DRL-Based URLLC-Constraint and Energy-Efficient Task Offloading for Internet of Health Things

Yixiao Wang, Huaming Wu, *Senior Member, IEEE*, Rutvij H. Jhaveri, *Senior Member, IEEE*, and Youcef Djenouri, *Senior Member, IEEE*

***Abstract*—Internet of Health Things (IoHT) is a promising e-Health paradigm that involves offloading numerous computational-intensive and delay-sensitive tasks from locally limited IoHT points to edge servers (ESs) with abundant computational resources in close proximity. However, existing computation offloading techniques struggle to meet the burgeoning health demands in ultra-reliable and low-latency communication (URLLC), one of the 5G application scenarios. This article proposes a Multi-Agent Soft-Actor-Critic-discrete based URLLC-constrained task offloading and resource allocation (MASACDUA) scheme to maximize throughput while minimizing power consumption on the remote side, considering the long-term URLLC constraints. The URLLC constraint conditions are formulated using extreme value theory, and Lyapunov optimization is employed to divide the problem into task offloading and computation resource allocation. MASAC-discrete and a queue backlog-aware algorithm are utilized to approach task offloading and computation resource allocation, respectively. Extensive simulation results demonstrate that MASACDUA outperforms traditional DRL algorithms under different IoHT points and data arrival rate intervals and achieves superior performance in delay, bound violation probability, and other characteristics related to URLLC.**

***Index Terms*—Internet of Health Things, multi-agent reinforcement learning, task offloading, URLLC.**

I. Introduction

THE Internet of Health Things (IoHT), an extension of Internet of Things (IoT) in the healthcare domain, is gaining significant popularity in numerous ways [1], [2], [3]. It is revolutionizing healthcare by enabling smart health solutions [4], [5], [6], such as real-time monitoring of physiological data, sensor patches with real-time respiration, human activity recognition and the development of sustainable wearable devices. Healthcare has become more prevalent due to the rapid adoption and large-scale deployment of IoT and substantial advancements in data generation and exchange [7], [8]. IoHT is reshaping traditional health systems in many unprecedented ways [9], [10], [11], enhancing data processing accuracy, strengthening reliability, and enabling convenient connections. However, from another perspective, there are still challenges related to life-demanding or computation-intensive tasks that require prompt processing of abundant data with strict quality of service (QoS) requirements, posing a significant challenge to the existing IoHT framework.

Nonetheless, due to the inherent limitations of computation resources in IoHT devices, tasks originating from these devices are often offloaded to remote cloud servers, resulting in unsatisfactory experiences, particularly when considering the stringent QoS demands of IoHT applications. These cloud servers are unable to meet the real-time processing and response requirements of IoHT services [12]. To address this issue, edge computing has emerged as a viable solution to process health-related data closer to its source, thereby minimizing latency and ensuring better QoS for IoHT applications.

One of the widely accepted application scenarios for 5G technology is ultra-reliable low-latency communication (URLLC). URLLC plays a vital role in upholding applications within the IoHT context [13] as certain tasks in IoHT are life-demanding and delay-sensitive. Conventional techniques fail to meet the specific requirements of IoHT in such cases. Therefore, the adoption of URLLC is essential to ensure the reliability and effectiveness of our model. The extreme value theory (EVT) [14] offers an effective means to characterize the features of URLLC by concentrating on the probability and statistics of bound violation events.

Multi-agent deep reinforcement learning (MADRL) [15] is an emerging technology that enables more than one agent to make decisions through interacting with the environment, without any prior knowledge. Considering the presence of numerous IoHT points in real-world scenarios, the adoption of MADRL becomes a natural choice. Heuristic approaches, although capable of dealing with multiple IoHT points, can only provide actions without a comprehensive policy, making them vulnerable to system disturbances. Thus, MADRL is considered as a more suitable approach

Manuscript received 10 March 2023; revised 2 June 2023 and 26 June 2023; accepted 17 July 2023. Date of publication 24 July 2023; date of current version 6 June 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62071327 and in part by Tianjin Science and Technology Planning Project under Grant 22ZYYYJC00020. *(Corresponding authors: Huaming Wu; Rutvij H. Jhaveri.)*

Yixiao Wang and Huaming Wu are with the Center for Applied Mathematics, Tianjin University, Tianjin 300072, China (e-mail: wang_yixiao@tju.edu.cn; whming@tju.edu.cn).

Rutvij H. Jhaveri is with the Department of Computer Science and Engineering, School of Technology, Pandit Deendayal Energy University, Gandhinagar 382007, India (e-mail: rutvij.jhaveri@sot.pdpu.ac.in).

Youcef Djenouri is with the NORCE Norwegian Research Center Oslo, 0368 Oslo, Norway, and also with the University of South-Eastern Norway, 3616 Kongsberg, Norway (e-mail: yodj@norceresearch.no).

Digital Object Identifier 10.1109/JBHI.2023.3297525

in this context. There are numerous options available when it comes to MADRL algorithms. One of the top choices currently is the multi-agent Soft Actor-Critic (SAC) algorithm [16], which has been widely employed in previous works. Unlike other DRL algorithms that solely focus on maximizing the reward function in the long run, SAC also maximizes the entropy of the action, ensuring a more balanced exploration and exploitation trade-off during the training process. Notably, the entropy parameter in SAC is treated as a trainable parameter [17]. However, SAC is not well-suited for discrete action space. Hence, we employ a variant of SAC known as SAC-discrete [18] combined with multi-agent to address this limitation. Our objective is to ensure stability in terms of data queuing and energy consumption, while also optimizing long-term throughput. We employ Lyapunov optimization to decompose a multi-stage optimization problem into many sub-problems in each time slot, achieving stability and optimizing performance in our IoHT system [19], [20].

In this article, we bring up a Multi-Agent Soft Actor-Critic-Discrete-based URLLC-constrained task offloading and resource allocation (MASACDUA) scheme for various IoHT points. The intention is to maximize the throughput of the system by offloading computation-intensive and delay-sensitive tasks, while ensuring long-term URLLC constraints. The main contributions of this article are summarized as follows:

- *URLLC-constraint Task Offloading Model:* We develop a task offloading model that takes into account the URLLC constraints. The proposed model is specifically designed to integrate the dynamic computing capabilities of base stations, which enhances its practicality for real-world implementation. By incorporating this feature, the model becomes well-suited for deployment in practical scenarios and can effectively adapt to the dynamic computing requirements of base stations.
- *MASACDUA Mechanism with Lyapunov Optimization:* To tackle the formulated problem, we propose a multi-agent SAC-discrete approach, which mitigates the inferior learning performance that can arise from unstable data arrivals and dynamic computing resources with varying numbers of IoHT points and BSs. Furthermore, we utilize Lyapunov optimization to achieve short-term optimization while adhering to long-term URLLC constraints, which may not be attainable within short time periods.
- *CTDE Execution:* To enhance the throughput from IoHT points and reduce the energy consumption of BSs, we further generalize the proposed centralized algorithm into a decentralized control setting. Particularly, each IoHT point acts as an independent agent with its own decentralized policy, which explores offloading decisions based on local observations. And when training the model, each agent can get the whole state.

II. Related Work

A. Task Offloading in IoHT

Task offloading is considered an essential direction for the IoHT system and has attracted significant attention from both academia and industry. For instance, Materwala et al. [21] proposed an algorithm for energy-aware offloading that minimizes the energy consumed by the patients' requests, which are computation-intensive but do not require real-time response. Wang et al. [13] proposed an energy-efficient scheme called UTO-EXP3 that employs multi-armed bandit (MAB) and EVT for task offloading in locally resource-limited IoHT points. Mukherjee et al. [22] aimed to minimize the average response time of tasks with different priorities that are scheduled in edge-assisted healthcare services with hard and soft deadlines at end-users and edge medical servers. Ren et al. [23] tackled the critical challenges of task offloading strategies by considering time, security, and reliability factors. They proposed a hierarchical network framework based on wireless body area networks that centralizes control but distributes computation, with the goal of enabling smart healthcare IoT applications.

B. Task Offloading Under URLLC Scenarios

In URLLC scenarios, the need for immediate and accurate communication often results in a flood of data requests, necessitating the use of task offloading and demanding higher levels of QoS. To address this challenge, Chen et al. [24] formulated an optimization problem for a parallel task offloading scenario aimed at reducing service delay. Their approach involves jointly finding the best solution, taking into account the computation resources of users and the sub-tasks assigned to multiple edge points in the vicinity. To optimize the solution, the authors consider normal tasks with minimal decomposition granularity. Dang et al. [25] proposed a novel edge network architecture that addresses the URLLC constraints. Specifically, the proposed architecture integrates communication allocation and computation offloading to reduce worst-case latency. This is achieved through consideration of factors such as user association, transmission power, and the processing rate of user equipment. Overall, their approach offers a unified solution that optimizes resource utilization and improves performance in URLLC-constrained environments. Wang et al. [26] focused on the down-link design in URLLC to identify transmission protocols that are latency-constrained and achieved a low output probability, while also translating the up-link procedure into an up-link budget. Liao et al. [27] proposed an intent-aware task offloading scheme for an air-ground combined vehicular edge computing (VEC) scenario, where they model the intent as maximizing long-term QoE while considering long-term URLLC constraints to increase the probability of task offloading success.

C. DRL Method Used for Task Offloading

Some researchers have investigated the MADRL-based task offloading without URLLC-constraint [28]. Li et al. [29], [30] expected long-term improvements for NOMA-enabled cooperative computation offloading, where a scattered network is adopted to enhance its stability whereas league learning is exploited to explore the environment collaboratively. Seid et al. [31] diminished the overall computation cost meanwhile guaranteeing the QoS requirements of IoT devices or UEs in the IoT network. Gao et al. [32] optimized multiple UAVs' trajectories to reduce the global synchronized communication overhead with ground users' offloading delay, energy efficiency as well as obstacle avoidance system. Jia et al. [33] proposed a multi-agent Ly-MAPPO to sustain each vehicle to maximize the logarithmic average data processing rate (LDPR) under long-term restrictions, which requires only local observation to give offloading policies and queue stability.

