Deep Reinforcement Learning for Delay Minimization in MEC-THz Networks with Finite Blocklength codes

Arian Ahmadi, Anders Høst-Madsen and Zixiang Xiong

2024 IEEE Globecom Workshops (GC Wkshps) | 979-8-3315-0567-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/GCWkshp64532.2024.11101710

***Abstract*—Communication at terahertz (THz) frequency bands is a promising solution for achieving extremely high data rates in 6G wireless networks while supporting the emerging Ultra-Reliable Low-Latency Communications (URLLC). To support the strict QoS requirements, it is preferable to deploy a multi-access edge computing (MEC) technology in the URLLC systems to improve the latency and reliability performance of wireless communication. Meanwhile, finite blocklength (FBL) coding has been established as a powerful technique to substantially enhance various QoS metrics for URLLC by enabling short-packet communications. However, guaranteeing ultra-reliable and low-latency MEC (URLL MEC) operating in FBL regime is very challenging due to uncertainties of wireless links, limited communications and computing resources, as well as dynamic network traffic. In this work, we investigate the combined offloading decisions, blocklength optimization, and computation and power allocation in a URLL MEC network operating with FBL codes, where the short-packet technique is adopted to meet the stringent QoS requirements of delay-sensitive computing services. Given the formulated problem is a mixed-integer non-linear programming, an innovative deep reinforcement learning (DRL) solution, utilizing the double deep Q-network framework is proposed to effectively enhance the optimization of offloading strategies within the MEC network. Then, by leveraging the optimized offloading decisions, the cost for each agent is determined through the optimization of either transmit power and blocklength or local computing resources. Simulation results show that the proposed DRL scheme can effectively reduce the system delivery delay in the MEC network and outperform the baseline approaches.**

I. Introduction

The forthcoming sixth-generation (6G) wireless cellular network supports a variety of new applications with ultra-high reliability and low latency (URLL) service requirements [1], [2]. Among these emerging applications include, connected and autonomous vehicles (CAVs), extended reality (XR), ivideo-driven machine–human interaction with strict quality-of-service (QoS) requirements on the end-to-end (E2E) delay (e.g., 1 ms) and reliability (e.g., 10^{-8} packet loss probability). Hence, ultra-reliable and low-latency communication (URLLC), which inherently incorporates reliability and latency as key factors in network design, stands as a prominent use case within 6G networks. To meet strict QoS requirements and achieve URLLC, multi-access edge computing (MEC) is an attractive solution to significantly reduce service delay by letting the user equipments (UEs) to offload their tasks to the base stations (BSs) in the network rather than the remote cloud center [3], [4].

This research was supported under Grants EECS-1923751 and EECS-1923803.

A. Ahmadi and A. Høst-Madsen are with the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, HI 96822 USA (Email: aahmadi, ahm@hawaii.edu).

Z. Xiong is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (Email: zx@ece.tamu.edu).

While promising, maintaining consistent performance in a MEC-assisted network presents significant challenges due to the inherent instability of wireless channel, stochastic task arrival, heterogeneity of edge computing servers and computing tasks. Therefore the performance of a MEC-assisted network not only depends on the delay violation probability but also is influenced by the decoding error probability. Specifically, to facilitate URLLC, it is necessary to have a relatively short coding blocklength. This means that the underpinning theory of finite blocklength (FBL) coding needs to be considered [5], [6]. In contrast to the infinite blocklength regime, where the blocklength is considered unlimited, the FBL regime involves a blocklength that is so short that data transmission can no longer be arbitrarily reliable [7]. This results in a trade-off when determining the optimal coding blocklength within the decision strategy. Therefore, it is necessary to determine the proper offloading decisions and recourse allocation for a MEC-assisted network in FBL regime to guarantee the system performance, including throughput, latency, and reliability.

In the literature, the majority of researches [8]–[12] investigate the MEC-assisted networks with joint communication and computation design. In [9], the authors propose a novel scheme to optimize the reliability of the MEC-assisted network by considering the distribution of E2E service delay, encompassing over-the-air transmission and edge computing delay. The authors in [8] present a multi-task learning based feed forward neural network model to achieve an optimal computation offloading strategy for the MEC-assisted network. The authors in [10] investigate user association problem by formulating the offloading delay for the MEC-assisted network with decoupled uplink/downlink association. The power allocation for delivery delay reduction in a device-to-device (D2D) enabled MEC scenario is studied in [11]. The authors in [12] propose a novel joint communication and computation load balancing scheme as well as resource allocations to minimize the E2E delay in the MEC-assisted internet-of-thing (IoT) network.

