Two-Timescale Learning-Based Task Offloading for Remote IoT in Integrated Satellite-Terrestrial Networks

Dairu Han, Qiang Ye, *Senior Member, IEEE,* Haixia Peng, *Member, IEEE,* Wen Wu, *Senior Member, IEEE,* Huaqing Wu, *Member, IEEE,* Wenhe Liao, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—In this paper, we propose an integrated satellite-terrestrial network (ISTN) architecture to support delay-sensitive task offloading for remote Internet of Things (IoT), in which satellite networks serve as a complement to terrestrial networks by providing additional communication resources, backhaul capacities, and seamless coverage. Under this architecture, we investigate how to jointly make offloading link selection and bandwidth allocation decisions for BSs and IoT users. Considering the differentiated decision-making time granularities, we formulate a two-timescale stochastic optimization problem to minimize the overall task offloading delay. To accommodate the two-timescale network dynamics and characterize state-action relations, we establish a hierarchical Markov decision process (H-MDP) framework with two separate agents tackling two-timescale network management decisions, and two evolved MDP-based subproblems are formulated accordingly. To efficiently solve the subproblems, we further develop a hybrid proximal policy optimization (H-PPO)-based algorithm. Specifically, a hybrid actor-critic architecture is designed to deal with the mixed discrete and continuous actions. In addition, an action mask layer and an action shaping function are designed to sample feasible task offloading decisions from the time-variant action set. Extensive simulation results have validated the superiority of the proposed ISTN architecture and the H-PPO-based algorithm, especially in scenarios with scarce spectrum resources and heavy traffic loads.**

***Index Terms*—Integrated satellite-terrestrial networks, remote IoT, task offloading, offloading link selection, bandwidth allocation, reinforcement learning.**

I. Introduction

With the advances in sensing technologies and artificial intelligence techniques, Internet of Things (IoT) emerges as an inter-networking paradigm to support a multitude of innovative applications and services in remote areas, such as intelligent agriculture, smart grid management, automated surface mining, and environment monitoring, etc [1]. For these vertical applications, a large portion of the obtained data or tasks needs to be transmitted to data centers or cloud servers for timely processing [2]. For instance, the lower delay in offloading real-time images and device status data of remote power electronics converters to cloud servers leads to more accurate automated control in smart grid. In this context, achieving low latency and seamless coverage is imperative for delay-sensitive IoT applications to facilitate precise control and prompt reaction, which also poses technical challenges especially for a remote IoT system. On one hand, improving the terrestrial backhaul capacity of existing base stations (BSs) in remote areas can be economically costly due to complex natural environment and long-range construction, thereby resulting in degraded radio access network performance; On the other hand, densely deployed small cells in urban areas can achieve ubiquitous coverage and low-latency communications, but they may not be feasible to be deployed in remote areas where the population density is low.

Alternatively, satellite networks can be a promising complementary solution to enhance terrestrial networks due to the intrinsic merits in high capacities and large footprints [3]. Recently, the rapid development of reusable launch systems and satellite maintenance technology paves the way for large-scale implementation of satellite networks. The proliferation of low earth orbit (LEO) constellations driven by SpaceX and OneWeb is currently reinventing network architectures for supporting low latency (around 30 ms), high-capacity (more than 10 Gbps per satellite), and global services [4]. The recent scientific literature has already demonstrated that satellite networks are capable of substantially improving the backhaul capacity for task offloading of remote BSs in case of terrestrial backhaul link failure or congestion [5], [6]. Meanwhile, LEO satellites can act as alternative radio access nodes to facilitate task offloading for remote users with poor user-to-BS (UTB) channel conditions or out of BS coverage [7]–[9]. To this end, the integrated satellite-terrestrial networks (ISTN) will inevitably play a pivotal role in the upcoming 6G era to bridge the digital divide across the globe and fulfill the surging demands for heterogeneous and flexible communications.

While the respective benefits of satellite networks in relieving the workload of terrestrial backhaul links and improving the performance of radio access networks have been proven in ISTN, the investigation of combining the both benefits to assist task offloading for remote IoT users is still missing. In this paper, we propose a novel ISTN architecture to support delay-sensitive task (e.g., content delivery, computation, etc) offloading for remote IoT users. In the proposed architecture, an LEO constellation is considered to provide additional resources and backhaul capacities for the terrestrial network

Dairu Han and Wenhe Liao are with the Department of Aeronautical and Astronautical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: {handairu, cnwho}@njust.edu.cn).

Qiang Ye is with the Department of Computer Science, Memorial University of Newfoundland, St. John's, NL A1B 3X5, Canada (e-mail: qiangy@mun.ca).

Haixia Peng is with the School of Information and Communication Engineering, Xi'an JiaoTong University, Xi'an 710049, China (e-mail: haixia.peng@xjtu.edu.cn).

Wen Wu is with the Frontier Research Center, Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: wuw02@pcl.ac.cn).

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).

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

with overloaded communication demands. All users equipped with dual radio transceivers are first scheduled to upload their tasks to BSs over sub-6 GHz band or LEO satellites over Ka-band [10]. Each BS then offloads the collected tasks to the core network via wired terrestrial backhaul links and satellite backhaul links over Ka-band, while each satellite offloads the collected tasks to the core network via satellite backhaul links. The task offloading decisions (including offloading link selection and bandwidth allocation) for user-to-satellite (UTS) and BS-to-satellite (BTS) transmissions are executed in a larger timescale compared with that for UTB transmissions.

In the ISTN architecture, how to design an efficient scheduling policy to minimize the long-term offloading delay of all tasks is a crucial but challenging issue: Firstly, it is non-trivial to rationally design a joint scheduling policy for UTB, UTS, and BTS task transmissions with differentiated channel characteristics and resource capabilities; Secondly, the offloading link selection decisions for UTB and UTS are interdependent since only one type of links can be associated with a user in a network operation window. The scheduling decisions for BTS transmissions determine the backhaul capacity of BSs, thus affecting the scheduling decisions for UTB transmissions. In addition, all BTS and UTS links share the same set of spectrum resources. Therefore, the specific scheduling decisions for UTB, UTS, and BTS task transmissions are correlated in different timescales; Thirdly, the variations of task arrivals and channel conditions in the coming future are not easy to be predicted, which causes uncertainty for real-time scheduling decision-making to maximize the long-term performance.

To address the above challenges, reinforcement learning (RL) has been considered as an efficient and future-proof solution for sequential decision-making with environment uncertainty due to its ability of learning in dynamic and complex systems [11], [12]. However, the traditional RL-based methods are generally suitable for obtaining a stationary solution to make proper decisions to a stochastic optimization problem, which cannot be applied directly to obtain the two-timescale network management decisions with coupled constraints. Therefore, we investigate how to develop a tailored RL-based solution to adaptively make task offloading scheduling decisions supporting the delay-sensitive tasks offloading for remote IoT.

In this paper, the main contributions are summarized as follows.

- We present a novel ISTN architecture to support delay-sensitive task offloading for remote IoT users. We formulate the problem of joint offloading link selection and bandwidth allocation for UTB, UTS, and BTS task transmissions as a two-timescale stochastic optimization problem with the objective of minimizing the overall task offloading delay, subject to coupled spectrum allocation constraints and backhaul capacity constraints.
- To cope with dynamic task arrivals and channel conditions, we propose a two-timescale hierarchical Markov decision process (H-MDP) framework to capture the state and action transitions, where two independent agents for different timescales are introduced and a novel reward design is proposed to achieve efficient learning. The formulated optimization problem is transformed into two MDP subproblems accordingly.
- We propose a hybrid proximal policy optimization (H-PPO)-based RL algorithm to solve the two subproblems. A novel hybrid Actor-Critic (H-AC) architecture is incorporated in the algorithm to deal with the mixed discrete and continuous action space. In addition, an action mask layer and an action shaping function are designed for agents to make proper action selections while interacting with the environment.

The remainder of this paper is organized as follows. Related works are reviewed in Section II. The system model and problem formulation are presented in Section III. In Section IV, the two-timescale H-MDP framework with problem transformation is established. The H-PPO-based algorithm is introduced in Section V. Simulation results are provided in Section VI, and the conclusion is drawn in Section VII.

II. Related Work

Recently, ISTN has drawn great attention in task offloading due to its ability in providing global and sustainable communications. In [13], LEO satellites were employed to assist data offloading for IoTs and an online Lyapunov-based algorithm was proposed to maximize the throughput. To relieve the load in capacity-limited terrestrial networks, the authors in [14] proposed a software-defined network based ISTN architecture, which can improve the user's quality-of-experience. By leveraging the LEO-backhauled small cell, Di *et al.* [15] introduced an ISTN architecture to optimize the overall data rate and the number of accessed users, which shows superior performance compared with the non-integrated networks. A pioneering work developed a cognitive service architecture for the 6G core network to meet the increasingly high requirement for quality of service [16]. In [17], a space-air-ground MEC solution was proposed to assist vehicles for task offloading in the scenario with dense buildings but scarce communication infrastructure. A collaborative communication scheme was proposed for supporting vehicle task offloading in [18], where the deployment of non-terrestrial networks was studied based on the prediction of vehicle distribution. In [19], satellites and BSs were integrated with non-orthogonal multiple access and orthogonal multiple access schemes to minimize the completion time of IoT tasks. A satellite storage-oriented handover scheme was presented in [20], which maps the satellite networks to the virtual space for better delay and handover performance. In [21], a traffic offloading scheme was proposed to intelligently direct traffic in ISTN for differentiated latency and throughput satisfaction. Different from the existing works, our work aims at minimizing the long-term offloading delay of all IoT tasks arriving dynamically in ISTN. In addition, we take advantage of satellite networks for both backhaul capacity improvement and connectivity enhancement to ameliorate the system performance of ISTN.

To deal with a complex and dynamic environment, RL-based algorithms have been widely applied to maximize the long-term performance of the ISTN. In [2], a deep risk-sensitive RL-based algorithm was proposed to make online

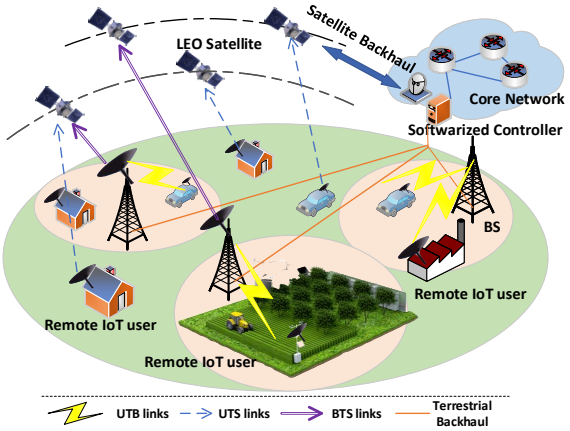


Fig. 1. System model of service offloading in the considered ISTN.

offloading decisions for IoT users. In [8], Zhou *et al.* studied the problem of data scheduling for IoT users in remote areas, in which a deep RL (DRL) -based algorithm was proposed to deal with the unknown channel conditions and solar infeed process. Focusing on traffic offloading with dynamic network traffic, Tang *et al.* [22] presented a DRL-based method with an improved delay-sensitive replay memory algorithm to minimize the packet delay. Considering the Markovian rainfall change and satellite movement, a multi-layer Ka/Q-band ISTN is introduced to obtain a high transmission rate via a DRL-based approach [9]. In [23], a DRL-based scheme with offline training and online decision-making is proposed to improve the system throughput and avoid frequent handovers among UTS links. Different from the existing works, the operations of satellite networks and terrestrial networks in our work differentiated in time granularity for a more practical realization of an ISTN system, due to which the traditional RL-based algorithms are infeasible to be directly applied for ISTN management.

