Learning-Based Joint Resource Slicing and Scheduling in Space-Terrestrial Integrated Vehicular Networks

Huaqing Wu, Jiayin Chen, Conghao Zhou, Junling Li, Xuemin (Sherman) Shen

***Abstract*—In this paper, we investigate the resource slicing and scheduling problem in the space-terrestrial integrated vehicular networks to support both delay-sensitive services (DSSs) and delay-tolerant services (DTSs). Resource slicing and scheduling are to allocate spectrum resources to different slices and determine user association and bandwidth allocation for individual vehicles. To accommodate the dynamic network conditions, we first formulate a joint resource slicing and scheduling (JRSS) problem to minimize the long-term system cost, including the DSS requirement violation cost, DTS delay cost, and slice reconfiguration cost. Since resource slicing and scheduling decisions are interdependent with different timescales, we decompose the JRSS problem into a large-timescale resource slicing subproblem and a small-timescale resource scheduling subproblem. We propose a two-layered reinforcement learning (RL)-based JRSS scheme to find the solutions to the subproblems. In the resource slicing layer, spectrum resources are pre-allocated to different slices via a proximal policy optimization-based RL algorithm. In the resource scheduling layer, spectrum resources in each slice are scheduled to individual vehicles based on dynamic network conditions and service requirements via matching-based algorithms. We conduct extensive trace-driven experiments to demonstrate that the proposed scheme can effectively reduce the system cost while satisfying service quality requirements.**

***Keywords*—space-terrestrial integrated vehicular networks, LEO satellite communication, resource slicing and scheduling, reinforcement learning, matching-based optimization**

Manuscript received Jun. 30, 2021; revised Aug. 23, 2021; accepted Aug. 28, 2021. Part of this work was presented at IEEE ICC 2021[1]. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and in part by Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS). The associate editor coordinating the review of this paper and approving it for publication was R. Wang.

H. Q. Wu, C. H. Zhou, X. M. (Sherman) Shen. Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: h272wu@uwaterloo.ca; c89zhou@uwaterloo.ca; sshen@uwaterloo.ca).

J. Y. Chen. Department of Electrical and Computer Engineering, the University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: jiayinchen@ece.ubc.ca).

J. L. Li. Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), and the Department of Electrical and Computer Engineering, University of Waterloo, Canada (e-mail: lijunling@cuhk.edu.cn).

I. Introduction

Connected and automated vehicles (CAVs) have been envisioned as a necessity in the future driverless era to enable a safe, efficient, and intelligent transportation system. To accommodate multifarious CAV services, the next-generation networks are expected to provide worldwide seamless coverage, enhanced network flexibility, and improved network reliability. Terrestrial networks alone can barely satisfy these requirements due to spectrum scarcity, high operational expenditure, and geographically-constrained infrastructure deployment. To fill this gap, space-terrestrial integrated vehicular networks (STIVNs) have emerged to utilize the complementary advantages of different network segments[2,3]. Specifically, low earth orbit (LEO) satellite networks are promising in providing high-bandwidth Internet connectivity due to the low orbit altitude and small signal attenuation. With the integration of space and terrestrial networks, the STIVN holds great potential to provide globally ubiquitous, flexible, and reliable network connectivity in a cost-effective way.

CAV services have diversified quality-of-service (QoS) requirements. For instance, delay-sensitive applications have stringent delay requirements, e.g., 0.5-2 ms for robotic aided surgery and 10 ms for augmented reality devices to offload processing tasks to a processing server[4]; while data-craving services are generally delay-tolerant and require high throughput, e.g., high definition (HD) map download and video streaming services. For flexible network resource management to satisfy differentiated QoS requirements, radio access network (RAN) slicing has emerged as a promising

solution[5]. By constructing multiple logically independent slices for different types of services on a shared physical network infrastructure, RAN slicing can realize service isolation to meet diversified service demands. In RAN slicing-based networks, network resources can be managed with different time granularities: large-timescale resource slicing and small-timescale resource scheduling. The large-timescale resource slicing, which is executed in each slicing window, determines resources allocated to each slice to guarantee service level agreement. The small-timescale resource scheduling is operated at each time slot to schedule resources in each slice to individual users for QoS satisfaction. Basically, each slicing window is composed of multiple time slots, the number of which can be flexibly adjusted in different scenarios.

In the literature, there exist some works investigating the space-terrestrial integrated networks and RAN slicing techniques, respectively. In Ref. [6], a framework of space-air-ground integrated moving cell, SAGECELL, is proposed to take the complementary advantages of space, aerial, and terrestrial networks to manage the traffic demands with limited network resources. Leveraging the cognitive ratio technology, Liang et al. propose an intelligent spectrum management framework based on software-defined networking (SDN) and artificial intelligence (AI) to enable spectrum sharing between satellite and terrestrial networks[7]. Spectrum sensing in the SAG network is investigated in Ref. [8], where a deep learning (DL)-based algorithm is investigated to enhance the spectrum sensing performance. In Ref. [9], a novel caching-assisted content distribution scheme is proposed in the space terrestrial integrated networks (STIN) to guarantee users' quality of experience. To fully utilize the computing capability of heterogeneous devices in the integrated networks, the joint optimization of radio resource allocation and bidirectional communication/computation task offloading is investigated in Ref. [10] to minimize the task completion time and satellite resource usage. RAN slicing has also attracted more and more attention from both industry and academia. In the industry, 3GPP standardizes the RAN slicing in 5G networks in Ref. [11] and specifies the concepts, use cases, and requirements for network slicing management in mobile networks in Ref. [12]. In academia, RAN slicing has also gained increasing attention. In Ref. [13], a hierarchical soft-slicing framework is proposed, including the network-level slicing and the gNB-level slicing, to support services with diversified QoS requirements. Focusing on resource slicing in the space-air-ground integrated vehicular networks (SAGVN), Lyu et al. propose an online control framework in Ref. [14] to make online decisions on the request admission and scheduling, unmanned autonomous vehicle (UAV) dispatching, and resource slicing for different services. Machine learning-based approaches have also been widely applied in solving resource slicing problems. In Ref. [15], an RAN slicing orchestration solution is proposed to provide latency and throughput guarantees via a multi-armed-bandit-based orchestrator. In Ref. [16], resource slicing is studied to maximize the eMBB data rate considering the constraints on ultra-reliable low latency communications (URLLC) reliability and an optimization-aided deep RL-based framework is proposed. Resource scheduling in RAN slicing is investigated in Ref. [17], where an intelligent resource scheduling strategy is proposed with a collaborative learning framework incorporating both DL and reinforcement learning (RL).

Despite the aforementioned existing works, various technical challenges associated with resource slicing and scheduling in the STIVN remain. First, most existing works consider resource slicing in networks where the available network resources are fixed, which cannot be directly applied to the STIVN. Due to satellite mobility, the number of satellites serving a target area changes with time, leading to time-varying LEO satellite spectrum resources in the STIVN. Second, most existing works study resource slicing and scheduling separately, and the joint optimization of resource slicing and scheduling has not been well investigated. The joint optimization is essential for vehicular QoS guarantee and the STIVN resource utilization improvement due to the interdependency between the two decisions. Third, considering the spatial-temporal variations of the vehicle density and service request arrival rate, designing a dynamic joint resource slicing and scheduling scheme to optimize the long-term performance is imperative yet challenging.

In this paper, we investigate the joint resource slicing and scheduling (JRSS) to manage the spectrum resources in the STIVN for supporting both delay-sensitive services (DSSs) and delay-tolerant services (DTSs). To cope with the dynamic vehicle density and service request arrival rate, the JRSS problem is formulated as a stochastic optimization problem to minimize the long-term overall system cost, including the DSS requirement violation cost, DTS delay cost, and slice reconfiguration cost. As the resource slicing and scheduling decisions are interdependent with different timescales, the formulated JRSS problem is intractable. To solve the problem, we propose a two-layered RL-based JRSS (*TLRL-JRSS*) scheme. Specifically, the JRSS problem is first decoupled into a large-timescale resource slicing subproblem and a small-timescale resource scheduling subproblem. In the resource slicing layer, spectrum resource slicing ratios for DSS and DTS slices are optimized in each slicing window. In resource scheduling layer, spectrum resources in each slice are scheduled to individual vehicles at each time slot by determining the vehicle-to-access point (AP) association and bandwidth allocation. The main contributions in this paper are summarized as follows.

• We study the joint design of spectrum resource slicing and scheduling in the STIVN, which is of significant impor-

Tab. 1 Summary of notations

$\mathcal{SAT}$, $\mathcal{TBS}$, $\mathcal{V}$, $\mathcal{AP}$	Set of LEO satellites, TBSs, vehicles, and all the APs, respectively
B, r, a	Spectrum resource, resource slicing ratio, vehicle-AP association, respectively
$T_{ap,v}^{rem}$	The remaining contact time between vehicle v and AP ap
ρ, λ	Vehicle density and service data packet arrival rate, respectively
$\varsigma^s, \varsigma_v^{w,s}(t)$	Size of each data packet and the requested data size to be delivered, respectively
P, g, ϕ	Transmission power, channel power gain, and spectral efficiency, respectively
$D_{v,th}$, ϕ_{th}	DSS delay requirement and the minimum spectral efficiency requirement, respectively
D^{prop}	Propagation delay for satellite communication links
$D_v^{w,s}(\varsigma_v^{w,s}(t))$	Delay for providing service s to vehicle v with data size $\varsigma_v^{w,s}(t)$ at time (w,t)
PC_t^w, DC_t^w, RC^w	DSS requirement violation cost, DTS delay cost, and slice reconfiguration cost, respectively
p_v^w, c_d, c_r	Unit cost for DSS requirement violation, DTS delay, and resource reconfiguration, respectively
α, β, ρ	Parameters controlling the relative importance of different types of costs
C_{sys}^w	Overall system cost in slicing window w
$\boldsymbol{\Psi}$, $\boldsymbol{\Xi}$, $\boldsymbol{\Pi}$	Set of all possible actions, states, and policies, respectively
$\boldsymbol{\xi}^w$	State in slicing window w
γ, ε	Discount factor and PPO clipping ratio, respectively

Notes: The notations can be used with subscripts and/or superscripts. Subscripts ap and v refer to AP $ap \in \mathcal{AP}$ and vehicular user $v \in \mathcal{V}$, respectively. Superscripts w and s represent slicing window $w \in \mathcal{W}$ and service $s \in \mathcal{S}$, respectively. If (t) is used after a notation, it refers to the notation at time slot $t \in \mathcal{T}$.

tance for CAV services with diversified QoS requirements. Specifically, we formulate the JRSS problem to investigate the interplay between resource slicing and scheduling scheme design, with the objective of minimizing the long-term overall system cost.

- We propose a *TLRL-JRSS* scheme to solve the JRSS problem. The JRSS problem is first decoupled into a large-timescale resource slicing subproblem and a small-timescale resource scheduling subproblem. The two subproblems are tightly-coupled. The resource slicing decisions pose resource constraints on the resource scheduling in each slice. On the other hand, the performance of resource scheduling can provide feedback for slicing decisions to facilitate slicing adjustment.

- In the resource slicing layer, a proximal policy optimization (PPO)-based RL algorithm is utilized to determine the spectrum resource slicing ratio, considering the impact of time-varying LEO satellite resources and service request arrival rates. In the resource scheduling layer, based on the dynamic network conditions and service requirements, matching-based optimization algorithms are proposed to solve the resource scheduling subproblem with low complexity.

The remainder of this paper is organized as follows. System model and problem formulation are given in section II. Section III presents the proposed *TLRL-JRSS* scheme, which includes the matching-based algorithms for the resource scheduling subproblem as presented in section IV, and the PPO-based RL algorithm for the resource slicing subproblem as discussed in section V. Performance evaluation is carried out in section VI to demonstrate the performance of the proposed scheme, followed by the conclusions and future works in section VII. Useful notations used throughout the paper are listed in Tab. 1.

