Learning-based Computation Offloading for IoRT through Ka/Q-Band Satellite-Terrestrial Integrated Networks

Tianjiao Chen, Jiang Liu, Qiang Ye, *Member, IEEE*, Weihua Zhuang, *Fellow, IEEE*, Weiting Zhang, Tao Huang, *Senior Member, IEEE*, Yunjie Liu

***Abstract*—In this paper, we propose a multi-layer Ka/Q-band satellite-terrestrial integrated network for the Internet of remote things (IoRT) to achieve a high transmission rate with communication robustness in dynamic network environments. Under this architecture, we investigate how to jointly manage the offloading path selection and resource allocation to offload computation-intensive and delay-sensitive tasks in the IoRT. Considering continuous low earth orbit (LEO) satellite movements and Markovian rainfall changes, the computation offloading problem is described as a Markov decision process (MDP) formulation with the objective of maximizing the number of offloaded tasks with satisfied delay requirements and minimizing the power consumption of the LEO satellites. A deep reinforcement learning (DRL) approach is leveraged to make optimal decisions by taking account of dynamic queues of IoRT devices, channel conditions that vary with rainfall intensities and satellite positions, and computing capabilities of ground stations. Extensive simulations are conducted to validate the effectiveness and superiority of our proposed scheme.**

***Index Terms*—Multi-band satellites, Internet of remote things, computation offloading, resource management, deep reinforcement learning.**

I. Introduction

THE past few years have witnessed the development and advancement of the Internet of things (IoT), which accommodates enormous mobile users and devices to be interconnected for information sharing [1]. To describe the situation where intelligent objects are located in remote areas or scattered in a wide geographical area, the concept of the Internet of remote things (IoRT) is emerging [2]. The IoRT is generally used to support monitoring and sensing services that terrestrial networks may not provide in deserted, forest, and oceanic areas, where the conventional communication infrastructures are inaccessible due to extreme geographical conditions and labor-intensive construction costs. In such cases, using satellite networks with high-performance backhaul links to support IoRT service deliveries is one of the promising and effective solutions [3], [4].

Depending on the orbital heights, satellites are divided into three categories: geostationary earth orbit (GEO), medium earth orbit (MEO), and low earth orbit (LEO) satellites. Each GEO satellite has a fixed position and covers almost half of the earth, supporting a stable regional communication. Compared with GEO satellites, MEO and LEO satellites have lower link delays and more available orbits [5]. In recent years, with the development of low-launching-cost LEO satellites, satellite companies such as SpaceX and OneWeb [6] plan to launch thousands of LEO satellites operating over high-frequency bands, aiming to deploy an ultra-dense constellation to provide low-delay and high-capacity communication services for ground users.

In recent years, emergency management has become one of the essential applications of IoRT due to frequent natural disasters [7]. In such a disaster scenario, IoRT devices not only provide data collection services, but also need to analyze the collected data to make rapid and intelligent disaster rescue decisions such as human body detection in ruins [8]. These decisions usually use deep learning to achieve image processing and video recognition [9]. However, IoRT devices cannot locally process these computation-intensive and time-sensitive tasks due to the limited computing and energy resources. Therefore, the computation tasks need to be transmitted through satellites to ground stations with abundant computing capabilities for processing. Since LEO satellites provide large-scale coverage and low-latency data transmission, computation offloading in IoRT can be effectively assisted by LEO satellite communications [10], [11]. In such a process, the transmission efficiency of satellite communications has a significant impact on the performance of computation offloading. Currently, Ka-band is usually used in satellite transmission to support high data-rate delivery [12]. With the continuous occupancy of the Ka-band, the residual available Ka-band resources are gradually shrunk. Therefore, Q-band starts to be exploited for more available bandwidth resources [13]. Hence Starlink makes efforts in launching Q-band satellites in 350 km orbits [14]. Nevertheless, the higher frequency band is affected by

This work was supported by the National Natural Science Foundation of China (No. 62171064), the Beijing Municipal Science and Technology Project (No. Z211100004421019) and the China Scholarship Council. *(Corresponding author: Jiang Liu.)*

Tianjiao Chen, Jiang Liu, Tao Huang and Yunjie Liu are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, P.R. China. They are also with the Purple Mountain Laboratories, Nanjing, 211111, P.R. China (e-mail: 2013210074@bupt.edu.cn; liujiang@bupt.edu.cn; htao@bupt.edu.cn; liuyj@chinaunicom.cn).

Qiang Ye is with the Department of Electrical and Computer Engineering and Technology, Minnesota State University, Mankato, MN 56001, USA (email: qiang.ye@mnsu.edu).

Weihua Zhuang is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (email: wzhuang@uwaterloo.ca).

Weiting Zhang is with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, 100044, P.R. China (e-mail: 17111018@bjtu.edu.cn).

more rain attenuation, which reduces the transmission rate [15]. Most of the existing works [16]–[19] have assumed the same portion of frequency bands to be used by satellite communications for data or computation offloading. However, using only Ka- or Q-band will cause the problems mentioned above due to their frequency band characteristics. Moreover, using the same portion of frequency bands in multi-layer LEO networks leads to inter-layer co-channel interference [20].

In this paper, to address the above problems, we propose a multi-layer Ka/Q-band satellite-terrestrial integrated network for the IoRT computation offloading. In this architecture, the LEO satellite network consists of Ka-band satellites at a lower altitude and Q-band satellites at a higher altitude. Each IoRT device offloads its computation tasks via a Ka- or Q-band LEO satellite, which forwards the tasks to a ground station for processing. However, new challenges emerge in the design of the computation offloading. On one hand, the available radio spectrum resources and rain attenuation of the different frequency bands need to be considered comprehensively under different rainfall intensities. Especially when the rainfall intensity at each device's location changes independently and dynamically, it is complicated to select a reasonable frequency band for each device to improve the overall transmission performance. On the other hand, the movements of LEO satellites affect the distance-related attenuation and change the available offloading paths for the IoRT devices. Therefore, the channel conditions and associated ground stations may change during the offloading of a single task, which makes the delay analysis intractable. Furthermore, due to the limited energy resources of satellites [19], the power consumption in satellite transmission needs to be considered during the computation offloading.

As is known, deep reinforcement learning (DRL) can be exploited to quickly solve task scheduling and resource management problems [21], [22]. Considering the dynamic network environment caused by the continuous satellite movements and the Markovian rainfall changes, we investigate how to develop a DRL solution to jointly manage the offloading path selection and resource allocation in the multi-layer Ka/Q-band satellite-terrestrial integrated networks. Specifically, the main contributions of this paper are summarized as follows:

- We propose a multi-layer Ka/Q-band satellite-terrestrial network to increase radio transmission resources and ensure communication robustness in different rainfall intensities for IoRT devices. Under this architecture, the dynamic computation offloading problem is studied;
- We formulate the computation offloading optimization problem to maximize the number of offloaded tasks with satisfied delay performance and minimize the power consumption of satellites. Then, we transform the formulated problem into a Markov decision process (MDP) to capture the continuous satellite movements and the Markovian rainfall changes;
- To effectively solve the transformed MDP problem with a large state space, a DRL algorithm based on a deep double Q network (D3QN) is proposed to make optimal decisions, including offloading path selection and resource allocation;
- Based on computer simulation results, the performance of our proposed D3QN-based computation offloading scheme significantly outperforms the benchmark schemes. We also find that the combined usage of Ka- and Q-bands improves the performance under different rainfall intensities.

The rest of this paper is organized as follows. In Section II, we introduce the recent related works. Afterward, the system model of the IoRT computation offloading in multi-layer Ka/Q-band satellite-terrestrial networks is elaborated in Section III. In Section IV, we formulate a joint optimization problem to solve the offloading path selection and resource allocation problem, and transform it into a MDP. We propose a D3QN-based algorithm in Sections V to solve the MDP problem. Simulation results are presented in Section VI, and finally, we conclude this work in Section VII.

II. Related Work

Computation offloading is effective in helping process tasks generated from energy- and computing-constrained IoT devices [23]. In an IoRT scenario, satellite communication is a promising way for task offloading and attracts significant attention from the industry and academia. By using the computing capabilities of LEO satellites, the energy-efficient mobile edge computing (MEC) for satellite-terrestrial IoT is studied in [24], in which tasks from IoT devices are offloaded to satellites for edge computing. Tang et al. present a computing framework considering LEO-MEC server and remote cloud server [25]. Based on the proposed framework, offloading decisions are jointly optimized to minimize the total energy consumption of ground users under the constraints of coverage time and computing capability of LEO satellites. In [26], Wang et al. consider a joint computation offloading and resource allocation problem in LEO satellite edge computing systems. Optimization algorithms based on the game theory and Lagrange multiplier method are proposed to solve the mixed-integer nonlinear programming problem. In [27], Cheng et al. study a joint computation offloading and resource allocation problem for space-air-ground integrated networks (SAGIN). Moreover, a DRL-based offloading method is proposed to allocate multi-dimensional SAGIN resources and learn dynamic network conditions.

Improving the satellite communication efficiency plays a vital role in ensuring the performance of computation offloading, and there are many related studies. In [16], an energy-efficient data collection problem considering time-varying uplinks in LEO satellite assisted IoT is proposed and solved by Lyapunov optimization theory. In [17], Ji et al. investigat a joint power and bandwidth allocation problem in a multi-user satellite downlink system based on the rain attenuation prediction. The influence of rainfall on the X-, Ka- and Q-bands in a downlink satellite system is discussed, and a beam controlling method is proposed to maximize site diversity gain [18]. In [19], a power allocation problem is investigated to extend the lifetime of the LEO satellite battery by sharing the workload of satellites, and a Q-learning approach is proposed to solve this problem. Most existing works focus on improving the wireless transmission

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3133110, IEEE Internet of Things Journal

efficiency in a single-layer LEO satellite network. To further enhance the capacity to meet the ever-increasing demand for data transmission, a multi-layer LEO network is a potential solution and starts to draw attention from researchers. For instance, with the expansion of the scale of LEO satellite constellation, terrestrial networks are integrated with ultra-dense LEO satellite networks [20]. Considering the inter-layer interference, a proper channel access mechanism is needed to achieve efficient data offloading. Futhermore, GEO satellites can be leveraged with a LEO satellite network to extend the communication coverage. A software-defined satellite-terrestrial network with control modules on GEO/MEO satellites is proposed, upon which a joint networking, computing, and caching resource orchestration problem is tackled using a deep Q-learning approach [28].