III. System Model and Problem Formulation

In this section, we first present the network model of the considered ISTN. Then, the detailed task offloading framework including offloading link selection and bandwidth allocation, channel model, and delay model is proposed. Finally, the problem formulation is presented. The definitions of main notations used in this paper are summarized in Table I.

A. Network Model

Consider an ISTN as shown in Fig. 1, which is composed of B BSs and an LEO satellite constellation with N satellites to cooperatively support round-the-clock task offloading and robust connection for U remote IoT users. All users are randomly distributed over the entire target scenario and need to offload collected tasks to the core network for further data processing. The deployment of ultra-dense LEO satellites (referred to as satellites throughout the rest of the paper for brevity) ensures seamless coverage for all BSs and users, which means each BS or user is always covered by more than one satellite. All users are equipped with dual radio transceivers and are able to directly connect with LEO satellites using very small aperture terminal or with BSs using antennas for terrestrial communications.[1] To improve the backhaul link capacity, all BSs equipped with high-gain satellite antennas can connect to the core network via terrestrial and satellite backhaul links. As satellite backhaul transmissions usually experience longer delay compared with terrestrial backhaul transmissions, terrestrial backhaul links are considered to have higher priority for task offloading of BSs. Therefore, each user can offload tasks through one of the following two paths:

1) Tasks are first offloaded to BSs over sub-6 GHz band, and then be transmitted to the core network through capacity-limited terrestrial backhaul links. If the terrestrial backhaul links are overloaded, tasks can be simultaneously uploaded to satellites over Ka-band and finally transmitted to the core network via satellite backhaul links. This path is called as user-BS offloading (UBO) path.
2) Tasks are directly offloaded to satellites over Ka-band and then forwarded to the core network through satellite

[1]The proposed architecture is independent of user radio types. A single-radio user can be directly extended to the ISTN by only connecting with BSs or satellites. In this work, only static users are considered.

Table I
Summary of Notations.

Notation	Description
$\mathcal{B}, \mathcal{N}, \mathcal{U}$	Sets of BSs, satellites, and users, respectively
$\mathcal{V}, \mathcal{T}$	Sets of time windows and time slots
$\boldsymbol{c}^{UTB}, \boldsymbol{c}^{BTS}, \boldsymbol{c}^{UTS}$	Association decisions for UTB, BTS, and UTS links, respectively
$\boldsymbol{a}^{UTB}, \boldsymbol{a}^{BTS}, \boldsymbol{a}^{UTS}$	Bandwidth allocation decisions for UTB, BTS, and UTS links, respectively
$T^{BTS}_{b,n}(v)$, $T^{UTS}_{u,n}(v)$	The remaining connection time of BTS links and UTS links at the beginning of time window v
$\boldsymbol{d}^{UTB}, \boldsymbol{d}^{BTS}, \boldsymbol{d}^{UTS}$	Distance between user and BS, distance between satellite and BS, and distance between user and satellite, respectively
$L^{BTS}_{b,n}(v,t)$, $L^{UTS}_{u,n}(v,t)$	Uplink path loss of BTS links and UTS links
$R^{UTB}_u(v,t)$	The achievable data rate of user u using terrestrial uplink at slot (v,t)
$R^{UTS}_u(v,t)$	The achievable data rate of user u using satellite uplink at slot (v,t)
$R^{BTS}_b(v,t)$	The achievable data rate of BS b using satellite uplink at slot (v,t)
$R^{BS}_b(v,t)$	The received data rate of BS b at slot (v,t)
G^S, G^B, G^U	Antenna gains of satellites, BSs, and users, respectively
p^{UTB}	Transmission power of users using terrestrial uplink
p^{BTS}, p^{UTS}	Transmission power of BSs and users using satellite uplink
$C^{BS}, C^E_b(v,t)$	The terrestrial backhaul capacity of each BS, the equivalent backhaul capacity of BS b at slot (v,t)
$\lambda, z_u(v,t), Z$	Task arrival rate parameter, overall size of arrived tasks of user u at slot (v,t), and size of each task, respectively
$I_u(v,t)$	The task backlog length of user u at the beginning of slot (v,t)
$W_u(v,t)$	The number of backlogged queue tasks of user u at the end of slot (v,t)
$M_u(v,t)$	The number of offloaded tasks of user u during slot (v,t)

backhaul links. This path is referred to as user-satellite offloading (USO) path. Several situations may lead to the selection of USO path, e.g., users located out of the coverage of BSs, better channel conditions of UTS links than UTB links, and more spectrum resources provided to users by satellites than BSs, etc.

A centralized control architecture is considered to increase the flexibility of network management with different time granularities of control [24]. All BSs and satellites are connected to a softwarized controller deployed near the edge of the core network to monitor the dynamic network conditions and conduct joint offloading link selection and bandwidth allocation decisions. In this work, we focus on the uplink process of task offloading. Considering the difference in operation time between satellite-based links (i.e., BTS and UTS links) and terrestrial links (i.e., UTB links), we adopt two different timescales in the ISTN system. Specifically, time is first divided into a sequence of time windows and the set of time windows is denoted by $\mathcal{V} = \{1, 2, \ldots, V\}$. BTS and UTS links are scheduled at the beginning of each time window. Then, each time window is further partitioned into multiple time slots with the equal and fixed length μ, indexed by $t \in \mathcal{T} = \{1, 2, \ldots, T\}$. Within a time window, UTB links are scheduled at the beginning of each slot, while the scheduling of BTS and UTS links remains unchanged. We assume that the whole system is time-slotted and quasi-static, which means the network topology remains constant within a slot due to the short duration but changes over slots. The propagation time for control signal is assumed negligible since it is much smaller as compared with the duration of time slot and time window.

B. Offloading Link Selection and Bandwidth Allocation

Denote $\mathcal{U} = \{1, 2, \ldots, U\}$ as the set of all users. The set of BSs is denoted by $\mathcal{B} = \{1, 2, \ldots, B\}$. Let $\mathcal{N} = \{1, 2, \ldots, N\}$ be the set of satellites. For notational simplicity, t-th slot in time window $v \in \mathcal{V}$ is denoted by (v, t).

1) For terrestrial links: At the beginning of each slot, the softwarized controller makes the offloading link selection and bandwidth allocation decisions for terrestrial links based on the global network information. Since the UTB link transmission is only feasible when user $u \in \mathcal{U}$ locates in the coverage of BS $b \in \mathcal{B}$, the set of available BSs that user u can be associated with is denoted by $\mathcal{B}_u$, where $\mathcal{B}_u = \{b | b \in \mathcal{B}, d_{u,b}^{UTB}(v,t) \leq d_{max}\}$, $d_{u,b}^{UTB}(v,t)$ is the distance between user u and BS b at slot (v,t), and d_{max} is the coverage radius of each BS. As typically adopted in 3GPP, we consider each user can be associated with a maximum of one BS at each slot (v,t). Let $c_u^{UTB}(v,t)$ be the integer association indicator variable, where $c_u^{UTB}(v,t) = b$, $\forall b \in \mathcal{B}_u$ indicates that user u is associated with BS b at slot (v,t), and $c_u^{UTB}(v,t) = 0$ means user u is not associated to any BSs.

All UTB links share the same spectrum resources, and frequency-division multiple access (FDMA) is adopted for transmission. Denote the bandwidth allocation decisions of UTB links by $a_{u,b}^{UTB}(v,t) \in [0,1]$, which represents the bandwidth fraction allocated to user u from BS b at slot (v,t). Then, we have

$$\sum_{u \in \mathcal{U}_b^{UTB}(v,t)} a_{u,b}^{UTB}(v,t) \leq 1, \ \forall b, v, t \tag{1}$$

where $\mathcal{U}_b^{UTB}(v,t) = \{u | u \in \mathcal{U}, d_{u,b}^{UTB}(v,t) \leq d_{max}\}$ is the set of users in the coverage of BS b at slot (v,t).

2) For satellite-based links: For BTS and UTS links, the softwarized controller makes offloading link selection and bandwidth allocation decisions following FDMA techniques at the beginning of each time window. Due to the high movement speed of satellites, BTS and UTS links are highly dynamic and the communication window is sporadic. Denote the remaining connection time of BTS and UTS links from BS b and user u to satellite $n \in \mathcal{N}$ at the beginning of time window v by $T_{b,n}^{BTS}(v)$ and $T_{u,n}^{UTS}(v)$, respectively. Associations of BTS and UTS links are feasible only when the remaining connection time is no shorter than the time window length, i.e., $T\mu$. In addition, BTS and UTS links under the minimum elevation angle will be hindered by natural barriers. Therefore, the sets of satellites that users u and BS b can associate with vary over different time windows, which are denoted by $\mathcal{N}_u^{UTS}(v) = \{n | n \in \mathcal{N}, T_{u,n}^{UTS}(v) \geq T\mu, k_{u,n}^{UTS}(v) \geq k_{min}\}$ and $\mathcal{N}_b^{BTS}(v) = \{n | n \in \mathcal{N}, T_{b,n}^{BTS}(v) \geq T\mu, k_{b,n}^{BTS}(v) \geq k_{min}\}$, respectively. In the above sets, $k_{b,n}^{BTS}(v)$ and $k_{u,n}^{UTS}(v)$ are the elevation angles from BS b and user u to satellite n at the beginning of time window v. k_{min} is the minimum elevation angle beyond which BTS and UTS links can be constructed.

Let $c_b^{BTS}(v)$ and $c_u^{UTS}(v)$ be the integer association indicator variable of BTS and UTS links, respectively, where $c_b^{BTS}(v) = n$, $\forall n \in \mathcal{N}_b^{BTS}(v)$ or $c_u^{UTS}(v) = n$, $\forall n \in \mathcal{N}_u^{UTS}(v)$ indicates that BS b or user u is associated with satellite n at time window v, and $c_b^{BTS}(v) = 0$ or $c_u^{UTS}(v) = 0$ denotes that there are no satellites associated with BS b or user u. Let $a_{b,n}^{BTS}(v) \in [0,1]$ and $a_{u,n}^{UTS}(v) \in [0,1]$ denote the bandwidth fraction allocated to BS b and user u from satellite n at time window v. All BTS and UTS links sharing the same spectrum resource pool, i.e.,

$$\sum_{b \in \mathcal{B}_n^{BTS}(v)} a_{b,n}^{BTS}(v) + \sum_{u \in \mathcal{U}_n^{UTS}(v)} a_{u,n}^{UTS}(v) \leq 1, \ \forall n, v \tag{2}$$

where $\mathcal{B}_n^{BTS}(v) = \{b | b \in \mathcal{B}, T_{b,n}^{BTS}(v) \geq T\mu, k_{b,n}^{BTS}(v) \geq k_{min}\}$ and $\mathcal{U}_n^{UTS}(v) = \{u | u \in \mathcal{U}, T_{u,n}^{UTS}(v) \geq T\mu, k_{u,n}^{UTS}(v) \geq k_{min}\}$ are sets of BSs and users that can connect with satellite n at the beginning of time window v.

