexible slicing consume significantly more resources, highlighting their limited ability to coordinate communication and caching between satellite and terrestrial networks.

In Fig. 8, we compare the average system cost of different resource reservation strategies with different environment



(a) CDF of resource cost. (b) CDF of system cost. (c) Resource consumptions.

Fig. 7. Performance under different reservation strategies.



(a) Different unit costs (b) Different elevation angles

Fig. 8. Performance under different environment settings.

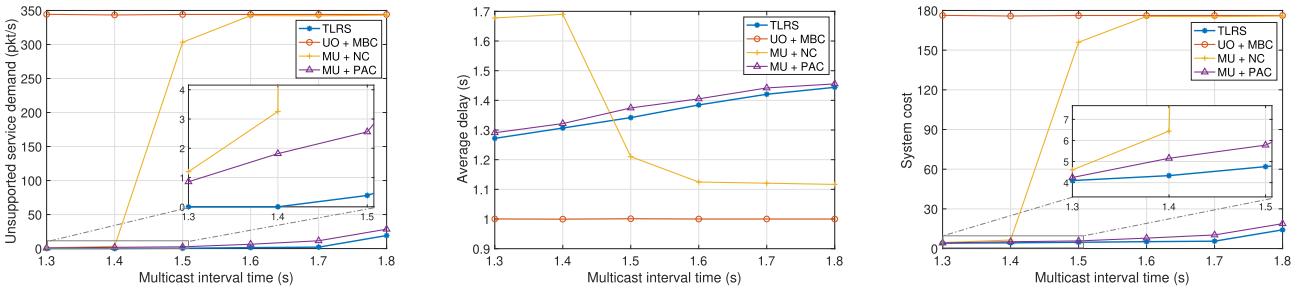
settings. Specifically, Fig. 8(a) illustrates the average system cost under various unit cost settings for both resources in both satellite and BS, shown as different combinations in the figure. We can observe that the proposed TLRS scheme consistently outperforms benchmark approaches across a wide range of parameter settings. Specifically, when the unit costs for satellite and BS are equal, our scheme reserves more satellite resources, leveraging the multicast capability to satisfy service requirements at a lower resource cost, resulting in a reduced average system cost. Conversely, when the satellite unit cost is much higher than that of the BS, our scheme flexibly adjusts resource reservation to avoid excessive costs, demonstrating the robustness of our scheme in different environment settings. In addition, the performance under different minimum elevation angles is shown in Fig. 8(b). When the minimum elevation angle is small, our proposed TLRS scheme achieves a slight improvement in overall system cost due to more efficient resource reservation, but the cost differences are inherently small. However, With the increment of minimum elevation angle, the overall cost of all strategies increases due to insufficient available resources. For example, at 40 degrees, our scheme shows a clear advantage, reducing system cost by approximately 11% compared to benchmark algorithms, thanks to the fine-grained resource reservation decisions by determining each AP's actions for long-term optimization.

Last, we evaluate the effectiveness of SBMA for caching and multicast decisions. We compare different benchmark algorithms, including

- *UO+MBC*: Supporting only unicast service mode with matching-based caching policy;
- *MU+NC*: Supporting both multicast and unicast service modes without any caching capability;
- *MU+PAC*: Supporting both multicast and unicast service modes with popularity-aware caching policy.

In Fig. 9, we compare performances of different caching and multicast policies with varying multicast interval times. As illustrated in Fig. 9(a), the proposed TLRS scheme accommodates the highest service demand due to its efficient caching and multicast strategies at each satellite. In particular, when the multicast interval is short, less resources are needed to multicast a file while meeting delay requirements, enabling more popular files to take advantage of multicast. However, as the multicast interval increases, the limited transmission time necessitates more resources to satisfy service requirements, which in turn leads to a higher level of unmet service demand. *MU+PAC* strategy performs slightly worse than the TLRS scheme since it only considers file popularity, failing to fully exploit the time-varying available resources and jointly leverage multicast and unicast capabilities of satellites. In addition, *MU+NC* strategy shows a sudden increase in unsupported service demand when the multicast interval increases. The main reason is that with a large multicast interval, the required queueing and transmission delay is reduced. Without satellite caching, the backhaul delay makes it difficult to satisfy service requirements, which leads to high unsupported service demand. For *UO+MBC* strategy without multicast, the unsupported service demand is high since unicast cannot support the huge service demand.