II. System Model and Problem Formulation

A. Network Model

As shown in Fig. 1, we consider an STIVN scenario where vehicles' service requests can be served by terrestrial base stations (TBSs) and/or LEO satellites. Denote the sets of LEO satellites, TBSs, and vehicles by $\mathcal{SAT}$, $\mathcal{TBS}$, and $\mathcal{V}$, respectively. $\mathcal{AP} = \mathcal{SAT} \cup \mathcal{TBS}$ denotes all the APs. Different spectrum frequencies are used for satellite-to-vehicle (S2V) and TBS-to-vehicle (T2V) communications to avoid co-channel interference. The total available spectrum bandwidth at each TBS (LEO satellite) is denoted by B_{TBS} (B_{SAT}). In this work, we adopt the control architecture proposed in Ref. [3], where the TBSs and satellites are controlled by a centralized SDN controller to conduct resource slicing and scheduling. In this work, we mainly focus on the RAN resource slicing and scheduling assuming that backhaul resources are sufficient to support all the service requests.

Considering the mobility of vehicles and LEO satellites, the remaining contact time between vehicle v and AP ap is denoted by $T_{ap,v}^{rem}$[1]. In the target scenario, the vehicle density within ap's coverage area is denoted by ρ_{ap}. At any given time instant, the vehicle density within different TBSs' coverage areas is different, while LEO satellites observe the same vehicle density since they all can cover the entire target area. In view of the vehicle mobility, at different time instants, ρ_{ap} changes for all $ap \in \mathcal{AP}$. In addition, due to satellite movement, the number of LEO satellites serving the target area changes with time, which can be calculated referring to Appendix B. Thus, the LEO satellite spectrum resource available for the target area also varies with time.

[1]Vehicles can report their locations and planned trajectories to APs, based on which $T_{ap,v}^{rem}$ can be calculated with the fixed deployment of TBSs and the trackable locations of satellites.

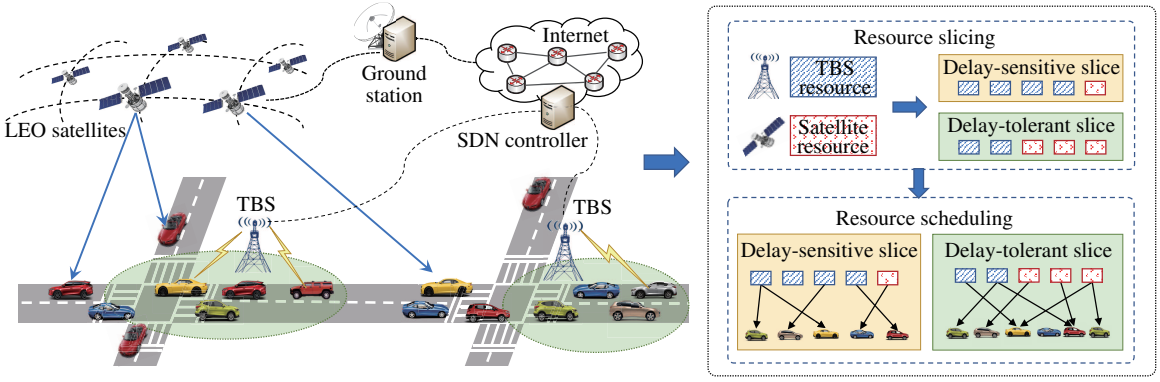


Fig. 1 Illustration of resource slicing and scheduling in the STIVN

B. Resource Slicing and Scheduling

As shown in Fig. 1, two slices are constructed in this work to support two types of services: 1) DSSs with maximum tolerable delay constraints; and 2) DTSs which require to minimize the overall service delay. The set of vehicular services is denoted by $\mathcal{S} = \mathcal{DSS} \cup \mathcal{DTS}$. Resource slicing and scheduling operate in a time-slotted manner. Time is partitioned into multiple slicing windows, denoted by $w \in \mathcal{W} = \{1, 2, \cdots, W\}$. Each slicing window is further partitioned into multiple time slots, denoted by $t \in \mathcal{T} = \{1, 2, \cdots, T\}$. Thus, the slicing and scheduling in different timescales can be described as follows.

1) Resource slicing: At the beginning of each slicing window w, the SDN controller makes resource slicing decisions $r_{ap}^{w,s} \in [0,1]$, i.e., the slicing ratio of spectrum resources allocated to slice s at AP ap. Within a slicing window, slicing decisions remain unchanged. Due to the variations in service request patterns and the mobility of vehicles and LEO satellites, the service demands and available spectrum resources change with time. Therefore, at the end of each slicing window, the SDN controller evaluates the system performance based on the feedback from APs, and adjusts the resource slicing decisions for the next slicing window.

2) Resource scheduling: Based on the resource slicing decisions, the available spectrum resources for slice s at AP ap within slicing window w is $B_{ap}^{w,s} = r_{ap}^{w,s} B_{ap}$. Then the resource scheduling is conducted at the beginning of each time slot to allocate spectrum resources to individual vehicular users based on the spatial-temporal variations in network topology, user mobility, and service requirements. The resource scheduling decisions include the association between vehicles and APs and the bandwidth allocation for the vehicle-AP communication links. Note that the resource scheduling decisions in different slicing windows are independent.

Recall that ρ_{ap} varies for different TBSs but keeps the same for all LEO satellites at any given time slot. Therefore, $r_{ap}^{w,s}$, $ap \in \mathcal{TBS}$ varies for different APs based on the dynamic traffic demand, while $r_{ap}^{w,s} = r_{ap'}^{w,s}$, $\forall ap, ap' \in \mathcal{SAT}$. Under this assumption, when new LEO satellites become available during slicing window w, the newly available satellites can follow the resource slicing ratio decisions made at the beginning of window w. In consequence, the resources of the new satellites can be utilized in the current slicing window without requiring re-calculation of the slicing ratio.

In the STIVN scenario, each vehicle requests service $s \in \mathcal{S}$ independently. Service data packet arrivals of each vehicle are assumed to follow a Poisson process. Let $\lambda^{w,s}$ denote the data packet arrival rate per vehicle for service s in slicing window w, and let the data size of each data packet be denoted by ς^s. Thus, at time slot t, the probability that vehicle v requests service s with data size $\varsigma_v^{w,s}(t)$ is

$$\Pr_{req}(\varsigma_v^{w,s}(t)) = \frac{(\lambda^{w,s})^{\varsigma_v^{w,s}(t)/\varsigma^s} \cdot e^{-\lambda^{w,s}}}{(\varsigma_v^{w,s}(t)/\varsigma^s)!}. \tag{1}$$

C. Communication Model

For notational simplicity, we use (w,t) to denote the t-th time slot in slicing window w. Let $a_{ap,v}^{w,s}(t)$ be the association indicator, where $a_{ap,v}^{w,s}(t) = 1$ when vehicle v is associated with ap for service s at time (w,t), and $a_{ap,v}^{w,s}(t) = 0$ otherwise. Denote by $B_{ap,v}^{w,s}(t)$ the spectrum resource allocated to ap-v communication link for service s at time (w,t). Since DSS requests generally have a small data size, the DSS data will not be separated for delivery from multiple APs, i.e.,

$$\sum_{ap \in \mathcal{AP}} a_{ap,v}^{w,DSS} \leqslant 1. \tag{2}$$

In addition, due to the stringent delay requirement, the handover should be avoided during DSS provisioning to eliminate the extra handover delay. Therefore, the ap-v association is feasible for DSSs only when the remaining contact time is no shorter than the maximum tolerable delay, i.e.,

$$a_{ap,v}^{w,DSS}(t) \leqslant \mathbb{1}_{T_{ap,v}^{rem}(t) \geqslant D_{v,th}^{w}(t)}, \tag{3}$$

where $D_{v,th}^{w}(t)$ is the maximum tolerable delay for vehicle v with DSS requests at time (w,t). $\mathbb{1}_{condition}$ is an indicator,

where $\mathbb{1}_{condition} = 1$ if the *condition* is true, and $\mathbb{1}_{condition} = 0$ otherwise.

Considering that DTSs generally have large data packet sizes, DTS requests can be served by multiple types of APs simultaneously. Without loss of generality, when vehicle v is covered by multiple APs of the same type (e.g., multiple TBSs or satellites), it can connect to at most one AP from the same network segment at each time slot, i.e.,

$$\sum_{l_i \in \mathcal{SAT}} a_{l_i,v}^{w,DTS}(t) \leqslant 1, \quad \sum_{b_k \in \mathcal{TBS}} a_{b_k,v}^{w,DTS}(t) \leqslant 1. \tag{4}$$

For vehicle v, the achievable signal-to-noise ratio (SNR), the spectral efficiency, and the achievable data rate of the ap-v communication link at time (w,t) are expressed as

$$\begin{cases} SNR_{ap,v}^{w}(t) = P_{ap,v} g_{ap,v}^{w}(t)/\sigma^2, \\ \phi_{ap,v}^{w}(t) = \log_2(1 + SNR_{ap,v}^{w}(t)), \\ R_{ap,v}^{w,s}(t) = B_{ap,v}^{w,s}(t)\phi_{ap,v}^{w}(t), \ \forall ap \in \mathcal{AP}, \ \forall v \in \mathcal{V}, \end{cases} \tag{5}$$

where σ^2, $P_{ap,v}$, and $g_{ap,v}^{w}(t)$ are the Gaussian noise power, the transmit power, and the channel power gain from ap to v, respectively. Note that $g_{ap,v}^{w}(t)$ consists of large-scale pathloss and small-scale channel fading: $g_{ap,v}^{w}(t) = (d_{ap,v}^{w}(t))^{-\delta} h_{ap,v}$, where $d_{ap,v}^{w}(t)$ is the distance between ap and v, δ is the pathloss exponent, and $h_{ap,v}$ is the channel fading. Specifically, the terrestrial channels follow Rayleigh fading due to the widely spread scatterers, and thus, we have $h_{ap,v} \sim Exp(1), ap \in \mathcal{TBS}$. For S2V communications, the line-of-sight (LoS) signal is a strong dominant component. Therefore, the S2V channels are considered as Rician fading channels [18], with the probability density function of the channel fading being

$$f(x) = \frac{K+1}{\Omega} \exp\left\{-K - \frac{(K+1)x}{\Omega}\right\} I_0\left(2\sqrt{\frac{K(K+1)x}{\Omega}}\right), \tag{6}$$

where K is the ratio between the power in the LoS path and the power in the scattered paths, Ω is the total power of the LoS and scattering signals, and $I_0(\cdot)$ is the modified Bessel function of the first kind with zero order.

For T2V communications, the propagation delay is negligible, while for S2V communications, the impact of the propagation delay is non-negligible due to the long communication distance. Considering the trackability of satellites, the propagation delay from LEO satellite l_i to vehicle v at time (w,t) can be obtained and denoted by $D_{l_i,v}^{prop,w}(t)$.

D. Problem Formulation

Considering the spatial-temporal variations in network conditions and service request arrival rates, the minimization of the long-term overall system cost is of significant importance, especially from the perspective of network operators. In this work, the overall system cost includes the DSS requirement violation cost, DTS delay cost, and slice reconfiguration cost.

1) DSS Requirement Violation Cost: When the DSS delay exceeds the maximum allowable delay $D_{v,th}^{w}(t)$, a penalty cost will incur to penalize constraint violation. Thus, the DSS delay violation penalty cost at time (w,t) is

$$PC_t^w = \sum_v p_v^w \mathbb{1}_{\varsigma_v^{w,DSS}(t)>0} \mathbb{1}_{D_v^{w,DSS}(\varsigma_v^{w,DSS}(t)) > D_{v,th}^{w}(t)}, \tag{7}$$

where $D_v^{w,s}(\varsigma_v^{w,s}(t))$ is the delay for providing service s to vehicle v with data size $\varsigma_v^{w,s}(t)$ at time (w,t), and p_v^w is the unit cost for violating the delay requirement for v in slicing window w (\$/number). p_v^w can also represent the priority of different vehicles, where a larger p_v^w indicates a higher service priority.

2) DTS Delay Cost: Since the service delay of DTS requests is significantly affected by the requested file size, average delay per unit data size is used to characterize the DTS delay performance. At time (w,t), the average DTS delay cost for all the vehicles with DTS requests is defined as

$$DC_t^w = \frac{\sum_v c_d [\mathbb{1}_{\varsigma_v^{w,DTS}(t)>0} \cdot D_v^{w,DTS}(\varsigma_v^{w,DTS}(t))]}{\sum_v \varsigma_v^{w,DTS}(t)}, \tag{8}$$

where c_d is the unit cost of DTS delay (\$/s).