The existing proposals on improving the transmission efficiency in the multi-layer LEO satellite networks mainly focus on using the Ka-band. However, the combined usage of Ka- and Q-bands can potentially increase the available radio spectrum resources and ensure communication robustness in different rainfall intensities. Further studies of computation offloading on a multi-layer LEO satellite network is required for joint path selection and resource allocation, especially regarding the combined usage of different frequency bands and in presence of changing rainfall intensities.

III. System Model

In this section, a multi-layer Ka/Q-band satellite-terrestrial network is presented. Then we propose an IoRT computation offloading framework, including the offloading path selection, communication model, task queue model, and task delay counters. The definitions of main notations used in this paper are summarized in Table I.

TABLE I: LIST OF NOTATIONS

Symbol	Description
N	Total Number of IoRT devices
M	Total Number of ground stations
V	Total Number of satellites
x, y	Horizontal plane coordinates
H_v	Altitude of satellites
C_m	Computing capability of ground stations
$P_n, P_{v,m}$	Transmission power
$G_{n,v}, G_{v,m}$	Antenna gain
$f_{n,v}$	Communications center frequency
$L^f_{n,v}, L^f_{v,m}$	Free space path loss
$L^p_{n,v}, L^p_{v,m}$	Rain attenuation
$R_{n,v} R_{v,m}$	Data rate of uplink/downlink
$h_{n,v}, h_{v,m}$	Channel magnitude
N_u	Noise of the uplink
N_d	Noise of the downlink
Z	Data size of each sub-task
$D_{n,i}$	Delay of tasks
$\boldsymbol{L}_n$	Location of IoRT device n
$\boldsymbol{L}_m$	Location of ground station m
$\boldsymbol{L}_v$	Position of satellite v
Ω_n	Rain intensity of IoRT device n
Ω_m	Rain intensity of ground station m
Γ_n	Waiting counter of IoRT device n
U_n	Latest delay counter of IoRT device n
B_n	Queue of IoRT device n
B_m	Queue of ground station m
p_n	Offloading path of IoRT device n
$W_{n,v}$	Bandwidth allocated to IoRT device n
$W_{v,m}$	Bandwidth allocated to ground station m

A. Network Scenario

As shown in Fig. 1, we consider a multi-layer Ka/Q-band satellite-terrestrial network, where IoRT devices can offload their computation tasks to ground stations via LEO satellites. To enhance the backhaul capacity, a multi-layer LEO satellite network is considered to provide multiple satellites covering remote areas simultaneously, thereby ensuring seamless coverage and offering additional communication resources for IoRT devices. Meanwhile, both the Ka- and Q-band are utilized by the LEO satellite network to maintain the channel's robustness under rainfall dynamics.

IoRT devices can access the computing resources placed at the server in the ground stations through either Ka-band satellites or Q-band satellites. According to the current satellite configurations, we assume each satellite is equipped with a transparent repeater with no onboard storage and processing capacity. Therefore, each IoRT device uploads computation tasks to the LEO satellites, and then each satellite directly forwards the tasks to the selected ground station according to the wireless channel conditions and the computing capacity of ground stations. To achieve high utilization of resources and meet QoS requirements, a software defined network (SDN) and network function virtualization (NFV) enabled controller is installed at a GEO satellite to centrally and dynamically make the computation offloading decisions over its coverage area. By collecting information about the task queues, rainfall intensities, and satellite positions, the controller can adjust the offloading path and allocate resources to each IoRT device.

B. Offloading Path Selection Model

Without loss of generality, a three-dimensional (3D) Euclidean coordinate is adopted to describe the location of IoRT devices, satellites and ground stations. A set of IoRT devices are randomly distributed on the ground. Denote the set of IoRT devices as $\mathcal{N} = \{1, 2, \ldots, N\}$, and the set of ground stations as $\mathcal{M} = \{1, 2, \ldots, M\}$. The location of IoRT device $n \in \mathcal{N}$ is denoted as $\boldsymbol{L_n} = (x_n, y_n, 0)$, where x_n and y_n are the horizontal plane coordinates of IoRT device n. Computation tasks generated by IoRT devices are offloaded through satellite links to ground stations, instead of being processed locally due to device computing and energy limitations in remote areas. Therefore, tasks are continuously generated and enter task queues of IoRT devices, waiting to be offloaded to computing servers deployed at ground stations. Similar to IoRT devices, the location of ground station $m \in \mathcal{M}$ is denoted by $\boldsymbol{L_m} = (x_m, y_m, 0)$. The computing servers of ground stations receive the offloaded tasks and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3133110, IEEE Internet of Things Journal

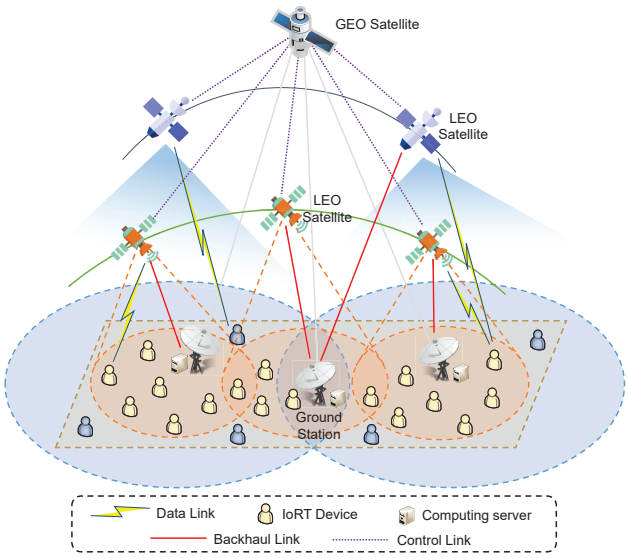


Fig. 1: An illustration of the multi-layer Ka/Q-band satellite-terrestrial network.

add them to their corresponding processing queues. The LEO satellite system consists of Ka-band satellites with an altitude of H_{Ka} km and Q-band satellites with an altitude of H_Q km (H_Q<H_{Ka}). At the same slot, we set V LEO satellites serving the IoRT devices. Let $\mathcal{V} = \{1, 2, \ldots, V\}$ denote the set of LEO satellites, where $\mathcal{V}_{Ka} = \{1, 2, \ldots, \psi\}$ ($\psi < V$) is the set of Ka-band satellites and $\mathcal{V}_Q = \{\psi + 1, \ldots, V\}$ is the set of Q-band satellites. Index ψ is used to distinguish these two kinds of satellites. The position of satellite $v \in \mathcal{V}$ is denoted by $\boldsymbol{L_v} = (x_v, y_v, H_v)$, where x_v and y_v are the horizontal plane coordinates of satellite v, and H_v is the altitude of satellite v. The system operates in a time-slotted fashion, and the time slot length and the time slot index are respectively denoted by τ and t. The locations of IoRT devices and ground stations are fixed, while the positions of satellites are fixed in each time slot for simplicity of analysis but vary among different time slots. Each satellite is assumed to be connected to multiple ground stations for the computation load balancing among ground stations.

In each time slot, a task offloading path is configured for each IoRT device, where satellite v for task transmission and ground station m for task processing are determined. The path of IoRT device n is denoted by $p_n = (v, m)$. The element $b_{n,v,m}(t) = 1$ indicates that IoRT device n is associated with satellite v and ground station m in time slot t, and $b_{n,v,m}(t) = 0$ otherwise. If $b_{n,v,m}(t) = 1$, the location relationship satisfies

$$\|\boldsymbol{L}_v(t) - \boldsymbol{L}_n(t)\|_2 \leqslant O_v \ \wedge \ \|\boldsymbol{L}_v(t) - \boldsymbol{L}_m(t)\|_2 \leqslant O_v, \quad \forall b_{n,v,m}(t) = 1 \tag{1}$$

where O_v is the coverage radius of satellite v. It should be noted that the current LEO satellite constellation guarantees seamless coverage of the ground. When a satellite v' moves to a position where it cannot cover any IoRT devices, $\mathcal{V}$ will remove satellite v' and join a newly arrived satellite in the same altitude to keep the number of available satellites unchanged.

C. Communication Model

1) Uplink Channel Model: The magnitude of uplink channel gain between IoRT device n and satellite v in time slot t is given by [29]

$$h_{n,v}(t) = \sqrt{\frac{G^N_{n,v} G^V_{n,v}}{L^f_{n,v}(t) L^p_{n,v}(t)}} \tag{2}$$

where $G^N_{n,v}$ and $G^V_{n,v}$ are the IoRT device transmitting antenna gain and the satellite receiving antenna gain, respectively, $L^f_{n,v}(t)$ is the free-space path loss, and $L^p_{n,v}(t)$ is the rain attenuation. The antenna gains can be calculated as follows [30]:

$$G^N_{n,v} = \phi \left(\frac{\pi f_{n,v} \varrho_n}{c} \right)^2 \text{ and } G^V_{n,v} = \phi \left(\frac{\pi f_{n,v} \varrho_v}{c} \right)^2 \tag{3}$$

where ϕ is the efficiency of antenna, ϱ is the diameter of antenna with a circular aperture or reflector, c is the speed of light and $f_{n,v}$ is the communications center frequency (in Hz) of uplinks.

The free-space path loss, $L^f_{n,v}(t)$, in time slot t is given by

$$L^f_{n,v}(t) = \left(\frac{4\pi \sigma_{n,v}(t)}{\lambda_{n,v}} \right)^2 \tag{4}$$

where $\lambda_{n,v} = c/f_{n,v}$, and $\sigma_{n,v}(t)$ is the slant range (in km) in time slot t.