In the other line of research, the body of work in [13]–[20] investigates several new methods based on deep neural networks (DNNs), deep reinforcement learning (DRL) and federated learning (FL) to optimize the performance of the MEC-assisted networks. The authors in [14] present a novel resource allocation scheme using multi-agent DRL in a vehicle-to-vehicle (V2V) communication network. In [15], the authors propose a DRL-based computation and communication resource allocation algorithm in a MEC-assisted railway IoT network.

The problem of joint power allocation and resource allocation for URLL communications in a vehicular network is proposed in [17]. The authors present a new distributed algorithm based on FL to estimate the tail distribution of the queue lengths. In [18], the authors investigate the problem of jointly optimizing resource and learning performance to reduce communication costs and improve learning performance in wireless FL systems. The authors in [20] focus on optimizing various aspects of FL, including weight compression, convergence analysis and iteration reduction over IoT networks.

While current schemes can effectively enhance the URLLC performance of MEC-assisted networks through appropriate radio resource allocation, the performance of MEC-assisted networks under finite blocklength (FBL) regime remains limited [21]–[23]. For example, in [22], the authors investigate the computation and power allocation problem for non-orthogonal multiple access MEC networks in the FBL regime. In fact, they propose a novel analytical scheme to approximate the average overall block error probability in finite blocklength regime. However, the impact of edge computing on task transmission time is not considered. Furthermore, the significant computational complexity introduced by the iterative methods can affect delay performance. This motivates us to devise a time-efficient resource management approach for MEC-assisted networks utilizing DRL.

In this paper, we investigate the resource allocation problem in the MEC-assisted network considering the FBL regime and aim to minimize the system delivery delay under both communication and computing constraints. The main contributions of our paper are summarized as follows:

- Instead of Shannon rate, we adopt the short packet coding rate to more accurately capture the rate loss in the FBL regime [5], [6], [24].
- We propose a novel scheme to optimize the reliability of the MEC-assisted network by minimizing the system delivery delay under both the wireless network and edge computing servers constraints.
- We formulate the multi-objective problem that aims to minimize the long-term delivery delay of a CAV in a decentralized manner by efficiently allocating offloading decisions, computation resources, transmit powers, and blocklength to CAVs across the radio access network (RAN).
- Given the proposed problem is a mixed integer non-linear programming (MINLP), a new solution based on DRL is proposed to jointly address the offloading decision problem, computing resource assignment, transmit power allocation and blocklength optimization across the MEC-assisted network. To this end, we cast our problem as a multi-agent DRL problem and solve it using double deep Q-network (DDQN).
- Comprehensive simulations are performed to show the superiority of the proposed algorithm through comparisons with those existing schemes for a network of CAVs as a case study.

The rest of the paper is organized as follows. Section II presents the system model. Section III and IV describe the problem formulation and the proposed solution. Simulation results are provided in Section V and conclusions are presented in Section VI.

II. System Model

Consider a single-cell multi-user MEC-assisted network consisting of a set $M \in \mathcal{M}$ CAVs and a BS as shown in Fig. 1. We consider a slotted orthogonal frequency division multiple access (OFDMA) transmission scheme where the length of a time slot is T. We also assume that each CAV m has its own dedicated sub-channel of bandwidth ω. Due to the stringent delay requirement of the CAVs, T is considered to be rather small to reduce the transmission delay. Next, we explain the transmission and computing latencies in details.