3) Slice Reconfiguration Cost: In different slicing windows, the SDN controller may need to adjust the spectrum resources allocated to different slices, rendering the slice reconfiguration cost[19]. Considering that resource release can be easily accomplished with negligible cost, the slice reconfiguration cost in slicing window w, denoted by RC^w, is expressed as

$$RC^w = \sum_{s \in \mathcal{S}} \sum_{ap \in \mathcal{AP}} c_r B_{ap} \max\{r_{ap}^{w,s} - r_{ap}^{w-1,s}, 0\}, \tag{9}$$

where c_r is the unit cost of resource reconfiguration (\$/Hz).

Combining (7)-(9), the overall system cost in slicing window w is defined as

$$C_{sys}^w = \alpha \frac{1}{T} \sum_{t=1}^{T} PC_t^w + \beta \frac{1}{T} \sum_{t=1}^{T} DC_t^w + \rho RC^w, \tag{10}$$

where parameters α, β, and ρ control the relative importance of the three types of costs.

The JRSS problem to minimize the long-term overall system cost is formulated as

$$\mathcal{P}_0: \quad \min_{\{\boldsymbol{r^w}, \boldsymbol{a_t^w}, \boldsymbol{b_t^w}\}} E\left[\lim_{W\to\infty} \frac{1}{W} \sum_{w=1}^{W} C_{sys}^w\right] \tag{11}$$

$$\text{s.t.} \quad 0 \leqslant r_{ap}^{w,s} \leqslant 1, \quad \sum_{s \in \mathcal{S}} r_{ap}^{w,s} = 1, \tag{11a}$$

$$a_{ap,v}^{w,s}(t) \in \{0,1\}, \tag{11b}$$

$$0 \leqslant B_{ap,v}^{w,s}(t) \leqslant r_{ap}^{w,s} B_{ap}, \tag{11c}$$

$$\sum_{v} a_{ap,v}^{w,s}(t) B_{ap,v}^{w,s}(t) \leqslant r_{ap}^{w,s} B_{ap}, \tag{11d}$$

$$a_{ap,v}^{w,s}(t) \leqslant \mathbb{1}_{\varsigma_v^{w,s}(t)>0} \mathbb{1}_{T_{ap,v}^{rem}(t)>0}, \tag{11e}$$

$$a_{ap,v}^{w,s}(t) \phi_{ap,v}^{w}(t) \geqslant a_{ap,v}^{w,s}(t) \phi_{th}, \tag{11f}$$

$$(2), (3), \text{and } (4),$$

where $\boldsymbol{r^w} = \{r_{ap}^{w,s}\}$, $\boldsymbol{a_t^w} = \{a_{ap,v}^{w,s}(t)\}$, $\boldsymbol{b_t^w} = \{B_{ap,v}^{w,s}(t)\}$, $\forall ap \in \mathcal{AP}, v \in \mathcal{V}, s \in \mathcal{S}, t \in \mathcal{T}$, and ϕ_{th} is the minimum spectrum efficiency requirement for correct data detection at the receiving vehicle. Constraint (11a) is the resource slicing constraint to ensure that resources allocated to all the slices should not exceed the resource capacity at each AP. Constraints (11b)-(11d) guarantee the feasibility of vehicle-AP association and bandwidth allocation decisions, where the resource slicing and scheduling decisions are tightly coupled. Constraint (11e) means that vehicle v can be associated with ap only when v has service requests and v is within ap's coverage area. Constraint (11f) ensures that a v-ap association is feasible only when the spectral efficiency satisfies $\phi_{ap,v}^{w}(t) \geqslant \phi_{th}$.

III. Design of the *TLRL-JRSS* Scheme

The formulated JRSS problem $\mathcal{P}_0$ belongs to the stochastic optimization due to the spatial-temporal variations in network conditions including vehicle density, service request arrival rates, and the available LEO satellite spectrum resources. In the highly dynamic STIVN, the lack of future network information makes it painstaking, if not impossible, to find the globally optimal solution via traditional optimization approaches. One potential solution is to apply RL-based methods to solve the problem without requiring knowledge of future network conditions. However, the resource slicing and scheduling decisions have different timescales and are tightly coupled, rendering traditional RL algorithms inefficient and difficult to converge. To solve the multi-timescale joint optimization problem with coupled constraints, we propose a *TLRL-JRSS* scheme. Specifically, problem $\mathcal{P}_0$ is the first decomposed into two subproblems: a resource scheduling subproblem and a resource slicing subproblem. In the resource scheduling layer, matching-based resource scheduling algorithms are proposed to determine the vehicle-AP association and bandwidth allocation for vehicular service requests at each time slot. In the resource slicing layer, we leverage a PPO-based RL algorithm to make resource slicing decisions in each slicing window. Fig. 2 shows the working diagram of the proposed *TLRL-JRSS* scheme.

A. Resource Scheduling Subproblem

According to (9), resource scheduling decisions $\boldsymbol{a_t^w}$ and $\boldsymbol{b_t^w}$ have no impact on the slice reconfiguration cost. Therefore, the objective of the resource scheduling subproblem is to minimize the sum of the DSS requirement violation cost and the DTS delay cost in each slicing window. With slice isolation, resource scheduling in DSS and DTS slices is independent. Considering that the information on future network conditions is not available, scheduling decisions $\boldsymbol{a_t^w}$ and $\boldsymbol{b_t^w}$ are optimized based only on the current network state to minimize the instantaneous service cost for DSS and DTS slices, respectively. Thus, resource scheduling in the DSS slice and the DTS slice can be respectively formulated as follows (for notation simplicity, we omit the slicing window superscript w in this subsection)

$$\mathcal{P}_{1-DSS}: \min_{\{a_{ap,v}^{DSS}(t), B_{ap,v}^{DSS}(t)\}} \sum_{v} p_v \mathbb{1}_{\varsigma_v^{DSS}(t)>0} \mathbb{1}_{D_v^{DSS}(\varsigma_v^{DSS}(t))>D_{v,th}(t)}$$
$$\text{s.t.} \quad (2), (3), \text{and } (11b)\text{-}(11f),$$

$$\mathcal{P}_{1-DTS}: \min_{\{a_{ap,v}^{DTS}(t), B_{ap,v}^{DTS}(t)\}} \frac{\sum_v c_d [\mathbb{1}_{\varsigma_v^{DTS}(t)>0} \cdot D_v^{DTS}(\varsigma_v^{DTS}(t))]}{\sum_v \varsigma_v^{DTS}(t)}$$
$$\text{s.t.} \quad (4) \text{ and } (11b)\text{-}(11f).$$

To solve problems $\mathcal{P}_{1-DSS}$ and $\mathcal{P}_{1-DTS}$, we first need to analyze the service delay performance. Due to the lack of knowledge of future network information, the scheduling decisions at each time slot can only be made based on the current/previous network status, to minimize the expected delay. For vehicle v requesting service s with data size $\varsigma_v^s(t)$, the expected service delay is expressed as

$$\begin{aligned} &D_v^s(\varsigma_v^s(t)) = \\ &\max \left\{ \sum_{b_k \in \mathcal{TBS}} a_{b_k,v}^s(t) \left(\frac{\varsigma_{b_k,v}^s(t)}{R_{b_k,v}^s(t)} + \mathbb{1}_{b_k,t}^{s,HO} \mathbb{1}_{t \neq t_0} D_{HO} \right) + \right. \\ &\left. \sum_{l_i \in \mathcal{SAT}} a_{l_i,v}^s(t) \left(\frac{\varsigma_{l_i,v}^s(t)}{R_{l_i,v}^s(t)} + \mathbb{1}_{l_i,t}^{s,HO} \left(D_{l_i,v}^{prop}(t) + \mathbb{1}_{t \neq t_0} D_{HO} \right) \right) \right\} + \\ &\left(1 - \max_{ap \in \mathcal{AP}} \{ a_{ap,v}^s(t) \} \right) D_{\max}, \end{aligned} \tag{12}$$

where $\varsigma_{ap,v}^s(t)$ is the data size scheduled to be delivered from ap to v at time slot t, which satisfies $\sum_{ap} \varsigma_{ap,v}^s(t) = \varsigma_v^s(t)$. For DSS requests satisfying constraint (2), we have $\varsigma_{ap,v}^{DSS}(t) = \varsigma_v^{DSS}(t)$ for the AP with $a_{ap,v}^{DSS}(t) = 1$. Since DTS requests can be served by different types of APs simultaneously according to (4), we have $0 \leqslant \varsigma_{ap,v}^{DTS}(t) \leqslant \varsigma_v^{DTS}(t)$. $\mathbb{1}_{ap,t}^{s,HO}$ is the handover indicator with $\mathbb{1}_{ap,t}^{s,HO} = \mathbb{1}_{a_{ap,v}^s(t)=1} \mathbb{1}_{a_{ap,v}^s(t-1)=0}$, t_0 denotes the time slot when the request starts being served, and D_{HO} is the handover delay. The last term in (12) indicates that when vehicle v is not associated with any APs, the expected service delay is $D_{\max}$, which is a sufficiently large number to penalize the unsuccessful service provisioning. More details on how to solve problems $\mathcal{P}_{1-DSS}$ and $\mathcal{P}_{1-DTS}$ will be illustrated in sections IV.A and IV.B, as shown in Fig. 2.

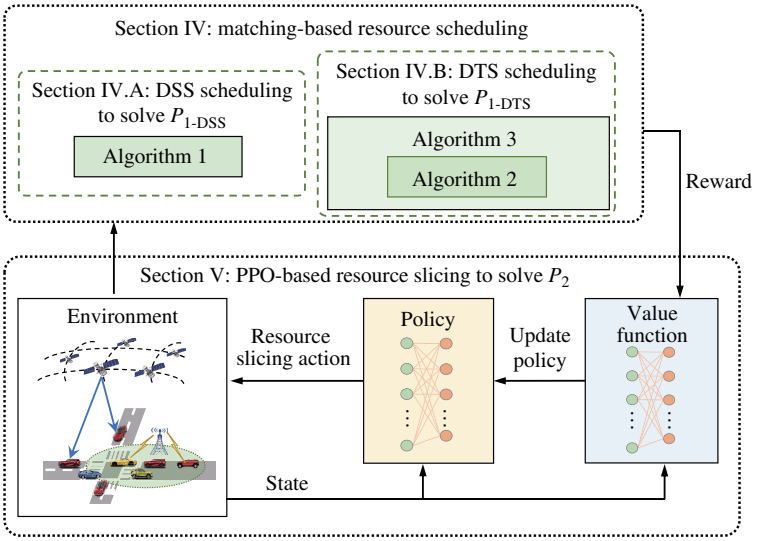


Fig. 2 Working diagram of the proposed *TLRL-JRSS* scheme

B. Resource Slicing Subproblem

Resource slicing decisions $\boldsymbol{r^w}$ are optimized for each slicing window w to minimize the long-term overall system cost, which is formulated as follows

$$\mathcal{P}_2: \quad \min_{\{\boldsymbol{r^w}\}} E\left[\lim_{W\to\infty}\frac{1}{W}\sum_{w=1}^{W} C_{sys}^{w}\right], \tag{13}$$
$$\text{s.t.} \quad (11a).$$

The objective of problem $\mathcal{P}_2$ is to design a resource slicing policy to minimize the expected long-term average system cost. Notice that for any given slicing decision, the resource scheduling within a slicing window can be performed as discussed in sections III.A and IV and the corresponding system cost can be calculated. However, the closed-form expression for the system cost performance is not available and the relationship between the sliced resources and the corresponding performance is unclear. To address this problem, problem $\mathcal{P}_2$ will be first formulated as a Markov decision process (MDP). Then a PPO-based RL algorithm will be proposed to solve the resource slicing subproblem, which will be presented in section V.