The transmission attenuation at t due to rain, $L^p_{n,v}(t)$, is affected by carrier frequency, elevation angle, altitude above sea level, and rainfall intensity. It is expressed as [31]

$$L^p_{n,v}(t) = \sigma^e_{n,v}(t) \gamma_v(t) \tag{5}$$

where $\sigma^e_{n,v}(t)$ (in km) is the effective path length of the wave in rain in time slot t, and $\gamma_v(t)$(dB/km) is the attenuation per kilometer in time slot t. The value of $\gamma_v(t)$ depends on the frequency and rain intensity $\Omega_n(t)$ (mm/h), and is calculated as [32]

$$\gamma_v(t) = \rho_v \cdot \Omega_n(t)^{\eta_v} \tag{6}$$

where ρ_v and η_v are the frequency-dependent coefficients.

According to the Shannon formula, the achievable uplink data rate, $R_{n,v}(t)$, (in bps) from IoRT device n to satellite v in time slot t is given by

$$R_{n,v}(t) = W_{n,v}(t) \log_2 \left(1 + \frac{P_n(t)[h_{n,v}(t)]^2}{W_{n,v}(t) N_u} \right) \tag{7}$$

where $W_{n,v}(t)$ is the bandwidth that satellite v allocates to device n, $P_n(t)$ is the transmission power of device n, and N_u is received noise power at the satellite.

2) Downlink Channel Model: Similar to the uplink channel model, the magnitude of downlink channel gain between satellite v and ground station m in time slot t is

$$h_{v,m}(t) = \sqrt{\frac{G^V_{v,m} G^M_{v,m}}{L^f_{v,m}(t) L^p_{v,m}(t)}} \tag{8}$$

where $G^V_{v,m}$ and $G^M_{v,m}$ are the satellite transmitting antenna gain and the ground station receiving antenna gain, respectively, $L^f_{v,m}(t)$ is the free-space path loss and $L^p_{v,m}(t)$ is the

rain attenuation.

Due to the transparent relay by satellites, the total noise power of the downlink, $N_{v,m}$, includes the noise power of uplink passing through the transparent repeater, given by

$$N_{v,m} = N_u G^{on} [h_{v,m}(t)]^2 + N_d \tag{9}$$

where G^{on} is gain of the on-board transparent repeater, N_d is the transmission background noise power between satellite v and ground station m.

Thus, the achievable downlink data rate, $R_{v,m}(t)$ (in bps), in time slot t is given by

$$R_{v,m}(t) = W_{v,m}(t) \log_2 \left(1 + \frac{P_{v,m}(t)[h_{v,m}(t)]^2}{W_{v,m}(t) N_{v,m}}\right) \tag{10}$$

where $W_{v,m}(t)$ is the bandwidth that satellite v allocates to ground station m, $P_{v,m}(t)$ is the transmission power of satellite v to ground station m.

3) Rain Intensity: The variation of rain intensity over time is modeled as a Markov process. The probability density function of rain intensity events at a point describes how rain intensity $\Omega(t)$ evolves from time t_0 to t, $t > t_0$, and is characterized by parameters Ω_m, $\sigma_{\ln \Omega}$ and γ, given by [33]

$$\begin{aligned} p(\Omega(t)|\Omega(t_0)) = &\left(\frac{1}{2\pi\sigma^2_{\ln\Omega}(\Delta t)}\right)^{\frac{1}{2}} \\ &\times \exp\left[-\frac{(\ln\Omega(t) - \ln\Omega_m(\Delta t))^2}{2\sigma^2_{\ln\Omega}(\Delta t)}\right], \end{aligned} \tag{11}$$

where $\Delta t = t - t_0$ and $\Omega_m(\Delta t)$ is the mean rain rate observed at the location. In (11), the time-dependent mean, $\Omega_m(\Delta t)$, and standard deviation, $\sigma_{\ln\Omega}(\Delta t)$, are given by

$$\Omega_m(\Delta t) = \Omega_m^{(1-\exp(-\gamma\Delta t))} \Omega_0^{\exp(-\gamma\Delta t)}, \Omega_0 \equiv \Omega(t_0) \tag{12}$$

and

$$\sigma_{\ln\Omega}(\Delta t) = \sigma_{\ln\Omega}\sqrt{1 - \exp(-2\gamma\Delta t)}. \tag{13}$$

D. Task Queue Model

1) Task queue of IoRT devices: Due to the movement of satellites, the ground stations that the IoRT device can connect to may change. Because the correlation between task data affects the result of processing, when a task does not complete the transmission and cannot connect to the current chosen station in the next slot, the task must wait for the next time when it can connect to this ground station. This leads to longer waiting delays. Consequently, we assume that each task can be divided into η sub-tasks, which can be offloaded to different ground stations for processing. The task is completed when all sub-tasks are finished, and the size of each sub-task is denoted by Z in bits. After completing the offloading path selection and bandwidth allocation, the IoRT devices start to offload the computation tasks. At time slot t, the number of sub-tasks that IoRT device n is able to offload is $a_n(t) = \min(R_{n,v}(t)\tau/Z, B_n(t))$, where $B_n(t)$ is the number of tasks in the queue of device n. The transmission of tasks in the queue and the generation of new tasks jointly decide queue update, expressed as

$$B_n(t+1) = \min\{B_n(t) - a_n(t) + \eta Y(t), B^{max}\} \tag{14}$$

where $Y_n(t)$ is a Bernoulli random variable, indicating $Y(t) = 1$ when a new task is generated at device n in time slot t, and $Y(t) = 0$ otherwise. B^{max} is the maximum queue length.

Then, we calculate the number of transmitted tasks $I_n(t)$ of IoRT device n in time slot t. According to the queue length, three cases need to be considered: the queue is empty (case 1: $B_n(t) = 0$), the head-of-line task has not been partially transmitted (case 2: $B_n(t) \neq 0$ and $B_n(t) \bmod \eta = 0$) and the head-of-line task has been partially transmitted (case 3: $B_n(t) \neq 0$ and $B_n(t) \bmod \eta \neq 0$) at the beginning of the slot t. When the queue is empty, $I_n(t)$ is 0. For case 2, the number of transmitted tasks is $\lfloor a_n(t)/\eta \rfloor$. For case 3, $\lfloor (a_n(t) - B_n(t) \bmod \eta)/\eta \rfloor$ tasks are completely transmitted after the transmission of the head-of-line task, and $I_n(t)$ is 0 when the head-of-line task is not completely transmitted in this slot. For the above three cases, the $I_n(t)$ is given by

$$I_n(t) = \lfloor \frac{a_n(t) - B_n(t) \bmod \eta}{\eta} \rfloor + \lceil \frac{B_n(t) \bmod \eta}{\eta} \rceil \tag{15}$$

where 'mod' is the remainder operation, $\lfloor \cdot \rfloor$ is the floor function, and $\lceil \cdot \rceil$ is the ceiling function.

2) Task queue of ground stations: Each ground station is equipped with a buffer containing received tasks waiting to be processed, and it is assumed that the buffer size is infinite. At the beginning of time slot t, ground station $m(t)$ will receive a_m sub-tasks from the satellites, given by

$$a_m(t) = \sum_{n \in \mathcal{N}} \sum_{v \in \mathcal{V}} a_n(t) b_{n,v,m}(t), \forall m \in \mathcal{M}. \tag{16}$$

The buffer occupancy of ground station m at slot $t+1$ is calculated as

$$B_m(t+1) = \max\{B_m(t) + a_m(t) - C_m, 0\} \tag{17}$$

where C_m is the computing capacity of ground station m, which represents the number of sub-tasks can be handled by the server of the ground station m at each slot.

E. Delay Counter of IoRT devices

Computation offloading is delay-sensitive. As mentioned, using the computation offloading technology, the total delay of each task depends on the delay of sub-tasks. For each sub-task, the total delay is composed of the waiting delay in the IoRT device's queue, the transmission delay, and the processing delay in the ground station. However, the stochastic task arrivals and dynamic variations of wireless channel capacity pose technical challenges in per-task delay modeling. Fortunately, the correlation between two consecutive tasks in the same device provides a way to determine the delays more effectively [34]. Suppose that two counters are created for the head-of-line task of device n : waiting counter Γ_n to present the number of slots between task arrival time slot and the current slot, and latest delay counter U_n to present the delay of its sub-task that returns the result at the latest. Since the size of computation results is usually much smaller than the size of input task data, the return transmission delay can be negligible.

1) Waiting counter: The waiting counter, Γ_n, records how long the head-of-line task has been waiting. The correlation

between two consecutive tasks is used to estimate the waiting time of all tasks in the queue [34]. We suppose that task k of IoRT device n arriving at T_k^A, is ready to be transmitted at T_k^R, and all its sub-tasks are transmitted at T_k^E. The waiting delay of this task is $d_k = T_k^E - T_k^A$.