A. Over-the-Air Transmission Delay

Considering the OFDMA mechanism, interference is ignored due to the exclusive allocation of subcarriers. If the CAV m decides to offload its task, it should first transmit it to the BS through wireless channels. Due to the wireless fading channel, some packets may not be decoded successfully at the BS, hence, re-transmission is needed. In this paper, we use a basic automatic repeat request (ARQ) transmission scheme in which a packet is retransmitted until the receiver acknowledges successful decoding. Since packets are short, we use Polyanskiy's FBL coding rate [5], [24]

$$k = nC(\gamma) - \sqrt{nV(\gamma)}Q^{-1}(\epsilon) + \frac{1}{2}\log n + O\left(1\right). \quad (1)$$

where γ is the SNR and $C\left(\gamma\right) = \omega \log_2(1+\gamma)$ is the channel capacity, k is the length of the packet and $V\left(\gamma\right)$ is channel dispersion given as follows

$$V(\gamma) = \omega^2 \left(1 - \frac{1}{(1+\gamma)^2}\right) \log_2^2 e. \quad (2)$$

We use this as an approximation by ignoring the $O(1)$ term. The SNR between CAV m and the BS for the j-th transmission, $\gamma_{m,j}$, is given by

$$\gamma_{m,j} = \frac{G_m P_m h_{m,j}}{\sigma^2}, \quad (3)$$

where P_m, G_m and σ^2 denote, respectively, the transmit power of CAV m, the antenna gain of CAV m and the noise power. In addition, $h_{m,j}$ represent the Rayleigh fading channel gain for the j-th transmission between CAV m and the BS. The channel gain $h_{m,j}$ is considered flat-fading over the bandwidth ω_m and constant during the transmission of one block. We allow the code rate to depend on the SNR. Since k is fixed between re-transmissions, we allow n to be variable through a function $n(\gamma)$. Then the error probability at the j-th transmission for the m-th CAV is

$$\epsilon_{m,j} = Q\left(\frac{n\left(\gamma_{m,j}\right)C\left(\gamma_{m,j}\right) + 0.5\log_2 n\left(\gamma_{m,j}\right) - k}{\sqrt{n\left(\gamma_{m,j}\right)V\left(\gamma_{m,j}\right)}}\right), \quad (4)$$

The re-transmission continue until no error happen
total number of transmission is therefore a random
The transmission delay for CAV m is then given by

$$\tau_m^{air} = \sum_{j=1}^{J} n\left(\gamma_{m,j}\right),$$

The aim is to find the optimum $n_m(\gamma)$ to mini
However, as the end goal is to minimize the syste
delay, $n_m(\gamma)$ will be found through a total system op
see Section III.

B. Computing Delay at Edge Computing Servers

Computing delay refers to the time needed for
a task at an edge computing server within the ME
network. The execution time of a task depends on the
processed and the tasks submitted, therefore, it can b
as a random variable. For instance, the executio
performing the object detection highly depends on
or level of details in the captured images, as well as th
processing resources of the edge server. Since only idle MEC server is considered for once scheduling, the waiting time for queue is not involved.

Let f_{max} denote the maximum computing-cycle frequency of edge processor for the BS. If CAV m offloads its task to the edge server the computation delay would be $\tau_m^{comp} = \frac{c_m}{f_m^{edge}}$, where c_m and f_m^{edge} denote the CPU cycle requirement of the task and average computation capacity of edge server, respectively.

In addition, if CAV m decides to process its task locally, the delay for local computation would vary based on the computation resources allocated for task processing, denoted by f_m^{loc}. Therefore, the local delay for the CAV is modeled as $\tau_m^{loc} = \frac{c_m}{f_m^{loc}}$.

It's important to note that higher resource utilization, whether in terms of transmit power or computation capacity, decreases the system delivery delay at the cost of reducing the blocklength $n(\gamma)$, but increasing the probability of error $\epsilon_{m,j}$ and the number of ARQ retransmissions. Hence, managing this trade-off requires careful handling through efficient offloading decision-making and precise optimization of both local computation and offloading strategies.