TABLE I
THE QUALITATIVE COMPARISON OF CURRENT LITERATURE, WHERE O&R REFERS TO THE USE OF BOTH OFFLOADING, AND ⊞ INDICATES THE UTILIZATION OF MULTI-MASAC

Paper	URLLC	O&R	DRL	Multi-Agent	IoHT	Energy-Efficiency	Data Throughput	Lyapunov	Heuristic
[21]	×	×	×	–	✓	✓	×	×	✓
[22]	×	×	×	–	✓	×	×	×	✓
[23]	×	✓	×	–	✓	×	×	×	✓
[13]	✓	×	×	–	✓	✓	×	✓	✓
[38]	✓	✓	×	–	×	✓	×	✓	✓
[24]	✓	✓	×	–	×	×	×	×	✓
[25]	✓	✓	×	–	×	✓	×	×	✓
[26]	✓	×	×	–	×	×	×	×	×
[27]	✓	×	×	–	×	×	✓	✓	✓
[30]	✓	✓	✓	×	×	✓	×	×	✓
[28]	✓	✓	✓	×	×	×	✓	×	✓
[29]	×	✓	✓	✓	×	✓	×	×	×
[31]	×	✓	✓	✓	×	✓	×	×	×
[32]	×	✓	✓	✓	×	×	×	×	✓
[33]	×	×	✓	✓	×	✓	×	✓	✓
[35]	×	×	✓	⊞	×	✓	×	×	×
[36]	×	×	✓	⊞	×	✓	✓	×	×
[37]	×	×	✓	⊞	×	×	✓	×	×
[39]	×	×	✓	✓	✓	×	✓	✓	×
[40]	×	×	×	–	✓	✓	×	×	✓
[41]	×	×	×	–	✓	×	×	×	✓
[42]	×	✓	×	–	✓	✓	×	×	✓
[43]	×	✓	✓	✓	✓	✓	×	×	×
ours	✓	✓	✓	✓	✓	✓	✓	✓	✓

Recently, a newly multi-agent maximum-entropy CTDE architecture named MASAC [34] has arrested attention from the IoT or edge computing academia. The utilization of entropy regularization in the reward function can effectively potentiate exploration, thus deterring the problem of over-fitting and pre-convergence. Many researchers have to integrate MASAC with edge computing scenarios. Wu et al. [35] constructed an edge-terminal collaboration model, where energy minimization and delay violation punishment are optimized through spectrum sharing and vehicle power control for task offloading. Wu et al. [36] proposed a method to minimize the average age of information and front-haul traffic loads in IoT networks by characterizing the average energy consumption during transmission from IoT sensors. This is done under the assumption of an effective wireless transmission condition. Yan et al. [37] proposed a consensus communication mechanism founded on counterfactual reasoning. Graph Attention Networks with the fully decentralized MASAC are utilized to reinforce the cooperation among agents.

D. A Qualitative Comparison

Table I presents a comprehensive comparison between our approach and related works with regards to various essential elements, e.g., URLLC, Offloading and Resource allocation (O&R), DRL, Multi-Agent, IoHT scenarios, energy-efficiency, data throughput, and Lyapunov-based and heuristic methods. To the best of our knowledge, our proposed approach is the first to integrate all the aforementioned factors into a unified framework, thereby distinguishing itself from prior works.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. Overall System Model

Fig. 1 illustrates the system architecture consisting of N IoHT points and $I + J$ BSs, where I BSs with larger computation resources and J BSs are BSs with small ones. In real scenarios, it is not reasonable to set only a single or just one type of edge server for the whole system, considering the distances and distributions between BSs and IoHT points. IoHT points include patients and doctors, financial services and medical institutions, and the devices of pharmaceutical enterprises and

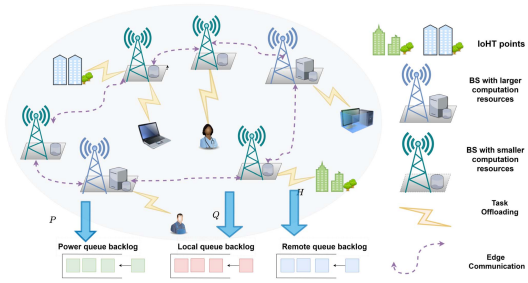


Fig. 1. Application scenario of the proposed scheme for IoHT.

TABLE II
NOTATIONS AND THEIR DEFINITIONS

Notations	Definition
$U_i(t)$	The attainable throughput for u_i
$Q_i(t)$	Local task queues for u_i
d_i^L	Queuing bounds for u_i
ϵ_i^L	Tolerable probabilities of bound violation for u_i
P_i^L	Extreme event occurrence probability for u_i
S_i^L	The excess value for u_i
$\mathbb{E}(S_i^L)$	Long-term conditional expectations for S_i^L
$\mathbb{E}(W_i^L)$	Long-term conditional expectations for the square of S_i^L
σ_i^L	The scale parameter of GPD distribution for u_i
$\sigma_i^{L,th}$	Threshold for σ_i^L
ξ_i^L	The shape parameter of GPD distribution for u_i
$\xi_i^{L,th}$	Threshold for ξ_i^L
P	Transmission power
N_0	Noise power
λ_j	Computation density of task data
κ	The computation power efficiency
$A_i(t)$	Task data arrival rate for u_i
γ_j	The threshold for power queue in long term
$Z_{i,j}(t)$	Amount of task data offloaded u_i to s_j
$W_{i,j}(t)$	Sub-channel bandwidth
$g_{i,j}(t)$	Channel gain
$R_{i,j}(t)$	Transmission rate
$d_{i,j}^O$	Queuing bounds for $H_{i,j}(t)$
$\epsilon_{i,j}^O$	Tolerable probabilities of bound violation for u_i
$P_{i,j}^O$	Extreme event occurrence probability for u_i
$S_{i,j}^O$	The excess value for $H_{i,j}(t)$
$\mathbb{E}(S_{i,j}^O)$	Long-term conditional expectations for $S_{i,j}^O$
$\mathbb{E}(W_{i,j}^O)$	Long-term conditional expectations for the square of $S_{i,j}^O$
$\sigma_{i,j}^O$	The scale parameter of GPD distribution for $H_{i,j}(t)$
$\sigma_{i,j}^{O,th}$	Threshold for $\sigma_{i,j}^O$
$\xi_{i,j}^O$	The shape parameter of GPD distribution for $H_{i,j}(t)$
$\xi_{i,j}^{O,th}$	Threshold for $\xi_{i,j}^O$
$f_{i,j}(t)$	Computation resources allocated by s_j for u_i
$p_j(t)$	The amount of computation power in s_j

online platforms. Each BS is co-located with an ES that provides both radio access and computational services, eliminating the need for assistance from a remote cloud or other BSs. The devices are irregularly distributed throughout the network and operate continuously.

The sets of IoHT points and servers are denoted as $\mathcal{U} = \{u_1, \ldots, u_i, \ldots, u_M\}$ and $\mathcal{S} = \{s_1, \ldots, s_j, \ldots, s_{I+J}\}$, respectively. We consider a time-slotted model that is characterized by a fixed duration of time slots denoted as τ, and a series of successive slots, where the set of all slots is defined as $\mathcal{T} = \{1, \ldots, t \ldots, T\}$. We assume that the channel information remains unchanged during each slot, while it may fluctuate dynamically between different slots. Meanwhile, the set of available BSs for each user, denoted as u_i, remains fixed across slots. In each slot, the user u_i autonomously decides whether to offload their tasks or not. The main notations in our article are summarized in Table II.

B. Traffic Model at IoHT Device Side

We presume that tasks arrive at user u_i randomly in each time slot and are subsequently offloaded to the selected BS for computation. The number of tasks arriving at u_i in the t-th time slot is denoted as $A_i(t)$ Mbits/s. To store data that has not yet been offloaded from u_i, we introduce the concept of the local task buffer. Specifically, each task buffer associated with u_i can be modeled as a data queue, and its backlog (i.e., the length of the local task buffer) is denoted as $Q_i(t)$, where

$$Q_i(t+1) = \max\{Q_i(t) - U_i(t) + \tau A_i(t), 0\}, \tag{1}$$

which satisfies the initial conditions: $Q_i(0) = 0, \forall u_i \in \mathcal{U}$ at $t = 0$. The transmission rate from u_i to s_j is given by:

$$R_{i,j}(t) = W_{i,j}(t) \log_2\left(1 + \frac{P g_{i,j}(t)}{N_0}\right), \tag{2}$$

where $W_{i,j}$ denotes the subchannel bandwidth allocated to each BS and shared among its connected UEs, P and N_0 are the transmission power, and noise power density, respectively, and $g_{i,j}(t)$ is the wireless channel gain between $i \in U$ and BS $j \in S$, including path loss and channel fading. We assume that all channels experience block fading.

Furthermore, we assume that the downlink transmission delay can be ignored due to its negligible cost compared with the offloaded tasks before computation. In the t-th time slot, $D_{i,j}(t)$ denotes the quantity of task data offloaded from point u_i to BS s_j and $R_{i,j}(t)$ denotes the achievable throughput of u_i in the same time slot, which can be formulated as:

$$z_{i,j}(t) = \min\{Q_i(t) + \tau A_i(t), \tau R_{i,j}(t)\}, \tag{3}$$

$$U_i(t) = \sum_{j=1}^{I+J} x_{i,j}(t) z_{i,j}(t). \tag{4}$$

C. Computation Model at the BS Side

A virtual task buffer is established at each BS to store the offloaded but not yet executed tasks from u_i. The execution of these offloaded tasks is carried out using the CPU provided by the respective BSs. The amount of task data produced by u_i and stored at BS s_j is denoted by $H_{i,j}(t)$. The allocation of CPU-cycle frequency will be explained later. Even though u_i does not transmit data to s_j, $f_{i,j}(t)$ can still be non-zero, and therefore the amount of data processed at s_j, denoted as $Y_{i,j}(t)$, is defined as:

$$Y_{i,j}(t) = \min\left\{H_{i,j}(t) + x_{i,j}(t) z_{i,j}(t), \frac{\tau f_{i,j}(t)}{\lambda_i}\right\}, \tag{5}$$

where λ_i denotes the computation density of the task data and satisfies the constraint $\sum_{i=1}^{N} f_{ij}(t) \le f_{j,\max}(t)$.