Each task queue conforms to the first-in-first-out (FIFO) rule. For consecutive tasks k and $k+1$ in the current queue, task $k+1$ starts to be transmitted only after task k is transmitted. The arrival epoch of task $k+1$ is $T_{k+1}^A = T_k^A + \Delta_{k,k+1}$, where $\Delta_{k,k+1}$ is the interval between the arrival time slots of the two tasks. The epoch when task k is completely transmitted is the time that task $k+1$ is ready to be transmitted, i.e., $T_{k+1}^S = T_k^E$. Hence, we have

$$\begin{aligned} d_{k+1} &= T_{k+1}^E - T_{k+1}^A \\ &= T_{k+1}^E - T_{k+1}^R + T_{k+1}^R - T_{k+1}^A \\ &= T_{k+1}^E - T_{k+1}^R + T_k^E - (T_k^A + \Delta_{k,k+1}) \\ &= T_{k+1}^E - T_{k+1}^R + D_k - \Delta_{k,k+1}. \end{aligned} \tag{18}$$

Task arrivals at each device is modeled as a stationary Bernoulli stochastic process, with the expectation of interval between two consecutive tasks $\mathbb{E}(\Delta_{k,k+1}) = \overline{\Delta}$. Consequently, the expectation of delay for task $k+1$ under T_{k+1}^R and T_{k+1}^E is

$$\mathbb{E}(d_{k+1}|T_{k+1}^R, T_{k+1}^E) = T_{k+1}^E - T_{k+1}^R + \mathbb{E}(d_k|T_k^R, T_k^E) - \overline{\Delta}. \tag{19}$$

The expected waiting time before a task is transmitted depends on the expected delay of the previous task $\mathbb{E}(d_k|T_k^R, T_k^E)$ and interval expectation $\overline{\Delta}$. Therefore, we set one waiting counter Γ_n for each IoRT device to record the expected delay of the current head-of-line task conditioned on the transmission epochs of previously tasks. Once the head-of-line task is completely transmitted, the waiting counter is reset to $\mathbb{E}(d_k|T_k^R, T_k^E) - \overline{\Delta}$ for the next task. When there are still tasks completely transmitted except for the head-of-line task in time slot t, for task $k_0 \in [k+1, k+I_n(t)-1]$, all the sub-tasks are transmitted in one time slot, i.e., $T_{k_0}^E - T_{k_0}^R = 0$. Therefore, for the head-of-line task $k+I_n(t)$ in the next time slot, the waiting counter is reset to $\mathbb{E}(d_k|T_k^R, T_k^E) - I_n(t)\overline{\Delta}$. If no task is completely transmitted in this slot, the waiting counter increases by 1.

The update of the waiting counter is given by

$$\Gamma_n(t+1) = \begin{cases} \Gamma_n(t) + 1, & I_n(t) = 0 \\ \Gamma_n(t) - I_n(t)\overline{\Delta}, & \text{otherwise} \end{cases} \tag{20}$$

where the first subequation indicates that the delay is increased by one for each slot time when the task is being transmitted; the second subequation indicates that a new task of IoRT device n is scheduled at the end of time slot t and its delay counter is initialized as $\Gamma_n(t) - I_n(t)\overline{\Delta}$. The waiting counter may be negative due to the estimated error between the true sampling interval of two consecutive tasks and their expectations. However, this potential negative value can be averaged out in a long run [34].

2) Latest delay counter: Waiting counter Γ_n represents the waiting time of the head-of-line task in the queue. When calculating the overall delay of an offloaded task, it is also necessary to know the processing delay in the ground station. At time slot t, the average processing time of the sub-tasks offloaded to the ground station m is

$$D_m(t) = \frac{B_m(t) + (1+a_m)/2}{C_m}. \tag{21}$$

The arrival of a task means that its sub-tasks arrive at the same time, thus the waiting counter of sub-tasks is equal to that of the task they belong to. For the sub-task transmitted to ground station m at time slot t, the total delay of the sub-task is $d_{sub} = \Gamma_n(t) + D_m(t)$.

Because the delay of each task depends on the delay of the sub-task that returns the result at the latest, we use the latest delay counter $U_n(t)$ to represent the maximum delay of the sub-tasks from the head-of-line task. Once the head-of-line task is completely transmitted, $U_n(t)$ is the delay of the head-of-line task from its arrival to the processing completion, and is reset to 0 for the next task. The update of the latest delay counter $U_n(t)$ is given by

$$U_n(t+1) = \begin{cases} \max(U_n(t), \Gamma_n(t) + D_m(t)), I_n(t) = 0 \\ 0, \text{otherwise} \end{cases} \tag{22}$$

where the first subequation indicates the maximum delay of the computed sub-task when the task is being transmitted; the second subequation indicates that, when no task is partially transmitted at the end of time slot t, the delay of the head-of-line task is reset to 0.

When $I_n(t) = 0$, no task is completely transmitted in time slot t, so the delay $D_{n,i}(t)$ will not be calculated and recorded. When $I_n(t)$ (>0) tasks of device n are completely offloaded to the ground station and processed, the total delay of task $i \in \mathcal{I}_n(t) = \{0, 1, \ldots, I_n(t)-1\}$ is

$$D_{n,i}(t) = \begin{cases} \max(U_n(t), \Gamma_n(t) + D_m(t)), i = 0 \\ \Gamma_n(t) - i\overline{\Delta} + D_m(t), \text{otherwise} \end{cases} \tag{23}$$

where the first subequation represents the delay of the head-of-line task; the second subequation represents the delay of other tasks that are completely offloaded in this time slot.

IV. Problem Formulation and Transformation

A. Problem Statement

We formulate our IoRT computation offloading problem in the multi-layer Ka/Q-band satellite-terrestrial network. The objective is to optimize the obtained average reward of whole computation offloading systems. Through observation, each IoRT device or ground station estimates the rain intensity at its location. Meanwhile, each device reports the states of its task queue and delay counters to the controller, and each ground station reports its task queue state. Based on the global state information, which includes task generation states of IoRT devices, communication channel states, and computing queue states of ground stations, the controller makes the offloading path selection and resource allocation decisions. In other words, the controller determines which satellite and ground station that the device should connect to and allocates the bandwidth, which decides how many tasks each IoRT device should offload to the ground station. Given

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3133110, IEEE Internet of Things Journal

the offloading decision, the computation tasks of IoRT devices are executed by the ground stations with computing capability, and immediate reward is obtained.

In the following, we formulate the computation offloading problem as an optimization problem. Then, to capture the network dynamics and the relationship between state and policy, we further describe the formulated optimization problem as an MDP formulation, where decisions are used to formalize the sequential offloading decision making in the considered network environment.

B. Problem Formulation

We aim to maximize the number of offloaded tasks while meeting delay requirements and to minimize the power consumption of the LEO satellites simultaneously.

$$\text{Maximize:} \sum_{t\in T}\sum_{n\in\mathcal{N}}\sum_{v\in\mathcal{V}}\sum_{m\in\mathcal{M}}\left(F_n(t)-\alpha P_{v,m}(t)\right) \tag{24}$$

$$\text{Subject to:} \quad \sum_{v\in\mathcal{V}}\sum_{m\in M} b_{n,v,m}(t)\leqslant 1, \forall n\in\mathcal{N} \tag{25}$$

$$\sum_{n\in\mathcal{N}} R_{n,v}(t)b_{n,v,m}(t)\leqslant R_{v,m}(t), \forall v\in\mathcal{V}, m\in\mathcal{M} \tag{26}$$

$$\sum_{n\in\mathcal{N}}\sum_{m\in\mathcal{M}} W_{n,v}(t)b_{n,v,m}(t)\leqslant W_v^*, \forall v\in\mathcal{V} \tag{27}$$

$$\sum_{n\in\mathcal{N}}\sum_{m\in\mathcal{M}} W_{v,m}(t)b_{n,v,m}(t)\leqslant W_v^*, \forall v\in\mathcal{V} \tag{28}$$

$$\sum_{n\in\mathcal{N}}\sum_{m\in\mathcal{M}} P_{v,m}(t)b_{n,v,m}(t)\leqslant P_v^*, \forall v\in\mathcal{V} \tag{29}$$

$$b_{n,v,m}(t)\in\{0,1\}, \forall n\in\mathcal{N}, v\in\mathcal{V}, m\in\mathcal{M} \tag{30}$$

$$\|\boldsymbol{L}_v(t)-\boldsymbol{L}_n\|_2\leqslant O_v \wedge \|\boldsymbol{L}_v(t)-\boldsymbol{L}_m\|_2\leqslant O_v, \quad \forall b_{n,v,m}(t)=1 \tag{31}$$

where $F_n(t)$ is the number of offloaded tasks with satisfied delay, α is a weight parameter to balance the trade-off between the power consumption and delay of the task completion. Constraint (25) represents that one device can connect only one satellite at time slot t. At each time slot, the amount of all uplink data and the amount of all downlink data at one satellite must satisfy the flow balance constraints in (26). Eqs. (27) and (28) are bandwidth constraints, (29) is power constraint of satellites, and (31) represents geographical constraints between satellite and an IoRT device (or ground station).

C. Problem Transformation with Markov-based Model

To capture the continuous satellite movements and the Markovian rainfall changes, we leverage a Markov decision process (MDP) framework to model the complex interactions among queueing, transmission, and computing processes in the IoRT system. In the MDP formulation, decisions are used to formalize the sequential decision making in a static or dynamic environment [35]. In addition, the environment is entirely observable, that is, any observation made at any time is sufficient for making the optimal decision. Basically, an MDP is defined by a tuple of state space S, decision space A, state transition probability function $P := S\times A\times S\to P$, and reward function $R := S\times A\to R$. We leverage the MDP method to formulate the IoRT computation offloading problem in the multi-layer Ka/Q-band satellite-terrestrial network.

1) Environment State: As discussed, each IoRT device, satellite and ground station periodically send the collected information to the controller in GEO satellite. By collecting such information, the agent (i.e., the controller) can obtain the environment state. The composite environment state $s(t)\in S$ at time slot t, is described as

$$s(t)=\big[\Omega_n(t),\Omega_m(t),B_n(t),\Gamma_n(t), U_n(t),B_m(t),\boldsymbol{L}_v(t)\big], \forall n\in\mathcal{N},\forall m\in\mathcal{M},\forall v\in\mathcal{V} \tag{32}$$

The environment state includes rain intensity of IoRT devices, $\Omega_n(t)$, rain intensity of ground stations, $\Omega_m(t)$, task queue length of IoRT devices, $B_n(t)$, task queue length of ground stations, $B_m(t)$, waiting delay counter of IoRT devices, $\Gamma_n(t)$, latest delay counter of IoRT devices, $U_n(t)$, and position of satellites, $\boldsymbol{L}_v(t)$.

2) Action: Based on the observed environment state $s(t)\in S$, the controller makes computation offloading decisions according to policy π. Denote the action space as A. Then, action $a(t)\in A$ taken by the IoRT devices and LEO satellites at time slot t is given by

$$a(t)=\big[p_n(t),W_{n,v}(t),W_{v,m}(t)\big], \quad \forall m\in\mathcal{M},\forall n\in\mathcal{N},\forall v\in\mathcal{V}. \tag{33}$$

The action includes offloading path selection for IoRT devices, $p_n(t)$, bandwidth allocated to IoRT devices, $W_{n,v}(t)$, and bandwidth allocated to ground stations, $W_{v,m}(t)$.