III. MULTI-OBJECTIVE PROBLEM FORMULATION

The main goal of this paper is to minimize the delay of completing task by jointly accounting the offloading decision strategy, computing and power allocation and the blocklength of each CAV over a specified time horizon, $\mathcal{T}$. The long-term expected delay for each CAV m, is formulated as follows:

$$\tau_m\left(\mathbf{n}_m(\gamma), \mathbf{P}_m, \mathbf{f}_m^{loc}, \mathbf{x}_m, \mathbf{y}_m\right) = \mathbb{E}\left[\lim_{t\to\infty} \frac{1}{t}\sum_{i=0}^{t}(x_{m,i}\tau_{m,i}^{loc} + y_{m,i}(\tau_{m,i}^{comp} + \tau_{m,i}^{air})\right], \quad (6)$$

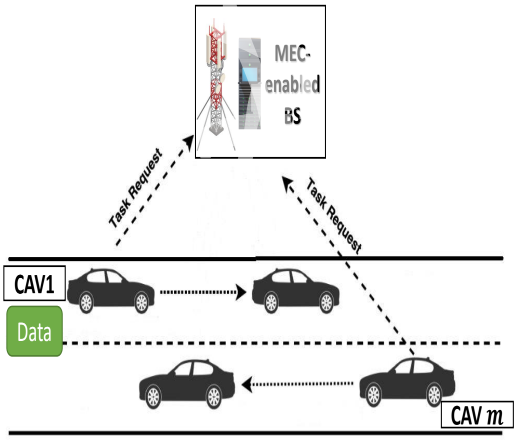

Fig. 1. System Model.

where $\mathbf{n}_m(\gamma)$, $\mathbf{P}_m$, $\mathbf{f}_m^{loc}$, $\mathbf{x}_m$ and $\mathbf{y}_m$ represent the vectors of blocklength, transmit powers, local computing resource allocation, edge server computation resource allocation, local computing and edge offloading decision of CAV m, respectively.

For any given CAV m, we model the decision-making problem (DMP), expressed as:

$$\begin{aligned} &\min_{\mathbf{n}_m(\gamma), \mathbf{P}_m, \mathbf{f}_m^{loc}, \mathbf{x}_m, \mathbf{y}_m} \tau_m \\ &\text{subject to:} \\ &\mathrm{C1}: \tau_m \le \tau_{\text{th}}, \\ &\mathrm{C2}: f_m^{loc} \le F_{\max,m}, \\ &\mathrm{C3}: x_m + y_m = 1, \qquad \forall m \in \mathcal{M}, \\ &\mathrm{C4}: x_m, y_m \in \{0,1\}, \qquad \forall m \in \mathcal{M}. \end{aligned} \quad (7)$$

Constraint C1 expresses that total time for processing the task must meet the maximum delay threshold determined by QoS requirement. Constraints C2 indicates the local computation capacity with the maximum threshold $F_{max,m}$. The constraints C3 and C4 indicate that the decision variables of the problem are all binary indicators.

The proposed optimization problem in (7) is a MINLP, hence, it is difficult to solve. Next, we develop a new efficient algorithm to solve this problem.

IV. PROPOSED DDQN ALGORITHM

When addressing the joint optimization problem in multi-user scenarios, various challenges come to light.

- The significant mobility observed in CAVs results in frequent and unpredictable shifts within the communication channel. This dynamic variability poses a considerable challenge for CAVs, impeding their ability to effectively make optimal real-time decisions.
- Due to constrained resources, the decision made by each individual CAV holds a consequential influence on the selection and actions of other CAVs within the network. Meaning that optimal decision-making by one CAV not only affects its own resource allocation and performance

Algorithm 1 Proposed DRL for Joint Offloading Decision and Computing and Communication Resource Allocation

Input: $\mathcal{M}$, $t = 0$

Output: $\mathbf{n}(\gamma), \mathbf{P}, \mathbf{f}^{loc} \mathbf{x}, \mathbf{y}$

1: Initialize the online and target networks for each agent m.
2: **while** (maximum number of iterations is not reached) **do**
3: $\quad \phi_m^{\text{online}} = \phi_m^{\text{target}}$,.
4: $\quad$ **for** each CAV m in $\mathcal{M}$ **do**
5: $\quad\quad$ **for** each time step t if $|\mathcal{I}_{m,t}| > 0$ **do**
6: $\quad\quad\quad$ Compute the offloading decision, $\mathbf{x}$ and $\mathbf{y}$, using ϕ_m^{online},
7: $\quad\quad\quad$ Interact with environment and solve (8) or (9) applying KKT conditions,
8: $\quad\quad\quad$ Save the experience $\mathbf{n}(\gamma), \mathbf{P}$ and $\mathbf{f}^{loc}$ in replay memory $\mathcal{O}_{m,t}$,
9: $\quad\quad\quad$ Calculate $\epsilon_{m,j}$ from (4),
10: $\quad\quad\quad$ **if** $\mathbb{P}(1 - \epsilon_{m,j}) = 1$ **then**
11: $\quad\quad\quad\quad$ Train the local model on $\mathcal{O}_{m,t}$,
12: $\quad\quad\quad\quad$ Transmit ϕ_m^{online} to the PS,
13: $\quad\quad\quad$ **else**
14: $\quad\quad\quad\quad$ return to step 7,
15: $\quad\quad\quad$ **end if**
16: $\quad\quad$ **end for**
17: $\quad$ **end for**
18: **end while**