C. Communication Model

1) For terrestrial links: Both large-scale and small-scale channel fading are considered. Denote the channel gain from user u to BS b at slot (v,t) by $h_{u,b}^{UTB}(v,t) = (d_{u,b}^{UTB}(v,t))^{-\alpha} g_{u,b}^2(v,t) \beta_{u,b}(v,t)$, where $\beta_{u,b}(v,t)$ indicates the shadowing effect which follows log-normal distribution, α denotes the path loss exponent, and $g_{u,b}(v,t) \sim \mathcal{CN}(0,1)$ represents the Rayleigh fading coefficient[2]. Since two adjacent

[2] $\mathcal{CN}$ denotes the Complex Gaussian distribution.

BSs in remote areas are generally separated with a long distance, the co-channel interference can be controlled within a low level and considered negligible. Then, based on Shannon's Theorem, the achievable rate from user u to BS b at slot (v,t) is expressed as

$$R_{u,b}^{U}(v,t) = a_{u,b}^{UTB}(v,t) B_0 \log_2 \left(1 + \frac{h_{u,b}^{UTB}(v,t) p^{UTB}}{a_{u,b}^{UTB}(v,t) B_0 N_0}\right) \tag{3}$$

where p^{UTB} is the uplink transmission power from user to BS, B_0 is the overall available spectrum resources of each BS over sub-6 GHz band, and N_0 is the additive white Gaussian noise power spectral density. Thus, the achievable rate of user u using terrestrial uplink at slot (v,t) is given by

$$R_u^{UTB}(v,t) = \begin{cases} R_{u,c_u^{UTB}(v,t)}^{U}(v,t), & \text{if } c_u^{UTB}(v,t) \neq 0, \\ 0, & \text{if } c_u^{UTB}(v,t) = 0. \end{cases} \tag{4}$$

Denote $R_b^{BS}(v,t)$ as the received data rate of BS b at slot (v,t), which is expressed as

$$R_b^{BS}(v,t) = \sum_{u \in \mathcal{U}_b^{UTB}} R_{u,b}^{U}(v,t), \forall b, v, t. \tag{5}$$

2) For satellite-based links: Denote satellite uplink path loss of BTS links and UTS links at slot (v,t) by $L_{b,n}^{BTS}(v,t)$ and $L_{u,n}^{UTS}(v,t)$. Different from terrestrial uplink channels, satellite uplink path loss is mainly composed of free-space path loss, polarization loss, and atmospheric loss. We denote the free-space path loss of BTS links and UTS links at slot (v,t) as $F_{b,n}^{BTS}(v,t)$ and $F_{u,n}^{UTS}(v,t)$, given by

$$F_{b,n}^{BTS}(v,t) = \left(\frac{4\pi d_{b,n}^{BTS}(v,t)}{\lambda^{SAT}}\right)^2 \tag{6}$$

and

$$F_{u,n}^{UTS}(v,t) = \left(\frac{4\pi d_{u,n}^{UTS}(v,t)}{\lambda^{SAT}}\right)^2 \tag{7}$$

where λ^{SAT} is the wavelength of the signal, $d_{b,n}^{BTS}(v,t)$ and $d_{u,n}^{UTS}(v,t)$ are the geographic distance from satellite n to BS b and user u at the beginning of slot (v,t). The polarization loss exists when the polarization of the receiving antenna is inconsistent with that of the incident plane wave, which is usually less than 3 dB [25]. The atmospheric loss is caused due to absorption and scattering by gas molecules in the atmosphere (e.g., rain attenuation), which can be predicted by measurement and statistics [26]. Denote the channel fading of BTS links and UTS links at slot (v,t) by $h_{b,n}^{BTS}(v,t)$ and $h_{u,n}^{UTS}(v,t)$. Since the line-of-sight (LoS) signal is generally a dominant component in satellite uplinks, Rician fading model is widely adopted. The probability density function of the channel fading is given by

$$f_{|h|^2}(x) = \frac{K+1}{\Omega} \exp\left\{-K - \frac{(K+1)x}{\Omega}\right\} I_0\left(2\sqrt{\frac{K(K+1)x}{\Omega}}\right) \tag{8}$$

where $h = h_{b,n}^{BTS}(v,t)$ or $h_{u,n}^{UTS}(v,t)$, K is the ratio between the power in the LOS path and in the non-LOS paths, Ω is the mean of the received local power, and $I_0(\cdot)$ is the modified Bessel function of the first kind with zero order. Since the antennas of BSs and users are generally of good directivity in Ka-band, thereby ensuring the side-lobe antenna gain is low and the co-channel interference is negligible. The achievable rates of BS b and user u served by satellite n at slot (v,t) are expressed as

$$R_{b,n}^{BTS}(v,t) = \log_2 \left(1 + \frac{p^{BTS} |h_{b,n}^{BTS}(v,t)|^2 G^S G^B L_{b,n}^{BTS}(v,t)}{a_{b,n}^{BTS}(v) B_1 N_0}\right) \cdot a_{b,n}^{BTS}(v) B_1 \tag{9}$$

and

$$R_{u,n}^{UTS}(v,t) = \log_2 \left(1 + \frac{p^{UTS} |h_{u,n}^{UTS}(v,t)|^2 G^S G^U L_{u,n}^{UTS}(v,t)}{a_{u,n}^{UTS}(v) B_1 N_0}\right) \cdot a_{u,n}^{UTS}(v) B_1 \tag{10}$$

where G^S, G^B, and G^U are the antenna gains of satellites, BSs, and users, respectively, p^{BTS} and p^{UTS} are the transmission power from BSs to satellites and from users to satellites, and B_1 is the overall amount of available spectrum resources of each satellite over Ka-band. The data rates of BS b and user u using satellite links are given by, respectively,

$$R_b^{BTS}(v,t) = \begin{cases} R_{b,c_b^{BTS}(v)}^{BTS}(v,t), \ \forall b, v, t, c_b^{BTS}(v) \neq 0, \\ 0, \ \forall b, v, t, c_b^{BTS}(v) = 0, \end{cases} \tag{11}$$

and

$$R_u^{UTS}(v,t) = \begin{cases} R_{u,c_u^{UTS}(v)}^{UTS}(v,t), \ \forall u, v, t, c_u^{UTS}(v) \neq 0, \\ 0, \ \forall u, v, t, c_u^{UTS}(v) = 0. \end{cases} \tag{12}$$

D. Delay Model

Considering the offloading paths and task arrival patterns, the delay model can be described as follows.

1) Queuing delay: In an ISTN scenario, the task arrival process of each user is assumed to follow Poisson process with rate parameter λ. Then, the probability that the cumulative data size $z_u(v,t)$ arrived at user u during slot (v,t) is

$$Pr(z_u(v,t)) = \frac{(\lambda\mu)^{z_u(v,t)/Z} \cdot e^{-\lambda\mu}}{(z_u(v,t)/Z)!} \tag{13}$$

where Z denotes the data size of each task [27], [28]. Since the uplink rate of each user is limited, the arrival tasks may not be completely offloaded by each user within a single slot. The remaining tasks are stored in the forwarding queue and wait to be transmitted in the following slots. Let $I_u(v,t)$ denote the task backlog length of user u at the beginning of slot (v,t). We denote $W_u(v,t)$ and $M_u(v,t)$ as the number of backlogged queue tasks of user u at the end of slot (v,t) and the number of offloaded tasks of user u during slot (v,t), respectively, given by

$$W_u(v,t) = \max\left\{I_u(v,t) - \lfloor \frac{R_u(v,t)\mu}{Z} \rfloor, 0\right\} \tag{14}$$

and

$$M_u(v,t) = \min\left\{I_u(v,t), \lfloor \frac{R_u(v,t)\mu}{Z} \rfloor\right\} \tag{15}$$

where $R_u(v,t) = R_u^{UTB}(v,t)$ when UTB link is allocated to user u, $R_u(v,t) = R_u^{UTS}(v,t)$ when UTS link is allocated to user u, and $\lfloor \cdot \rfloor$ is the floor function. Then, the task backlog length of user u at the beginning of slot $(v, t+1)$ is given by

$$I_u(v,t+1) = W_u(v,t) + \frac{z_u(v,t)}{Z}. \tag{16}$$

Given the number of backlogged queue tasks of user u $W_u(v,t)$, the total queuing delay of all tasks of user u in slot (v,t) is given by $W_u(v,t)\mu$.

2) Delay of USO path: For USO path, both transmission delay and propagation delay are considered. Given the data rate of each user $R_u^{UTS}(v,t)$ and the offloaded task number $M_u(v,t)$ at slot (v,t), the uplink delay of all tasks offloaded by user u is calculated as

$$\begin{aligned} o_u^{USO}(v,t) =& M_u(v,t)\left(\frac{Z}{R_u^{UTS}(v,t)} + o_{pg}\right), \\ & \forall u, v, t, c_u^{UTS}(v) \neq 0 \end{aligned} \tag{17}$$

where o_{pg} is the round-trip propagation delay for each task. Note that we neglect the feeder link transmission delay of satellites since the capacity of feeder link is usually much higher than that of user link in satellite communication.

3) Delay of UBO path: The delay of UBO path consists of two components, UTB link delay and backhaul delay. Similarly as derived above, UTB link delay of all tasks offloaded by user u at slot (v,t) is expressed as

$$o_u^{UTB}(v,t) = M_u(v,t)\left(\frac{Z}{R_u^{UTB}(v,t)}\right),\ \forall u, v, t, c_u^{UTB}(v,t) \neq 0. \tag{18}$$

As BSs may access to satellites for additional backhaul links, the equivalent backhaul capacity of BS b at slot (v,t) is given by $C_b^E(v,t) = C^{BS} + R_b^{BTS}(v,t)$, where C^{BS} is the capacity of terrestrial backhaul link. We ignore the propagation delay of terrestrial backhaul links and UTB links due to the short transmission distance. Then, the equivalent backhaul delay of all tasks offloaded by user u at slot (v,t) is expressed as

$$\begin{aligned} o_u^{BAU}(v,t) =& o_{pg} M_u(v,t) \frac{R_{c_u^{UTB}(v,t)}^{BTS}(v,t)}{C_{c_u^{UTB}(v,t)}^{E}(v,t)} \\ &+ M_u(v,t) \frac{Z}{C_{c_u^{UTB}(v,t)}^{E}(v,t)},\ \forall u, v, t, c_u^{UTB}(v,t) \neq 0 \end{aligned} \tag{19}$$

where the first term on the right side of (19) is the satellite backhaul delay, and the second term is the terrestrial backhaul delay. When $R_{c_u^{UTB}(v,t)}^{BTS}(v,t) = 0$, BS $b = c_u^{UTB}(v,t)$ only uses terrestrial backhaul links, and the equivalent backhaul propagation delay does not exist. Then, the uplink delay of all offloaded tasks by user u at slot (v,t) through UBO Path is calculated as

$$o_u^{UBO}(v,t) = o_u^{UTB}(v,t) + o_u^{BAU}(v,t), \forall u, v, t, c_u^{UTB}(v,t) \neq 0. \tag{20}$$

Given the queuing delay, the delay of UBO Path, and the delay of USO Path, the total offloading delay of all tasks of each user u at slot (v,t) is calculated as

$$o_u(v,t) = \begin{cases} W_u(v,t)\mu + o_u^{USO}(v,t), & \forall u, v, t, c_u^{UTS}(v) \neq 0, \\ & c_u^{UTB}(v,t) = 0, \\ W_u(v,t)\mu + o_u^{UBO}(v,t), & \forall u, v, t, c_u^{UTS}(v) = 0, \\ & c_u^{UTB}(v,t) \neq 0, \\ W_u(v,t)\mu, & \text{otherwise.} \end{cases} \tag{21}$$