3) Reward: The controller takes action $a(t)$ based on observed environment state $s(t)$, and then the environment returns an immediate reward, $r(t)$, to the controller. Policy π includes the offloading path selection and the resource allocation. Based on the received reward, the controller updates π until the learning algorithm converges in the learning stage. As indicated in (24), the delay requirement and power consumption should be simultaneously satisfied to guarantee the performance of computation offloading. Thus, to maximize the number of offloaded tasks with satisfied delay performance and to minimize the power consumption of the LEO satellites at time slot t, we define the following two reward elements

$$r_d(t)=\sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{I}_n(t)}\log_2\left(\frac{D_{max}+\beta}{D_{n,i}(t-1)+\beta}\right) \tag{34}$$

$$r_p(t)=\sum_{v\in\mathcal{V}}\sum_{m\in\mathcal{M}}\log_2\left(\frac{P_{max}+\mu}{P_{v,m}(t-1)+\mu}\right) \tag{35}$$

where $r_d(t)\geq 0$ when the delay requirement is satisfied, and $r_p(t)\geq 0$ when the power constraint is satisfied, otherwise negative $r_d(t)$ and $r_p(t)$ are obtained. The total reward is

$$r(t)=r_d(t)+\alpha r_p(t). \tag{36}$$

The main goal of the controller is to make the optimal offloading decision to maximize its average reward in the satellite system according to the rain intensity state, task queue state, delay counter state, and satellite position state. Accordingly, the objective of MDP formulation is to maximize

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3133110, IEEE Internet of Things Journal

the expected sum of rewards

$$\mathcal{R}(\pi) = \lim \sup_{T\to\infty} \frac{1}{T}\mathbb{E}_\pi \left[\sum_{t=0}^{T-1} r(t)\right] \tag{37}$$

where $\mathbb{E}[\cdot]$ is the expected reward of the controller in the long run.

V. DRL-Based Solution

In this section, a DRL based algorithm is designed to solve the formulated MDP problem. First, the basic framework of the DRL is introduced. Then, the details of leveraging the D3QN approach to solve the proposed optimization problem are presented.

A. Algorithm Design

In the preceding MDP formulation, the controller learns policy $\pi : S \to A$ which maximizes the long-term reward $\mathcal{R}$. A common method to obtain approximate solutions to problems with continuous states and actions is to quantify the state and action space and then use finite state dynamic programming (DP) techniques. In our stochastic network environment, the formulated MDP problem has a large dimensional space, while DP has the curse of dimensionality. Therefore, in view of the unknown dynamic satellite network environment, we adopt the model-free DRL learning optimal decision approach [36], [37] to solve the problem.

In an unknown stochastic environment, by using RL method, the optimal policy, π^*, is obtained to maximize the long-term reward for the controller. For policy $\pi : s(t) \to a(t)$, the value of state $s(t)$ can be expressed as

$$V(s(t),\pi) = \mathbb{E}_{a(t)\sim\pi(s(t))}\left[Q(s(t),a(t),\pi)\right] \tag{38}$$

where $(s(t), a(t))$ is the state-action value function under policy π and is defined as

$$Q(s(t),a(t),\pi) = \mathbb{E}_\pi\left[r(t)|s(t),a(t)\right]. \tag{39}$$

Based on the Bellman equation, the Q-value function can be computed recursively by

$$\begin{aligned} &Q(s(t),a(t),\pi) \\ &= \mathbb{E}_{s(t+1)}\left[r(t) + Q(s(t+1),a(t+1),\pi)|s(t),a(t)\right]. \end{aligned} \tag{40}$$

The optimal policy, π^*, corresponding to the policy of action selection gives maximum $Q^*(s(t), a(t))$ for state $s(t)$: $\pi^*(s(t)) = \arg\max_{a(t)} Q^*(s(t), a(t))$. Therefore, the optimal policy, π^*, can be obtained from the following optimal Bellman equation:

$$\begin{aligned} &Q^*(s(t),a(t)) \\ &= \mathbb{E}_{s(t+1)}\left[r(t) + \gamma \max Q^*(s(t+1),a(t+1))|s(t),a(t)\right]. \end{aligned} \tag{41}$$

To solve the optimal Bellman equation, Q-learning is a widely used traditional reinforcement learning (RL) method. However, Q-learning has the curse of dimensionality and cannot solve complex problems with a large multi-dimensional discrete or continuous state-action space. By combining reinforcement learning with deep neural networks (DNNs), DQN is proposed to overcome the curse of dimensionality. In DQN, a parameterized version of the Q-value function is used for approximation $Q(s(t), a(t); \theta) \approx Q^*(s(t), a(t))$.

The DQN utilizes a target network alongside an online network to stabilize the overall network performance. The two neural networks are the same, except that the target network with parameters θ^- is copied from the online network with parameters θ every τ^- step. Therefore, parameters of the online network are updated at each time step, and parameters of the target network are updated in every τ^- steps from the online network and then remain unchanged in all other steps.

In one-step learning, parameters θ is updated by iteratively minimizing a sequence of loss functions. The ith loss function is defined as

$$L_i(\theta_i) = \mathbb{E}_{s(t),a(t),r(t),s(t+1)}\left[\left(y_i^{DQN} - Q\left(s(t),a(t);\theta_i\right)\right)^2\right] \tag{42}$$

where y_i^{DQN} is the target of Q-value function and is defined as

$$y_i^{DQN} = r(t) + \gamma \max_{a(t+1)\in\mathcal{A}} Q\left(s(t+1),a(t+1);\theta_i^-\right). \tag{43}$$

To stabilize the learning, experience tuples $[s(t), a(t), r(t), s(t+1)]$ are collected and stored in a replay memory. During the training process, the experience of mini-batch is randomly sampled and input into the network.

Although DQN can speed up the convergence of the learning process, it is easily overestimated because the same values are used to select and evaluate actions. Therefore, by decoupling the selection and evaluation of the target Q value, the Double DQN (DDQN) algorithm is further proposed to alleviate this problem. The DDQN is realized by replacing the target y_i^{DQN} by the following target y_i^{DDQN}:

$$\begin{aligned} y_i^{DDQN}(\theta_i) &= r(t) \\ &+ Q\left(s(t+1), \arg\max_{a(t+1)\in\mathcal{A}} Q(s(t+1),a(t+1);\theta_i);\theta_i^-\right). \end{aligned} \tag{44}$$

Two DQN networks, θ and θ^-, are learned in DDQN. For each update, The DDQN uses one of the DQN networks to determine the policy, and uses the other to determine its value.

Furthermore, to achieve better strategy evaluation, we introduce a dueling structure based on the DDQN algorithm in our work. In the original DQN, we only update the Q value observed during training. This leads to a slower learning rate because we do not yet learn the Q value of the action that has not been taken. The duel structure separates the state value and the action advantage value into two streams, one stream outputs scalar state value $V(s(t))$, and the other stream outputs advantage vector $A(s(t), a(t))$ whose dimension is equal to the number of actions. Then, by combining $V(s(t))$ and $A(s(t), a(t))$, the Q-value function $Q(s(t), a(t))$ can be estimated as follows:

$$\begin{aligned} Q(s(t),a(t)) &= V(s(t)) \\ &+ \left(A(s(t),a(t)) - \frac{1}{|A|}\sum_{a(t)\in\mathcal{A}} A(s(t),a(t+1))\right) \end{aligned} \tag{45}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3133110, IEEE Internet of Things Journal

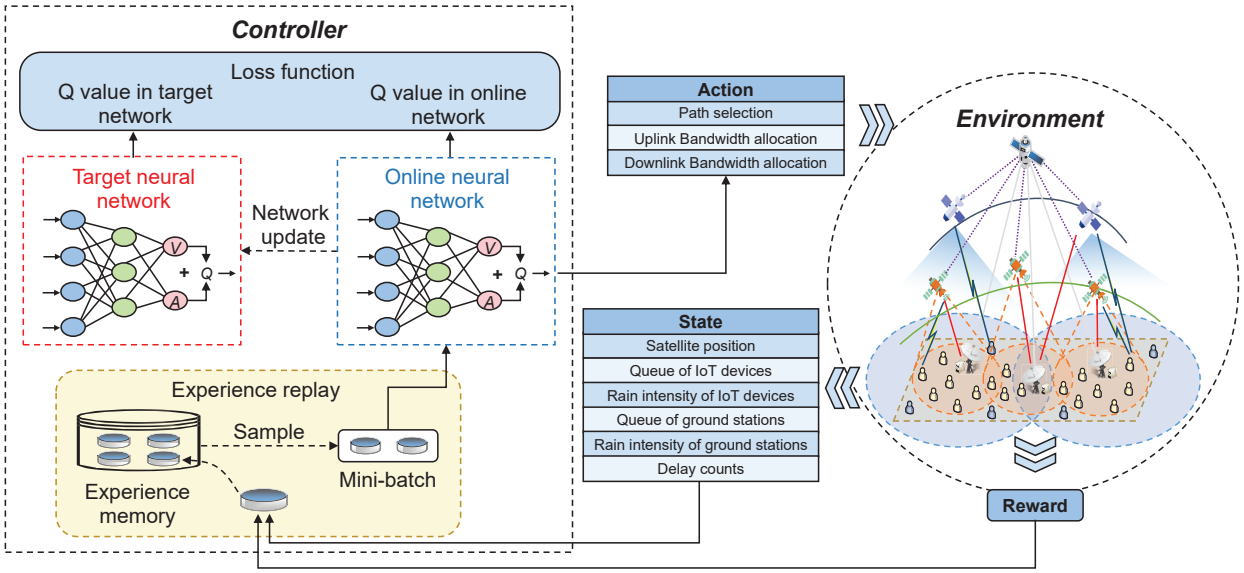


Fig. 2: Reinforcement learning with D3QN strategy.

where $|A|$ is the size of action space. Dueling DDQN (D3QN) subtracts the average of the advantage value to help the network separate the advantage value and the state value, thereby effectively reducing the overestimation and improving the performance of DDQN.