but also ripples through the network, influencing the decisions and performance of other CAVs.

- As the number of CAVs grows in the network, the offloading decision complexity increases exponentially. This escalating complexity underscores the need for scalable and efficient approaches to address the evolving demands of CAV networks.

To address the issues mentioned above, we propose a multi-agent DDQN algorithm to solve the secure offloading and computing and communication resource allocation problem. The details of the algorithm are elaborated in the following section.

A. *Sketch of double deep Q-network*

Given the aforementioned challenges, employing traditional optimization methods to address the dynamic optimization problem described in equation (7) is not feasible. Model-free DRL is a useful tool for handling the DMP and learning the optimal solutions in dynamic environments. Hence, the DMP is formulated as a Markov Decision Process (MDP). Particularly, we model our problem as a multi-agent DDQN problem. For each CAV (DRL agent), we have following components:

- *State space:* the state space for each agent m, denoted by s_m, is defined as
$s_m = \{\mathcal{I}_m, h_{m,j}\}$, where $\mathcal{I}_m$ is the length of the task queue of CAV m.
- *Action space:* The action space of agents, denoted by $\mathcal{A}$, includes the offloading decisions, blocklengths, computing frequencies, and transmit powers.
- *Cost function:* The objective function defined in (7) depends on the value of $\mathbf{n}(\gamma)$ and P_m when offloading to edge server and f_m^{loc} if local computation is selected. Hence, in order to accurately capture the benefits of a specific offloading decision within the cost function, it is imperative to carefully optimize these variables. Therefore, when CAV m performs its task locally, i.e., $x_m = 1$, the cost would be calculated by solving the following optimization problem:

$$\min_{f_m^{loc}} \tau_m^{loc} \quad \text{subject to: C1, C2.} \tag{8}$$

If CAV m offloads its task to the edge server, i.e., $y_m = 1$, the blocklength, the transmit power and computing frequency at the edge server would be optimized by solving the following optimization problem:

$$\min_{\mathbf{n}_m(\gamma), P_m} \tau_m^{comp} + \tau_m^{air} \quad \text{subject to: C1.} \tag{9}$$

TABLE I
SIMULATION PARAMETERS

Notation	Parameter	Value
M	Number of CAVs	30
P_m	Transmit power of a CAV	10 mW
G_m	Antenna gains	1
f_{max}	Computing frequency	30 GHz
N_0	Noise power spectral density	-204 dBm/Hz [25]
ω	Total system bandwidth	3 GHz [25]
k	Packet length	256 bits
τ_{th}	Service latency requirement	10-100 ms

By modeling the cost function as an optimization problem, we can not only optimize local resource utilization and enforce system constraints, but also provide the agent with an accurate estimation of the quality of offloading decisions. It's important to note that both equations (8) and (9) represent convex optimization problems with respect to the variables f_m^{loc} and $\mathbf{n}_m(\gamma)$, P_m, respectively. These types of optimization problems can be effectively solved using standard software tools, e.g., Karush-Kuhn-Tucker (KKT) conditions. After finding the optimal computing and communication resource allocation vectors, $\mathbf{f}_m^{loc}$, $\mathbf{P}_m$ as well as the blocklenght vector $\mathbf{n}_m(\gamma)$, first we calculate the packet error probability, $\epsilon_{m,j}$, defined in (4). If $\mathbb{P}(1 - \epsilon_{m,j}) = 1$, we feed the optimal computing and communication resource allocation vectors into the DDQN framework as the immediate cost function.