IV. Matching-Based Algorithms for Resource Scheduling Layer Subproblem

A. DSS Resource Scheduling

Recall that handover is discouraged during the DSS provisioning to avoid handover delay. According to (12), for a DSS request with data size $\varsigma_v^{DSS}(t)$, the expected service delay can be simplified as

$$D_v^{DSS}(\varsigma_v^{DSS}(t)) = \max\left\{\sum_{b_k\in\mathcal{TBS}} a_{b_k,v}^{DSS}(t)\frac{\varsigma_v^{DSS}(t)}{R_{b_k,v}^{DSS}(t)},\right.$$
$$\left.\sum_{l_i\in\mathcal{SAT}} a_{l_i,v}^{DSS}(t)\left(\frac{\varsigma_v^{DSS}(t)}{R_{l_i,v}^{DSS}(t)}+D_{l_i,v}^{prop}(t)\right)\right\}+$$
$$\left(1-\max_{ap\in\mathcal{AP}}\{a_{ap,v}^{DSS}(t)\}\right)D_{\max}. \tag{14}$$

Therefore, given association decisions, if $\max_{ap\in\mathcal{AP}}\{a_{ap,v}^{DSS}(t)\}=0$, the request will not be served and a penalty should be imposed; Otherwise, the required bandwidth to satisfy the delay constraint $D_v^{DSS}(\varsigma_v^{DSS}(t)) \leqslant D_{v,th}(t)$ can be calculated as

$$B_{ap,v}^{DSS,req}(t) = \sum_{b_k\in\mathcal{TBS}} a_{b_k,v}^{DSS}(t)\frac{\varsigma_v^{DSS}(t)}{D_{v,th}(t)\phi_{b_k,v}(t)}+$$
$$\sum_{l_i\in\mathcal{SAT}} a_{l_i,v}^{DSS}(t)\frac{\varsigma_v^{DSS}(t)}{[D_{v,th}(t)-D_{l_i,v}^{prop}(t)]\phi_{l_i,v}(t)}. \tag{15}$$

The goal of DSS resource scheduling is to minimize the DSS requirement violation cost, which can be rewritten as

$$PC_t = \sum_v p_v \mathbb{1}_{\varsigma_v^{DSS}(t)>0}\mathbb{1}_{D_v^{DSS}(\varsigma_v^{DSS}(t))>D_{v,th}(t)} =$$
$$\sum_v p_v \mathbb{1}_{\varsigma_v^{DSS}(t)>0} - \sum_v p_v \mathbb{1}_{\varsigma_v^{DSS}(t)>0}\mathbb{1}_{D_v^{DSS}(\varsigma_v^{DSS}(t))\leqslant D_{v,th}(t)}.$$

Therefore, minimizing PC_t is equivalent to maximizing $\overline{PC}_t = \sum_v p_v \mathbb{1}_{\varsigma_v^{DSS}(t)>0}\mathbb{1}_{D_v^{DSS}(\varsigma_v^{DSS}(t))\leqslant D_{v,th}(t)}$.

To find the optimal $a_{ap,v}^{DSS}(t)$ and $B_{ap,v}^{DSS}(t)$ to maximize $\overline{PC}_t$ while satisfying the constraints in problem $\mathcal{P}_{1-DSS}$, we first reformulate $\mathcal{P}_{1-DSS}$ into a matching problem. Construct a bipartite graph $\mathcal{G}=(\mathcal{V}_t,\mathcal{AP}_t,\mathcal{E}_t)$, where $\mathcal{V}_t$ is the set of requesting vehicles, $\mathcal{AP}_t$ is the set of APs, and $\mathcal{E}_t$ is the set of edges connecting vertices in $\mathcal{V}_t$ and $\mathcal{AP}_t$. For an edge (v,ap) connecting v and ap ($v\in\mathcal{V}_t, ap\in\mathcal{AP}_t$), we have $(v,ap)\in\mathcal{E}_t$ if and only

if constraints (3), (11e), and (11f) are satisfied. Therefore, the DSS resource scheduling problem is transformed into a matching problem to find a matching $\mathcal{M} \subseteq \mathcal{G}$ to maximize $\overline{PC}_t$, i.e., the sum of the penalty for matched vehicular requests. To satisfy constraints (2) and (11b)-(11d), each vehicle can be matched with at most one AP, while each AP can be matched with multiple vehicles based on its available bandwidth resources. Thus, a matching can be formally presented as

Definition 1 Given two disjoint sets, $\mathcal{V}_t$ of the requesting vehicles, and $\mathcal{AP}_t$ of the APs, a matching $\mathcal{M}$ is a mapping from the set $\mathcal{V}_t \cup \mathcal{AP}_t$ into the set of all subsets of $\mathcal{V}_t \cup \mathcal{AP}_t$ such that for every $v \in \mathcal{V}_t$ and $ap \in \mathcal{AP}_t$, we have: 1) $\mathcal{M}(v) \subseteq \mathcal{AP}_t$ and $\mathcal{M}(ap) \subseteq \mathcal{V}_t$; 2) $v \in \mathcal{M}(ap) \Leftrightarrow ap \in \mathcal{M}(v)$; and 3) $|\mathcal{M}(v) \cap \mathcal{AP}_t| \leqslant 1, \sum_{v \in \mathcal{M}(ap)} B_{ap,v}^{DSS,req}(t) \leqslant r_{ap}^{w,DSS} B_{ap}$.

Then we propose a knapsack-based matching (KBM) algorithm to solve the DSS resource scheduling problem.

Step 1: For each vehicle, construct a preference list by sorting APs based on ascending order of $B_{ap,v}^{DSS,req}(t)$. In other words, vehicles prefer APs that provide higher spectral efficiency.

Step 2: Each vehicle proposes to its current most favorite AP and then removes this AP from its preference list.

Step 3: Each AP checks all the received proposals, including the new proposals and those accepted in previous iterations. Given the set of proposals, each with a weight ($B_{ap,v}^{DSS,req}(t)$) and a value (p_v), each AP needs to determine which proposals to accept such that the total value can be maximized without violating the total weight constraint ($r_{ap}^{w,DSS} B_{ap}$). This is a 0∼1 Knapsack problem, which can be solved based on the dynamic programming approach[20].

Step 4: For all the rejected vehicles, go to Step 2. The matching process terminates when all the vehicles are matched or all the APs bandwidth resources are allocated.

The overall process for DSS resource scheduling is illustrated in Algorithm 1. First, we need to check whether there are previously generated requests that have not been completed and still satisfy the delay constraint. Bandwidth resources are first allocated to these users based on previously decided association decisions (Lines 5-10). Then we apply the KBM algorithm to determine the association and bandwidth allocation for newly generated service requests with the remaining spectrum resources $B_{ap,rem}^{DSS}(t)$ (Lines 11-14).

B. DTS Resource Scheduling

The goal of DTS scheduling at each time slot is to minimize the expected service delay given in (12). With known $a_{ap,v}^{DSS}(t)$ and $B_{ap,v}^{DSS}(t)$, the optimal $\varsigma_{ap,v}^{DTS}(t)$ can be calculated based on Lemma 1 in Ref. [1], as shown below

$$\varsigma_{l_i,v}^{DTS}(t) = \frac{\mathbb{1}_{l_i} R_{l_i,v}^{DTS}(t) \max\{\varsigma_v^{DTS}(t) - D_{diff}(t)\mathbb{1}_{b_k} R_{b_k,v}^{DTS}(t), 0\}}{R_{l_i,v}^{DTS}(t) + \mathbb{1}_{b_k} R_{b_k,v}^{DTS}(t)},$$

Algorithm 1 DSS resource scheduling algorithm

$\mathcal{U}_{rem}^{DSS}(t)$: the set of vehicles whose previously generated DSS requests have not finished before time slot t and the delay constraint has not been violated.
$\mathcal{U}_{new}^{DSS}(t)$: the set of vehicles generating DSS requests at time slot t.
Δ: the duration of one time slot.
Initialization: $t = 0, \mathcal{U}_{rem}^{DSS}(t) = \emptyset$.
Bandwidth allocation for previously generated requests:
for $v \in \mathcal{U}_{rem}^{DSS}(t)$ **do**
  $a_{ap,v}^{DSS}(t) = a_{ap,v}^{DSS}(t-1)$;
  Update the remaining data size
  $\varsigma_v^{DSS}(t) = \varsigma_v^{DSS}(t-1) - \sum_{ap} a_{ap,v}^{DSS}(t-1) R_{ap,v}^{DSS}(t-1)\Delta$;
  Update the delay budget $D_{v,th}(t) = D_{v,th}(t-1) - \Delta$;
  Calculate $B_{ap,v}^{DSS}(t) = B_{ap,v}^{DSS,req}(t)$ based on (15);
  Update the remaining bandwidth resources for ap:
  $B_{ap,rem}^{DSS}(t) = r_{ap}^{w,DSS} B_{ap} - \sum_{v \in \mathcal{U}_{rem}^{DSS}(t)} a_{ap,v}^{DSS}(t) B_{ap,v}^{DSS}(t)$;
end
Bandwidth allocation for newly generated requests:
for $v \in \mathcal{U}_{new}^{DSS}(t)$ **do**
  Calculate $B_{ap,v}^{DSS,req}(t), \forall ap \in \mathcal{AP}_t$ based on (15);
  Apply the **KBM** algorithm to determine $a_{ap,v}^{DSS}(t)$ and $B_{ap,v}^{DSS}(t)$;
  Let $t = t+1$, update $\mathcal{U}_{rem}^{DSS}(t)$, and go back to Line 6;
end
Output: $a_{ap,v}^{DSS}(t)$ and $B_{ap,v}^{DSS}(t)$, $\forall ap \in \mathcal{AP}_t, v \in \mathcal{V}_t, t \in \mathcal{T}$

$$\varsigma_{b_k,v}^{DTS}(t) = \frac{\mathbb{1}_{b_k} R_{b_k,v}^{DTS}(t)[\varsigma_v^{DTS}(t) + \mathbb{1}_{l_i} D_{diff}(t) R_{l_i,v}^{DTS}(t)]}{\mathbb{1}_{l_i} R_{l_i,v}^{DTS}(t) + R_{b_k,v}^{DTS}(t)}, \quad (16)$$

where

$$D_{diff}(t) = \mathbb{1}_{l_i,t}^{DTS,HO} \left(D_{l_i,v}^{prop}(t) + \mathbb{1}_{t \neq t_0} D_{HO} \right) - \mathbb{1}_{b_k,t}^{DTS,HO} \mathbb{1}_{t \neq t_0} D_{HO},$$

$\mathbb{1}_{b_k} = 1$ when $a_{b_k,v}^{DTS}(t) = 1$, otherwise $\mathbb{1}_{b_k} = 0$, $\mathbb{1}_{l_i} = 1$ when $a_{l_i,v}^{DTS}(t) = 1$ and

$$\frac{\varsigma_v^{DTS}(t)}{R_{b_k,v}^{DTS}(t)} + \mathbb{1}_{b_k,t}^{DTS,HO} \mathbb{1}_{t \neq t_0} D_{HO} > \mathbb{1}_{l_i,t}^{DTS,HO} \left(D_{l_i,v}^{prop}(t) + \mathbb{1}_{t \neq t_0} D_{HO} \right),$$

otherwise $\mathbb{1}_{l_i} = 0$.

To minimize the expected service delay, we first analyze the impact of bandwidth allocation on the delay performance with given association decisions. A diminishing gain effect is revealed as shown in the following lemma, the proof of which can be found in Appendix B.

Lemma 1 (Diminishing Gain Effect) For bandwidth allocation in each AP (i.e., TBS or satellite), with more bandwidth resources allocated to a vehicular user, the delay performance gain (i.e., delay decrement) diminishes.