In this study, we develop a DRL offloading algorithm based on D3QN to solve the proposed MDP formulation presented in Section IV. In the D3QN-based strategy, by combining the double DQN and the dueling structure, the approximating function can be trained, and the Q value of each action at each state can be calculated. The optimal offloading policy can then be obtained by selecting the action with the maximal Q value. The procedure of D3QN-based strategy is shown in Fig. 2. Specifically, the controller acts as the decision agent and aims to make the optimal decision to maximize the average reward of the considered system while meeting the delay requirements of IoRT devices. The decision includes the offloading path selection decision and the resource allocation decision. The reward consists of power consumption and the delay requirement.

DRL algorithms usually have very high time complexity and space complexity [38]. Therefore, it is impractical to conduct neural network training directly on the GEO satellite with limited computing resources. To solve this challenge, the proposed D3QN-based offloading and resource allocation algorithm can be placed on the ground station connected to the GEO satellite for offline training [39]. The detailed procedure of the proposed D3QN-based algorithm is presented in Algorithm 1. Specifically, transition samples are first collected by the GEO satellite from the running system. Then GEO satellite transmits these samples to the ground station, and the ground station stores them into a replay buffer for learning. The offline training procedure starts with randomly initializing D3QN and copying the weights of D3QN to its target network. For each training episode, the global network state is initialized. The controller collects the global state information, consisting of positions of the satellites, the rain intensity and data queue state of IoRT devices, the rain intensity and data queue state of ground stations, and the delay counter state. Each episode includes T time slots (steps). In each time slot, an action (including path selection action and bandwidth allocation action) is selected to provide offloading services for each IoRT device based on the current global state information. During the learning process, the ϵ-greedy policy is adopted to select actions from the estimated Q-value, $Q(s(t), a(t))$. In ϵ-greedy policy, the controller has a probability of $1-\epsilon$ to choose best action a^* and a probability of ϵ to randomly select the action. After selecting the action, immediate reward $r(t)$ is obtained, and the next state $s(t+1)$ can be observed. Then, the current state and action along with the next state and obtained reward are stored into the replay memory as a four-tuple $[s(t), a(t), r(t), s(t+1)]$. The D3QN is trained by randomly sampling a mini-batch from the replay memory. The weight parameter for training D3QN is updated using (44). The target network of D3QN is updated slowly, with a small target update rate, τ^-, as the control parameter. τ^- is the number of steps in each update cycle. Each episode ends when T time slots are reached. The total episode reward is the accumulation of immediate rewards of all time slots within the episode. Through iterative training, the D3QN agent eventually learns to obtain high rewards by making optimal offloading decisions. After the offline training, the ground station directly applies the proposed algorithm to the controller in the GEO satellite, seamlessly monitoring the system state, making timely decisions, executing actions, and further interacting with the system.

B. Complexity and Convergence Analysis

We first provide the complexity analysis of the proposed D3QN algorithm. Denote ω as the total number of training iterations. The time complexity of the proposed Algorithm 1 is $\mathcal{O}(\omega)$. For each training iteration, the computation efficiency

Algorithm 1: Dueling Double DQN based Offloading Algorithm for IoRT Satellite Networks

Input: replay memory size $\mathcal{D}_r$, minibatch size $\mathcal{D}_b$, greedy ϵ, pre-train episodes $episode$, pre-train steps $step$, learning rate ς and discount factor γ.

Initialize replay memory $\mathcal{D}$;
▷ Data Collection
The controller deployed on the GEO satellite collects desirable information from the running system and transmits the information to the ground satation;
Pre-process and store the transition samples into the replay buffer $\mathcal{D}$;
▷ Offline Training
The ground station initializes parameters θ with random values ;
Update the target network θ^- with $\theta^- = \theta$;
for $episode = 1, 2, \ldots$ **do**
    Initialize enviroment and receive initial state $s(0)$;
    for $step = 1, 2, \ldots, T$ **do**
        Select action $a(t)$ with: $a(t) = \epsilon - greedy(Q(s(t), a(t); \theta))$;
        Obtain the immediate reward $r(t)$ according to Eq. (36) ;
        Observe the next state $s(t+1)$;
        Store the tuple $< s(t), a(t), r(t)(s(t), a(t)), s(t+1) >$ into reply memory $\mathcal{D}$;
        Randomly sample a mini-batch of tuples from $\mathcal{D}$;
        Update the Q-network weights with θ by minibatch gradient descent according to Eq. (44);
        Replace target parameters $\theta^- = \theta$ every τ^- steps;
    Update ϵ with $\epsilon = \max(\epsilon_{\min}, \epsilon * decay)$;
▷ Online Agent Plug
The ground station plugs the trained D3QN into the controller in GEO satellite;
while *Observed state s from the system* **do**
    Select the action a with the highest Q value: $a = \arg\max_{a \in \mathcal{A}} Q(s, a; \theta)$;
    The controller sends the offloading command to LEO satellites;
    LEO satellites execute the action in the system;

Output: $Q(\cdot)$with weight θ

of the proposed algorithm can be calculated based on the complexity for training the adopted neural network parameters. In calculating neural network parameters, the agent (i.e., the controller) utilizes convolutional neural networks (CNN) to generate an action. Then, denote J as the number of convolutional layers, n_e as the size of feature map, n_k as the size of kernel, and k as the number of filters. According to [40], the time complexity for a convolution layer is diven by $T_c = \mathcal{O}\left(\sum_{j=1}^{J} (n_e)^2 (n_k)^2) k_j k_{j-1}\right)$. Moreover, denoting c_j as the number of neural units in fully-connected layer c, the time complexity for a fully-connected layer is $T_f = \mathcal{O}\left(\sum_{j=1}^{J} c_j c_{j-1}\right)$. Therefore, the total time complexity of Algorithm 1 is $\mathcal{O}\left(\omega(T_c + T_f)\right)$.

DRL algorithms rely heavily on hyperparameters, such as learning rate, discount rate, and the hidden layer structure. Therefore, it is challenging to use analytical methods to verify the convergence of the proposed D3QN method. Currently, most of the literature verifies the optimal configuration of hyperparameters for a specific problem via trial and error. Based on the experimental results, the optimality and convergence are then further analyzed and proved. Therefore, similar to [41], in this study, we limit the convergence analysis by providing simulation results in Section VI. Through an appropriate hyperparameter setting, convergence can be achieved in the proposed D3QN algorithm.

VI. Numerical Results and Discussion

In this section, extensive simulation results are presented to evaluate the performance of our proposed D3QN-based IoRT computation offloading scheme in the multi-layer Ka/Q-band satellite-terrestrial network.

A. Simulation Framework

Our simulations are conducted on a satellite assisted-IoRT scenario which consists of eight IoRT devices, two ground stations, and four LEO satellites in two layers. In the simulations, each time slot lasts 4 s.

The IoRT devices are distributed in an area of $(700\ km, 100\ km)$ and the two ground stations are distributed in this area to serve the devices. For satellite parameter settings, we use the Satellite Tool Kit and set the height of Ka-band satellites at 700 km and Q-band satellites at 350 km [14], the minimum elevation angle at $40°$. Due to the constraint of the elevation angle, handover occurs when a satellite moves out of the service range over time, and we set up four satellites in service over this area at each slot. To ensure at least one ground station for each satellite to serve the IoRT devices, we set the distance between the Q-band satellites as 380 km and that between Ka-band satellites as 860 km. The uplink/downlink bandwidth of a Ka-band satellite is 1 MHz, and that of a Q-band satellite is 1.5 MHz. The noise density for space-terrestrial communications is -174 dBm/Hz. The rain attenuation coefficients ρ and η of Ka-band satellites are 0.2588 and 0.9392, and those of Q-band are 0.6600 and 0.8084, respectively [32]. The vertical equivalent path length of rain is 4.5 km [30], [42]. The efficiency and diameter of satellite antenna are 0.6 and 2 m, respectively [30]. If not specified, the defaulted task arrival rate at each device is set at $1/slot$, the data size of each task is 3000 KB, and each task can be divided into 3 sub-tasks of the same size. The queue length of each device is 6, which means that up to 18 sub-tasks are in the queue simultaneously. The maximum delay of each task is 7 slots. The transmit power of each IoRT device is set as 10 W, and the efficiency and diameter of the IoRT device antenna are 0.55 and 1 m, respectively. The defaulted computing capacity of each ground station is 12 tasks/τ. The efficiency and diameter of the ground station antenna are 0.6 and 4 m, respectively. We set 4 levels of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3133110, IEEE Internet of Things Journal

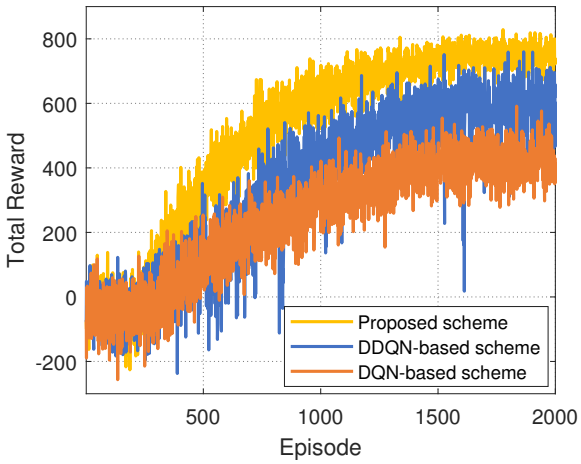


Fig. 3: The convergence performance of the proposed algorithm.