B. *DDQN training phase*

Consider the immediate cost of each CAV m obtained from the solution of the (8) and (9) as $R_m(s, a)$. Using Bellman equation, the action-state value is:

$$Q_m(s,a) = R_m(s,a) + \gamma \sum_{s' \in \mathcal{S}} W_{ss'}(a) \max_{a' \in \mathcal{A}} Q_m^*(s', a'), \tag{10}$$

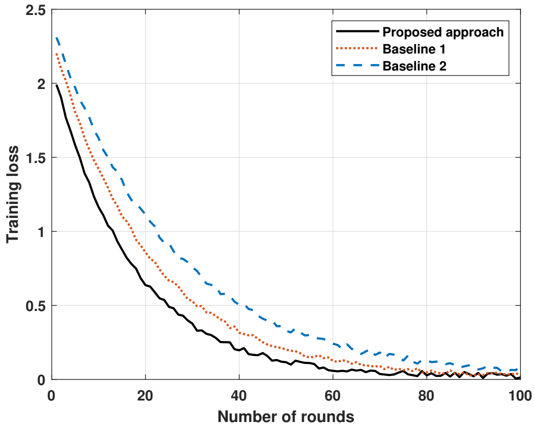

Fig. 2. Training loss versus the number of rounds.

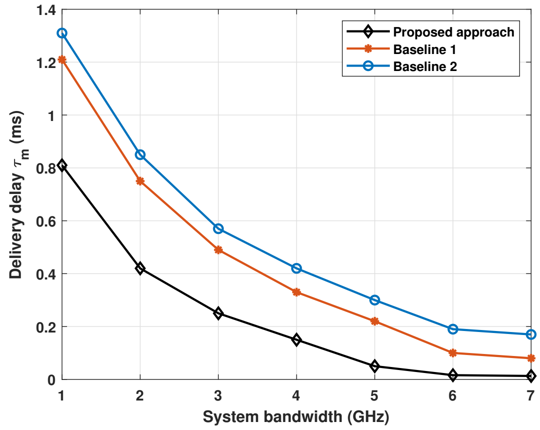

Fig. 3. Delivery delay versus the bandwidths.

where $\mathcal{S}, W_{ss'}(a)$, and $0 < \gamma < 1$ are the set of states, the transition probability function, and the discount factor, respectively. To avoid the need for a complete model of the environment, eliminate the calculation of the transition probability function, and achieve a more stable learning process, we employ DDQN in this work. In particular, we adopt DDQN to find a solution to maximize the state-action function $Q_m^*(s', a')$. Each agent m has two neural networks working alongside each other, one called *online network* with parameters ϕ_m^{online} and the other called *target network* with parameters ϕ_m^{target}. At each training iteration the target value for training the online network in device m is calculated as:

$$Z_m = R_m(s,a) + \gamma Q_m(s', \max_{a' \in \mathcal{A}} Q_m^*(s', a'; \phi_m^{\text{online}}), \phi_m^{\text{target}}) \tag{11}$$

The proposed algorithm is summarized in Algorithm 1.

V. Simulation Results

In this section, we evaluate the performance of the proposed scheme for delay optimization in the MEC-assisted network shown in Fig. 1. The performance was evaluated by averaging the results over sufficiently large Monte Carlo runs. We compare the performance of the proposed method with two baseline approaches. The first baseline approach, hereinafter referred to as "Baseline 1", uses DQN scheme to solve the proposed problem. The second baseline, hereinafter referred to as "Baseline 2", uses random offloading decision scheme with the proposed computing and power allocation. Simulation parameters are summarized in Table I.

The convergence curve of the proposed DRL method is shown in Fig. 2. As shown in Fig. 2, we observe that the loss function decreases rapidly for all three methods, showing the fast convergence of the proposed scheme as well as two other baselines. The results show that the proposed scheme successfully converges within reasonably small number of epochs and outperforms two other baseline schemes.