The task buffer dedicated to storing the tasks of user u_i at the BS can be modeled as a queue. However, the BS-side information such as the queue backlog $H_{i,j}(t)$ and allocated CPU-cycle frequency $f_{i,j}(t)$ are unknown to u_i. Nevertheless, u_i can establish a virtual remote queue $H_{i,j}(t)$ locally for BS s_j. This virtual queue evolves as follows:

$$H_{i,j}(t+1) = \max\{H_{i,j}(t) - Y_{i,j}(t) + x_{i,j}(t) z_{i,j}(t), 0\}, \tag{6}$$

which satisfies the initial conditions: $H_{i,j}(0) = 0, \forall s_j \in \mathcal{S}, \forall u_i \in \mathcal{U}$ at $t = 0$

D. Power Consumption Model at the BS Side

The power consumption of s_j for remote execution is:

$$p_j(t) = \sum_{i=1}^{N} \kappa \left(f_{i,j}(t)\right)^3, \tag{7}$$

where κ is the switched capacitance of s_j's execution CPU, determined by the hardware implementation. Similarly, $p_j(t)$ should satisfy:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} p_j(t) \le \gamma_j, \forall s_j \in \mathcal{S}, \tag{8}$$

where γ_j is the time-average power threshold.

E. URLLC Constraints

The end-to-end latency encountered by a device is influenced by the choice of execution approach. In the scenario where IoHT devices offload tasks to BSs, the experienced end-to-end latency encompasses the following components: i) queuing delay within the local task buffer; ii) uplink and downlink delay; iii) queuing delay at the remote location; iv) computational delay at the remote location. Neglecting the downlink feedback delay of computational results is deemed reasonable, as the data size of computational results is typically smaller compared to that of offloaded tasks. This assumption has also been adopted in prior works [13], [44].

URLLC from both IoHT points and ESs requires rigorous restrictions on the queuing delay which holds a large proportion of the end-to-end delay. As a result, URLLC constraints must be enforced on both the local and remote sides. According to Little's Law, the average queuing delay is proportional to the ratio of the average queue length to the average data arrival rate [45]. Therefore, the average queuing delays for the local task $\mathcal{D}_i$ and the remote task $\mathcal{D}_{i,j}$ can be expressed as follows:

$$\mathcal{D}_i = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \frac{Q_i(t)}{\tilde{A}_i(t-1)} < d_i^L, \tag{9}$$

$$\mathcal{D}_{i,j} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \frac{H_{i,j}(t)}{\tilde{z}_i(t-1)} < d_{i,j}^O, \tag{10}$$

where the time-average data arrival rates of local and remote task buffers denoted by $\tilde{A}_i(t-1) = \frac{1}{t}\sum_{m=0}^{t-1} A_i(m)$ and $\tilde{Z}_{i,j}(t-1) = \frac{1}{t}\sum_{m=0}^{t-1} x_{i,j}(m) z_{i,j}(m)$, respectively. The local and remote task buffers have corresponding queuing delay bounds d_i^L and $d_{i,j}^O$, respectively. Focusing solely on the average queuing delay may result in the occurrence of extreme events where the queuing delay surpasses the upper bound, which is undesirable for URLLC. Therefore, in order to elaborate on the restrictions, we need to quantify these extreme events. We define excess values as: $S_i^L(t) = \max\{Q_i^L(t) - \tilde{A}_i(t-1)d_i, 0\}$ for $Q_i(t)$ and $S_{i,j}^O(t) = \max\{S_{i,j}^O(t) - \tilde{z}_{i,j}(t-1)d_{i,j}, 0\}$ for $H_{i,j}(t)$. Then, we can define indicators for the occurrence of these extreme events as $\mathbb{I}_i = \mathbb{I}\{S_i^L(t) > 0\}$ for $Q_i(t)$ and $\mathbb{I}_{i,j} = \mathbb{I}\{S_{i,j}^O(t) > 0\}$.

Naturally, the long-term URLLC constraint is self-evidently characterized by constraints on the probability of extreme event

occurrence which can be formulated as follows:

$$\bar{P}_i^L = \lim_{T\to\infty} \frac{1}{T} \sum^{T} \Pr\left(S_i^L(t) > 0\right) \leq \varepsilon_i^L, \tag{11}$$

$$\bar{P}_{i,j}^O = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \Pr\left(S_{i,j}^O(t) > 0\right) \leq \varepsilon_{i,j}^O, \tag{12}$$

where $\varepsilon_i^L \ll 1$ and $\varepsilon_{i,j}^O \ll 1$ are the tolerable probabilities of bound violation. Moreover, it is imperative to consider the statistical properties of $S_i^L(t)$ and $S_{i,j}^O(t)$. To achieve this, we employ the EVT and leverage the Pickands-Balkman-de Haan Theorem [14] to characterize the tail distribution and statistical features of $\mathbb{I}_i$ and $\mathbb{I}_{i,j}$.

Specifically, the conditional excess distribution function (CEDF) of $\mathbb{S}i^L$ and $\mathbb{S}i, j^O$ can be approximated using a Generalized Pareto Distribution (GPD). In this regard, we assume a GPD with parameters σ and ξ. The first and second moments of the aforementioned GPD can be expressed as $M_f(\sigma,\xi) = \frac{\sigma}{1-\xi}$ and $M_s(\sigma,\xi) = \frac{2\sigma^2}{(1-\xi)(1-2\xi)}$, respectively.

The CEDF for u_i can be denoted as follows:

$$\bar{F}(s_i^L) = \frac{P(S_i^L(t) > s_i^L)}{P(S_i^L(t) > 0)}, \tag{13}$$

where $\sigma_i^L \leq \sigma_i^{L,th}$ and $\xi_i^L \leq \xi_i^{L,th}$, to ensure the reliability and latency restrictions.

The statistical properties of GPD and the relationship between its two parameter thresholds can be leveraged to establish constraints on the long-term time-average conditional expectations for both the first and second moment of the excess value, as follows:

$$\begin{aligned}\mathbb{E}(S_i^L) &= \lim_{T\to+\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[S_i^L(t) \mid S_i^L(t) > 0\right] \\ &\leq M_f(\sigma_i^{L,th}, \xi_i^{L,th}),\end{aligned} \tag{14}$$

$$\begin{aligned}\mathbb{E}(W_i^L) &= \lim_{T\to+\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[W_i^L(t) \mid S_i^L(t) > 0\right] \\ &\leq M_s(\sigma_i^{L,th}, \xi_i^{L,th}),\end{aligned} \tag{15}$$

where $W_i^L(t) = [S_i^L(t)]^2$.

The CEDF for $H_{i,j}(t)$ can be written as:

$$\bar{F}(s_{i,j}^O) = \frac{P(S_{i,j}^O(t) > s_{i,j}^O)}{P(S_{i,j}^O(t) > 0)}, \tag{16}$$

which follows the GPD $G(s_{i,j}^O; \sigma_{i,j}^O, \xi_{i,j}^O)$. The thresholds $\sigma_{i,j}^O \leq \sigma_{i,j}^{O,th}$ and $\xi_{i,j}^O \leq \xi_{i,j}^{O,th}$. We enforce the constraints on the time-average conditional first and second moment as:

$$\begin{aligned}\mathbb{E}(S_{i,j}^O) &= \lim_{T\to+\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[S_{i,j}^O(t) \mid S_{i,j}^O(t) > 0\right] \\ &\leq M_f(\sigma_{i,j}^{O,th}, \xi_{i,j}^{O,th}),\end{aligned} \tag{17}$$

$$\begin{aligned}\mathbb{E}(W_{i,j}^O) &= \lim_{T\to+\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[W_{i,j}^O(t) \mid S_{i,j}^O(t) > 0\right] \\ &\leq M_s(\sigma_{i,j}^{O,th}, \xi_{i,j}^{O,th}),\end{aligned} \tag{18}$$

where $W_{i,j}^O(t) = [S_{i,j}^O(t)]^2$.

F. Problem Formulation

Maximizing throughput alone cannot guarantee satisfactory performance, even if the queuing delay on the IoHT point side is reduced. Focusing solely on average queuing delay is insufficient for meeting the strict URLLC requirements, and may result in frequent occurrences of extreme events.

As far as we know, IoHT applications rely heavily on the availability of high throughput and low latency in time-varying network conditions. Therefore, we seek task offloading and resource allocation to formulate the problem for maximizing the long-term throughput of all IoHT points while satisfying long-term URLLC constraints, as follows:

$$\boldsymbol{\mathcal{P}_1}: \max_{\{\mathbf{x},\mathbf{f}\}} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{I+J} x_{i,j}(t) z_{i,j}(t), \tag{19}$$

$$\text{s.t.} \sum_{j=1}^{I+J} x_{ij}(t) = 1, \quad \forall u_i \in \mathcal{U}, \forall t \in \mathcal{T}, \tag{19a}$$

$$\sum_{i=1}^{N} f_{ij}(t) \leq f_{j,\max}(t), \forall s_j \in \mathcal{S}, \forall t \in \mathcal{T}, \tag{19b}$$

$$x_{ij}(t) \in \{0,1\}, f_{ij}(t) \geq 0, \forall u_i \in U, \forall s_j \in S, \tag{19c}$$

$$\text{Constraints (8), (11)} - \text{(18)}, \tag{19d}$$

where constraints in (19a), (19b), and (19c) ensure that each IoHT point can only select one BS for remote execution in each time slot and that the selected BS's CPU-cycle frequency is within the available frequency and power limits. Constraint (19d) places bounds on the long-term violation probability, as well as the conditional mean and second moment of the excess values of local and remote task queues. Directly solving problem $\boldsymbol{\mathcal{P}_1}$ is challenging due to the long-term constraints. Therefore, we employ Lyapunov optimization.