The average delay performance is provided in Fig. 9(b). Without multicast, the average delay is low since there is no waiting time for each multicast event. For *MU+NC* strategy, when the multicast interval is small, the delay requirements can be easily met with the available resources, but the backhaul delay leads to a higher delay compared to other strategies. As the multicast interval time increases, the delay requirement cannot be satisfied without caching, resulting in more unsupported service demand. Therefore, the average delay decreases since only the accommodated service demand (mainly from US) is considered. When both multicast and caching are incorporated, the delay performance becomes similar across both policies, where the proposed TLRS scheme notably achieves a lower average delay,



(a) Unsupported service demand. (b) Average delay. (c) Average system cost.

Fig. 9. Performance for different caching and multicast policies.

benefiting from SBMA for jointly optimizing caching and multicast decisions.

Finally, the average system cost performance is shown in Fig. 9(c), where the proposed TLRS scheme outperforms other strategies. The dominant factor contributing to the overall cost is the unsupported service demand. In this case, the performance of *MU+PAC* strategy approaches that of the proposed TLRS scheme. On the other hand, strategies without either multicast or caching have large system costs, especially under a large multicast interval time. This demonstrates that the proposed TLRS scheme provides adaptive and efficient resource slicing in cognitive STVN.

VIII. Conclusion and Future Work

In this paper, we have investigated a multi-dimensional resource slicing problem for HD map distribution in cognitive STVN. By fully leveraging the merits of satellite and terrestrial networks, we developed a novel resource slicing scheme by establishing MS and US. To cope with time-varying available resources and spatiotemporal dynamics in service demands, we proposed the TLRS scheme, which is a hybrid data-model co-driven approach that jointly optimizes slicing window length, communication and caching resource reservation, and multicast decisions. The proposed scheme efficiently manages resources from both satellite and terrestrial networks, resulting in reduced delay, low slice reconfiguration cost, enhanced resource utilization, and robust adaptability to time-varying service demands while satisfying service requirements.

Although the current resource slicing focuses on HD map delivery with two cooperative slices, cognitive STVN is expected to provision multiple heterogeneous services simultaneously. Our framework can be naturally extended to this setting by introducing additional slices for different services, which may cooperate or remain isolated depending on their service requirements. By enabling cooperation among slices at the outer layer, the proposed TLRS framework can jointly optimize heterogeneous requirements and performance trade-offs. This extension enhances the generalizability of TLRS and makes it applicable to realistic multi-service scenarios. On the other hand, it may face increased complexity in accommodating multiple heterogeneous services. In particular, the determination of slicing window length needs to account for the distinct characteristics of different services, such as latency sensitivity, reliability requirements, and traffic dynamics. This could lead to additional challenges, including heterogeneity of service requirements, resource competition among slices, prioritization across different service types, and increased coordination overhead. These factors may complicate both the design of efficient window adaptation policies and the overall orchestration of communication and caching resources. In future work, we will further investigate resource slicing strategies to support multifarious vehicular services, ensuring effectiveness across more extensive network scenarios while reducing complexity.

References

[1] C. Wu, Z. Liu, F. Liu, T. Yoshinaga, Y. Ji, and J. Li, "Collaborative learning of communication routes in edge-enabled multi-access vehicular environment," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 4, pp. 1155–1165, Dec. 2020.

[2] H. Wang et al., "Architectural design alternatives based on cloud/edge/fog computing for connected vehicles," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2349–2377, 2020.

[3] L. Chen et al., "Milestones in autonomous driving and intelligent vehicles: Part I: Control, computing system design, communication, HD map, testing, and human behaviors," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 9, pp. 3673–3685, Jan. 2020.

[4] F. Wu, W. Yang, J. Lu, F. Lyu, J. Ren, and Y. Zhang, "RLSS: A reinforcement learning scheme for HD map data source selection in vehicular NDN," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 10777–10791, Jul. 2022.