Figure. 3 illustrates how the delivery delay is affected by the allocation of different bandwidth. As the bandwidth increases, the delivery delay decreases due to the fact that each CAV

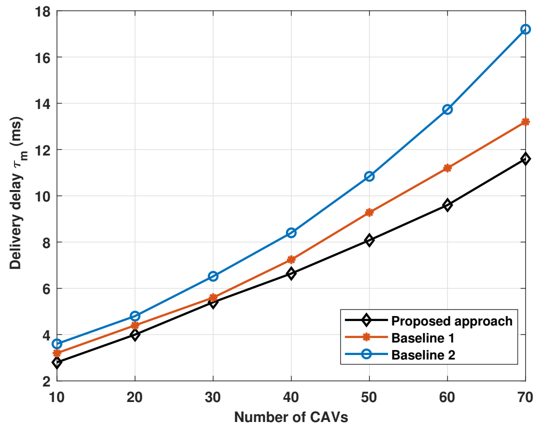

Fig. 4. Delivery delay versus the number of CAVs.

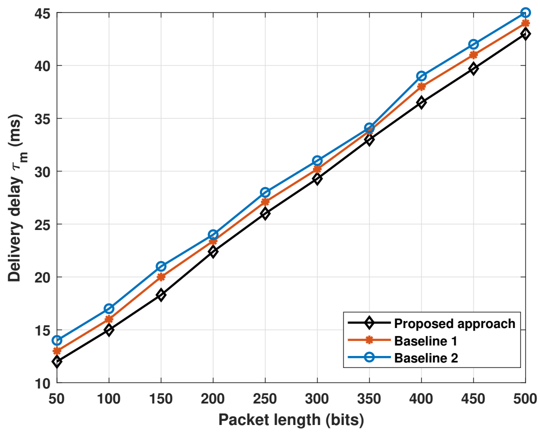

Fig. 5. Delivery delay versus the packet length.

requires to dedicate small amount of power to achieve higher transmission rates. Therefore, the delivery delay is reduced. The results in Fig. 3 indicate the superior performance of the presented scheme compared to the baseline methods. For instance, in a MEC-assisted network with assigned bandwidth

ω = 4 GHz, the performance gains yielded by the proposed algorithm are up to 40% and 55%, respectively, compared to baseline methods 1 and 2.

In Fig. 4, the delivery delay versus the network size is shown for the proposed scheme and the two baseline methods. It is clear that the delivery delay increases as more CAVs exist in the network. The results in Fig. 4 highlights that the proposed scheme can yield up to 13% and 28% performance gain, when M = 50, compared to the baseline 1 and 2, respectively. Furthermore, Fig. 4 also shows the scalability of the proposed method. For example, with the delivery delay of 12 ms, the proposed scheme can support up to 70 CAVs, which is 11% and 32% higher compared to baselines 1 and 2, respectively.

Figure 5 presents how the delivery delay varies with the packet length k. As Fig. 5 shows the delivery delay increases as packet length increases. This is because for a larger packet, the additional time for the transmission, propagation, queuing, and processing is required. The results in Fig. 5 also show that for the same packet length, the proposed DRL scheme can achieve the minimum delivery delay compared with two other baselines. For instance, with the packet length of 300 bits, the performance gains yielded by the proposed algorithm are up to 3% and 6% , respectively, compared to baselines 1 and 2, respectively.

VI. Conclusion

In this paper, we considered a URLL MEC-based 6G THz CAV network with ARQ schemes operating in the FBL regime. We aim to minimize the delivery delay by optimally selecting the offloading decisions, transmit power, computing recourse allocation and blocklength of each CAV while fulfilling the delay budget requirement. To solve the proposed MINLP, we developed a new algorithm that adopted DDQN. More specifically, DDQN is used to determine the offloading decisions of the CAVs. Given the offloading decisions, the computing capacity, transmit power and blocklength of the CAVs is optimized to minimize the delivery delay. Then, we feed the results into the DDQN framework as the immediate cost function to optimize the offloading decisions. Simulation results have confirmed the effectiveness of the proposed scheme to those comparative algorithms.

References

[1] A. Ahmadi and O. Semiari, "Reinforcement learning for optimized beam training in multi-hop terahertz communications," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[2] A. Ahmadi, *Machine Learning for Latency and Reliability Optimization in Next-Generation Wireless Networks*, Ph.D. thesis, University of Colorado Colorado Springs, 2023.

[3] Q. Pham et al., "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020.