IV. Problem Transformation and Solutions

A. Problem Transformation

Based on Lyapunov optimization, we decompose $\boldsymbol{\mathcal{P}_1}$, which has tight URLLC constraints, into several sub-problems. These sub-problems can be optimized for sure in each time slot, while satisfying long-term objectives. Virtual queues are employed to transform the long-term URLLC constraints into stability restrictions. Specifically, we introduce three virtual queues as state variables that measure the behavior of local systems with respect to (11), (14), and (15), respectively.

$$Q_i^{L,(P)}(t+1) = \max\left\{Q_i^{L,(P)}(t) + \mathbb{I}\left\{S_i^L(t) > 0\right\} - \epsilon_i^L, 0\right\}, \tag{20}$$

$$\begin{aligned}Q_i^{L,(S)}(t+1) = \max\Big\{&Q_i^{L,(S)}(t) + \mathbb{I}\left\{S_i^O(t) > 0\right\} \times, \\ &\left(S_i^L(t+1) - M_f(\sigma_i^{L,th}, \xi_i^{L,th})\right), 0\Big\}\end{aligned} \tag{21}$$

$$Q_i^{L,(W)}(t+1) = \max\Big\{Q_i^{L,(W)}(t) + \mathbb{I}\left\{S_i^L(t)>0\right\}\times, \left(W_i^L(t+1) - M_s(\sigma_i^{L,th}, \xi_i^{L,th})\right), 0\Big\} \tag{22}$$

where $Q_i^{L,(P)}(t)$, $Q_i^{L,(S)}(t)$, and $Q_i^{L,(W)}(t)$ denote the deviations from the tolerable probabilities of bound violation, the long-term time-average conditional expectations for the first and second moment of the excess value of the local task queue, respectively.

Similarly, for constraints (12), (17), and (18), we respectively introduce three virtual queues as follows:

$$H_{i,j}^{O,(P)}(t+1) = \max\Big\{H_{i,j}^{O,(P)}(t) + \mathbb{I}\left\{S_{i,j}^O(t)>0\right\} - \epsilon_i^L, 0\Big\}, \tag{23}$$

$$H_{i,j}^{O,(S)}(t+1) = \max\Big\{H_{i,j}^{O,(S)}(t) + \mathbb{I}\left\{S_{i,j}^O(t)>0\right\} \times \left(S_{i,j}^O(t+1) - M_f(\sigma_{i,j}^{O,th}, \xi_{i,j}^{O,th})\right), 0\Big\} \tag{24}$$

$$H_{i,j}^{O,(W)}(t+1) = \max\Big\{H_{i,j}^{O,(W)}(t) + \mathbb{I}\left\{S_{i,j}^O(t)>0\right\} \times \left(W_{i,j}^O(t+1) - M_s(\sigma_{i,j}^{O,th}, \xi_{i,j}^{O,th})\right), 0\Big\} \tag{25}$$

where $H_{i,j}^{O,(P)}$, $H_{i,j}^{O,(S)}$ and $H_{i,j}^{O,(W)}$ denote the deviations from the tolerable probabilities of bound violation, the time-average conditional expectations for the first and second moment of the excess value of the remote task queue, respectively.

According to [20], it is proven that if the virtual queues satisfy mean rate stability, then their corresponding constraints are also satisfied. For instance, for the virtual queue $H_{i,j}^O(t)$, its mean rate stability requires that if $\lim_{T\to\infty}\mathbb{E}[H_{i,j}^O(t)]/T = 0$, then the constraint (12) is satisfied.

For each BS, we build the power queue:

$$P_j(t+1) = \max\{P_j(t) - \gamma_j + p_j(t)\}, \forall s_j \in S \tag{26}$$

Thus, problem $\boldsymbol{\mathcal{P}_1}$ can be transformed into problem $\boldsymbol{\mathcal{P}_2}$:

$$\boldsymbol{\mathcal{P}_2}: \max_{\{\mathbf{x},\mathbf{f}\}} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{I+J} x_{i,j}(t) z_{i,j}(t) \tag{27}$$

$$\text{s.t. Constraints (19a)} - \text{(19c)} \tag{27a}$$

$$\text{Constraints (1), (6), (20)} - \text{(26)}$$

$$\text{stay mean rate stable} \tag{27b}$$

Using the drift-plus-penalty algorithm of Lyapunov optimization [20], $\boldsymbol{\mathcal{P}_2}$ transformed into a sequence of deterministic sub-problems in the short term that can be solved by each device. We can further decouple $\boldsymbol{\mathcal{P}_2}$ into two sub-problems based on the involved variables:

$$\boldsymbol{\mathcal{SP}_1}: \max_{\{x_{i,j}(t)\}} F\left(x_{i,j}(t)\right) \tag{28}$$

$$\text{s.t. Constraints (19a), (19c)} \tag{28a}$$

$$\boldsymbol{\mathcal{SP}_2}: \max_{\{f_{i,j(t)}\}} K\left(f_{i,j}(t)\right) \tag{29}$$

$$\text{s.t. Constraint (19b)} \tag{29a}$$

where $K(f_{i,j}(t))$ and $F(x_{i,j}(t))$ are written as:

$$K\left(f_{i,j}(t)\right) = \alpha_H \sum_{i=1}^{N} Y_{i,j}(t)\left(x_{i,j}(t) z_{i,j}(t) + H_{i,j}(t)\right) - \alpha_P P_j(t) p_j(t) \gamma_j,$$

$$F\left(x_{i,j}(t)\right) = \left(\alpha + \alpha_{L,Q}\tau A_i(t) + \alpha_{L,Q} Q_i(t)\right) \times \sum_{j=1}^{I+J} x_{i,j}(t) z_{i,j}(t) - \alpha_{L,P}$$

$$\left(Q_i^{L,(P)}(t) - \epsilon_i^L\right)\mathbb{I}\left(S_i^L(t)>0\right) - \alpha_{L,S} Q_i^{L,(S)}(t)\cdot\Big(S_i^L(t+1) - M_f(\sigma_i^{L,th}, \xi_i^{L,th})\Big)\mathbb{I}\left(S_i^L(t)>0\right) - \alpha_{L,W} Q_i^{L,(W)}(t)\cdot\mathbb{I}\left(S_i^L(t)>0\right)$$

$$\left(W_i^L(t+1) - M_s(\sigma_i^{L,th}, \xi_i^{L,th})\right) - \alpha_{O,H}\sum_{j=1}^{I+J} H_{i,j}(t) x_{i,j}(t) z_{i,j}(t)$$

$$- \alpha_{O,P}\sum_{j=1}^{I+J}\left(Q_{i,j}^{O,(P)}(t) - \epsilon_{i,j}^O\right)\mathbb{I}\left(S_{i,j}^O(t)>0\right) - \alpha_{O,S}\sum_{j=1}^{I+J} Q_{i,j}^{O,(s)}(t)$$

$$\left(S_{i,j}^O(t+1) - M_f(\sigma_{i,j}^{O,th}, \xi_{i,j}^{O,th})\right)\cdot\mathbb{I}\left(S_{i,j}^O(t)>0\right) - \alpha_{o,W}\sum_{j=1}^{I+J} Q_{i,j}^{o,(W)}(t)\left(W_{i,j}^O(t+1) - M_f(\sigma_{i,j}^{O,th}, \xi_{i,j}^{O,th})\right) \times \mathbb{I}\left(S_{i,j}^O(t)>0\right).$$

Our objectives are to maximize the throughput of all IoHT points and minimize the overall energy consumption of BSs. However, traditional approaches cannot be used to solve $\boldsymbol{\mathcal{P}_2}$ due to its complexity. It involves both continuous variables (f) and discrete variables (x) and typically involves multiple IoHT points. Apparently, it is naive to traverse all possible (x) for terrible time complexity $O(TN^{I+J})$

B. MASACDUA Scheme

The proposed MASACDUA scheme is depicted in Fig. 2, which is comprised of MASAC-based task offloading and CPU resource allocation.

1) Resource Allocation: We propose a heuristic algorithm for resource allocation [28], as summarized in Algorithm 1.

In Algorithm 1, equal computational resources $f_{i,j}^{pre}(t)$ are first allocated to all available IoHT points, whereas the remaining resources $f_{i,j}^{re}$ are reassigned based on the remote queue backlog $H_{i.j}(t)$ and power consumed by the BSs $p_j(t)$. At first, we initialize the $\mathcal{U}_j(t)$, $num_j(t)$ and $\Delta f_{j,\max}(t)$. Then, we allocate $p_{ij}(t) f_{j,max}(t)$ evenly to all IoHT points in $\mathcal{U}_j(t)$ (line 2∼4), then select the IoHT point with the u_{i^*} on the strength of maximum target value $K(\kappa_{i,j}(t))$ (line 9∼10),

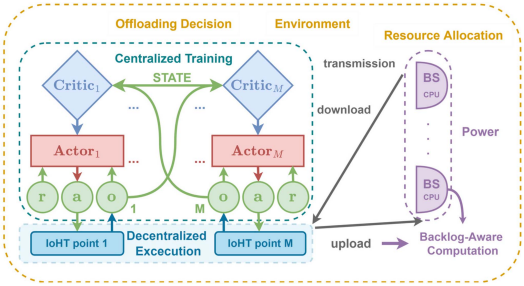


Fig. 2. Overall structure of the MASACDUA framework.

Algorithm 1: Computation Resource Allocation.