Due to the diminishing gain, allocating a lot of bandwidth resources to the same user is undesirable. To minimize the

Algorithm 2 Greedy-based bandwidth allocation with known $\boldsymbol{a_t}$ for DTS

$\Delta D_{ap,v}(\Delta B_{ap})$: the delay performance gain when ap allocates additional bandwidth resource ΔB_{ap} to vehicle v.
Initialization: $\Delta D_{ap,v}(\Delta B_{ap})=0$, $B^{DTS}_{ap,rem}=r^{w,DTS}_{ap}B_{ap}$, $B^{DTS}_{ap,v}=0$, $\forall ap\in\mathcal{AP}, v\in\mathcal{V}$.
for $v\in\mathcal{V}$ **do**
  Find the associated APs (at most one TBS and one LEO satellite) based on $\boldsymbol{a_t}$;
  if *the associated AP satisfies* $B^{DTS}_{ap,rem}>0$ **then**
    Calculate the corresponding $\Delta D_{ap,v}(\Delta B_{ap})$;
  end
end
$(ap^*,v^*)=\arg\max_{ap,v}\Delta D_{ap,v}(\Delta B_{ap})$;
$B^{DTS}_{ap^*,v^*}=B^{DTS}_{ap^*,v^*}+\Delta B_{ap^*}$;
Update $\Delta D_{ap,v^*}(\Delta B_{ap})$ for vehicle v^*;
$B^{DTS}_{ap^*,rem}=B^{DTS}_{ap^*,rem}-\Delta B_{ap^*}$;
if $B^{DTS}_{ap^*,rem}<=0$ **then**
  $\Delta D_{ap^*,v}(\Delta B_{ap})=0, \forall v\in\mathcal{V}$;
end
Go to Line 7 and repeat until bandwidth resources deplete;
Calculate $D^{DTS}_v(\varsigma^{DTS}_v)$ based on (12) with known $\boldsymbol{a_t}$ and the obtained $B^{DTS}_{ap,v}$;
Output: $D^{DTS}_v(\varsigma^{DTS}_v)$ and $B^{DTS}_{ap,v},\forall ap\in\mathcal{AP}, v\in\mathcal{V}$

overall expected service delay, i.e., maximizing the overall delay performance gain, we implement bandwidth allocation in the units of sub-channels with bandwidth ΔB_{ap}. In specific, we propose a greedy-based bandwidth allocation algorithm with known $\boldsymbol{a}_t$ considering the diminishing gain effect, as shown in Algorithm 2 ((t) is omitted in the algorithm for notational simplicity).

The next step is to optimize $a^{DSS}_{ap,v}(t)$, $\forall ap\in\mathcal{AP}, v\in\mathcal{V}, t\in\mathcal{T}$ to minimize the overall delay for all the requesting vehicles. Note that problem $\mathcal{P}_{1-DTS}$ is non-convex due to the binary constraints and the interdependent association decisions for different vehicles. In addition, the closed-form expression for $D^{DTS}_v(\varsigma^{DTS}_v(t))$ is not available when using Algorithm 2 for bandwidth allocation. Therefore, it is highly complex to solve this problem by utilizing the conventional centralized exhaustive method, especially in a dense network. Next, we develop a matching-based algorithm with externalities to solve problem $\mathcal{P}_{1-DTS}$.

Similar to section IV.A, we can construct a weighted bipartite graph $\mathcal{G}'=(\mathcal{V}_t,\mathcal{AP}_t,\mathcal{E}_t)$. For an edge (v,ap) connecting vehicle v and ap ($v\in\mathcal{V}_t, ap\in\mathcal{AP}_t$), we have $(v,ap)\in\mathcal{E}_t$ if and only if constraints (11e)-(11f) are satisfied. Therefore, the DTS scheduling problem is transformed into a matching problem to find a matching $\mathcal{M}\subseteq\mathcal{G}'$ to minimize the sum of the expected delay for all vehicles. The definition of $\mathcal{M}$ is similar to Definition 1, with condition *3)* modified as $|\mathcal{M}(v)\cap\mathcal{SAT}_t|\leqslant 1$, $|\mathcal{M}(v)\cap\mathcal{TBS}_t|\leqslant 1$, $\mathcal{AP}_t=\mathcal{SAT}_t\cup\mathcal{TBS}_t$ to satisfy the association constraints (4) and (11b).

Algorithm 3 Swap-based matching algorithm for association optimization

Step 1: Initialization Phase
Let $\mathcal{M}=\emptyset$.
for $v\in\mathcal{V}_t$ **do**
  Match v with the most preferred TBS and LEO satellite with the largest $\phi_{b_k,v}$ and $\phi_{l_i,v}$. $\mathcal{M}=\mathcal{M}\cup\{(v,b_k),(v,l_i)\}$;
end
Step 2: Swap Matching Phase
for $ap\in\mathcal{AP}_t$ *with ascending order of* $|\mathcal{M}(ap)|$ **do**
  $\mathcal{M}'_v=\mathcal{M}\setminus\{(v,\mathcal{M}(v))\}\cup\{(v,ap)\},\forall v\in\mathcal{V}_t$;
  Calculate $U_{sys}(\mathcal{M}'_v),\forall v\in\mathcal{V}_t$ based on **Definition 2** and **Algorithm 2**;
  $v^*=\arg\min_v U_{sys}(\mathcal{M}'_v)$;
  if $U_{sys}(\mathcal{M}'_{v^*})<U_{sys}(\mathcal{M})$ **then**
    $\mathcal{M}=\mathcal{M}'_{v^*}$ and go back to Line 7;
  end
end
Search for a pair $(v_1,ap_1),(v_2,ap_2)\in\mathcal{M}$ satisfying $ap_1,ap_2\in\mathcal{SAT}_t$ or $ap_1,ap_2\in\mathcal{TBS}_t$;
Let $\mathcal{M}=\mathcal{M}^{v_2,ap_2}_{v_1,ap_1}$ if $U_{sys}(\mathcal{M}^{v_2,ap_2}_{v_1,ap_1})<U_{sys}(\mathcal{M})$;
Go back to Line 12 and repeat until no swapping pairs can be found;
Output: $\mathcal{M}$

In the matching between vehicles and APs, vehicle v's preference over AP ap is determined by the achievable delay performance, which is affected by other vehicles associated with ap. This type of matching is called the matching game with externalities[21], where the preferences of vehicles not only depend on the APs that they are matched with, but also the other vehicles associated with the same AP. This is different from the conventional matching games in which players have fixed preference lists. Therefore, we propose a swap-based matching algorithm to solve the problem.

Definition 2 Given a matching $\mathcal{M}$, the system delay function is $U_{sys}(\mathcal{M})=\sum_{(v,\mathcal{M}(v))\in\mathcal{M}}D^{DTS}_v(\varsigma^{DTS}_v(t))$, where $D^{DTS}_v(\varsigma^{DTS}_v(t))$ can be obtained according to Algorithm 2.

Definition 3 Given a matching $\mathcal{M}$ and two pairs $(v_1,ap_1),(v_2,ap_2)\in\mathcal{M}$, a swap matching is

$$\mathcal{M}^{v_2,ap_2}_{v_1,ap_1}=\mathcal{M}\setminus\{(v_1,ap_1),(v_2,ap_2)\}\cup\{(v_1,ap_2),(v_2,ap_1)\}.$$

The procedure of our proposed swap-based matching algorithm for the association optimization is shown in Algorithm 3. In the initialization phase, a greedy-based association method is applied, where each vehicle is matched with the APs with the best channel qualities. This may, however, lead to load imbalance among APs. Therefore, in the swap matching phase, we first sort the APs based on the ascending order of the number of associated vehicles. Starting from the least crowded AP, each vehicle checks whether it can switch its currently associated AP to this AP to reduce the system delay. Then the vehicle with the minimum system delay is selected for AP association swapping. The iterations stop when

no AP association swapping can be found to further reduce the system delay. Then, swapping pairs are searched among different vehicles, and the swap matching is performed if the system delay can be further reduced. The iterations continue until no swapping pairs can be found.

V. PPO-Based RL Algorithm for Resource Slicing Layer Subproblem

First, we model the resource slicing problem $\mathcal{P}_2$ as an MDP. Define the set of all possible *actions* as $\boldsymbol{\Psi}$. Corresponding to (13), the action is to determine the spectrum resource slicing ratios. Thus, the action in slicing window w is

$$\boldsymbol{r}^w = \{r_{ap}^{w,s} \mid \forall ap \in \mathcal{AP}, s \in \mathcal{S}\} \in \boldsymbol{\Psi}. \tag{17}$$

Note that the action space is continuous with $\boldsymbol{r}^w \in [0,1]^{|\mathcal{AP}|\times|\mathcal{S}|}$, which should satisfy constraint (11a).

The set of all possible *states* is denoted by $\boldsymbol{\Xi}$. The system state contains information on service data packet arrival rate, vehicle density, the number of LEO satellites covering the target area[2], and the spectrum resource slicing decisions in the previous slicing window. Denote by $\overline{N}_L^w$ the average number of LEO satellites covering the target area in slicing window w. We define the state in slicing window w as

$$\boldsymbol{\xi}^w = \left\{\{\lambda_s^w\}_{s\in\mathcal{S}}, \{\rho_{ap}^w\}_{ap\in\mathcal{AP}}, \overline{N}_L^w, \boldsymbol{r}^{w-1}\right\} \in \boldsymbol{\Xi}. \tag{18}$$

When choosing action $\boldsymbol{r}^w$ in state $\boldsymbol{\xi}^w$, the probability that the system evolves into state $\boldsymbol{\xi}^{w+1}$ is the state transition probability (STP), denoted by $\boldsymbol{P}(\boldsymbol{\xi}^{w+1}|\boldsymbol{\xi}^w, \boldsymbol{r}^w)$. To evaluate the performance of taking an action with the given state, the *reward* function is defined as $-C_{sys}^w(\boldsymbol{\xi}^w, \boldsymbol{r}^w)$, which can be obtained based on (10) and the resource scheduling algorithms proposed in section IV.

Since the action space is continuous, we assume that the action $\boldsymbol{r}^w$ is drawn from a stochastic policy $\boldsymbol{\pi}(\boldsymbol{r}|\boldsymbol{\xi}) = \Pr(\boldsymbol{r}^w = \boldsymbol{r}|\boldsymbol{\xi}^w = \boldsymbol{\xi})$, which is a mapping from the state to the probabilities of taking actions. Let $\boldsymbol{\Pi}$ be the set of all policies. Then, the problem of minimizing the long-term average system cost in problem $\mathcal{P}_2$ is approximated by minimizing the long-term accumulated discounted reward, i.e.,

$$\mathcal{P}_2': \quad \max_{\boldsymbol{\pi}\in\boldsymbol{\Pi}} E\left[\lim_{W\to\infty}\sum_{w=1}^{W} -\gamma^w C_{sys}^w(\boldsymbol{\xi}^w, \boldsymbol{r}^w)\middle|\boldsymbol{\pi}\right], \tag{19}$$
$$\text{s.t.} \quad (11a),$$

[2]Since the available spectrum resources in different satellites are known a priori and are the same in this work, the average number of available satellites is sufficient to characterize the average available satellite resources for the target area. However, if the available resources vary for different satellites, we should use the average available satellite resources instead of the average number of available satellites to describe the system status.

where γ is the discount factor representing that more emphasis is put on the current reward than the future reward. Note that problem $\mathcal{P}_2$ can be well approximated by $\mathcal{P}_2'$ as the discount factor approaches one [22,23]. Considering the lacking information on the STP and the continuous action space, we adopt a policy gradient (PG)-based RL algorithm to efficiently solve the problem without requiring the STP information.

Basically, PG methods directly update the policy to maximize the expected total reward using policy gradient, which is estimated by using data collected from the environment. PG methods have a low data efficiency because data collected using the current policy can be used for only one gradient update, after which new data is required to estimate the gradient with respect to the updated parameters[24]. To improve data efficiency, importance sampling techniques[25] can be applied such that the policy gradient can be calculated using an old policy and then recalibrated by the policy ratio

$$\varphi_\theta = \frac{\boldsymbol{\pi}_\theta(\boldsymbol{r}|\boldsymbol{\xi})}{\boldsymbol{\pi}_{\theta_{old}}(\boldsymbol{r}|\boldsymbol{\xi})}, \tag{20}$$

where $\boldsymbol{\pi}_\theta$ is a policy parameterized by θ.