[5] *C-V2X Use Cases Volume II: Examples and Service Level Requirements*, 5GAA, Munich, Germany, Aug. 2020.

[6] H. Wu, M. He, X. Shen, W. Zhuang, N. Dao, and W. Shi, "Network performance analysis of satellite-terrestrial vehicular network," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 16829–16844, May 2024.

[7] O. Y. Kolawole, S. Vuppala, M. Sellathurai, and T. Ratnarajah, "On the performance of cognitive satellite–terrestrial networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 668–683, Dec. 2017.

[8] M. He, H. Wu, X. Shen, and W. Zhuang, "Cooperative resource scheduling for environment sensing in satellite–terrestrial vehicular networks," *IEEE Internet Things J.*, vol. 12, no. 6, pp. 6734–6748, Mar. 2025.

[9] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 1–30, 1st Quart., 2022.

[10] C. Jiang and Z. Li, "Decreasing big data application latency in satellite link by caching and peer selection," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2555–2565, Oct. 2020.

[11] D. Jiang et al., "QoE-aware efficient content distribution scheme for satellite-terrestrial networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 443–458, Jan. 2023.

[12] 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.

[13] Y. Liu, T. Ma, X. Qin, H. Zhou, and X. S. Shen, "Reconfigurable RAN slicing for ultra-dense LEO satellite networks via DRL," *IEEE Trans. Cognit. Commun. Netw.*, vol. 11, no. 1, pp. 566–580, Feb. 2025.

[14] H. Shen, Y. Tian, T. Wang, and G. Bai, "Slicing-based task offloading in Space-air-ground integrated vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4009–4024, May 2024.

[15] J. Li, K. Xue, D. S. L. Wei, J. Liu, and Y. Zhang, "Energy efficiency and traffic offloading optimization in integrated satellite/terrestrial radio access networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2367–2381, Apr. 2020.

[16] Y. Shi, Y. Cao, J. Liu, and N. Kato, "A cross-domain SDN architecture for multi-layered space-terrestrial integrated networks," *IEEE Netw.*, vol. 33, no. 1, pp. 29–35, Jan. 2019.

[17] D. Chen et al., "Resource cube: Multi-virtual resource management for integrated satellite-terrestrial industrial IoT networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11963–11974, Oct. 2020.

[18] S. Fu, B. Wu, S. Wu, and F. Fang, "Multi-resources management in 6G-oriented terrestrial-satellite network," *China Commun.*, vol. 18, no. 9, pp. 24–36, Sep. 2021.

[19] T. de Cola and I. Bisio, "QoS optimisation of eMBB services in converged 5G-satellite networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12098–12110, Oct. 2020.

[20] M. He, H. Wu, C. Zhou, and X. Shen, "Resource slicing with cross-cell coordination in satellite-terrestrial integrated networks," *Proc. IEEE ICC*, pp. 2501–2506, Jun. 2024.

[21] L. Liu, Z. Zhang, H. Zhang, Y. Zhang, and Y. Xu, "Two-timescale dynamic resource management in smart-grid powered heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 2681–2695, Apr. 2024.

[22] G. Ding, J. Yuan, G. Yu, and Y. Jiang, "Two-timescale resource management for ultrareliable and low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 70, no. 5, pp. 3282–3294, May 2022.

[23] X. Li et al., "Self-adjusting network slicing for dynamic heterogeneous task offloading in UAV-enabled mobile edge computing," *IEEE Trans. Cognit. Commun. Netw.*, early access, May 2, 2025, doi: 10.1109/TCCN.2025.3565942.

[24] J. Zhao, Y. Wang, H. Lu, Z. Li, and X. Ma, "Interference-based QoS and capacity analysis of VANETs for safety applications," *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2448–2464, Mar. 2021.

[25] N. Ramakrishnan and T. Soni, "Network traffic prediction using recurrent neural networks," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Orlando, FL, USA, Dec. 2018, pp. 187–193.