1: Initialize $\mathcal{U}_j(t) = \{u_i \in \mathcal{U} | H_{i,j}(t) > 0\}, num_j(t) = size(\mathcal{U}_J(t))$, and $\Delta f_{j,\max}(t) = (1 - p_j(t)) f_{j,\max}$
2: **for** IoHT point $i \in \mathcal{U}$ **do**
3: $f^{pre}_{i,j}(t) = p_j(t) f_{j,\max}(t)/num_j(t)$
4: **end for**
5: **for** s_j in $\mathcal{U}_j(t)$ **do**
6: **while** $\mathcal{U}_j(t) \neq 0$ and $\Delta f_{j,\max}(t) > 0$ **do**
7: $\kappa_{i,j}(t) = \min\{\Delta f_{j,\max}(t), \frac{\lambda_i}{\tau}[H_{i,j}(t)]\}$
8: $u_{i^*} = \arg\max_{u \in \mathcal{U}_j(t)} K(\kappa_{i,j}(t))$
9: **if** $\kappa_{i^*,j}(t) \geqslant f^{pre}_{i^*,j}(t)$ **then**
10: $f^{re}_{i^*,j}(t) = \kappa_{i^*,j}(t) - f^{pre}_{i^*,j}(t)$
11: $\Delta f_{j,\max}(t) = \Delta f_{j,\max}(t) - f^{pre}_{i^*,j}(t)$
12: **else**
13: $f^{pre}_{i^*,j}(t) = 0$
14: $\Delta f_{j,\max}(t) = \Delta f_{j,\max}(t) + f^{pre}_{i^*,j}(t) - \kappa_{i,j}(t)$
15: $f^{pre}_{i^*,j}(t) = \kappa_{i,j}(t)$
16: **end if**
17: $f_{i^*,j}(t) = f^{pre}_{i^*,j}(t) + f^{re}_{i^*,j}(t)$
18: $\mathcal{U}_j(t) = \mathcal{U}_j(t) \setminus u_{i^*}$
19: **end while**
20: **end for**

where $\kappa_{i,j}(t)$ is the largest computation resources wanted by u_i. And we get the $f_{i^*,j}(t)$ based on whether $\kappa_{i^*,j}(t)$ exceeds the $\kappa_{i^*,j}(t)$. If $\kappa_{i^*,j}(t)$ exceeds $f^{pre}_{i^*,j}(t)$, then we set $f^{re}_{i^*,j}(t) = \kappa_{i^*,j}(t) - f^{pre}_{i^*,j}(t)$ (line 9∼11). On the contrary, we set $f^{re}_{i^*,j}(t) = 0$ and $f^{pre}_{i^*,j}(t) = \kappa_{i^*,j}(t)$ (line 13∼15). In the two situations, we regenerate $\Delta f_{j,\max}(t)$ and $f^{pre}_{i^*,j}(t)$ (line 11∼14). Finally, u_{i^*} is removed from $\mathcal{U}_j(t)$ (line 18). The iteration halts when $\mathcal{U}_j(t) = \emptyset$ or $f_{j,max}(t) = 0$. Apparently, the allocation of computing resources is based on $f_{i,j}(t)$, which means a larger queue backlog or lower power consumption by the BSs receives computation resources more probably. The worst-case scenario involves $M \times N$ iterations.

2) MAMDP Model: The problem $\boldsymbol{\mathcal{SP}_1}$ can be represented as an observable MAMDP with the following components: $\langle n, \mathcal{S}, \mathcal{A}_1, \ldots, \mathcal{A}_n, \mathcal{O}_1, \ldots, \mathcal{O}_n, \mathcal{R}_1, \ldots, \mathcal{R}_n, \pi_1, \ldots, \pi_n, P \rangle$. We assume that N agents interact with the environment characterized by a set of states $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \ldots \times \mathcal{A}_n$. In each time slot, each agent receives its own private observation $\mathcal{O}_i$ and takes its own action $\pi_i : \mathcal{O}_i \to \mathcal{A}i$, and receives a reward $\mathcal{R}_i : \mathcal{S} \times \mathcal{A}_i \times \mathcal{S}' \to \mathcal{R}'_i$. Then, the environment transitions to a new state with probability $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S}' \to [0, 1]$.

Thus, we define the observation, action space, and reward function for each IoHT point in the t-th time slot as follows:

- *Observation Space $\mathcal{O}_i(t)$:* The network state at the beginning of each time slot is determined by the queue information, which is represented by:

$$\mathcal{O}_i(t) = \Big[Q_i(t), Q_i^{L,(P)}(t), Q_i^{L,(S)}(t), Q_i^{L,(W)}(t), H_{i,j}(t), H_{i,j}^{O,(P)}(t), H_{i,j}^{O,(S)}(t), H_{i,j}^{O,(W)}(t)\Big], \tag{30}$$

  which consists of the queue information, along with the virtual queue information. It should be noted that the power queue is excluded from the queue information in the MDP model, as we believe it is not relevant for the offloading decision.
- *Action Space $\mathcal{A}_i(t)$:* It is defined as the set of servers from which the IoHT point U_i can choose for task offloading. Therefore, the action space is represented by the vector $[x_{i,1}(t), \ldots, x_{i,I+J}(t)]$, where $x_{i,j}(t) = 1$ if server S_j is selected by device U_i for offloading in the t-th time slot, and $x_{i,j}(t) = 0$ otherwise.
- *Reward Function $\mathcal{R}_i(t)$:* The reward $\Gamma(x_{i,j}(t))$ of device U_i selecting BS S_j in the t-th time slot is set to maximize the optimization objective of $\boldsymbol{\mathcal{P}_1}$.

Unlike the previous methods, this approach does not require the mean and variance of a certain action. Additionally, the architecture of the Q-network has been modified to output Q-values for all possible actions with only states as input, instead of the previous approach where one Q-value was calculated for input including states and all actions.

3) Masac-Discrete: Each IoHT point is controlled by a dedicated agent, which is equipped with an actor network represented by $a_i(t) = \pi_i(o_i(t))$, two critic networks ($Q^i_j(s_t)$, where $j = 1, 2$), and their corresponding target networks ($Q^{i-}_j(s_{t+1})$, where $j = 1, 2$). Additionally, the agent is equipped with an experience replay buffer B_i. Following the CTDE structure, IoHT points can get the state for the training of their critic network, after that, each point executes their action based on the actor network with their local information.

The CTDE process is simply demonstrated as follows: During the training process, each agent shares its private observation $o_i(t)$ and action $a_i(t)$ with the environment, and the resulting state s_t is returned to all agents. This enables the simultaneous exchange of private information among all agents. The critic network of each agent is trained with the joint states and actions that include the observations of all agents. During the action-choosing process, each agent uses only its private observation $o_i(t)$ to execute its chosen action.

The pseudo-code of the proposed algorithm is listed in Algorithm 2. This algorithm encompasses two distinct phases, namely, the initialization phase (line 1 ∼ 8) and the DRL phase (line 6 ∼ 29), which can be further subdivided into the step process (line 9 ∼ 19) and the training process (line 20 ∼ 29).

- *The initialization phase:* All networks' parameters are set with the truncated uniform random numbers. The experience replay buffer's size is set to 10,000. At the beginning of each episode, the state and each observation $o_i(t)$ are initialized as all zero.

- *The step process:* Based on the first phase, each IoHT point attains its action and observations next time. Founded on each IoHT point's action and observation, we obtain each reward at time t. At last, we concatenate the $a_i(t)$ and $o_i(t+1)$ to get the $a(t)$ and $s(t+1)$ correspondingly.
- *The training process:* If it is time to train the network, like collecting enough data in the replay buffer, then a random mini-batch of transitions consisting of the current state, action taken by the agent, resulting reward, and next state $\{s^k, a_i^k, r_i^k, s^{k+1}\}, k = 1, 2, \ldots, B$ is sampled from the experience replay buffer B_i, where B denotes the batch size. These transitions are fed to the neural network for calculating the gradient.

The parameters of each critic network are updated in line 23, by minimizing the loss function $L_{Q_{i,j}}$ of critic i from the transition.

$$L_{Q_{i,j}} = \frac{1}{B}\sum_{k=1}^{B}\left[y_i^k - Q_{i,j}\left(s^k \mid \delta_{i,j}^Q\right)\right]^2, \; j = 1, 2, \tag{31}$$

where y_i^k is the value function target of the agent and can be defined as:

$$y_i^k = s^k + \gamma \times \Big[\pi_{\theta_i^\pi}(a_i^k)\min_{j=1,2}\left(Q_{i-,j}(s^{k+1} \mid \delta_{i-,j}^Q)\right) - \alpha \log\left(\pi_{\theta_i^\pi}\left(a_i^{k+1} \mid s^{k+1}\right)\right)\Big], \tag{32}$$

where $Q_{i-,j}(\cdot \mid \delta_{i-,j}^Q)$ represents the j-th target critic function of agent i, and γ is the discount factor.

The parameters of each actor network are updated in line 24, by minimizing the loss function $L_{\pi i}$ of actor i from the transition.

$$E\left[\pi_{\theta_i^\pi}\left(a_i^k\right)\left[\alpha \log \pi_{\theta_i^\pi}\left(a_i^k \mid o_i^k\right) - Q_{i,j}\left(s^k \mid \theta_{i,j}^Q\right)\right]\right], \tag{33}$$

where a suitable α has a prodigious impact on our scheme's performance, which can be adjusted by the algorithm itself rather than by hand.

The parameters of each α coefficient are updated in line 25, according to [16], [17], by minimizing the loss function $L_{\alpha i}$:

$$L_{\alpha i} = \pi_{\theta_i^\pi}(a_i^k)[-\alpha(\log \pi_{\theta_i^\pi}\left(a_i^k \mid o_i^k\right) + \bar{H})]. \tag{34}$$

Finally, following three successive updates, we have reached a point where we can effectively implement soft updates on the parameters of the target networks, specifically in line 26. By employing the soft update method, we are able to achieve a relatively smoother estimation of $Q_{i-,j}(s^{k+1} \mid \delta_{i-,j}^Q)$, thereby contributing to the stabilization of our scheme.