Considering that PG methods suffer from high variance, which often leads to destructively large policy updates during learning, PG methods can still be extremely unstable. To improve performance reliability and data efficiency of learning, trust region policy optimization (TRPO) was proposed[26], which, however, has high complexity and poor scalability. PPO, which is a variant of TRPO, was proposed in Ref. [27] to optimize a clipped objective function. Specifically, the PPO algorithm aims to find the optimal policy that maximizes the following function.

$$L^{clip}(\theta) = E_{\boldsymbol{\xi},\boldsymbol{r}}\left[\min\left(\varphi_\theta A^{\boldsymbol{\pi}_{\theta_{old}}}, \text{clip}\left(\varphi_\theta, 1-\varepsilon, 1+\varepsilon\right)A^{\boldsymbol{\pi}_{\theta_{old}}}\right)\right], \tag{21}$$

where $A^{\boldsymbol{\pi}_{\theta_{old}}}$ is the advantage function for policy $\boldsymbol{\pi}_{\theta_{old}}$ and can be expressed as

$$A^{\boldsymbol{\pi}_{\theta_{old}}}(\boldsymbol{\xi}^w, \boldsymbol{r}^w) = -C_{sys}^w(\boldsymbol{\xi}^w, \boldsymbol{r}^w) + \gamma V_\vartheta\left(\boldsymbol{\xi}^{w+1}\right) - V_\vartheta\left(\boldsymbol{\xi}^w\right). \tag{22}$$

$V_\vartheta(\boldsymbol{\xi}^w)$ is the state-value function parameterized by ϑ

$$V_\vartheta(\boldsymbol{\xi}^w) = E\left\{\sum_{k=0}^{\infty} -\gamma^k C_{sys}^{w+k+1}\middle|\boldsymbol{\xi}^w\right\}. \tag{23}$$

ε in (21) is a clipping hyperparameter. The second term in (21), $\text{clip}\left(\varphi_\theta, 1-\varepsilon, 1+\varepsilon\right)A^{\boldsymbol{\pi}_{\theta_{old}}}$, clips the ratio φ_θ at $1-\varepsilon$ or $1+\varepsilon$ depending on whether the advantage $A^{\boldsymbol{\pi}_{\theta_{old}}}$ is negative or positive. The clipping removes incentives for the new policy to get far from the old policy, which improves performance stability with controllable policy updates.

Algorithm 4 PPO-based resource slicing algorithm

Initialize policy $\boldsymbol{\pi}$ with parameter θ.
Initialize value function V with parameter ϑ.
Initialize buffer $\mathcal{D}$.
for $k \in \{1,2,\cdots\}$ **do**
  for $w \in \{1,2,\cdots,W\}$ **do**
    Observe state $\boldsymbol{\xi}^w$;
    Take action $\boldsymbol{r}^w$ based on policy $\boldsymbol{\pi}^k = \boldsymbol{\pi}(\theta_k)$;
    Compute reward $-C^w_{sys}$ and observe new state $\boldsymbol{\xi}^{w+1}$;
    Record $(\boldsymbol{\xi}^w, \psi^w, -C^w_{sys}, \boldsymbol{\xi}^{w+1})$ into buffer $\mathcal{D}$;
  end
  Collect set of trajectories $\mathcal{D}_k$ by running policy $\boldsymbol{\pi}^k$;
  Compute advantage estimates $\{A^{\boldsymbol{\pi}_k}\}^W_{w=1}$ based on current value function V_{ϑ_k};
  Update the policy by maximizing the PPO-clip objective

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k| W} \sum_{\tau \in \mathcal{D}_k} \sum_{w=0}^{W} \min\left(\frac{\boldsymbol{\pi}_\theta(\boldsymbol{r}|\boldsymbol{\xi})}{\boldsymbol{\pi}_{\theta_k}(\boldsymbol{r}|\boldsymbol{\xi})} A^{\boldsymbol{\pi}_{\theta_k}}(\boldsymbol{\xi}^w, \boldsymbol{r}^w), \right.$$
$$\left. \text{clip}\left(\frac{\boldsymbol{\pi}_\theta(\boldsymbol{r}|\boldsymbol{\xi})}{\boldsymbol{\pi}_{\theta_k}(\boldsymbol{r}|\boldsymbol{\xi})}, 1-\varepsilon, 1+\varepsilon \right) A^{\boldsymbol{\pi}_{\theta_k}}(\boldsymbol{\xi}^w, \boldsymbol{r}^w) \right)$$

  via gradient ascent methods;
  Fit value function by regression on mean-squared error

$$\vartheta_{k+1} = \arg\min_{\vartheta} \frac{1}{|\mathcal{D}_k| W} \sum_{\tau \in \mathcal{D}_k} \sum_{w=0}^{W} \left(V_\vartheta(\boldsymbol{\xi}^w) + C^w_{sys} \right)^2$$

  via gradient descent methods;
end

In this work, we leverage the PPO algorithm for resource slicing due to its outstanding performance and low complexity. The overall PPO-based resource slicing algorithm is demonstrated in Algorithm 4. First, the policy and the value function are initialized with parameters θ and ϑ, respectively. The buffer memory $\mathcal{D}$ is used to store the trajectories (i.e., a sequence of transitions $(\boldsymbol{\xi}^w, \boldsymbol{r}^w, -C^w_{sys}, \boldsymbol{\xi}^{w+1})$) of interacting with the environment (Lines 1∼3). Then, at each iteration, the SDN controller observes the state, takes actions based on the current policy, computes the reward, and observes the new state (Lines 4∼9). After collecting a set of trajectories, the advantage estimates can be calculated based on (22). Then the policy is updated based on the PPO-clip objective and the value function is updated to minimize the mean-square error (Lines 11∼13).

VI. Performance Evaluation

In this section, we conduct trace-driven simulations to evaluate the performance of the proposed *TLRL-JRSS* scheme. We adopt the Didi Chuxing GAIA Initiative dataset, which includes taxi GPS traces within the second ring road in Xi'an from 1 October 2016 to 31 October 2016 (31 days)[28]. The simulation scenario includes 10 LTE TBSs and 2 LEO satellite orbits. We consider Starlink satellites with an orbit height of 550 km, and each orbit has 60 satellites. The minimum communication elevation angle is set to be 30°. Therefore, at each time instant, there could be 2 or 3 satellites available for the target simulation scenario in each orbit. Following[29], LEO satellite communication parameters are set to be $P_{l_i,v} =$ 10 dBW with a transmission antenna gain of 20 dBi and a receiver gain of 30 dBi. The transmit power of LTE TBSs is set to be 28 dBm. The pathloss exponents for T2V and S2V communications are set to be 3.5 and 2.5, respectively[18]. At each time slot, each vehicle generates DSS and DTS service requests following Poisson processes with mean λ^w_{DSS} and λ^w_{DTS}, respectively. λ^w_{DSS} and λ^w_{DTS} vary in different slicing windows and are randomly chosen from $[1,10]$ requests/s. Main simulation parameters are summarized in Tab. 2. Notice that the exemplary weight parameters for the DSS requirement violation cost, the DTS delay cost, and the slice reconfiguration cost are set to be $\{1,1\,000,1\}$[3], but the values of these weighting factors can be adjusted based on the preference of the network operator.

Tab. 2 Simulation parameters

$D_{v,th}$, ϕ_{th}	10 ms, 5 dB
λ^w_{DSS}	[1, 10] requests/s
λ^w_{DTS}	[1, 10] requests/s
ς_{DSS}, ς_{DTS}	1 500 byte, 10 Mbit/s
B_{b_k}, B_{l_i}	100 MHz, 500 MHz
Duration of a slicing window	10 min
Duration of a scheduling time slot	1 s
$p^w_v(t)$, c_d, c_r	\$1/number, \$1/s, \$1/MHz
α, β, ρ	1, 1 000, 1

First, we evaluate the effectiveness of the matching-based resource scheduling algorithms in the *TLRL-JRSS* scheme. The following benchmark algorithms are considered for performance comparison:

- *Best SNR association (BSA):* All the vehicles are associated with the APs with the best SNR;
- *Random association (RA):* All the vehicles are randomly associated with the APs;
- *Equal bandwidth allocation (EBA):* The bandwidth resources of APs are equally allocated to all the associated vehicles;
- *Min-max bandwidth allocation (MMBA):* the bandwidth resources of APs are allocated such that the maximum delay experienced by all the associated vehicles is minimized.

For BSA and RA algorithms, the bandwidth allocation in

[3] Since the average delay per unit data size is a very small number, the weight for DTS delay cost is set to be 1 000 to make different types of costs in the same order of magnitude.

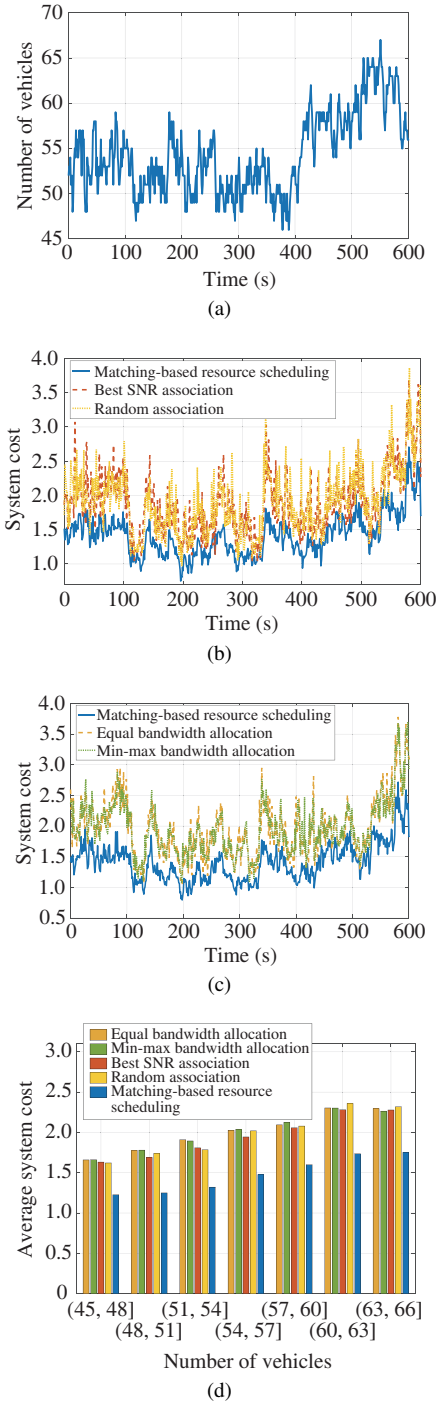


Fig. 3 System costs for different association and bandwidth allocation algorithms: (a) number of vehicles; (b) comparison of association algorithms; (c) comparison of bandwidth allocation algorithms; (d) average system cost vs. different numbers of vehicles

DSS and DTS slices is optimized following Algorithms 1 and 2, respectively. For EBA and ETBA algorithms, the optimization of the vehicle-AP association in DSS and DTS slices is the same as Algorithms 1 and 3, respectively.

Fig. 3 shows the system cost, including the DSS requirement violation cost and the DTS delay cost, in a slicing window with different association and bandwidth allocation algorithms. The number of vehicles at different time slots within the slicing window is depicted in Fig. 3(a). As shown in Fig. 3(b), the system cost varies with time due to the highly dynamic number of vehicles and service requests. Although the BSA and RA algorithms can achieve good cost performance at some time slots, the proposed matching-based resource scheduling algorithm always outperforms the benchmark algorithms with the lowest cost, especially when the number of vehicles increases. The reason is that the proposed matching-based scheduling algorithm considers not only the channel quality, but also the resource constraints and potential competition among vehicles to guarantee balanced user association with good delay performance.

Fig. 3(c) shows the system cost of different bandwidth allocation algorithms in a slicing window. Similar to the results in Fig. 3(b), the proposed matching-based resource scheduling algorithm significantly outperforms the EBA and MMBA algorithms. For the EBA algorithm, the impact of vehicles' channel conditions and the requested data size is ignored, leading to unsatisfactory delay performance for vehicles with bad channel conditions or large requested data sizes. The MMBA algorithm, on the other hand, allocates bandwidth resources to ensure that all the service requests in the same slice can experience the same delay performance. Therefore, the existence of users with bad channel conditions can significantly degrade the overall system performance. The proposed matching-based resource scheduling algorithm considers each service request's delay requirement and the diminishing gain effect to guarantee superior delay performance. To better demonstrate the performance of different algorithms, the average system cost performance is also provided with different numbers of vehicles, as shown in Fig. 3(d). With an increasing number of vehicles, the average system costs for all the resource scheduling algorithms show an increasing trend due to the resource competition. However, the proposed matching-based resource scheduling algorithm always outperforms the benchmark resource scheduling algorithms with different numbers of vehicles.