E. Problem Formulation

Due to spatial-temporal network environment variations, it is imperative for network operators to minimize the time-average task delay over VT slots while satisfying the backhaul capacity and spectrum allocation constraints. Thus, the problem is formulated as

$$\text{P0:} \min_{\substack{\{\mathbf{c}^{UTB}, \mathbf{c}^{BTS}, \mathbf{c}^{UTS}, \\ \mathbf{a}^{UTB}, \mathbf{a}^{BTS}, \mathbf{a}^{UTS}\}}} \lim_{V\to\infty} \frac{1}{VT} \sum_{v=1}^{V} \sum_{t=1}^{T} \sum_{u=1}^{U} o_u(v,t) \tag{22}$$

$$\text{s.t. } (1) \text{ and } (2),$$

$$a_{u,b}^{UTB}(v,t) \in [0,1], c_u^{UTB}(v,t) \in \mathcal{B}_u \cup \{0\},\ \forall u, b, v, t, \tag{22a}$$

$$a_{b,n}^{BTS}(v) \in [0,1], c_b^{BTS}(v) \in \mathcal{N}_b^{BTS}(v) \cup \{0\},\ \forall b, n, v, \tag{22b}$$

$$a_{u,n}^{UTS}(v) \in [0,1], c_u^{UTS}(v) \in \mathcal{N}_u^{UTS}(v) \cup \{0\},\ \forall u, n, v, \tag{22c}$$

$$c_u^{UTB}(v,t) c_u^{UTS}(v) = 0,\ \forall u, v, t, \tag{22d}$$

$$R_b^{BS}(v,t) \leq C_b^E(v,t),\ \forall b, v, t \tag{22e}$$

where $\mathbf{c}^{UTB} = \{c_u^{UTB}(v,t)\}$, $\mathbf{c}^{BTS} = \{c_b^{BTS}(v)\}$, $\mathbf{c}^{UTS} = \{c_u^{UTS}(v)\}$, $\mathbf{a}^{UTB} = \{a_{u,b}^{UTB}(v,t)\}$, $\mathbf{a}^{BTS} = \{a_{b,n}^{BTS}(v)\}$, $\mathbf{a}^{UTS} = \{a_{u,n}^{UTS}(v)\}$, $\forall u \in \mathcal{U}$, $\forall b \in \mathcal{B}$, $\forall n \in \mathcal{N}$, $\forall v \in \mathcal{V}$, $\forall t \in \mathcal{T}$. Constraints (1), (2), and (22a) - (22c) guarantee the feasibility of offloading link selection and bandwidth allocation decisions. Constraint (22d) implies that each user can only be associated with at most one BS or satellite within each slot, which shows the coupling relations between associations of UTB and UTS links. Constraint (22e) guarantees that the received data rate of each BS cannot exceed the capacity of backhaul link.

IV. Two-Timescale H-MDP Framework and Problem Transformation

The formulated problem P0 is a mixed-integer nonlinear optimization problem with dynamic task arrivals. In the considered ISTN, obtaining the optimal solution via per-slot real-time optimization without future network information is non-trivial. To accommodate network dynamics for performance optimization over time, one potential solution is to apply RL-based methods. However, traditional RL-based methods are usually suitable for long-term stochastic decision-making problems without constraints, which may not work well for problems of multi-timescales with coupled constraints. In this case, it is pivotal to develop a tailored solution to the underlying problem. In this section, we first propose a two-timescale H-MDP framework to capture the problem dynamics. Then, a

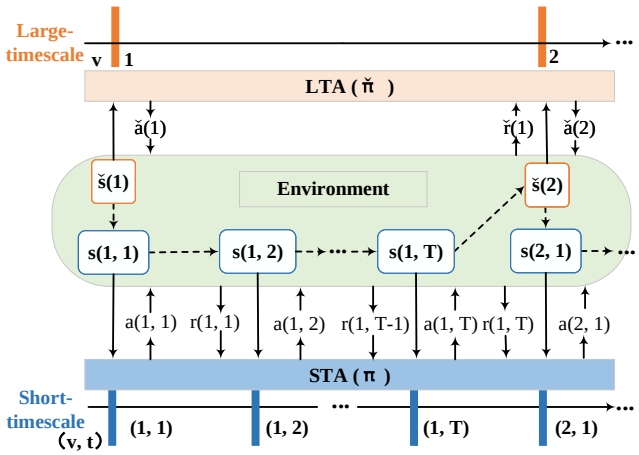


Fig. 2. Illustration of the proposed two-timescale H-MDP structure.

problem transformation is presented with two agents operated in different timescales.

A. Two-Timescale H-MDP

As shown in Fig. 2, we extend the standard MDP setup [2] to a two-timescale H-MDP structure, which includes a large-timescale agent (LTA) and a short-timescale agent (STA) jointly interacting with the environment. Specifically, at the beginning of each time window v, LTA on large-timescale state $\check{\boldsymbol{s}}(v)$ takes a large-timescale action $\check{\boldsymbol{a}}(t)$ following its policy $\check{\boldsymbol{\pi}}$. At each slot (v,t), after observing a short-timescale state observation $\boldsymbol{s}(v,t)$ including the processed large-timescale action $\check{\boldsymbol{a}}(t)$, STA takes a short-timescale action $\boldsymbol{a}(v,t)$ by sampling from its policy $\boldsymbol{\pi}$. The environment then immediately returns a short-timescale reward $r(v,t)$ to STA and yields transitions to a new short-timescale state $\boldsymbol{s}(v,t+1)$ according to state transition probability $Pr(\boldsymbol{s}(v,t+1)|\boldsymbol{a}(v,t),\boldsymbol{s}(v,t))$. After STA executes the short-timescale actions for T slots, a large-timescale reward $\check{r}(v)$ including the cumulative T-step short-timescale reward is received by LTA. Concurrently, the large-timescale state proceeds to $\check{\boldsymbol{s}}(v+1)$ according to $Pr(\check{\boldsymbol{s}}(v+1)|\check{\boldsymbol{a}}(v),\check{\boldsymbol{s}}(v))$. The objectives of LTA and STA are to maximize the expected return $\mathbb{E}_{\check{\boldsymbol{\pi}}}[\sum_{v\geq 1}\gamma^{v-1}\check{r}(v)]$ and $\mathbb{E}_{\boldsymbol{\pi}}[\sum_{t\geq 1}\sum_{v\geq 1}\gamma^{((v-1)T+t-1)}r(v,t)]$, through which the best policy will be learned. In the above expected return functions, $\gamma\in[0,1]$ is the discount factor representing the preference between the current reward and the future reward.

Remark 1: The proposed H-MDP is to efficiently solve the problem with two-timescale actions, which is different from previous works related to hierarchical RL [29], [30]. The difference is two-fold: (1) Actions in the H-MDP are designed specifically for different timescales and the short-timescale policy is a sub-policy conditioned on $\check{\boldsymbol{\pi}}$, and (2) The tailored reward $\check{r}(v)$ and $r(v,t)$ ensures the stable coordination of LTA and STA, which will be illustrated later in Section IV-B.

B. Problem Transformation with H-MDP

In the ISTN, STA and LTA are implemented by the soft-warized controller. The specific definitions of states, actions, and rewards of H-MDP are illustrated as follows.

1) Action: Corresponding to (22), as offloading link selection and bandwidth allocation decisions are made at different timescales, actions for LTA and STA are defined as

$$\begin{aligned}&\check{\boldsymbol{a}}(v)=[c_b^{BTS}(v),c_u^{UTS}(v),a_{b,n}^{BTS}(v),a_{u,n}^{UTS}(v)],\ \text{and}\\&\boldsymbol{a}(v,t)=[c_u^{UTB}(v,t),a_{u,b}^{UTB}(v,t)],\forall u,b,n.\end{aligned}\tag{23}$$

Note that the action space of STA and LTA both contain discrete actions and continuous actions.

2) State: For STA, the decision-making at slot (v,t) depends on the equivalent backhaul capacity of each BS, the task backlog length, the distance between user u and BS b, and the channel state of UTB links. Meanwhile, constraint (22d) implies that decision $c_u^{UTB}(v,t)$ is made under the guidance of $c_u^{UTS}(v)$. Thus, the system state of STA at slot (v,t) is denoted by

$$\boldsymbol{s}(v,t)=[C_b^E(v,t),I_u(v,t),d_{u,b}^{UTB}(v,t),h_{u,b}^{UTB}(v,t),c_u^{UTS}(v)].\tag{24}$$

For LTA, the decision-making at time window v depends on the remaining contact time of UTS and BTS links, distance from satellite n to BS b and user u, the task backlog length, the elevation angles of BTS links and UTS links, and the channel state of UTB, BTS, and UTS links. To better capture the dynamic channel condition in the large timescale, the average channel states of UTB, BTS, and UTS links during time window $v-1$ are also observed by LTA. Thus, the system state of LTA in time window v is denoted by

$$\begin{aligned}\check{\boldsymbol{s}}(v)=[&T_{u,n}^{UTS}(v),T_{b,n}^{BTS}(v),d_{b,n}^{BTS}(v),d_{u,n}^{UTS}(v),k_{u,n}^{UTS}(v),\\&\frac{1}{T}\sum_{t=1}^{T}\left(h_{u,b}^{UTB}(v-1,t),h_{b,n}^{BTS}(v-1,t),h_{u,n}^{UTS}(v-1,t)\right),\\&I_u(v,t),h_{u,b}^{UTB}(v,1),h_{b,n}^{BTS}(v,1),h_{u,n}^{UTS}(v,1),k_{b,n}^{BTS}(v)].\end{aligned}\tag{25}$$

3) Reward: For STA, to evaluate the performance of action taken under the observed state and minimize the overall delay, the reward function is defined as

$$r(v,t)=-\sum_{u=1}^{U}o_u(v,t),\ \forall u,c_u^{UTS}(v)=0.\tag{26}$$

For LTA, the reward for taking action $\check{\boldsymbol{a}}(t)$ consists of two parts. The first part is the overall delay of users who use USO path for service offloading. The second part is the cumulative T-step STA reward during time window v, which is the evaluation of action $\check{\boldsymbol{a}}(t)$ on STA. Therefore, the reward function is defined as

$$\check{r}(v)=-\sum_{t=1}^{T}\sum_{u=1}^{U}o_u(v,t),\forall u\in\mathcal{U}.\tag{27}$$

Based on the proposed H-MDP, problem P0 can be decomposed into two MDP-based subproblems for LTA and STA, which are formulated as

$$\begin{aligned}\text{P1:}\max_{\check{\boldsymbol{\pi}}}\ &\mathbb{E}\left[\lim_{V\to\infty}\frac{1}{VT}\sum_{v=1}^{V}\gamma^{v-1}\check{r}(v)\right]\\&\text{s.t. }(2),(22b),\text{ and }(22c)\end{aligned}\tag{28}$$