Based on the analysis presented, the time complexity of Algorithm 2 can be expressed as $O(ETM)$, as it involves carrying out three types of loops and multiplying their respective lengths. Additionally, the space complexity can be expressed as $O(M(I+J))$, since there are M IoHT points and $I+J$ ESs.

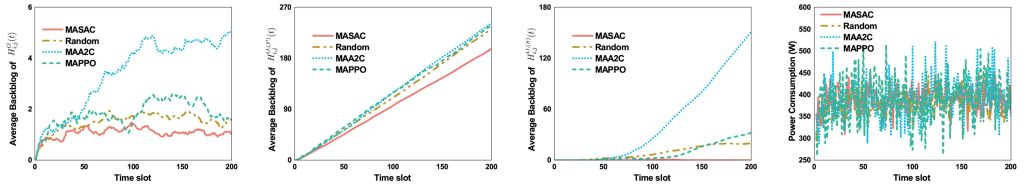
Algorithm 2: MASAC-Discrete.

1: **for** IoHT point i in $\mathcal{U}$ **do**
2: Initialize actor network $\pi_i(\cdot)$, critic network $Q_i(\cdot)$ with parameters θ_i^π and θ_i^Q;
3: Initialize target network $Q_{i-}(\cdot)$ with parameters θ_{i-}^Q;
4: Initialize experience replay buffer B_i;
5: **end for**
6: **for** episode from 1 to E **do**
7: Initialize state $s(t)$
8: Initialize observation $o_i(t)$ for IoHT point in $\mathcal{U}$
9: **for** t from 1 to T **do**
10: **for** IoHT point i in $\mathcal{U}$ **do**
11: Get action $a_i(t)$ according to $a_i(t) = \pi_i(o_i(t))$
12: Get reward $r_i^k(t)$ according to (30)
13: Get observation $o_i(t+1)$
14: **end for**
15: Get joint action $a(t)$ by concatenate $a_i(t)$ together
16: Get state $s(t+1)$ by concatenate $o_i(t+1)$ together
17: **for** IoHT point i in $\mathcal{U}$ **do**
18: Store transition $\{s(t), a_i(t), s^k(t), s(t+1)\}$ into
19: experience replay buffer B_i
20: **if** training process begins **then**
21: Sample a mini-batch of M transitions from B_i
22: Update critic network according to: $\theta_{i,j}^Q \leftarrow \theta_{i,j}^Q - lr^c \nabla_{\theta_{i,j}^Q} L_{Q_{i,j}}(\theta_{i,j}^Q)$, $j = 1, 2$
23: Update actor network according to: $\theta_i^\pi \leftarrow \theta_i^\pi - lr^a \nabla_{\theta_i^\pi} L_{\pi i}(\theta_i^\pi)$
24: Update temperature according to: $\theta_i^\alpha \leftarrow \theta_i^\alpha - lr^\alpha \nabla_{\alpha_i} L_{\alpha i}(\alpha_i)$
25: Update target networks according to: $\theta_{i,j-}^Q \leftarrow r\theta_{i-}^Q + (1-r)\theta_{i,j-}^Q$, $j = 1, 2$.
26: **end if**
27: **end for**
28: **end for**
29: **end for**

V. Performance Evaluation

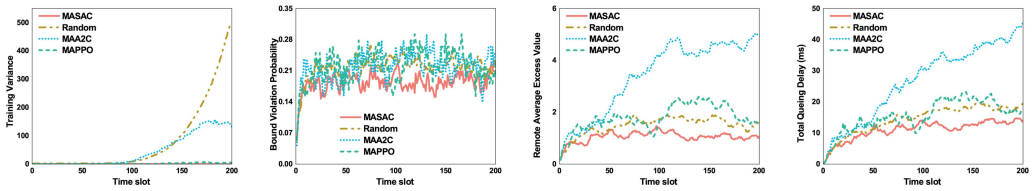
A. Parameter Setting

We maintain a fixed number of 2 BS with larger computation resources and 4 BS with smaller ones in our system. The available computational resources of ESs in the BS for IoHT points fluctuate irregularly within a limited range during 200 time slots with 0.1 s intervals. Specifically, we set the number of IoHT points to 6, and the data arrival rate $a_i(t)$ varies within the interval [7.5, 8.5] Mbits/s. We configure the transmission power to 20 dBm, sub-channel bandwidth to 1 MHz, and channel gain to 3.4×10^{-12}. The data is randomly generated based on the formulations outlined in Section III. Our simulations are conducted on both laptops equipped with NVIDIA 4 G GeForce RTX 3050 and a workstation equipped with 11 G GeForce RTX 2080 Ti. The parameter values mentioned above, along with others, are summarized in Table III. To evaluate the performance of our algorithms, we adjust the number of IoHT points (Scene I), computation density (Scene λ), the length of data arrival rates interval (Scene A) and the sub-bandwidth interval (Scene W).



(a) $H_{i,j}(t)$ (b) $H_{i,j}^{O,(P)}(t)$ (c) $H_{i,j}^{O,(S)}(t)$ (d) Power consumption of BS(w)

Fig. 3. Comparison of $H_{i,j}(t)$, $H_{i,j}^{O,(P)}(t)$ and $H_{i,j}^{O,(S)}(t)$ and power consumption of BS(w) over time slots.



(a) Training Variance (b) Bound Violation Probability (c) Remote Average Excess Value (d) Total Queuing Delay

Fig. 4. TV, BVP, RAEV and TQD over time slots.

TABLE III
SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
N	6	τ	0.1s
I	4	J	2
T	200	N_0	-118dBm
$g_{i,j}(t)$	3.5×10^{-12}	$A_i(t)$	[7.5, 8.5] Mbits/s
d_i^L	0.01	ϵ_i^L	0.1
$\sigma_i^{L,th}$	2.5 Mbits	$\xi_i^{L,th}$	0.2
P	20 dBm	W	[0.8, 1.2] MHz
$d_{i,j}^O$	0.1	$\epsilon_{i,j}^O$	0.01
$\sigma_{i,j}^{O,th}$	3 Mbits	$\xi_{i,j}^{O,th}$	0.2
λ_j	1000 $cycles/bit$	κ	$10^{-27} W \cdot s^3/cycle^3$
$f_{1,max}(t)$	[11.8, 13.8] GHz	$f_{2,max}(t)$	[10.6, 12.6] GHz
$f_{3,max}(t)$	[8.8, 9.8] GHz	$f_{4,max}(t)$	[8.6, 9.6] GHz
$f_{5,max}(t)$	[8.5,9.5] GHz	$f_{6,max}(t)$	[9.2, 8.2] GHz

TABLE IV
THE AVERAGE REDUCTION OF MASAC-DISCRETE OVER OTHER THREE ALGORITHMS OVER TIME SLOT

Indexes	Sub-indexes	Random	MAA2C	MAPPO
Remote Queues	$H_{i,j}(t)$	28.61%	68.78%	41.07%
	$H_{i,j}^{O,(P)}(t)$	21.08%	20.65%	21.64%
	$H_{i,j}^{O,(S)}(t)$	98.70%	99.85%	98.62%
	RAEV	28.58%	69.25%	47.31%
System Performance	TQD	23.24%	58.52%	28.98%
	P	-2.27%	0.23%	0.07%
	TV	99.74%	99.54%	87.66%
	BVP	24.31%	24.31%	25.00%

B. Baselines

In this study, we compare our proposed MASAC-discrete with three other algorithms, namely, Random, MAA2C, and MAPPO, the latter two are both fine-tuned Actor-Critic-based DRL algorithms. When considering the computation resources, they all use the Algorithm 1.

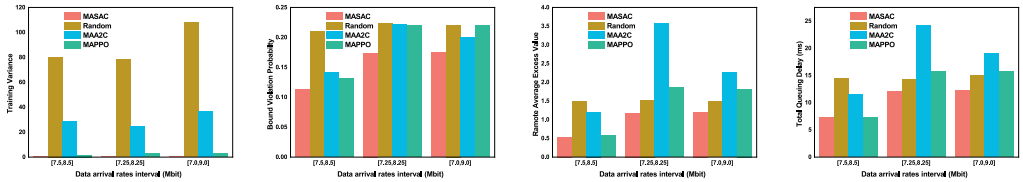
- *Random:* Each IoHT point randomly chooses its offloading decisions without knowing the state and decisions of other IoHT points.
- *MAPPO [46]:* It is similar to our proposed algorithm, MASAC-discrete, as it employs the CTDE structure and interacting PPO algorithms between IoHT points.
- *MAA2C:* It combines the CTDE structure with widely-used Advantage Actor-Critic (A2C).

C. Temporal Characteristics of the System

We compare MASAC-discrete with three other algorithms from the long-term perspective of remote queues and system performance. The corresponding results can be seen in Figs. 3 and 4, and a detailed numerical analysis is provided in Table IV.

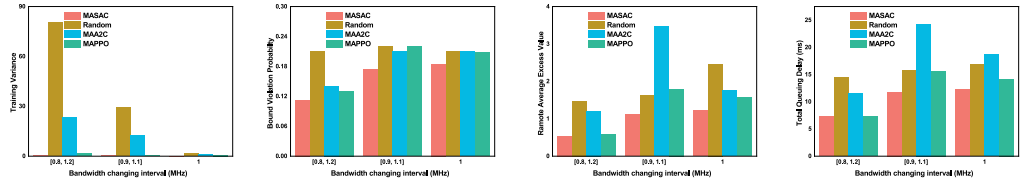
For remote queues, the MASAC-discrete is effective in reducing $H_{i,j}(t)$, with a minimum reduction of 28.61% and a maximum reduction of 68.78%. This indicates a significant reduction in congestion during the transmission process. Our MASAC-discrete algorithm demonstrates superior performance in reducing $H_{i,j}^{O,(P)}(t)$, achieving reductions of 21.08%, 20.65% and 21.64% for Random, MAA2C and MAPPO, respectively. As for $H_{i,j}^{O,(S)}(t)$, MASAC-discrete outperforms the other three algorithms by 90%. This indicates that MASAC-discrete effectively reduces the extent of bound violation as well as transmission congestion and probability of bound violation.