[26] Y. Lin, M. Wang, X. Zhou, G. Ding, and S. Mao, "Dynamic spectrum interaction of UAV flight formation communication with priority: A deep reinforcement learning approach," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 3, pp. 892–903, Sep. 2020.

[27] D. Wu and R. Negi, "Effective capacity: A wireless link model for support of quality of service," *IEEE Trans. Wireless Commun.*, vol. 24, no. 5, pp. 630–643, May 2003.

[28] C.-S. Chang, *Performance Guarantees in Communication Networks*. London, U.K.: Springer, 2000.

[29] M. Grant and S. Boyd. (2014). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.1*. [Online]. Available: https://cvxr.com/cvx/

[30] J. Zhao, Y. Liu, K. K. Chai, Y. Chen, and M. Elkashlan, "Many-to-many matching with externalities for device-to-device communications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 1, pp. 138–141, Feb. 2017.

[31] B. Di, L. Song, and Y. Li, "Sub-channel assignment, power allocation, and user scheduling for non-orthogonal multiple access networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7686–7698, Nov. 2016.

[32] Y. Tang and S. Agrawal, "Discretizing continuous action space for on-policy optimization," in *Proc. AAAI Artif. Intell.*, vol. 34, Montreal, QC, Canada, Dec. 2020, pp. 5981–5988.

[33] W. Masson, P. Ranchod, and G. Konidaris, "Reinforcement learning with parameterized actions," in *Proc. AAAI Artif. Intell.*, Phoenix, AZ, USA, Feb. 2016, pp. 1934–1940.

[34] J. Xiong et al., "Parametrized deep Q-networks learning: Reinforcement learning with discrete-continuous hybrid action space," 2018, *arXiv:1810.06394*.

[35] Z. Fan, R. Su, W. Zhang, and Y. Yu, "Hybrid actor-critic reinforcement learning in parameterized action space," in *Proc. Int. Joint Conf. Artif. Intell.*, Macao, China, Aug. 2019, pp. 2279–2285.

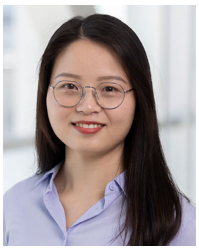
[36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[37] I. Leyva-Mayorga et al., "Non-geostationary orbit constellation design for global connectivity," in *Proc. Non-Geostationary Satell. Commun. Syst.*, 2022, pp. 237–267.

[38] T. T. de Almeida, L. de Carvalho Gomes, F. M. Ortiz, J. G. R. Júnior, and L. H. M. K. Costa, "Comparative analysis of a vehicular safety application in NS-3 and veins," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 620–629, Jan. 2022.

[39] T. Wang, Y. Deng, Z. Yang, Y. Wang, and H. Cai, "Parameterized deep reinforcement learning with hybrid action space for edge task offloading," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 10754–10767, Mar. 2024.

Mingcheng He (Member, IEEE) received the B.Sc. and M.Eng. degrees from Shanghai Jiao Tong University, Shanghai, China, in 2017 and 2020, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, ON, Canada, in 2024. He is currently a Post-Doctoral Fellow with the University of Waterloo. His research interests include network slicing in satellite-terrestrial integration networks and artificial intelligence for future wireless networks.



Huaqing Wu (Member, IEEE) received the B.E. and M.E. degrees from Beijing University of Posts and Telecommunications, Beijing, China, in 2014 and 2017, respectively, and the Ph.D. degree from the University of Waterloo, ON, Canada, in 2021. She is currently an Assistant Professor with the Department of Electrical and Software Engineering, University of Calgary, AB, Canada. Her current research interests include B5G/6G, space-air-ground integrated networks, the Internet of vehicles, mobile/edge computing/caching, and artificial intelligence (AI) for future networking. She received the N2Women: Rising Stars in Computer Networking and Communications Award in 2024, the Best Paper Awards from IEEE GLOBECOM 2018, Chinese Journal on Internet of Things in 2020, the IEEE GLOBECOM 2022, and the IEEE GLOBECOM 2024. She was the Symposium Co-Chair of IEEE GLOBECOM 2024 on Communication QoS, the Reliability and Modeling Symposium, and the Keynote and Panel Co-chair of IEEE INFOCOM Workshop on Pervasive Network Intelligence for 6G Network from 2022 to 2024. She serves as an Associate Editor for IEEE COMMUNICATIONS SURVEYS AND TUTORIALS in 2024, IEEE NETWORK in 2023, and *Security and Safety* Journal in 2021.