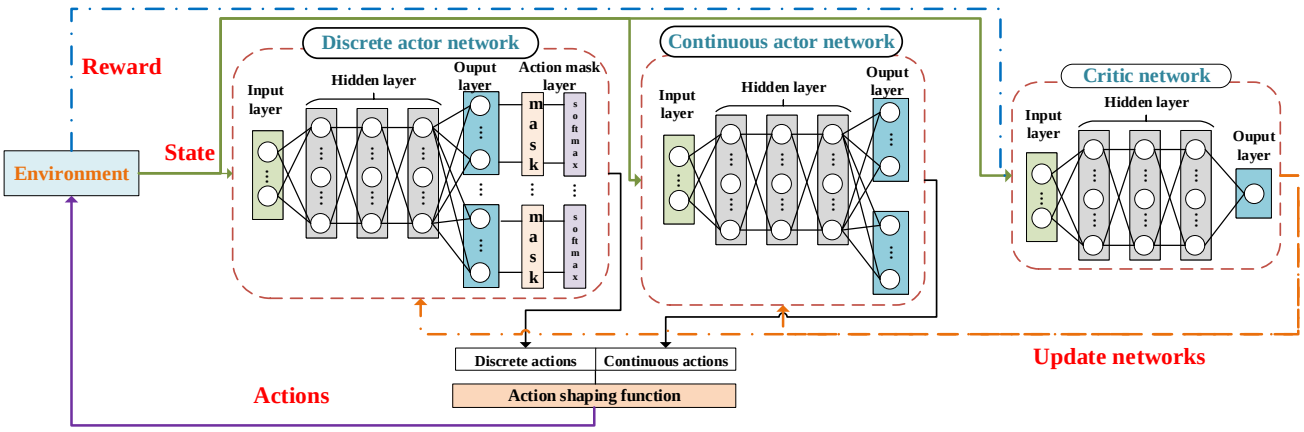


Fig. 3. An overview of the proposed H-AC architecture.

and

$$\text{P2:} \max_{\boldsymbol{\pi}} \quad \mathbb{E}\left[\lim_{V\to\infty}\frac{1}{VT}\sum_{v=1}^{V}\sum_{t=1}^{T}\gamma^{((v-1)T+t-1)}r(v,t)\right] \tag{29}$$
$$\text{s.t.} \quad (1), (22a), (22d), \text{ and } (22e).$$

It can be found that P0 can be approximated by P1 when policy $\boldsymbol{\pi}$ is fixed and discount factor γ is close to one. Meanwhile, solving P0 can be approximated by solving P2 when policy $\check{\boldsymbol{\pi}}$ is fixed and discount factor γ is close to one. In addition, the system state of STA at slot (v,t) includes the actions chosen by LTA at time window (v). The reward function of LTA contains the reward function of STA. Therefore, jointly training STA and FTA to get the optimal objective value of P1 is approximately equal to obtaining the optimal decision variables for P0. An intuitive idea is to train LTA and STA separately by leveraging off-policy RL-based methods, which can make full use of training data through the experience replay technique. However, one fundamental issue occurs when off-policy RL-based methods are applied. The Markov property of H-MDP will not be held since the state transition of LTA is not only dependent on actions of LTA but also impacted by the subsequent actions of STA. If LTA is trained by sampling trajectories from the experience replay buffer, the policy of STA is changing, which results in a non-stationary training process.

Obviously, when policy $\check{\boldsymbol{\pi}}$ is fixed, the optimization of policy $\boldsymbol{\pi}$ leads to the improvement in the objective value of P2. Meanwhile, the objective value of P1 can be improved by optimizing $\check{\boldsymbol{\pi}}$ when policy $\boldsymbol{\pi}$ is fixed. To this end, P0 can be optimized monotonically under the premise of two conditions: (1) policy $\boldsymbol{\pi}$ is optimized monotonically with fixed policy $\check{\boldsymbol{\pi}}$; (2) policy $\check{\boldsymbol{\pi}}$ is optimized monotonically with fixed policy $\boldsymbol{\pi}$. Therefore, to improve the stationarity of training process, the on-policy RL algorithms are considered, which use the data sampled by the current policy for updating. Moreover, STA and LTA are trained based on the two above conditions. Specifically, when STA or LTA samples trajectories for policy updating, the policies of STA or LTA should keep unchanged.

V. H-PPO-Based Algorithm

Although P0 can be tackled by optimizing policy $\check{\boldsymbol{\pi}}$ and policy $\boldsymbol{\pi}$ iteratively, it is still challenging to directly solve P1 and P2 by applying typical RL algorithms. On one hand, it is intractable to improve the objective value and guarantee the constraints simultaneously with unknown state transition probabilities. On the other hand, most RL algorithms are designed for either continuous or discrete action space, while both problems have parameterized action space [31]. Specifically, it requires LTA and STA first to select offloading link selection actions from a discrete list of actions and then choose the continuous bandwidth allocation actions under constraints (1) and (2). Some RL-based solutions directly convert the discrete action into a continuous action space or transform continuous actions into discrete ones, which will inevitably lead to the curse of dimensionality or lose the advantages of fine-grained control.

In this section, we first select the proper DRL algorithm as a primary solution. Then, we propose an H-PPO-based algorithm with an H-AC architecture to efficiently solve P1 and P2 to get the optimal policy for P0.

A. Primary Algorithm Design

As stated in the previous section, the on-policy DRL algorithms should be considered to improve the stationarity of policy updating process. In the ISTN network, the state transition probability is unknown due to the stochastic network environment. Furthermore, the setup of H-MDP indicates that the sizes of action space and state space of STA and LTA are infinite. Therefore, the model-free policy gradient (PG)-based DRL algorithm, which makes action decisions according to actions' probabilities [32], is adopted to efficiently solve P1 and P2.

The core idea of the PG methods is to control the action policy of the agent through the parameter $\boldsymbol{\theta}$, which is expressed as $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. As the expected value can be estimated through the statistical trajectory collected from the environment, action policy can be updated by repeatedly estimating the gradient. However, one big downside of the PG method is the low sample efficiency since the collected trajectory following the

current policy can be used only once for gradient update, which means once the parameter $\boldsymbol{\theta}$ is updated, new trajectories sampled following the new updated policy are required to estimate the gradient. Meanwhile, PG methods suffer from high variance due to the long sample trajectory and large reward value scale of different states, which leads to an unstable learning process and invalid policy updates.

To ameliorate the above drawbacks, trust region policy optimization (TRPO) [33] is proposed to improve performance reliability and data efficiency of learning by leveraging importance sampling techniques and regularization of KL divergence. Stem from TRPO, PPO [34] is then proposed as a state-of-art on-policy RL method which is much simpler for implementation. Basically, PPO enables multiple epochs of minibatch updates by optimizing a surrogate objective function given by

$$L^{clip}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_t \left[\min \left(\zeta_t(\boldsymbol{\theta}) \hat{A}_t, \text{clip} \left(\zeta_t(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon \right) \hat{A}_t \right) \right] \tag{30}$$

where $\zeta_t(\boldsymbol{\theta})$ is the probability ratio calculated by

$$\zeta_t(\boldsymbol{\theta}) = \frac{\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\boldsymbol{\pi}_{\boldsymbol{\theta}_{old}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}. \tag{31}$$

Here, ϵ is a hyperparameter that measures the degree of deviation between the new policy and the old policy.

B. H-PPO-Based Algorithm Design

Generally, PPO is implemented based on the AC architecture, where one actor learns a stochastic policy $\boldsymbol{\pi}$ and one critic works as an estimator of the state-value function $V(\boldsymbol{s})$. However, directly applying the basic DNN model to make action selections is not suitable for parameterized action space and cannot meet constraints for P1 and P2. Inspired by [35], we design an H-AC architecture with an action mask layer and an action shaping function to generate suitable actions for LTA and STA when interacting with the environment. The proposed H-AC architecture is shown in Fig. 3.

1) H-AC architecture design: Considering that action space of LTA and STA are similar with each other, we take STA as an example to elucidate the implementation of the proposed H-AC architecture. For notation simplicity, $(v,t), \forall v \in \mathcal{V}, t \in \mathcal{T}$ is replaced by $(t), 1 \leq t \leq VT$ to represents all slots in the H-AC architecture design. Different from basic AC architecture, we devise two independent actor networks with respective parameters $\boldsymbol{\theta}^d$ and $\boldsymbol{\theta}^c$ to make discrete actions and continuous actions in parallel for STA in the proposed H-AC architecture. The short-timescale policy $\boldsymbol{\pi}$ is decomposed into discrete policy $\boldsymbol{\pi}^d$ (choosing discrete action $\boldsymbol{a}^d(t)$) and continuous policy $\boldsymbol{\pi}^c$ (choosing continuous action $\boldsymbol{a}^c(t)$), where $\boldsymbol{\pi}(\boldsymbol{a}(t)|\boldsymbol{s}(t)) = \boldsymbol{\pi}^d(\boldsymbol{a}^d(t)|\boldsymbol{s}(t))\boldsymbol{\pi}^c(\boldsymbol{a}^c(t)|\boldsymbol{s}(t))$. For discrete actor network, the observed state $\boldsymbol{s}(t)$ is mapped to U heads through shared hidden layers. Each head produces $(N+1)$ digits which are then passed to the softmax function to generate a $(N+1)$-dimension vector $\boldsymbol{\pi}_u^d(\boldsymbol{s}(t))$ = $[\boldsymbol{\pi}_u^d(0|\boldsymbol{s}(t)), \cdots, \boldsymbol{\pi}_u^d(N|\boldsymbol{s}(t))]$. Parameters of $\boldsymbol{\pi}_u^d(\boldsymbol{s}(t))$ are the probability value of possible discrete actions that can be selected for user u. The discrete action for user u to take at slot (t) is sampled from the $\boldsymbol{\pi}_u^d(\boldsymbol{s}(t))$ distribution. To this end, the probability value of taking discrete action $\boldsymbol{a}^d(t)$ for all users is $\boldsymbol{\pi}^d(\boldsymbol{a}^d(t)|\boldsymbol{s}(t)) = \prod_{u=1}^{U} \boldsymbol{\pi}_u^d(c_u^{UTB}(t)|\boldsymbol{s}(t))$. For continuous actor network, the stochastic policy $\boldsymbol{\pi}^c$ is generated by outputting the means and variances of a total number of U Gaussian distributions for all continuous actions.

Similar to the typical PPO method, a single critic network with parameter $\boldsymbol{\phi}$ is adopted in the proposed H-AC architecture. To stabilize and smooth the learning process with multi-dimension action space, generalized advantage estimator for calculating advantage function $\hat{A}(t)$ is implied in (30), which is given as

$$\hat{A}(t) = \delta(t) + (\gamma\eta)\delta(t+1) + \cdots + (\gamma\eta)^{VT-t+1}\delta(VT-1). \tag{32}$$

Here $\delta(t) = r(t) + \gamma V_{\boldsymbol{\phi}}(s(t+1)) - V_{\boldsymbol{\phi}}(s(t))$, η is a discount hyperparameter. $V_{\boldsymbol{\phi}}(s(t))$ is the state-value function approximated by the critic network.