The effectiveness of resource slicing in the proposed *TLRL-JRSS* scheme is also evaluated. We compare the *TLRL-JRSS* scheme with the resource allocation scheme without resource slicing (i.e., without resource reservation for different types of services). The number of vehicles in the scenario is shown in Fig. 3(a). Comparing with Fig. 4(a) and Fig. 4(b), we can see that when the number of vehicles in the scenario is small (i.e., light traffic demand), the resource allocation scheme without slicing can achieve similar system cost performance with the proposed *TLRL-JRSS* scheme. However, with a number of vehicles, the DSS delay requirements cannot be guaranteed for the scheme without slicing because the data-intensive DTSs

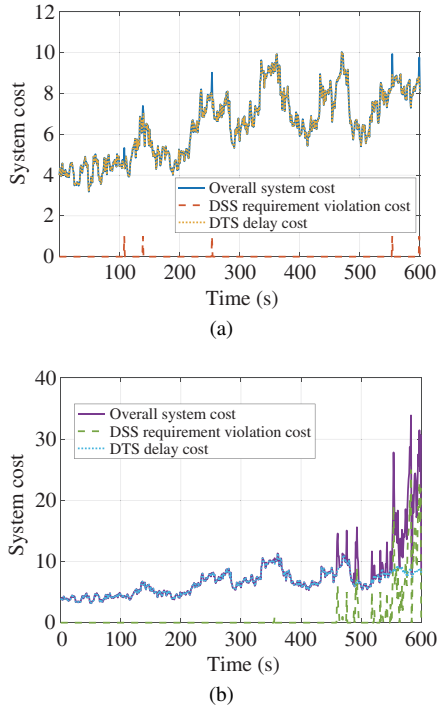


Fig. 4 System cost for the *TLRL-JRSS* scheme and the resource allocation algorithm without slicing: (a) proposed *TLRL-JRSS* scheme; (b) resource allocation without slicing

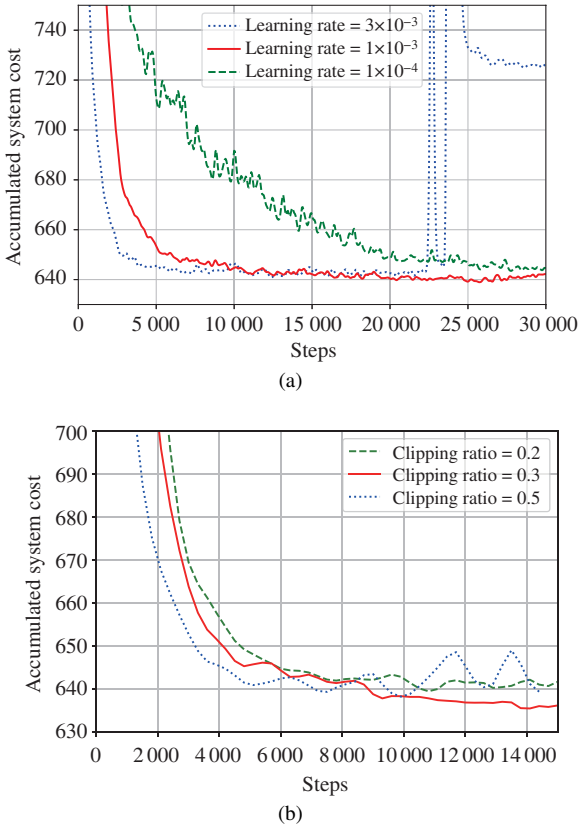


Fig. 5 Convergence performance of the *TLRL-JRSS* scheme with different learning rates and clipping ratios: (a) impact of learning rates; (b) impact of clipping ratios

may exhaust the limited resources. On the contrary, with resource slicing and reservation, the *TLRL-JRSS* scheme can effectively guarantee the DSS delay requirements under different traffic demand conditions without deteriorating the performance of the DTSs.

When applying the PPO-based RL algorithm to make resource slicing decisions, the convergence performance is shown in Fig. 5. The accumulated system cost over training steps with different learning rates is shown in Fig. 5(a). We can observe that all the curves converge to a similar overall accumulated system cost. With a larger learning rate, the system cost performance converges faster. However, when the learning rate becomes too large, performance collapse might happen, e.g., as shown in the case with learning rate being 3×10^{-3}. Therefore, in the remaining part of this section, the learning rate for the PPO-based algorithm is set to be 1×10^{-3} [4]. The impact of the PPO clipping ratio ε on the convergence performance is shown in Fig. 5(b). With a larger ε, the system cost converges faster since the policy can be updated more aggressively in each step, but the large policy update leads to unstable performance. On the contrary, a small ε enforces conservative policy updates, leading to slower performance convergence and non-optimal converged system cost. Therefore, ε is set to be 0.3 in the remaining part of this section.

We also compare the system cost (the overall cost and the corresponding DSS violation cost, DTS delay cost, and slice reconfiguration cost) of the proposed *TLRL-JRSS* scheme with that of the proportional slicing scheme. In the proportional slicing scheme, resource slicing ratios are determined based on the proportion of resource demands from the DSS slice and the DTS slice. Fig. 6(a) shows the number of vehicles in the target scenario in 35 slicing windows, and the costs of the two schemes are depicted in Figs. 6(b)-6(d). As shown in Fig. 6(b), with different values of ρ (the weighting factor for slice reconfiguration cost), the slicing decisions made by the proportional slicing scheme keep unchanged, leading to a substantial difference in the overall system cost. On the other hand, the *TLRL-JRSS* scheme can adjust the slicing ratio decisions based on the cost considerations to reduce the overall cost. For instance, when more emphasis is put on the slice reconfiguration cost with $\rho = 10$, the resource slicing decisions made by the *TLRL-JRSS* scheme have a lower slice reconfiguration cost than the case with $\rho = 1$, as shown in Figs. 6(c) and 6(d). We can also observe that the proposed learning-based *TLRL-JRSS* scheme can significantly outper-

[4]Note that the learning rate can also be adjusted during the training process by using existing learning rate tuning methods, e.g., the adaptive learning rate method [30] or the cyclical learning rate method[31].

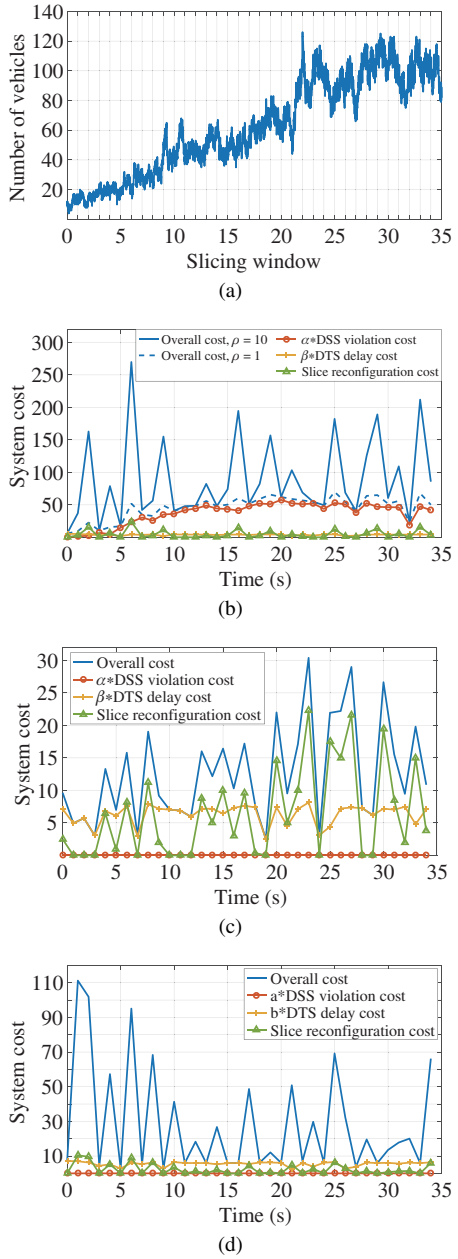
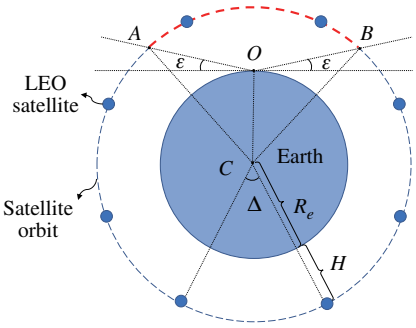


Fig. 6 Performance comparison between the *TLRL-JRSS* scheme and the proportional slicing scheme: (a) number of vehicles; (b) proportional slicing scheme; (c) *TLRL-JRSS* scheme, $\rho = 1$; (d) *TLRL-JRSS* scheme, $\rho = 10$

form the proportional slicing scheme in terms of the overall system cost. Specifically, the overall system costs of the *TLRL-JRSS* scheme for the cases with $\rho = 1$ and $\rho = 10$ are 72.57% and 55.07%, on average, less than that of the proportional slicing scheme, and the maximum overall cost reduction can reach 95.59% and 98.22%, respectively.

VII. Conclusions and Future Works

In this paper, we have investigated the JRSS problem in the STIVN to support DSSs and DTSs with diversified QoS requirements. In specific, we have proposed the *TLRL-JRSS* scheme to jointly optimize the spectrum resource slicing and scheduling in different timescales with the objective of minimizing the long-term system cost. By adopting the proposed scheme, heterogeneous network resources in the STIVN can be efficiently exploited to fully unleash their differential merits in supporting diversified services. The proposed scheme is also adaptive to time-varying network conditions without requiring future information. Besides, we believe the principle of integrating optimization approaches and RL algorithms can be valuable for other resource management problems in future dynamic and complicated networks. For future work, we will further consider the joint optimization of communication, caching, and computing resource slicing and scheduling for diversified service provisioning with enhanced network performance.



R_e and H: Earth radius and satellite altitude
ε: the minimum elevation angle
N_{LEO}: number of LEO satellites in each orbit
O: the observation point

Fig. 7 Illustration of a satellite orbit

Appendix

A) Calculation of the number of an available satellites To justify the model for the number of available satellites, we illustrate a satellite orbit as shown in Fig. 7. In an orbit with N_{LEO} LEO satellites, the angular separation between two adjacent LEO satellites is $\Delta = 2\pi/N_{LEO}$. When the required minimum elevation angle for communication is ε, a satellite is available for the observation point only when it is located within the red arc $\overset{\frown}{AB}$. The distance between O and A (or B) can be calculated as

$$R_e^2 + d_{OA}^2 - 2R_e d_{OA}\cos\left(\varepsilon + \frac{\pi}{2}\right) = (R_e + H)^2 \Rightarrow$$

$$d_{OA} = \sqrt{H^2 + 2HR_e + R_e^2\sin^2\varepsilon} - R_e\sin\varepsilon.$$

The angle $\angle ACO$ is expressed as

$$\angle ACO = \arccos\frac{R_e^2 + (R_e + H)^2 - d_{OA}^2}{2R_e(R_e + H)} =$$

$$\arccos \frac{R_e - R_e \sin^2 \varepsilon + \sin \varepsilon \sqrt{H^2 + 2HR_e + R_e^2 \sin^2 \varepsilon}}{R_e + H}.$$

Therefore, the number of available satellites in an orbit is

$$N_{avail} = \frac{2\angle ACO}{\Delta} = \frac{2\angle ACO \cdot N_{LEO}}{2\pi}.$$

B) Proof of Lemma 1 For notational simplicity, we omit (t) and the superscript DTS in the proof. Let $\boldsymbol{b}_0$ be the initial bandwidth allocation result. The optimal content delivery ratio and the corresponding expected service delay can be calculated based on (16) and (12), respectively. Taking T2V communications as an example, for vehicle v which is connected with TBS b_k with the given $\boldsymbol{a}$ and $\boldsymbol{b}_0$, its expected service delay is

$$D_v(\boldsymbol{a}, \boldsymbol{b}_0) = \frac{\varsigma_{b_k,v}}{R_{b_k,v}} = \frac{\varsigma_v + \mathbb{1}_{l_i} D_{diff} R_{l_i,v}}{\mathbb{1}_{l_i} R_{l_i,v} + B^0_{b_k,v} \phi_{b_k,v}},$$

where $B^0_{b_k,v}$ is the bandwidth allocated from TBS b_k to vehicle v with the given $\boldsymbol{b}_0$.