Conghao Zhou (Member, IEEE) received the B.Eng. degree from Northeastern University, Shenyang, China, the M.Sc. degree from the University of Illinois Chicago, Chicago, IL, USA, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada. He was a Post-Doctoral Fellow with the University of Waterloo. He is currently a Professor with the School of Telecommunications Engineering, Xidian University, China. His current research interests include immersive communications, AI for networking, and space-air-ground integrated networks.

Shisheng Hu (Member, IEEE) received the B.Eng. and M.A.Sc. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2018 and 2021, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2021. He is currently a Post-Doctoral Fellow with the University of Waterloo. His research interests include AI for wireless networks and networking for AI.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, the Internet of Things, 5G and beyond, and vehicular networks. 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 an International Fellow of the Engineering Academy of Japan. He received the "West Lake Friendship Award" from Zhejiang in 2023, the President's Excellence in Research from the University of Waterloo in 2022, Canadian Award for Telecommunications Research from Canadian Society of Information Theory (CSIT) in 2021, the R.A. Fessenden Award from IEEE, Canada in 2019, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, the James Evans Avant Garde Award from the IEEE Vehicular Technology Society in 2018, the Joseph LoCicero Award in 2015, and the Education Award from the IEEE Communications Society (ComSoc) in 2017), and the Technical Recognition Award from Wireless Communications Technical Committee in 2019, and the AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award from the University of Waterloo in 2006 and the Premier's Research Excellence Award (PREA) from the Province of Ontario, Canada, in 2003. He serves/served as the General Chair for the 6G Global Conference 2023, and ACM Mobihoc 2015, the Technical Program Committee Chair/Co-Chair for IEEE Globecom 2024, 2016, and 2007, the IEEE Infocom 2014, the IEEE VTC 2010 Fall, and the Chair for the IEEE ComSoc Technical Committee on Wireless Communications. He is the past President of the IEEE ComSoc, the Vice President for Technical and Educational Activities, the Vice President for Publications, a Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a Member of the IEEE Fellow Selection Committee of the ComSoc. He served as the Editor-in-Chief for IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*.

Weihua Zhuang (Fellow, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Dalian Maritime University, China, and the Ph.D. degree in electrical engineering from the University of New Brunswick, Canada.

She is currently a University Professor and a University Research Chair in electrical and computer engineering, University of Waterloo, Canada. Her current research focuses on network architecture, algorithms and protocols, and service provisioning in future telecommunication systems. She is a Fellow of the Royal Society of Canada (RSC), Canadian Academy of Engineering (CAE), and the Engineering Institute of Canada (EIC). She is an Elected Member of the Board of Governors of the IEEE Vehicular Technology Society (VTS). She was a recipient of the 2021 Women's Distinguished Career Award from the IEEE Vehicular Technology Society, the R.A. Fessenden Award from IEEE Canada in 2021, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2021, and the Technical Recognition Award in Ad Hoc and Sensor Networks from the IEEE Communications Society in 2017. In 2021, she was recognized by the University of Waterloo for her exceptional scholarly achievement and international pre-eminence through the designation "University Professor". She was a Tier I Canada Research Chair in Wireless Communication Network from 2010 to2024, the VTS President from 2023 to 2024, an Editor-in-Chief of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2007 to 2013, an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2005 to 2009, the General Co-Chair of the IEEE/CIC International Conference on Communications in China (ICCC) in 2021, the Technical Program Committee (TPC) Chair/Co-Chair of IEEE Vehicular Technology Conference in Fall 2016 and Fall 2017, and the TPC Symposia Chair of the IEEE Globecom 2011, and an IEEE Communications Society Distinguished Lecturer from 2008 to 2011.