For j-th iteration of training, the continuous policy $\boldsymbol{\pi}^c$ and discrete policy $\boldsymbol{\pi}^d$ are updated separately by maximizing their respective surrogate objective functions,

$$\begin{aligned} \boldsymbol{\theta}_{j+1}^c = \arg\max_{\boldsymbol{\theta}^c} \frac{1}{|\mathcal{D}|VT} \sum_{\sigma \in \mathcal{D}} \sum_{t=1}^{VT} \min \Big(& \zeta_t(\boldsymbol{\theta}^c) \hat{A}^{\boldsymbol{\theta}^c}(t), \\ & \text{clip} \left(\zeta_t(\boldsymbol{\theta}^c), 1-\epsilon, 1+\epsilon \right) \hat{A}^{\boldsymbol{\theta}^c}(t) \Big) \end{aligned} \tag{33}$$

and

$$\begin{aligned} \boldsymbol{\theta}_{j+1}^d = \arg\max_{\boldsymbol{\theta}^d} \frac{1}{|\mathcal{D}|VT} \sum_{\sigma \in \mathcal{D}} \sum_{t=1}^{VT} \min \Big(& \zeta_t(\boldsymbol{\theta}^d) \hat{A}^{\boldsymbol{\theta}^d}(t), \\ & \text{clip} \left(\zeta_t(\boldsymbol{\theta}^d), 1-\epsilon, 1+\epsilon \right) \hat{A}^{\boldsymbol{\theta}^d}(t) \Big) \end{aligned} \tag{34}$$

via gradient ascent methods, where $\mathcal{D}$ is the buffer memory. The parameter $\boldsymbol{\phi}$ of critic network is updated by minimizing the loss function, i.e.,

$$\boldsymbol{\phi}_{j+1} = \arg\min_{\boldsymbol{\phi}} \frac{1}{|\mathcal{D}|VT} \left(V_{\boldsymbol{\phi}}(\boldsymbol{s}(t)) - \hat{R}(\boldsymbol{s}(t)) \right)^2 \tag{35}$$

where $\hat{R}(\boldsymbol{s}(t))$ is the cumulative rewards starting from state $\boldsymbol{s}(t)$.

2) Action mask layer design: Due to the distance and communication window length constraints, the available discrete action sets of STA and LTA vary at different states. Added to that, constraint (22d) needs to be guaranteed and requires the valid discrete action set for STA at different states. However, the output size of the discrete actor network is fixed, which is incorrect for action selection. Therefore, we design an action mask layer between the output layer and softmax layer for each head in discrete actor network. Each action mask layer helps to avoid sampling invalid actions by adding a large negative number (e.g., -1×10^{10}) to the outputted logits of the invalid actions [36].

3) Action shaping function design: Although constraints (22a) - (22d) are satisfied by adopting the action mask layer, the resource constraint (22e) is still left to be guaranteed. In this context, we design an action shaping function to generate feasible actions for STA when interacting with the environment. In the action shaping function, a U-dimension binary coding vector (in which the u-th parameter is equal to 0

Algorithm 1: H-PPO-Based Algorithm

1 Initialize the actor networks and critic networks for STA and LTA with parameters $\boldsymbol{\theta}^c$, $\boldsymbol{\theta}^d$, $\boldsymbol{\phi}$, $\check{\boldsymbol{\theta}}^c$, $\check{\boldsymbol{\theta}}^d$, and $\check{\boldsymbol{\phi}}$;
2 Initialize buffer $\mathcal{D} \leftarrow \emptyset$, $\check{\mathcal{D}} \leftarrow \emptyset$;
3 **foreach** *iteration* **do**
4 Initialize environment;
5 **for** $v \in \{1, 2, \cdots, V\}$ **do**
6 Observe state $\check{\boldsymbol{s}}(v)$;
7 Select action $\check{\boldsymbol{a}}(v)$ based on $\check{\boldsymbol{a}}^c(v) \sim \check{\boldsymbol{\pi}}^c(\cdot|\check{\boldsymbol{s}}(v))$ and $\check{\boldsymbol{a}}^d(v) \sim \check{\boldsymbol{\pi}}^d(\cdot|\check{\boldsymbol{s}}(v))$;
8 Get action $\check{\boldsymbol{a}}(v)$ processed by action shaping function;
9 Execute action $\check{\boldsymbol{a}}(v)$;
10 **for** $t \in \{1, 2, \cdots, T\}$ **do**
11 Observe state $\boldsymbol{s}(v,t)$;
12 Select action $\boldsymbol{a}(v,t)$ based on $\boldsymbol{a}^c(v,t) \sim \boldsymbol{\pi}^c(\cdot|\boldsymbol{s}(v,t))$ and $\boldsymbol{a}^d(v,t) \sim \boldsymbol{\pi}^d(\cdot|\boldsymbol{s}(v,t))$;
13 Get action $\boldsymbol{a}(v,t)$ processed by action shaping function;
14 Execute action $\boldsymbol{a}(v,t)$;
15 Receive reward $r(v,t)$ and observe new state $\boldsymbol{s}(v,t+1)$;
16 Store $(\boldsymbol{s}(v,t), \boldsymbol{a}(v,t), r(v,t), \boldsymbol{s}(v,t+1), \boldsymbol{a}^c(v,t), \boldsymbol{a}^d(v,t))$ into buffer $\mathcal{D}$;
17 Receive reward $\check{r}(v)$ and observe new state $\check{\boldsymbol{s}}(v+1)$;
18 Store $(\check{\boldsymbol{s}}(v), \check{\boldsymbol{a}}(v), \check{r}(v), \check{\boldsymbol{s}}(v+1), \check{\boldsymbol{a}}^c(v), \check{\boldsymbol{a}}^d(v))$ into buffer $\check{\mathcal{D}}$;
19 Read trajectories in buffer $\mathcal{D}$, $\boldsymbol{\pi}^c_{old} \leftarrow \boldsymbol{\pi}^c$, $\boldsymbol{\pi}^d_{old} \leftarrow \boldsymbol{\pi}^d$;
20 Update the policy $\boldsymbol{\pi}^c$ with eq. (33) via gradient ascent methods;
21 Update the policy $\boldsymbol{\pi}^d$ with eq. (34) via gradient ascent methods;
22 Update the critic network parameter $\boldsymbol{\phi}$ with eq. (35) via gradient descent methods;
23 Read trajectories in buffer $\check{\mathcal{D}}$, $\check{\boldsymbol{\pi}}^c_{old} \leftarrow \check{\boldsymbol{\pi}}^c$, $\check{\boldsymbol{\pi}}^d_{old} \leftarrow \check{\boldsymbol{\pi}}^d$;
24 Update the policy $\check{\boldsymbol{\pi}}^c$ similarly with eq. (33) via gradient ascent methods;
25 Update the policy $\check{\boldsymbol{\pi}}^d$ similarly with eq. (34) via gradient ascent methods;
26 Update the critic network parameter $\check{\boldsymbol{\phi}}$ similarly with eq. (35) via gradient descent methods;
27 Reset buffer $\check{\mathcal{D}} \leftarrow \emptyset$, $\mathcal{D} \leftarrow \emptyset$

when $c_u^{UTB}(v,t) = 0$; otherwise the u-th parameter is equal to 1) and a shaping function are included. The core idea is that the selected continuous action $\boldsymbol{a}^c(v,t)$ is first passed through a U-dimension binary coding vector to get a feasible action. Then, if constraint (22e) is not satisfied, the selected continuous action processed by coding vector is shaped by the shaping function until $R_b^{BS}(v,t) = C_b^E(v,t),\ \forall b, v, t$. Specifically, the achievable rate, i.e., $R_{u,b}^U(v,t)$, from each user u to BS b at slot (v,t) calculated with the selected continuous action is first divided by $R_b^{BS}(v,t)/C_b^E(v,t)$. Then, the continuous action for each user can be derived by solving (3) using Newton's method [37]. Note that the action shaping function is only applied when interacting with the environment to get the reward.

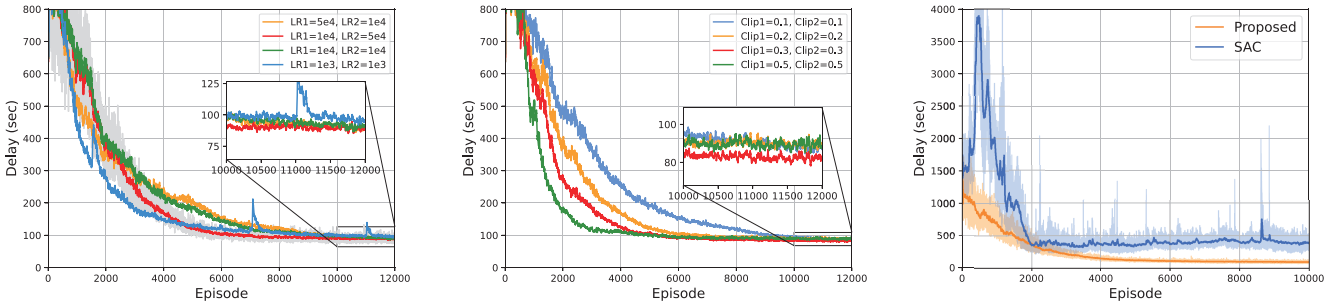
Based on the design of the H-AC architecture, the action mask layer and the action shaping function, we propose an H-PPO-based algorithm to efficiently solve P0. The detailed procedure is demonstrated in Algorithm 1. First, the actor networks and critic networks of STA and LTA are initialized with parameters $\boldsymbol{\theta}^c$, $\boldsymbol{\theta}^d$, $\boldsymbol{\phi}$, $\check{\boldsymbol{\theta}}^c$, $\check{\boldsymbol{\theta}}^d$, and $\check{\boldsymbol{\phi}}$, respectively. Buffer $\check{\mathcal{D}}$ and $\mathcal{D}$ are initialized for storing trajectories of LTA and STA, respectively (Lines 1-2). Then, during each iteration, trajectories for LTA, i.e., a sequence of transitions $(\check{\boldsymbol{s}}(v), \check{\boldsymbol{a}}(v), \check{r}(v), \check{\boldsymbol{s}}(v+1), \check{\boldsymbol{a}}^c(v), \check{\boldsymbol{a}}^d(v))$, and trajectories for STA, i.e., a sequence of transitions $(\boldsymbol{s}(v,t), \boldsymbol{a}(v,t), r(v,t), \boldsymbol{s}(v,t+1), \boldsymbol{a}^c(v,t), \boldsymbol{a}^d(v,t))$, are obtained through interactions between the softwarized controller and the environment (Lines 4-18). Based on the acquired trajectories, the actor and critic networks of STA and LTA are updated in parallel via gradient ascent (descent) methods (Lines 19-27). Note that to make timely decisions, STA and LTA are first trained offline by Algorithm 1 until LTA can obtain a good value of the episode reward (accumulation of immediate rewards of each time window within one episode). After the offline training, the softwarized controller then directly utilizes the trained LTA and STA to make offloading link selection and bandwidth allocation decisions in the ISTN.

VI. Simulation Results

In this section, simulation results are presented to demonstrate the performance of the proposed H-PPO-based algorithm and ISTN architecture for supporting task offloading.

A. Simulation Setup

Our simulations are conducted in a remote scenario where a satellite constellation and four BSs cooperatively provide service offloading for 40 users. We use the satellite tool kit to construct the whole topology system. All users are randomly distributed in a square area of 4 km $\times$ 4 km. All BSs are evenly distributed in the target area with a coverage radius of 1 km. The latitude and longitude of the center of the target area are set to be $(32.5^\circ N, 118.6^\circ E)$. The satellite constellation consists of 60×40 LEO satellites with an inclination of 96.5° orbiting at the height of 550 km, which ensures the seamless coverage of the target area with at least 3 satellites. The simulated time horizon is from 2022-6-8 08:00:00 to 2022-6-8 08:10:00. The minimum elevation angle is set to be 30° for guaranteeing the transmission quality. The backhaul capacity of each BS is 20 Mbps [15], and each task has the same size of 0.5 Mb. At each slot, the task generation of each user follows a Poisson process with $\lambda = 8$. For UTB links, the transmit power of each user is set as 0.5 W, and pathloss exponent is set as 3.5 [28]. The bandwidth for sub-6 GHz band communication is set to 20 MHz. For UTS and BTS links, the transmit power of each BS and user are set as 10 W and 3 W according to the terminal of Starlink [38], respectively. Since the satellite constellation is responsible for providing service globally, only a portion of communication resources is allocated to the target area. We set the available bandwidth for Ka-band communication as 10 MHz. The antenna gain of users, BSs, and satellites are set as 30 dBi, 40 dBi, and 43 dBi [38]. Rician fading is considered for UTS and BTS links with $K = 7$ whilst normalized Rayleigh fading is adopted for UTB links [39]. The atmospheric loss and the polarization loss are set as 0.5 dB and 1 dB [9].