For a new bandwidth allocation decision $\boldsymbol{b}'$, in which TBS b_k allocates an extra bandwidth of $\Delta B_{b_k,v}$ to vehicle v (the other allocation decisions keep the same with $\boldsymbol{b}_0$), the expected service delay for vehicle v is

$$D_v(\boldsymbol{a}, \boldsymbol{b}') = \frac{\varsigma_v + \mathbb{1}_{l_i} D_{diff} R_{l_i,v}}{\mathbb{1}_{l_i} R_{l_i,v} + (B^0_{b_k,v} + \Delta B_{b_k,v}) \phi_{b_k,v}}.$$

When the value of $\mathbb{1}_{l_i}$ keeps unchanged for decisions $\boldsymbol{b}_0$ and $\boldsymbol{b}'$, the delay performance gain (i.e., delay decrement) is

$$\Delta D_{b_k,v}(\Delta B_{b_k,v}) = D_v(\boldsymbol{a}, \boldsymbol{b}_0) - D_v(\boldsymbol{a}, \boldsymbol{b}') = \frac{(\varsigma_v + \mathbb{1}_{l_i} D_{diff} R_{l_i,v}) \phi_{b_k,v} \Delta B_{b_k,v}}{(\mathbb{1}_{l_i} R_{l_i,v} + B^0_{b_k,v} \phi_{b_k,v})(\mathbb{1}_{l_i} R_{l_i,v} + (B^0_{b_k,v} + \Delta B_{b_k,v}) \phi_{b_k,v})}.$$

If the value of $\mathbb{1}_{l_i}$ changes, i.e., $\mathbb{1}_{l_i} = 1$ for $\boldsymbol{b}_0$ and $\mathbb{1}_{l_i} = 0$ for $\boldsymbol{b}'$, then the delay performance gain is

$$\Delta D_{b_k,v}(\Delta B_{b_k,v}) = \frac{\varsigma_v + D_{diff} R_{l_i,v}}{R_{l_i,v} + B^0_{b_k,v} \gamma_{b_k,v}} - \frac{\varsigma_v}{(B^0_{b_k,v} + \Delta B_{b_k,v}) \gamma_{b_k,v}} = \frac{\Delta B_{b_k,v} \gamma_{b_k,v} (\varsigma_v + D_{diff} R_{l_i,v}) - R_{l_i,v} [\varsigma_v - D_{diff} B^0_{b_k,v} \gamma_{b_k,v}]}{[R_{l_i,v} + B^0_{b_k,v} \gamma_{b_k,v}][(B^0_{b_k,v} + \Delta B_{b_k,v}) \gamma_{b_k,v}]}.$$

For the above two cases, the second derivative of $\Delta B_{b_k,v}$ is negative for each of them, which means the delay performance gain is a concave function of $\Delta B_{b_k,v}$. In other words, when TBS b_k allocates more bandwidth to vehicle v, the delay performance gain diminishes. Similarly, when considering bandwidth allocation from each satellite to a vehicle, the diminishing gain effect also exists given that the others' allocation decisions are fixed, which can conclude the proof.

REFERENCES

[1] WU H, CHEN J, ZHOU C, et al. Load- and mobility-aware cooperative content delivery in SAG integrated vehicular networks[C]//Proceedings of the IEEE International Conference on Communications. Piscataway: IEEE Press, 2021.

[2] DI B, SONG L, LI Y, et al. Ultra-dense LEO: integration of satellite access networks into 5G and beyond[J]. IEEE Wireless Communications, 2019, 26(2): 62-69.

[3] WU H, CHEN J, ZHOU C, et al. Resource management in space-air-ground integrated vehicular networks: SDN control and AI algorithm design[J]. IEEE Wireless Communications, 2020, 27(6): 52-60.

[4] 3GPP. 3GPP service requirements for cyber-physical control applications in vertical domains; stage 1 (release 17)[S]. 2019.

[5] SHEN X, GAO J, WU W, et al. AI-assisted network-slicing based next-generation wireless networks[J]. IEEE Open Journal of Vehicular Technology, 2020, 1(1): 45-66.

[6] ZHOU Z, FENG J, ZHANG C, et al. SAGECELL: software-defined space-air-ground integrated moving cells[J]. IEEE Communications Magazine, 2018, 56(8): 92-99.

[7] LIANG Y C, TAN J, JIA H, et al. Realizing intelligent spectrum management for integrated satellite and terrestrial networks[J]. Journal of Communications and Information Networks, 2021, 6(1): 32-43.

[8] LIU R, MA Y, ZHANG X, et al. Deep learning-based spectrum sensing in space-air-ground integrated networks[J]. Journal of Communications and Information Networks, 2021, 6(1): 82-90.

[9] JIANG D, WANG F, LV Z, et al. Qoe-aware efficient content distribution scheme for satellite-terrestrial networks[J]. IEEE Transactions on Mobile Computing, 2021, 4.

[10] WANG G, ZHOU S, NIU Z. Radio resource allocation for bidirectional offloading in space-air-ground integrated vehicular network[J]. Journal of Communications and Information Networks, 2019, 4(4): 24-31.

[11] 3GPP. Technical specification group services and system aspects; telecommunication management; study on management and orchestration of network slicing for next generation network (release 15)[S]. 2018.

[12] 3GPP. Technical specification group services and system aspects; management and orchestration; concepts, use cases and requirements (release 17)[S]. 2021.

[13] LI J, SHI W, YANG P, et al. A hierarchical soft RAN slicing framework for differentiated service provisioning[J]. IEEE Wireless Communications, 2020, 27(6): 90-97.

[14] LYU F, YANG P, WU H, et al. Service-oriented dynamic resource slicing and optimization for space-air-ground integrated vehicular networks[J]. IEEE Transactions on Intelligent and Transportation Systems, 2021, 4(99): 1-15.

[15] ZANZI L, SCIANCALEPORE V, SAAVEDRA A G, et al. LACO: a latency-driven network slicing orchestration in beyond-5G networks[J]. IEEE Transactions on Wireless Communications, 2021, 20(1): 667-682.

[16] ALSENWI M, TRAN N H, BENNIS M, et al. Intelligent resource slicing for eMBB and URLLC coexistence in 5G and beyond: a deep reinforcement learning based approach[J]. IEEE Transactions on Wireless Communications, 2021, 20(7): 4585-4600.

[17] YAN M, FENG G, ZHOU J, et al. Intelligent resource scheduling for 5G radio access network slicing[J]. IEEE Transactions on Vehicular Technology, 2019, 68(8): 7691-7703.

[18] DU J, JIANG C, WANG J, et al. Resource allocation in space multiaccess systems[J]. IEEE Transactions Aerospace and Electronics Systems, 2017, 53(2): 598-618.

[19] WU W, CHEN N, ZHOU C, et al. Dynamic RAN slicing for service-oriented vehicular networks via constrained learning[J]. IEEE Journal

on Selected Areas in Communications, 2021, 39(7): 2076-2089.
[20] MARTELLO S, PISINGER D, and TOTH P. Dynamic programming and strong bounds for the 0-1 knapsack problem[J]. Management Science, 1999, 45(3): 414-424.
[21] ZHAO J, LIU Y, CHAI K K, et al. Many-to-many matching with externalities for device-to-device communications[J]. IEEE Wireless Communications Letter, 2017, 6(1): 138-141.
[22] HE H, SHAN H, HUANG A, et al. Edge-aided computing and transmission scheduling for LTE-U-enabled IoT[J]. IEEE Transactions on Wireless Communications, 2020, 19(12): 7881-7896.
[23] BISHOP C J, FEINBERG E A, ZHANG J. Examples concerning abel and cesaro limits[J]. Journal of Mathematical Analysis and Applications, 2014, 420(2): 1654-1661.
[24] PENG X B, ABBEEL P, LEVINE S, et al. Deepmimic: example-guided deep reinforcement learning of physics-based character skills[J]. ACM Transactions on Graphics (TOG), 2018, 37(4): 1-14.
[25] METELLI A M, PAPINI M, FACCIO F, et al. Policy optimization via importance sampling[J]. arXiv preprint arXiv:1809.06098, 2018.
[26] SCHULMAN J, LEVINE S, ABBEEL P, et al. Trust region policy optimization[J]. Computer Science, 2015: 1889-1897.
[27] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.
[28] DIDI CHUXING GAIA INITIATIVE[EB].
[29] ASSEMBLY, ITU RADIOCOMMUNICATION. Satellite system characteristics to be considered in frequency sharing analyses within the fixed-satellite service[R]. 2002.
[30] DAUPHIN Y N, VRIES H D, BENGIO Y. Equilibrated adaptive learning rates for non-convex optimization[J]. arXiv preprint arXiv:1502.04390, 2015.
[31] SMITH L N. Cyclical learning rates for training neural networks[C]//Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV). Piscataway: IEEE Press, 2017.

About the Authors

Huaqing Wu (S'15-M'21) received the Ph.D. degree from University of Waterloo, Ontario, Canada, in 2021. She received the B.E. and M.E. degrees from Beijing University of Posts and Telecommunications, Beijing, China, in 2014 and 2017, respectively. She received the prestigious Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellowship Award in 2021. She is currently a postdoctoral research fellow at the McMaster University, Ontario, Canada. Her current research interests include vehicular networks with emphasis on edge caching, wireless resource management, space-air-ground integrated networks, and application of artificial intelligence (AI) for wireless networks. She received the Best Paper Award at IEEE GLOBECOM 2018.

Jiayin Chen received the Ph.D. degree from University of Waterloo, Ontario, Canada, in 2021, and the B.E. degree and the M.S. degree in the School of Electronics and Information Engineering from Harbin Institute of Technology, Harbin, China, in 2014 and 2016, respectively. She is currently a postdoctoral research fellow at the University of British Columbia, British Columbia, Canada. Her research interests are in the area of vehicular networks and machine learning, with current focus on Intelligent Transport System and big data.

Conghao Zhou (S'19) received the B.E. degree from Northeastern University, Shenyang, China, in 2017, and the M.Sc. degree from the University of Illinois at Chicago, Chicago, IL, USA, in 2018. He is currently pursuing the Ph.D. degree with the department of electrical and computer engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include space-air-ground integration networks and machine learning in wireless networks.

Junling Li [corresponding author] (IEEE S'18) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada in 2020. She is currently a Joint Postdoctoral Research Fellow at Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), University of Waterloo, and the Chinese University of Hong Kong, Shenzhen. She received the M.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2016, and the B.S. degree from Tianjin University, Tianjin, China, in 2013. Her interests include game theory, machine learning, software-defined networking, network function virtualization, and vehicular networks. She received the Best Paper Award at the IEEE/CIC International Conference on Communications in China (ICCC) in 2019.

Xuemin (Sherman) Shen (M'97-SM'02-F'09) received the Ph.D. degree in Electrical Engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular ad hoc and sensor networks. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.

Dr. Shen received the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society (ComSoc), and Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013). He has also received the Excellent Graduate Supervision Award in 2006 from University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, IEEE Globecom'07, and the Chair for the IEEE ComSoc Technical Committee on Wireless Communications. Dr. Shen is the President Elect of the IEEE ComSoc. He was the Vice President for Technical and Educational Activities, Vice President for Publications, Member-at-Large on the Board of Governors, Chair of the Distinguished Lecturer Selection Committee, and Member of IEEE Fellow Selection Committee of the ComSoc. Dr. Shen served as the Editor-in-Chief of the IEEE IoT JOURNAL, IEEE Network, and IET Communications.