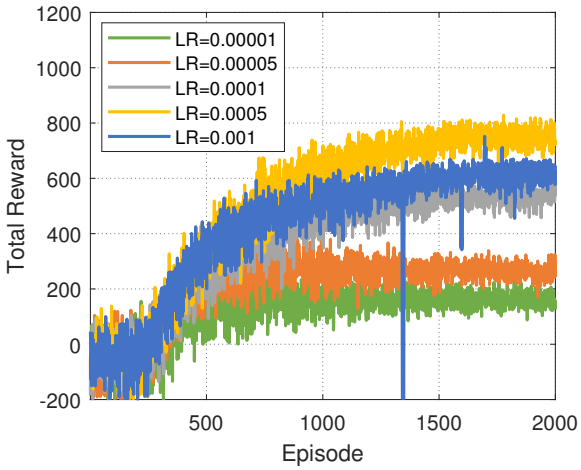


Fig. 4: Training process with different learning rate settings.

rain intensity $[0, 6, 12, 18]$ mm/h, and the default probability of being sunny is 0.74.

The parameter settings for the experiment are set with reference to [41]. For the proposed D3QN-based offloading algorithm, the model is trained in iterations. The D3QN consists of an input layer (50 neurons), three hidden layers (64, 32 and 32 neurons), and an output layer ($|A|$ neuron). Here, $|A|$ denotes the number of possible actions of the controller. The development environment of D3QN is built on the Tensorflow framework [43]. The size of memory replay is 2×10^4, and the mini-batch sample size is 128. The value of ϵ is initialized to 1 and decays with a rate of 0.95 per epoch until it reaches 0.1. The discount factor is set to 0.85, and the learning rate is set to 0.0005. Rectified linear unit (ReLU) function is adopted as the activation function of all hidden layers, and Adam optimizer is adopted in the DNN training to minimize the mean square error (MSE) loss. The performance results are collected under the computation offloading policy after 2000 learning episodes, where each episode has 100 time slots. The reward is accumulated in an episode. To ensure estimation accuracy of the performance results, all numerical results are obtained through multiple experiments.

In order to demonstrate the performance of our scheme, we use the following benchmark algorithms:

- *DQN-based scheme* - At each time slot, the GEO satellite learns the offloading decision based on the DQN method to maximize its long-term reward;
- *DDQN-based scheme* - At each time slot, the GEO satellite learns the offloading decision based on the DDQN method to to maximize its long-term reward.

To further explore the advantages of the combined usage of Ka- and Q-bands, we consider the following alternative satellite band setting scheme:

- *Q/Ka scheme* - The LEO satellite system consists of Q-band satellites in 700 km and Ka-band satellites in 350 km. The bandwidth of each Q-band satellite and Ka-band satellite is 1.5 MHz and 1 MHz, respectively;
- *Q/Q scheme* - In the LEO satellite system, both layers of satellites transmit data in Q-band. Because of more available resources in the Q-band, the bandwidth setting of the Q/Q scheme is the same as that of our proposed architecture. The bandwidth of each Q-band satellite is 1.5 MHz at 350 km and 1 MHz at 700 km;
- *Ka/Ka scheme* - In the LEO satellite system, both layers of satellites transmit data in Ka-band. Due to the limited resources of Ka-band, the satellites in different layers share a bandwidth of 1 MHz. The bandwidth of each Ka-band satellite is 0.5 MHz at 350 km and 0.5 MHz at 700 km.

To evaluate the effectiveness of the D3QN based scheduling scheme, we evaluate the performance of our proposed algorithm and the benchmark schemes using the following metrics:

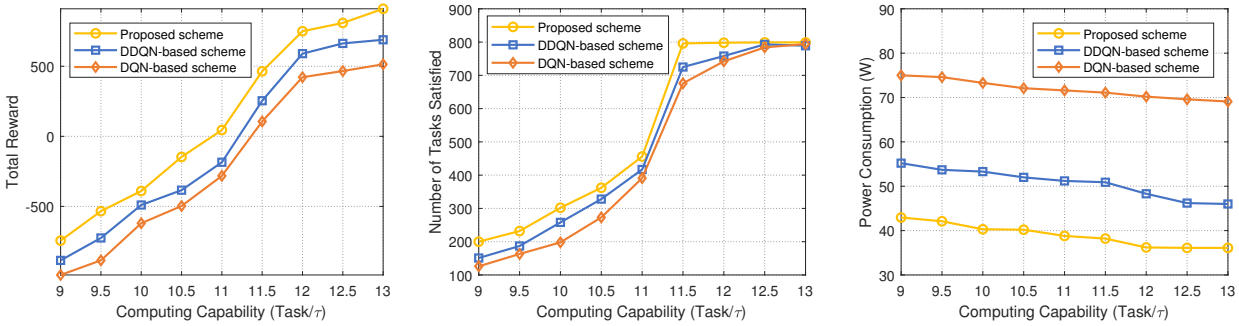
- *The total reward*, which is the sum of the system reward obtained by computation offloading during the simulation time;
- *The number of offloaded tasks with satisfied delay constraint*, which is the number of tasks that meet the delay requirements and are successfully offloaded to the ground stations;
- *The power consumption*, referring to the power consumed by all LEO satellites when transmitting tasks to ground stations.

B. Performance Analysis

Based on the parameter setting, we carry out simulations to evaluate the D3QN-based computation offloading algorithm of the proposed satellite-assisted IoRT. The simulations focus on the performance in two aspects. We first test the performance of the learning stage to learning the D3QN, DDQN and DQN-based model in terms of the average reward of all the IoRT devices. Then, the learning models are tested under different available resources and different environmental conditions to measure the performance of the proposed scheme.

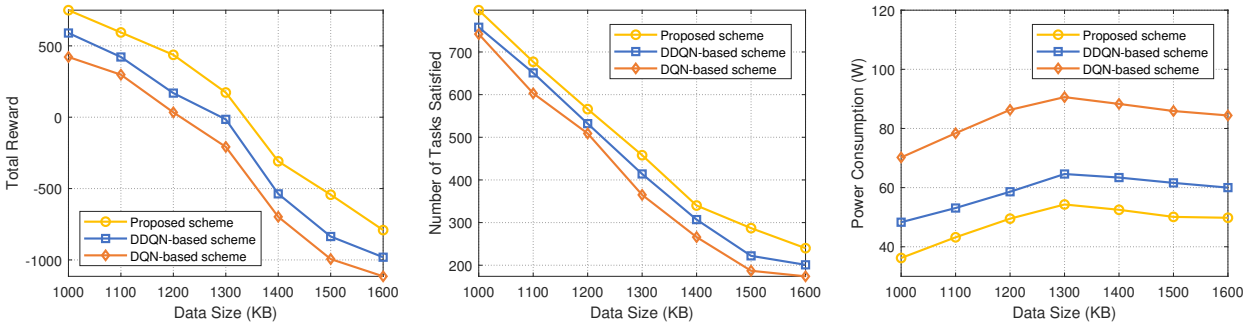
In Fig. 3, we study the convergence performance of the proposed scheme compared with the DDQN and DQN based benchmark algorithms. The first few hundred sets are relatively small and then tend to a relatively stable high value. Once

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3133110, IEEE Internet of Things Journal



(a) Total reward (b) Number of tasks satisfied (c) Power consumption

Fig. 5: Impact of the computing capacity on the computation offloading performance.



(a) Total reward (b) Number of tasks satisfied (c) Power consumption

Fig. 6: Impact of the data size on the computation offloading performance.

20000 experiences are saved in the replay memory buffer, the learning phase of the three algorithms starts to update the parameters of participants and critics. Therefore, the total reward of each set fluctuates sharply at the beginning of the learning phase and then increases with the gradual optimization of the parameters. With the help of duel structure, the learning convergence speed of the scheduling scheme based on D3QN is the fastest, reaching convergence in about 1200 sets. Our scheme can also get a better reward than the benchmark scheme. In addition, compared with the scheme based on DQN, the scheme based on DDQN has faster learning speed and better practicability.

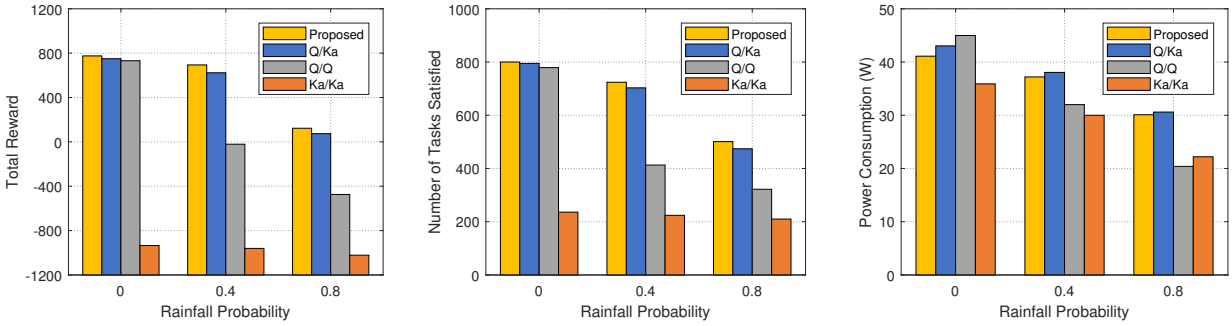
In Fig. 4, we study the effect of learning rate on algorithm performance. With an increase of the learning rate, the algorithm has a faster convergence speed. A low learning rate leads to a slow convergence speed, making the agent unable to learn from experiences. However, fast convergence may cause the agent to fall into the local optimum rather than the global optimum. Therefore, when setting the learning rate, it should not be too high or too low. In the following simulation, we need to find the appropriate learning rate for different resource and environment parameters to obtain moderate convergence speed and good learning stability.

Next, we test the performance of different computation offloading algorithms under different indicators to verify the effectiveness of our proposed D3QN algorithm. In Fig. 5, we evaluate the performance of different computation offloading algorithms under different computing capabilities of ground stations. For all schemes, the average reward of all IoRT devices increase with the computing capacity and finally tends to be stable. Compared with other benchmark schemes, the proposed scheme always obtains higher rewards. Specifically, when the processing performance of the ground station is 12 task/τ, the reward of the scheme is 27.1% and 77.7% higher than that of the scheme based on DDQN and DQN, respectively. In addition, with the increase of processing performance at the ground station, both the average reward and the number of tasks meeting the delay requirements increase. In contrast, the average power consumption of satellites decreases.