[4] A. Ahmadi, A. Høst-Madsen, and Z. Xiong, "Latency and energy minimization in NOMA-assisted MEC network: A federated deep reinforcement learning approach," *arXiv preprint arXiv:2405.04012*, 2024.

[5] Y. Polyanskiy, H V. Poor, and Sergio Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[6] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, "Quasi-static multiple-antenna fading channels at finite blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4232–4265, 2014.

[7] S. S Kowshik, K. Andreev, A. Frolov, and Y. Polyanskiy, "Energy efficient coded random access for the wireless uplink," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4694–4708, 2020.

[8] B. Yang, X. Cao, J. Bassey, X. Li, and L. Qian, "Computation offloading in multi-access edge computing: A multi-task learning approach," *IEEE transactions on mobile computing*, vol. 20, no. 9, pp. 2745–2762, 2020.

[9] A. Ahmadi, O. Semiari, M. Bennis, and M. Debbah, "Variational autoencoders for reliability optimization in multi-access edge computing networks," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2022, pp. 752–757.

[10] A. Al-Shuwaili and A. Lawey, "Latency reduction for mobile edge computing in hetnets by uplink and downlink decoupled access," *IEEE Wireless Communications Letters*, vol. 10, no. 10, pp. 2205–2209, 2021.

[11] U. Saleem, Y. Liu, S. Jangsher, X. Tao, and Y. Li, "Latency minimization for d2d-enabled partial computation offloading in mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4472–4486, 2020.

[12] Y. Sun, J. Xu, and Sh. Cui, "User association and resource allocation for mec-enabled iot networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 10, pp. 8051–8062, 2022.

[13] E. Mustafa, J. Shuja, K. Bilal, S. Mustafa, T. Maqsood, F. Rehman, and A. Khan, "Reinforcement learning for intelligent online computation offloading in wireless powered edge networks," *Cluster Computing*, vol. 26, no. 2, pp. 1053–1062, 2023.

[14] X. Li, L. Lu, W. Ni, A. Jamalipour, D. Zhang, and H. Du, "Federated multi-agent deep reinforcement learning for resource allocation of vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 8, pp. 8810–8824, 2022.

[15] J. Xu, B. Ai, L. Chen, Y. Cui, and N. Wang, "Deep reinforcement learning for computation and communication resource allocation in multiaccess MEC assisted railway IoT networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23797–23808, 2022.

[16] A. Taleb Zadeh Kasgari, W. Saad, M. Mozaffari, and H. V. Poor, "Experienced deep reinforcement learning with generative adversarial networks (gans) for model-free ultra reliable low latency communication," *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 884–899, 2020.

[17] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, 2019.

[18] H. Chen, Sh. Huang, D. Zhang, M. Xiao, M. Skoglund, and H. V. Poor, "Federated learning over wireless iot networks with optimized communication and resources," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16592–16605, 2022.

[19] J. Pei, Sh. Li, Zh. Yu, L. Ho, W. Liu, and L. Wang, "Federated learning encounters 6g wireless communication in the scenario of internet of things," *IEEE Communications Standards Magazine*, vol. 7, no. 1, pp. 94–100, 2023.

[20] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in iot," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5986–5994, 2019.

[21] J. Jalali, A. Khalili, R. Berkvens, and J. Famaey, "Joint offloading policy and resource allocation in irs-aided MEC for IoT users with short packet transmission," in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*. IEEE, 2023, pp. 1–7.

[22] H. Tang, B. He, Y. Zhou, L. Yang, and J. Chen, "Joint computation and power allocation for NOMA enabled MEC networks in the finite blocklength regime," *IET Communications*, vol. 15, no. 20, pp. 2552–2563, 2021.

[23] Y. Yang, Y. Hu, and M. Gursoy, "Reliability-optimal designs in MEC networks with finite blocklength codes and outdated CSI," in *2021 17th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2021, pp. 1–6.

[24] A. Fengler, A. Lancho, and Y. Polyanskiy, "Coded orthogonal modulation for the multi-antenna MAC," in *2023 12th International Symposium on Topics in Coding (ISTC)*. IEEE, 2023, pp. 1–5.

[25] A. Ekti, A. Boyaci, et al., "Statistical modeling of propagation channels for terahertz band," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2017, pp. 275–280.