In terms of system performance, it is evident that MASAC achieves reductions of 23.24%, 58.52% and 28.98% in the total queuing delay (TQD), which is defined as the combined value of the local queuing delay and the remote queuing delay, when compared to the Random, MAA2C, and MAPPO approaches, respectively. MASAC also exhibits a remarkable reduction in training variance (TV), exceeding 85% compared to the other three algorithms. Furthermore, MASAC-discrete achieves substantial improvements in remote average excess value (RAEV) compared to Random, MAA2C, and MAPPO, with lifts of 28.58%, 69.25%, and 47.31% respectively. It is noteworthy that MASAC-discrete outperforms all three algorithms by more than 20% in terms of bound violation probability (BVP), highlighting its ability to effectively reduce the occurrence of extreme events and their impact.



(a) Training Variance (b) Bound Violation Probability (c) Remote Average Excess Value (d) Total Queuing Delay

Fig. 5. TV, BVP, RAEV and TQD over time slots with different data arrival rates interval.



(a) Training Variance (b) Bound Violation Probability (c) Remote Average Excess Value (d) Total Queuing Delay

Fig. 6. TV, BVP, RAEV and TQD over time slots with different bandwidth changing intervals.

It is noticeable that the widely-used MAPPO algorithm struggles to effectively search within the large action spaces in our system. Additionally, the performance of MAA2C falls short of expectations when compared to Random. However, our algorithm outperforms Random with superior stability and efficiency. This success can be attributed to our algorithm's ability to explore a diverse range of action policy choices compared to Random. By optimizing both entropy and rewards over a long period.

D. Impact of Data Arrival Rates

We narrow the data arrival rates interval gradually, while their mean remains constant. As depicted in Fig. 5, the wider the data arrival rates interval, roughly the worse the performance for all four algorithms. MASAC exhibits significant reductions compared to the other four algorithms in these four indexes. This indicates that MASAC-discrete can perform still well even when the data arrival rates change. The outstanding impact is observed in BVP, TQD, RAEV, and especially TV, where all reductions exceed 19%, and the top is more than 60%, suggesting that MASAC can reduce the probability and extent of bound violation occurrence however the data arrival rates change. Similarly, our algorithm is the most stable compared to the other three algorithms.

E. Impact of Sub-Channel Bandwidth

We gradually narrow the sub-channel bandwidth interval from [0.8, 1.2] to [0.9, 1.1] Mbits, ultimately settling on a channel bandwidth of 1 Mbit. As depicted in Fig. 6, the wider the sub-channel bandwidth interval, the higher the training variance. Our algorithm remains the stablest. our scheme exhibits significant improvements compared to other four algorithms across all three performance indexes. The outstanding impact is observed in BVP by 19.04%, 17.79%, and 18.99% compared to Random, MAA2C, and MAPPO, respectively. In TQD, the reduction compared to Random, MAA2C and MAPPO is 25.29%, 29.81% and 22.40%, respectively. In RAEV, the reduction exceeds 33%. All these three comparisons confirm that MASAC can effectively reduce the probability and extent of bound violation occurrence, regardless of changes in data arrival rates.

TABLE V
THE AVERAGE REDUCTION OF MASAC-DISCRETE OVER OTHER THREE ALGORITHMS UNDER DIFFERENT SCENARIOS

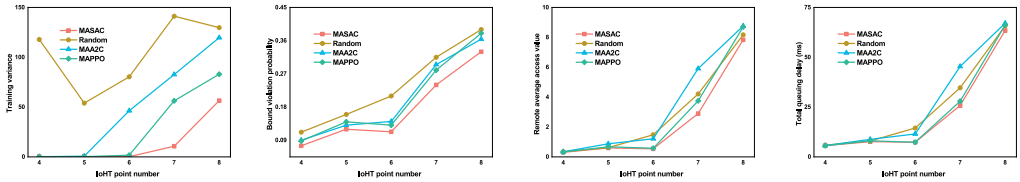
Index	Scene	Random	MAA2C	MAPPO
BVP	A	22.37%	19.48%	22.12%
	M	24.31%	15.47%	16.30%
	λ	15.47%	19.82%	16.57%
	W	19.04%	17.79%	18.99%
TQD	A	19.00%	29.84%	24.76%
	M	12.06%	13.18%	10.06%
	λ	12.46%	24.53%	17.25%
	W	25.29%	29.81%	22.40%
RAEV	A	23.19%	60.94%	37.21%
	M	15.75%	34.81%	18.79%
	λ	16.02%	48.88%	32.60%
	W	36.68%	51.47%	33.79%
TV	A	99.40%	98.26%	79.13%
	M	89.63%	54.36%	31.80%
	λ	94.04%	77.84%	46.55%
	W	95.44%	98.54%	53.47%

TABLE VI
COMPARISON OF ALGORITHM 1 WITH (35) AND (36)

RAA	RAEV	BVP	TQD	TV	PV	P	$H_{i,j}(t)$	$H_{i,j}^{O_i(P)}(t)$	$H_{i,j}^{O_i(S)}(t)$
Eq. (35)	1.951	0.402	17.778	0.185	516.507	336.179	1.981	253.981	0.000
Eq. (36)	1.653	0.427	15.774	0.166	510.482	353.292	1.687	243.515	0.000
ours	1.057	0.165	11.101	0.211	269.030	388.062	1.068	96.444	0.115

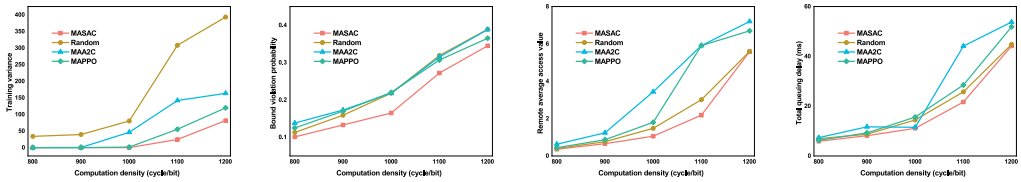
F. Impact of IoHT Points

As illustrated in Fig. 7 and Table V, with the increase in the number of IoHT points, all these algorithms demonstrate a decline in performance within the system due to the diminished availability of resources for each IoHT point. However, it is remarkable that MASAC-discrete continues to exhibit strong performance. For instance, when compared to the other three algorithms, MASAC-discrete achieves a reduction of over 10% in TQD, and over 15% in both BVP and RAEV. In terms of TV, MASAC-discrete only experiences a decrease of 54.36% and



(a) Training Variance (b) Bound Violation Probability (c) Remote Average Excess Value (d) Total Queuing Delay

Fig. 7. TV, BVP, RAEV and TQD over time slots with different IoHT points.



(a) Training Variance (b) Bound Violation Probability (c) Remote Average Excess Value (d) Total Queuing Delay

Fig. 8. TV, BVP, RAEV and TQD over time slots with different computation density.

31.80% when compared to MAA2C and MAPPO, respectively. We attribute this success to the fact that as the number of IoHT points increases, the state spaces and action complexity expand significantly, thereby making the search process more challenging.

G. Impact of Computation Density

As depicted in Fig. 8 and Table V, we observe that as the computation density increases, the remote congestion ($H_{i,j}(t)$) decreases. This is because the BSs process less data transmitted from IoHT points, resulting in a decrease in the model's performance. Specifically, when compared to the Random, MAA2C, and MAPPO approaches, our proposed model exhibits a reduction of 15.47%, 19.82%, and 16.57%, respectively, in terms of BVP. For TQD, the reduction is 12.46%, 24.53%, and 17.25%, respectively. In terms of RAEV, the reduction is 16.02%, 48.88%, and 32.60%, respectively. These results highlight the superior performance of our proposed model in mitigating congestion and improving various performance metrics compared to the alternative approaches under consideration.

H. Impact of Resource Allocation Algorithm (RAA)

We conduct a comparative study between Algorithm 1, and two alternative algorithms for CPU cycle frequency allocation. The selection of these alternative algorithms will be based on the ratio of non-zero elements present in the matrix $H_{i,j}(t)$, which can be determined as follows:

$$f_{i,j}(t) = \left[\frac{\mathbb{I}\{H_{i,j}(t) > 0\}}{\sum_{i=1}^{N} \mathbb{I}\{H_{i,j}(t) > 0\}} \right] f_{j,\max}, \tag{35}$$

or allocate the resources proportionally by $H_{i,j}(t)$ as follows:

$$f_{i,j}(t) = \left[\frac{H_{i,j}(t)}{\sum_{i=1}^{N} H_{i,j}(t)} \right] f_{j,\max}. \tag{36}$$

As depicted in Table VI, when employing the easily understood RAA, both the BVP and $H_{i,j}^{O,(P)}$ increase by approximately 2.5 times. This suggests a significant increase in the occurrence of bound violation events. Additionally, the excess value at the queue (RAEV) increases by 84.58% and 56.38% according to (35) and (36), respectively. Moreover, Algorithm 2 demonstrates a remarkable improvement in TQD, reducing it by 60.01% and 42.09% compared to (35) and (36), respectively, Furthermore, $H_{i,j}(t)$ is reduced by 85.49% and 57.96% compared to (35) and (36), respectively. Although Algorithm 2 can incur a little instability for the entire system, and power consumption has raised by 15.43% and 9.84%, our Power Variance (PV) has significantly decreased by 52.09% and 52.67% compared to (35) and (36), respectively. This indicates that Algorithm 2 can significantly improve the system performances.

VI. Conclusion

In this article, we propose a novel MASACDUA scheme specifically designed for IoHT scenarios, which leverages the MASAC-discrete algorithms to effectively tackle task offloading problems while considering the constraints imposed by URLLC requirements. Extensive simulation results demonstrate that MASACDUA yields substantial reductions in remote congestion $H_{i,j}(t)$, RAEV, TQD, and BVP, while maintaining the same level of data throughput and power consumption in BSs. These findings serve as compelling evidence for the efficiency of the MASACDUA scheme in enhancing system performance within IoHT scenarios. Furthermore, a comparative evaluation of our scheme across four fluctuated scenarios highlights the superior performance and robustness of MASACDUA in handling varying conditions and further validates its effectiveness as compared to alternative methods.