In Fig. 6, we evaluate the performance of different task scheduling schemes with different task packet sizes. With the increase of task size, the average reward and the number of tasks meeting delay requirements decrease, while the average power consumption increases. In addition, our scheme achieves better performance than the benchmark schemes. For example, when the packet size is 1100 KB, the average reward of the proposed scheme is 40.7% and 99.3% higher than that of the DDQN and DQN based scheme, respectively.

For computation offloading, we compare the performance of the proposed multi-layer Ka/Q-band LEO satellite network architecture with the Ka/Ka scheme, Q/Ka scheme and Q/Q scheme. Due to the difference of how Ka-band and Q-band are affected by rainfall, we compare the performance under different rainfall probability parameters. In Fig. 7, the rainfall

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3133110, IEEE Internet of Things Journal



(a) Total reward (b) Number of tasks satisfied (c) Power consumption

Fig. 7: Impact of the rainfall probability on the computation offloading performance.

probability parameter varies among 0, 0.4 and 0.8. With the parameter increases, the probability of high intensity rainfall is greater. It can be seen that the performance of the Ka/Q scheme is the best. The performance of the Q/Ka scheme is slightly worse, but is similar to that of the Ka/Q. This is because after the exchange between the orbital height of the Q-band and the Ka-band satellites, the free-space path loss of the Ka-band increases and that of the Q-band decreases. Hence the overall transmission rate decreases only by 2% when there is no rain. Due to the limitation of available bandwidth, the transmission rate of the Ka/Ka scheme is greatly reduced and its performance is the worst. When there is no rainfall, the performance of the Q/Q scheme is similar to that of the Q/Ka due to the slight increase of free-space path loss. When the rainfall increases, the performance of the Q-band decreases sharply as compared with the Ka-band. As the rainfall intensity increases, the power consumption decreases with the number of successfully transmitted sub-tasks. When the rain intensity is 0, the power consumption is mainly affected by the free-space path loss. When the rain intensity increases and the number of successfully transmitted sub-tasks decreases, the power consumption is also affected by the actual amount of data transmitted. Therefore, the Ka/Q scheme balances available bandwidth, free-space path loss and rain attenuation, and has the best performance among the four schemes.

To sum up, from the simulation results, the proposed scheme can effectively achieve convergence through continuous training. Compared with the benchmark algorithms, the algorithm based on D3QN has a faster convergence speed and better practicability. In addition, under different parameter settings, our scheme outperforms the benchmark scheme in terms of reward, the number of tasks with satisfied delay constraints and satellite power consumption. Compared with the other schemes with different frequency band settings, the proposed scheme adopts appropriate settings of two types of satellites, which effectively improves the reward of the computation offloading system.

VII. Conclusion

In this paper, we consider a multi-layer Ka/Q-band satellite-terrestrial network to provide IoRT devices for more available bandwidth resources and communication robustness in different rainfall intensities. Under this scenario, we formulate the dynamic computation offloading optimization problem and transform it into an MDP to maximize the number of offloaded tasks while meeting delay requirements and minimize the power consumption of satellites. Then, to efficiently solve the transformed MDP formulation, we develop a D3QN-based offloading algorithm to find the solution. The solution allows the controller deployed at the GEO satellite to make optimal decisions, including the satellite/ground station selection and bandwidth resource allocation based on channel conditions, satellite positions, and ground station computation capabilities. Finally, we conduct extensive experiments to validate the effectiveness and superiority of our proposed offloading scheme.

References

[1] H. Li, K. Ota, and M. Dong, "LS-SDV: Virtual network management in large-scale software-defined IoT," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1783–1793, 2019.

[2] Z. Jia, M. Sheng, J. Li, D. Niyato, and Z. Han, "LEO-satellite-assisted UAV: Joint trajectory and data collection for Internet of remote things in 6G aerial access networks," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9814–9826, 2021.

[3] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[4] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[5] M. Sheng, D. Zhou, R. Liu, Y. Wang, and J. Li, "Resource mobility in space information networks: Opportunities, challenges, and approaches," *IEEE Network*, vol. 33, no. 1, pp. 128–135, 2019.

[6] B. Elizabeth, "Small satellite market observations," in *Proc. Annu. AIAA/USU Conf. Samll Satell.*, Logan, UT, USA, Aug. 2015, pp. 1–5.

[7] M. De Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, "Satellite communications supporting Internet of remote things," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 113–123, 2016.

[8] P. Rudol and P. Doherty, "Evaluation of human body detection using deep neural networks with highly compressed videos for UAV search and rescue missions," *Springer, Cham*, 2019.

[9] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.

[10] 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 Transactions on Wireless Communications*, vol. 20, no. 2, pp. 911–925, 2021.

[11] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, "Satellite-terrestrial integrated edge computing networks: Architecture, challenges, and open issues," *IEEE Network*, vol. 34, no. 3, pp. 224–231, 2020.

[12] R. Deng, B. Di, S. Chen, S. Sun, and L. Song, "Ultra-dense LEO satellite offloading for terrestrial networks: How much to pay the satellite operator?" *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6240–6254, 2020.

[13] S. Tani, K. Motoyoshi, H. Sano, A. Okamura, H. Nishiyama, and N. Kato, "An adaptive beam control technique for Q band satellite to maximize diversity gain and mitigate interference to terrestrial networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 1, pp. 115–122, 2019.

[14] https://www.starlink.com/.

[15] A. Kelmendi, A. Hrovat, M. Mohori, and A. Vilhar, "Prediction model of fade duration statistics for satellite communications at Ka and Q - bands," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 8, pp. 5519–5531, 2019.

[16] 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 Transactions on Green Communications and Networking*, vol. 3, no. 3, pp. 806–816, 2019.

[17] Z. Ji, Y. Wang, W. Feng, and J. Lu, "Delay-aware power and bandwidth allocation for multiuser satellite downlinks," *IEEE Communications Letters*, vol. 18, no. 11, pp. 1951–1954, 2014.

[18] S. Tani, K. Motoyoshi, H. Sano, A. Okamura, H. Nishiyama, and N. Kato, "An adaptive beam control technique for diversity gain maximization in LEO satellite to ground transmissions," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–5.

[19] H. Tsuchida, Y. Kawamoto, N. Kato, K. Kaneko, S. Tani, S. Uchida, and H. Aruga, "Efficient power control for satellite-borne batteries using Q-learning in low-earth-orbit satellite constellations," *IEEE Wireless Communications Letters*, vol. 9, no. 6, pp. 809–812, 2020.

[20] 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 Transactions on Wireless Communications*, vol. 18, no. 1, pp. 47–62, 2019.

[21] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, and J. Rao, "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open Journal of Vehicular Technology*, vol. 1, no. 1, pp. 45–66, Jan. 2020.

[22] W. Wu, N. Chen, C. Zhou, M. Li, X. Shen, W. Zhuang, and X. Li, "Dynamic RAN slicing for service-oriented vehicular networks via constrained learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2076–2089, Jul. 2021.

[23] W. Huang, K. Ota, M. Dong, T. Wang, S. Zhang, and J. Zhang, "Result return aware offloading scheme in vehicular edge networks for IoT," *Computer Communications*, vol. 164, pp. 201–214, 2020.

[24] Z. Song, Y. Hao, Y. Liu, and X. Sun, "Energy efficient multi-access edge computing for terrestrial-satellite Internet of things," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[25] Q. Tang, Z. Fei, B. Li, and Z. Han, "Computation offloading in LEO satellite networks with hybrid cloud and edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9164–9176, 2021.

[26] B. Wang, T. Feng, and D. Huang, "A joint computation offloading and resource allocation strategy for LEO satellite edge computing system," in *Proc. IEEE Int. Conf. Commun. Tech. (ICCT)*, Nanning, China, Oct. 2020, pp. 649–655.

[27] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1117–1129, 2019.

[28] C. Qiu, H. Yao, F. R. Yu, F. Xu, and C. Zhao, "Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5871–5883, 2019.

[29] D. Zhou, M. Sheng, R. Liu, Y. Wang, and J. Li, "Channel-aware mission scheduling in broadband data relay satellite networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 5, pp. 1052–1064, 2018.

[30] Z. S. Grard Maral, Michel Bousquet, *Satellite Communications Systems: Systems, Techniques and Technology, Sixth Edition*. John Wiley and Sons, 2020.

[31] "Propagation data and prediction methods required for the design of earth-space telecommunication systems," *Document P.618-12 Rec. ITU-R*, 2015.

[32] "Specific attenuation model for rain for use in prediction methods," *Document P.838-3 Rec. ITU-R*, 2005.

[33] R. M. Manning, "A unified statistical rain-attenuation model for communication link fade predictions and optimal stochastic fade control design using a location-dependent rain-statistics database," *International Journal of Satellite Communications*, vol. 8, no. 1, pp. 11–30, 1990.

[34] H. He, H. Shan, A. Huang, Q. Ye, and W. Zhuang, "Edge-aided computing and transmission scheduling for LTE-U-enabled IoT," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 7881–7896, 2020.

[35] H. Peng and X. Shen, "Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2416–2428, 2020.

[36] Z. Xiong, Y. Zhang, W. Y. B. Lim, J. Kang, D. Niyato, C. Leung, and C. Miao, "UAV-assisted wireless energy and data transfer with deep reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 85–99, 2021.

[37] H. Li, K. Ota, and M. Dong, "Deep reinforcement scheduling for mobile crowdsensing in fog computing," *ACM Transactions on Internet Technology (TOIT)*, vol. 19, no. 2, pp. 1–18, 2019.

[38] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, and L.-C. Wang, "Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges," *IEEE Vehicular Technology Magazine*, vol. 14, no. 2, pp. 44–52, 2019.

[39] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[40] Y. Liu, H. Wang, M. Peng, J. Guan, and Y. Wang, "An incentive mechanism for privacy-preserving crowdsensing via deep reinforcement learning," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[41] N. Zhao, Y. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5141–5152, 2019.

[42] "Topography for earth-to-space propagation modelling," *Document P.1151 Rec. ITU-R*, 2001.

[43] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.