(a) Training process with different learning rates (b) Training process with different clipping ratios (c) Training process of SAC and proposed algorithm

Fig. 4. The convergence performance of the proposed algorithm.

For STA, we deploy a three-layer fully-connect neural network (FCNN) with $[512, 512, 256]$ neurons for the discrete actor network, a three-layer FCNN with $[512, 256, 64]$ neurons for the continuous actor network, and a three-layer FCNN with $[512, 256, 64]$ neurons for the critic network. For LTA, we deploy a three-layer FCNN with $[1024, 512, 256]$ neurons for the discrete actor network, a three-layer FCNN with $[1024, 512, 64]$ neurons for the continuous actor network, and a three-layer FCNN with $[1024, 256, 64]$ neurons for the critic network. Tanh function is adopted as the activation function for the FCNNs. Additionally, parameters of all networks are trained by adopting Adam optimizer. Each episode consists of 120 slots or 40 time windows. To demonstrate the effectiveness of the proposed H-PPO-based algorithm, we use the following benchmark algorithms for comparison:

- *SAC*: In this algorithm, LTA and STA learn the offloading link selection and bandwidth allocation policy based on the SAC method [40] with the same action mask layer and action shaping function.
- *Equal bandwidth allocation and best channel condition association* (EB): The second benchmark is a general algorithm for access association and resource allocation, which is based on neither RL nor an optimization approach. Specifically, in this algorithm, each user is first associated to the BS or satellite with the best channel condition. Each BS is associated to the satellite with the best channel condition.[3] Then, the available spectrum resources of BSs or satellites are equally allocated to their associated users or BSs and users.
- *Random*: In this algorithm, the offloading link selection and bandwidth allocation decisions for UTS and BTS links are first selected with the same probability. Based on the above decisions, the offloading link selection and bandwidth allocation decisions for UTB links are then randomly selected in the feasible decision space.

B. Performance Evaluation

The performance evaluation of the proposed H-PPO-based algorithm contains two stages. We first show the learning procedure of the proposed algorithm. Then, the well-learned models are employed and tested under different available network resources to validate the superiority of our developed algorithm.

1) Convergence performance: The convergence performance of the proposed H-PPO-based algorithm is shown in Fig. 4. The average delay (i.e., the overall offloading delay of all tasks divided by the amount of time slots) with different learning rates is illustrated by Fig. 4(a), where LR1 is the learning rate for LTA and LR2 is the learning rate for STA. To highlight the convergence trend, points of delay are processed by a twenty-point moving average. Additionally, we plot the curve of average delay with $LR1 = 0.0001, LR2 = 0.0005$ (i.e., grey curve) to show the learning process. We can observe that all curves decline with the increase of the number of training episodes and gradually saturate at their corresponding optimal values, which verifies the convergence property of the proposed H-PPO-based algorithm. Moreover, a large learning rate leads to a fast convergence speed of delay. However, unstable convergence performance will be resulted with too large learning rate. In addition, it can be observed that setting a larger learning rate for STA than LTA results in efficient convergence performance, which can be taken as a reference for learning rate setup in the simulations with different resources and environment conditions. The average delay with different clip ratios is illustrated by Fig. 4(b), where Clip1 is the clip ratio for LTA and Clip2 is the clip ratio for STA. Note that points of delay are also processed by a twenty-point moving average. It can be seen that the average delay converges faster with a larger clip ratio due to the more aggressive update of policy in each iteration. However, the average delay with too large clip ratio is apt to premature convergence and trap in local optimum. Therefore, we choose 0.3 as the clip ratios of STA and LTA in the remaining simulations.

To further study the convergence and the effectiveness of the proposed H-PPO-based algorithm, we compare the convergence performance of the proposed H-PPO-based algorithm with the SAC algorithm as shown in Fig. 4(c). Ten independent random seeds are deployed for the experiments to show the performance in a stochastic environment. The mean average delay value processed by a twenty-point moving average is represented as solid curves, whilst the min-max bounds of performance are indicated by light-colored regions. It can be

[3]To avoid the situation that all BSs are associated with a single satellite, each satellite can only be linked with less than a total of two BSs, which have more tasks to be offloaded.

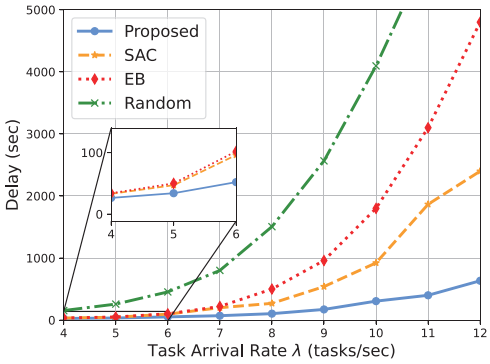
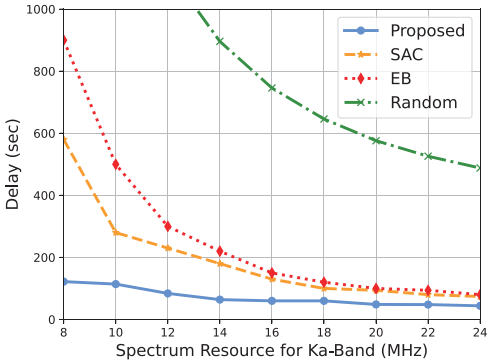
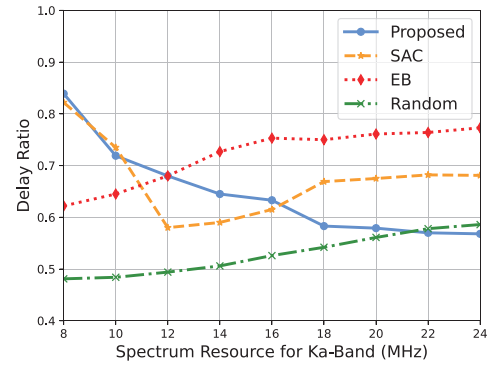


Fig. 5. Impact of task arrival rate on delay performance



(a) The average delay performance

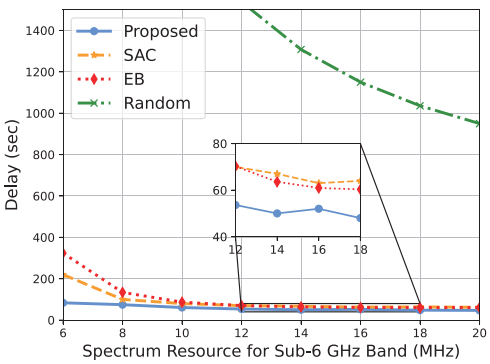


(b) The delay ratio performance

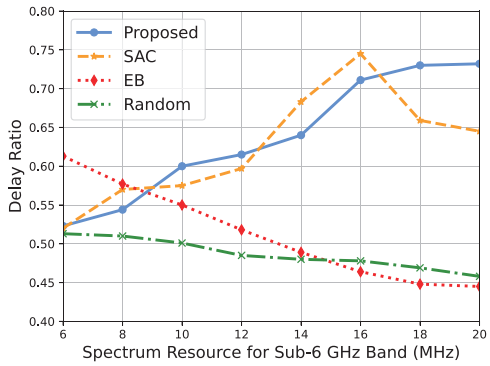
Fig. 6. Performance versus the available Ka-band spectrum resources.

observed that the converged value achieved by our proposed algorithm is 74.5% lower than that of the SAC algorithm. Although the SAC algorithm can achieve optimal level of performance after 2000 episodes with certain seeds, the performance of the proposed algorithm shows smaller variance and higher stability in the training process. The reason is that the proposed algorithm updates the policy of STA and LTA in an on-policy manner, which eliminates the bias incurred by applying the off-policy RL methods.

2) Impact of task arrival rate: Fig. 5 shows the performance of different algorithms with respect to the task arrival rate. As expected, the average delay increases with larger



(a) The average delay performance



(b) The delay ratio performance

Fig. 7. Performance versus the available sub-6 GHz band spectrum resources.

task arrival rates since the unchanged amount of resources is shared by more tasks. In addition, our proposed H-PPO-based algorithm achieves the lowest average delay among all algorithms. The superiority of our proposed algorithm is more remarkable in heavy task scenarios. Specifically, the average delay achieved by the proposed algorithm is 72.8% lower than that by the best benchmark algorithm with $\lambda = 12$.

3) Impact of spectrum resources: The amount of spectrum resources is of the utmost importance in deciding the performance of different algorithms in the ISTN. Fig. 6 shows the performance of different algorithms with respect to available Ka-band spectrum resources. As shown in Fig. 6(a), our proposed algorithm outperforms all benchmark algorithms on average delay, especially with inadequate spectrum resources. For instance, the average delay of the proposed algorithm for the case with $B_1 = 8$ MHz is 78.9% lower than that of the best benchmark algorithm. Meanwhile, Fig. 6(b) depicts the delay ratio (i.e., the average delay achieved by LTA divided by the average delay achieved by STA) versus the available Ka-band spectrum resources. With the increase of Ka-band spectrum resources, the ratio value of the proposed algorithm shows an overall trend of decrease. This indicates that the proposed algorithm allocates more spectrum resources to BSs to alleviate the shortage of backhaul capacity, which is the reason for the superior performance of the proposed algorithm shown in Fig. 6(a).

Figure 7(a) shows the average delay performance of dif-

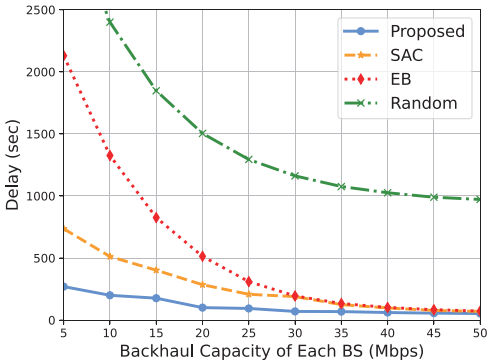


Fig. 8. Delay performance versus the available backhaul capacity of each BS.