References

[1] Z. Lian, Q. Zeng, W. Wang, T. R. Gadekallu, and C. Su, "Blockchain-based two-stage federated learning with non-IID data in IoMT system," *IEEE Trans. Computat. Social Syst.*, 2022, early access, Nov. 21, 2022, doi: 10.1109/TCSS.2022.3216802.

[2] A. Khamparia, D. Gupta, V. H. C. de Albuquerque, A. K. Sangaiah, and R. H. Jhaveri, "Internet of health things-driven deep learning system for detection and classification of cervical cells using transfer learning," *J. Supercomputing*, vol. 76, no. 11, pp. 8590–8608, Jan. 2020.

[3] G. Yenduri, R. Kaluri, T. R. Gadekallu, M. Mahmud, and D. J. Brown, "Blockchain for software maintainability in healthcare," in *Proc. 24th Int. Conf. Distrib. Comput. Netw.*, 2023, pp. 420–424.

[4] A. R. Javed, M. U. Sarwar, M. O. Beg, M. Asim, T. Baker, and H. Tawfik, "A collaborative healthcare framework for shared healthcare plan with ambient intelligence," *Hum.-Centric Comput. Inf. Sci.*, vol. 10, pp. 1–21, 2020.

[5] R. A. Haraty, B. Boukhari, and S. Kaddoura, "An effective hash-based assessment and recovery algorithm for healthcare systems," *Arabian J. Sci. Eng.*, vol. 47, pp. 1523–1536, 2021.

[6] M. Rizwan et al., "Risk monitoring strategy for confidentiality of healthcare information," *Comput. Elect. Eng.*, vol. 100, 2022, Art. no. 107833.

[7] A. Basharat, M. M. B. Mohamad, and A. Khan, "Machine learning techniques for intrusion detection in smart healthcare systems: A comparative analysis," in *Proc. IEEE 4th Int. Conf. Smart Sensors Appl.*, 2022, pp. 29–33.

[8] M. R. Naqvi, M. Aslam, M. W. Iqbal, S. Khuram Shahzad, M. Malik, and M. U. Tahir, "Study of block chain and its impact on Internet of Health Things (IoHT): Challenges and opportunities," in *Proc. IEEE Int. Congr. Hum.- Comput. Interact., Optim. Robot. Appl.*, 2020, pp. 1–6.

[9] M. K. Hasan et al., "Fischer linear discrimination and quadratic discrimination analysis–based data mining technique for Internet of Things framework for healthcare," *Front. Public Health*, vol. 9, 2021, Art. no. 1354.

[10] T. M. Ghazal, M. K. Hasan, S. N. H. Abdullah, K. A. Abubakkar, and M. A. Afifi, "IoMT-enabled fusion-based model to predict posture for smart healthcare systems," *Comput. Mater. Continua*, vol. 71, no. 2, pp. 2579–2597, 2022.

[11] B. Han, R. H. Jhaveri, H. Wang, D. Qiao, and J. Du, "Application of robust zero-watermarking scheme based on federated learning for securing the healthcare data," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 2, pp. 804–813, Feb. 2023.

[12] J. Yao and N. Ansari, "Task allocation in fog-aided mobile IoT by Lyapunov online reinforcement learning," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 2, pp. 556–565, Jun. 2020.

[13] Z. Zhou et al., "Learning-based URLLC-aware task offloading for Internet of Health Things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 396–410, Feb. 2021.

[14] S. Coles, J. Bawa, L. Trenner, and P. Dorazio, *An Introduction to Statistical Modeling of Extreme Values*, vol. 208. Berlin, Germany: Springer, 2001.

[15] Y. Yang and J. Wang, "An overview of multi-agent reinforcement learning from game theoretical perspective," 2020, *arXiv:2011.00583*.

[16] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[17] T. Haarnoja et al., "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.

[18] P. Christodoulou, "Soft actor-critic for discrete action settings," 2019, *arXiv:1910.07207*.

[19] H. Wu, J. Chen, T. N. Nguyen, and H. Tang, "Lyapunov-guided delay-aware energy efficient offloading in IIoT-MEC systems," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 2117–2128, Feb. 2023.

[20] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synth. Lectures Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010.

[21] H. Materwala and L. Ismail, "Energy-aware edge-cloud computation offloading for smart connected health," in *Proc. IEEE 8th Int. Conf. Future Internet Things Cloud*, 2021, pp. 144–150.

[22] M. Mukherjee et al., "Delay-sensitive and priority-aware task offloading for edge computing-assisted healthcare services," in *Proc. IEEE Glob. Commun. Conf.*, 2020, pp. 1–5.

[23] J. Ren, J. Li, H. Liu, and T. Qin, "Task offloading strategy with emergency handling and blockchain security in SDN-empowered and fog-assisted healthcare IoT," *Tsinghua Sci. Technol.*, vol. 27, no. 4, pp. 760–776, 2022.

[24] J. Chen, C. Feng, D. Feng, and S. Meng, "Multi-edge computing offloading for ultra-reliable and low-latency communication," in *Proc. IEEE 13th Int. Conf. Wireless Commun. Signal Process.*2021, pp. 1–6.

[25] D. Van Huynh et al., "URLLC edge networks with joint optimal user association, task offloading and resource allocation: A digital twin approach," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7669–7682, Nov. 2022.

[26] J. Wang, Y. Ma, N. Yi, and R. Tafazolli, "On URLLC downlink transmission modes for MEC task offloading," in *Proc. IEEE 91st Veh. Technol. Conf.*, 2020, pp. 1–5.

[27] H. Liao et al., "Learning-based intent-aware task offloading for air-ground integrated vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5127–5139, Aug. 2021.

[28] C. Pan et al., "Asynchronous federated deep reinforcement learning-based URLLC-aware computation offloading in space-assisted vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7377–7389, Jul. 2023.

[29] Z. Li, M. Xu, J. Nie, J. Kang, W. Chen, and S. Xie, "NOMA-enabled cooperative computation offloading for blockchain-empowered Internet of Things: A learning approach," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2364–2378, Feb. 2021.

[30] J. Yun, Y. Goh, W. Yoo, and J.-M. Chung, "5G multi-RAT URLLC and eMBB dynamic task offloading with MEC resource allocation using distributed deep reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20733–20749, Oct. 2022.

[31] A. M. Seid, G. O. Boateng, B. Mareri, G. Sun, and W. Jiang, "Multi-agent DRL for task offloading and resource allocation in multi-UAV enabled IoT edge network," *IEEE Trans. Netw. Serv. Manage.*, vol. 18, no. 4, pp. 4531–4547, Dec. 2021.

[32] A. Gao, Q. Wang, W. Liang, and Z. Ding, "Game combined multi-agent reinforcement learning approach for UAV assisted offloading," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 12888–12901, Dec. 2021.

[33] Y. Jia, C. Zhang, Y. Huang, and W. Zhang, "Lyapunov optimization based mobile edge computing for Internet of Vehicles systems," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7418–7433, Nov. 2022.

[34] Z. Wang, Y. Zhang, C. Yin, and Z. Huang, "Multi-agent deep reinforcement learning based on maximum entropy," in *Proc. IEEE 4th Adv. Inf. Manage., Communicates, Electron. Automat. Control Conf.*, 2021, pp. 1402–1406.

[35] D. Wu, T. Liu, Z. Li, T. Tang, and R. Wang, "Delay-aware edge-terminal collaboration in green internet of vehicles: A multiagent soft actor-critic approach," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 2, pp. 1090–1102, Jun. 2023.

[36] X. Wu, X. Li, J. Li, P. C. Ching, V. C. M. Leung, and H. V. Poor, "Caching transient content for IoT sensing: Multi-agent soft actor-critic," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5886–5901, Sep. 2021.

[37] Y. Yan, B. Zhang, C. Li, and C. Su, "Cooperative caching and fetching in D2D communications — A fully decentralized multi-agent reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16095–16109, Dec. 2020.

[38] C.-F. Liu, M. Bennis, M. Debbah, and H. V. Poor, "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4132–4150, Jun. 2019.

[39] X. Yuan et al., "A DQN-based frame aggregation and task offloading approach for edge-enabled IoMT," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 3, pp. 1339–1351, May/Jun. 2023.

[40] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "BEdge-Health: A decentralized architecture for edge-based IoMT networks using blockchain," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11743–11757, Jul. 2021.

[41] M. T. Khan, L. Barik, A. Adholiya, S. S. Patra, A. N. Brahma, and R. K. Barik, "Task offloading scheme for latency sensitive tasks in 5G IoHT on fog assisted cloud computing environment," in *Proc. IEEE 3rd Int. Conf. Emerg. Technol.*, 2022, pp. 1–5.

[42] Y. Qiu, H. Zhang, and K. Long, "Computation offloading and wireless resource management for healthcare monitoring in fog-computing-based Internet of Medical Things," *IEEE Internet Things J.*, vol. 8, no. 21, pp. 15875–15883, Nov. 2021.

[43] P. Lin, Q. Song, F. R. Yu, D. Wang, and L. Guo, "Task offloading for wireless VR-enabled medical treatment with blockchain security using collective reinforcement learning," *IEEE Internet Things J.*, vol. 8, no. 21, pp. 15749–15761, Nov. 2021.

[44] Z. Wang et al., "Energy-aware and URLLC-aware task offloading for Internet of Health Things," in *Proc. IEEE Glob. Commun. Conf.*, 2020, pp. 1–6.

[45] S. M. Ross, "Introduction to probability models," *Technometrics*, vol. 40, pp. 78–78, 1975.

[46] C. Yu et al., "The surprising effectiveness of PPO in cooperative multi-agent games," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2022, pp. 24611–24624.