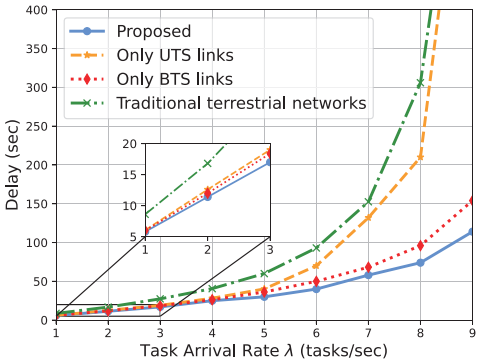


Fig. 9. The average delay performance versus task arrival rate of different network architectures.

ferent algorithms with respect to available sub-6 GHz band spectrum resources. The backhaul capacity of each BS is set as 60 Mbps. The results show that the obtained average delays by the proposed algorithms, SAC algorithm, and EB algorithm first decrease sharply with the increase of the available sub-6 GHz band spectrum resources, and almost saturate at small values after $B_0 = 14$ MHz. The reason is that the spectrum resources allocated to UTB links are reduced once constraint (22e) is not satisfied, which sets a rate bound for UTB links. It can also be seen that our proposed algorithm surpasses the benchmarks over the entire horizontal axis. Specifically, when $B_0 = 20$ MHz, the 40.5%, 45%, and 90.9% improvements in the average delay are achieved by the proposed algorithm compared with benchmark algorithms, i.e., SAC, EB, and random, respectively. In addition, the delay ratio versus the available sub-6 GHz band spectrum resources is illustrated by Fig. 7(b). The ratio value of the proposed algorithm shows an overall trend of increase, which endorses that the proposed algorithm can make wise decisions to adapt to the increase of sub-6 GHz band spectrum resources.

Figure 8 shows the impact of the available terrestrial backhaul capacity of each BS on the average delay. With the increase of the available terrestrial backhaul capacity, the obtained average delays by all algorithms decrease. In addition, we can observe that the average delay is improved slowly when C^{BS} is larger than 40 MHz, which implies that favorable backhaul resources should be arranged in the ISTN to gain a high resources utilization rate. Moreover, our proposed algorithm outperforms all benchmarks in the average delay. For example, the average delay obtained by the proposed algorithm is 63.3% lower than that by the best benchmark algorithm with $C^{BS} = 5$ MHz.

4) Performance evaluation with different network architectures: In addition to the comparisons with three offloading link selection and bandwidth allocation benchmark algorithms, we also conduct the experiment to demonstrate the effectiveness of the proposed ISTN architecture with three different network architectures: (1) satellites are limited to providing connections for users only, which is called as "only UTS links"; (2) satellites are limited to providing connections for BSs only, which is called as "only BTS links"; and (3) only BSs are available to provide service offloading for users, which is called as "traditional terrestrial network". For fairness, in this network architecture, the available bandwidth of Ka-band is added to that of sub-6 GHz band, and the backhaul of each BS is set as 50 Mbps. As depicted in Fig. 9, the proposed architecture achieves lower average delay compared with the other three network architectures. Specifically, the average delay obtained by the proposed architecture is improved by 26.3% compared with that of the best benchmark architecture in the heaviest traffic scenario. The reason is that the proposed architecture can not only mitigate the scarcity of backhaul capacity but also provide UTS links as a supplement for users with bad UTB channel conditions. Moreover, although the backhaul capacity in traditional terrestrial networks is increased to 50 Mbps, it still gains the worst performance among all network architectures, which endorses the indispensable function of satellites in the future networks.

VII. Conclusion

In this paper, we have proposed an ISTN architecture to support task offloading for remote IoT users. An H-PPO-based algorithm has been proposed to make offloading link selection and bandwidth allocation decisions in real-time to minimize the overall task offloading delay while accommodating the dynamic task arrivals and channel conditions without future information. From the numerical results, the proposed H-PPO-based algorithm can reduce the delay with different task arrival rates and spectrum resources by 63.87% and 84.23% on average compared with SAC and EB algorithms. Furthermore, the proposed ISTN architecture can achieve up to 75.68% delay reduction compared with the terrestrial network only. The ISTN architecture can also be applied to serve remote IoT users to achieve low-latency task offloading in scenarios with heavy traffic loads. For the future work, we will further investigate the task offloading scheduling for mobile users in the ISTN to cope with the spatial-temporal variations of user density.

References

[1] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato, O. Dobre, and H. V. Poor, "6G Internet of Things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, Jan. 2022.

[2] C. Zhou, W. Wu, H. He, P. Yang, F. Lyu, N. Cheng, and X. Shen, "Deep reinforcement learning for delay-oriented IoT task scheduling in SAGIN," *IEEE Trans. Wireless. Commun.*, vol. 20, no. 2, pp. 911–925, Feb. 2021.

[3] M. Centenaro, C. E. Costa, F. Granelli, C. Sacchi, and L. Vangelista, "A survey on technologies, standards and open challenges in satellite IoT," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1693–1720, 3rd Quart. 2021.

[4] M. Giordani and M. Zorzi, "Non-terrestrial networks in the 6G era: Challenges and opportunities," *IEEE Network*, vol. 35, no. 2, pp. 244–251, Mar./Apr. 2021.

[5] K. Liolis, A. Geurtz, R. Sperber, D. Schulz, S. Watts, G. Poziopoulou, B. Evans, N. Wang, O. Vidal, B. Tiomela Jou *et al.*, "Use cases and scenarios of 5G integrated satellite-terrestrial networks for enhanced mobile broadband: The SaT5G approach," *Int. J. Satell. Commun. Netw.*, vol. 37, no. 2, pp. 91–112, 2019.

[6] X. Fang, W. Feng, T. Wei, Y. Chen, N. Ge, and C.-X. Wang, "5G embraces satellites for 6G ubiquitous IoT: Basic models for integrated satellite terrestrial networks," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 14 399–14 417, Sept. 2021.

[7] 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. 12 098–12 110, Oct. 2020.

[8] D. Zhou, M. Sheng, Y. Wang, J. Li, and Z. Han, "Machine learning-based resource allocation in satellite networks supporting Internet of remote things," *IEEE Trans. Wireless. Commun.*, vol. 20, no. 10, pp. 6606–6621, Oct. 2021.

[9] T. Chen, J. Liu, Q. Ye, W. Zhuang, W. Zhang, T. Huang, and Y. Liu, "Learning-based computation offloading for IoRT through Ka/Q-band satellite-terrestrial integrated networks," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12 056–12 070, July 2022.

[10] M. Sadek and S. Aissa, "Personal satellite communication: technologies and challenges," *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 28–35, Dec. 2012.

[11] 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.

[12] Q. Ye, W. Shi, K. Qu, H. He, W. Zhuang, and X. Shen, "Joint RAN slicing and computation offloading for autonomous vehicular networks: A learning-assisted hierarchical approach," *IEEE Open J. Veh. Technol.*, vol. 2, pp. 272–288, June 2021.

[13] H. Huang, S. Guo, W. Liang, K. Wang, and A. Y. Zomaya, "Green data-collection from geo-distributed IoT networks through low-earth-orbit satellites," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 3, pp. 806–816, Sept. 2019.

[14] C. Niephaus, J. Mödeker, and G. Ghinea, "Toward traffic offload in converged satellite and terrestrial networks," *IEEE Trans. on Broadcast.*, vol. 65, no. 2, pp. 340–346, June 2019.

[15] B. Di, H. Zhang, L. Song, Y. Li, and G. Y. Li, "Ultra-dense LEO: Integrating terrestrial-satellite networks into 5G and beyond for data offloading," *IEEE Trans. Wireless. Commun.*, vol. 18, no. 1, pp. 47–62, Jan. 2019.

[16] Y. Li, J. Huang, Q. Sun, T. Sun, and S. Wang, "Cognitive service architecture for 6G core network," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 7193–7203, Oct. 2021.

[17] L. Zhao, K. Yang, Z. Tan, X. Li, S. Sharma, and Z. Liu, "A novel cost optimization strategy for SDN-enabled UAV-assisted vehicular computation offloading," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3664–3674, June 2021.

[18] N. Lin, L. Fu, L. Zhao, G. Min, A. Al-Dubai, and H. Gacanin, "A novel multimodal collaborative drone-assisted VANET networking model," *IEEE Trans. Wireless. Commun.*, vol. 19, no. 7, pp. 4919–4933, July 2020.

[19] Z. Gao, A. Liu, C. Han, and X. Liang, "Max completion time optimization for Internet of things in LEO satellite-terrestrial integrated networks," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9981–9994, June 2021.

[20] L. Zhao, C. Wang, K. Zhao, D. Tarchi, S. Wan, and N. Kumar, "INTERLINK: A digital twin-assisted storage strategy for satellite-terrestrial networks," *IEEE Trans. Aerosp. Electron. Syst.*, 2022, doi:10.1109/TAES.2022.3169130.

[21] W. Abderrahim, O. Amin, M.-S. Alouini, and B. Shihada, "Latency-aware offloading in integrated satellite terrestrial networks," *IEEE open j. Commun. Soc.*, vol. 1, pp. 490–500, April 2020.

[22] F. Tang, H. Hofner, N. Kato, K. Kaneko, Y. Yamashita, and M. Hangai, "A deep reinforcement learning-based dynamic traffic offloading in space-air-ground integrated networks (SAGIN)," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 276–289, Jan. 2022.

[23] Y. Cao, S.-Y. Lien, and Y.-C. Liang, "Deep reinforcement learning for multi-user access control in non-terrestrial networks," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 1605–1619, Mar. 2021.

[24] D. Han, W. Liao, H. Peng, H. Wu, W. Wu, and X. Shen, "Joint cache placement and cooperative multicast beamforming in integrated satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 3131–3143, Mar. 2022.

[25] N. Saeed, A. Elzanaty, H. Almorad, H. Dahrouj, T. Y. Al-Naffouri, and M.-S. Alouini, "Cubesat communications: Recent advances and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1839–1862, 3rd Quart. 2020.

[26] A. Dissanayake, J. Allnutt, and F. Haidara, "A prediction model that combines rain attenuation and other propagation impairments along earth-satellite paths," *IEEE Trans. Antennas Propag.*, vol. 45, no. 10, pp. 1546–1558, Oct. 1997.

[27] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing-based iot," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2146–2153, June 2018.

[28] H. Wu, J. Chen, C. Zhou, J. Li, and X. Shen, "Learning-based joint resource slicing and scheduling in space-terrestrial integrated vehicular networks," *Journal of Communications and Information Networks*, vol. 6, no. 3, pp. 208–223, Sept. 2021.

[29] S. Li, R. Wang, M. Tang, and C. Zhang, "Hierarchical reinforcement learning with advantage-based auxiliary rewards," in *Proc. NIPS*, vol. 32, 2019.

[30] H. Peng and X. Shen, "Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2416–2428, Oct. 2020.

[31] W. Masson, P. Ranchod, and G. Konidaris, "Reinforcement learning with parameterized actions," in *Proc. Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[32] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[33] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. PMLR*, 2015, pp. 1889–1897.

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

[35] Z. Fan, R. Su, W. Zhang, and Y. Yu, "Hybrid actor-critic reinforcement learning in parameterized action space," *arXiv preprint arXiv:1903.01344*, 2019.

[36] S. Huang and S. Ontañón, "A closer look at invalid action masking in policy gradient algorithms," *arXiv preprint arXiv:2006.14171*, 2020.

[37] A. Gil, J. Segura, and N. M. Temme, *Numerical methods for special functions*. SIAM, 2007.

[38] FCC. Space exploration holdings LCC. (2021) ses-lic-intr2021–02141. Accessed June 2022. [Online]. Available: https://fcc.report/IBFS/SES-LIC-INTR2021-02141.

[39] Q. Huang, M. Lin, J.-B. Wang, T. A. Tsiftsis, and J. Wang, "Energy efficient beamforming schemes for satellite-aerial-terrestrial networks," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3863–3875, June 2020.

[40] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.