DRQN-based Task Offloading in UAV-assisted Mobile Edge Computing Environments with Hidden Channel Conditions

Taewon Song
Department of Internet of Things
Soonchunhyang University
Asan, Korea
twsong@sch.ac.kr

***Abstract*—Mobile edge computing (MEC) with unmanned aerial vehicle (UAV) assistance offers promising solutions for reducing latency in 5G networks. However, uncertainty in task processing requirements poses challenges for efficient task offloading. This paper introduces a deep recurrent Q-network (DRQN) based algorithm for task offloading in UAV-assisted MEC environments with hidden channel conditions. We formulate the problem as a partially observable Markov decision process (POMDP), where the UAV decides whether to process tasks locally or offload them to a cloud server without complete information about the channel state. Our approach leverages temporal dependencies in the environment to make informed decisions under uncertainty. Simulations demonstrate that our DRQN-based algorithm outperforms previous methods, including DQN-based, offload-only, and local-only strategies. This work contributes to the development of robust task offloading strategies for dynamic edge computing environments in 5G and beyond networks.**

***Index Terms*—5G mobile networks, task offloading, POMDP, DRQN, mobile edge computing, network softwarization**

I. Introduction

The advent of 5G technology has increased demand for efficient wireless networks, with mobile edge computing (MEC) emerging as a crucial solution for reducing latency [1]. In remote areas, unmanned aerial vehicles (UAVs) assist MEC systems by acting as "flying gateways", offering flexible deployment and wide coverage [2], [3]. Researches on UAV-assisted MEC have focused on trajectory design [4], [5], resource management [6], and computation offloading [5], [7]. Recent works have applied deep reinforcement learning (DRL) to optimize these processes [8]. In addition, our previous work introduced an opportunistic task offloading algorithm (OTO-MEC) based on deep Q-network (DQN), showing significant improvements over traditional methods [9].

However, a critical challenge remains: uncertainty in channel conditions between the UAV and the cloud server. To address this, we propose PORTO-MEC, a partially observable recurrent task offloading algorithm. We formulate the problem as a partially observable Markov decision process (POMDP) and construct a deep recurrent Q-network (DRQN) model to make informed offloading decisions under uncertainty of channel conditions.

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1F1A1076069), and in part by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2021RIS-004).

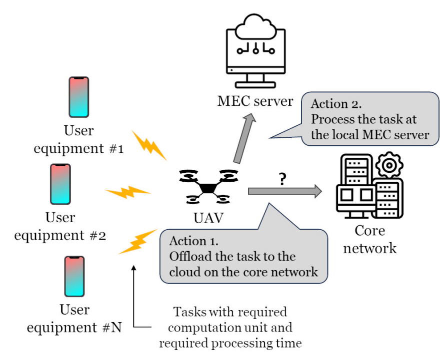

Fig. 1. System illustration for PORTO-MEC. UE has a number of tasks that consist of required computation units and required processing times.

II. System Description and MDP Formulation

A. System and Channel Model

Fig.1 illustrates a UAV-assisted MEC system where multiple UEs send tasks to a UAV for processing. We model the wireless channel between the UAV and the core network using a two-state Gilbert-Elliot model [10], [11], with $\mathbf{C} = 0, 1$ representing bad and good states. Although UAV-MEC and UAV-UE connections are assumed to be stable, the channel state of UAV-core network is unknown so that the link quality may affect cloud offloading efficiency.

B. Markov Decision Process Formulation

The MDP model is an appropriate mathematical decision-making framework. In this subsection, we present an MDP model for offloading decision.

1) State Space: The state space $\mathbf{S}$ is defined as a combination of several components:

$$\mathbf{S} = \mathbf{F} \times \mathbf{T} \times \mathbf{U} \times \mathbf{N}, \tag{1}$$

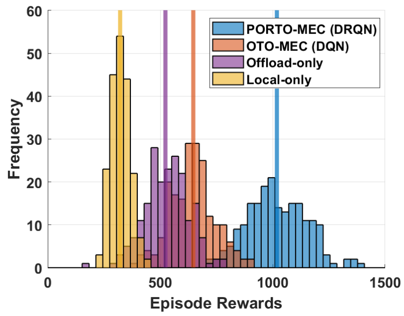

Fig. 2. Histograms of PORTO-MEC and comparison algorithms. Each algorithm is tested for 200 different episodes. Vertical bars represent the average rewards for each algorithm.

where $\mathbf{F}$ represents available computation units, $\mathbf{T}$ denotes remaining epochs, $\mathbf{U}$ describes queued tasks, and $\mathbf{N}$ represents the next candidate task. $\mathbf{U}$ can be expressed as $\mathbf{U} = \prod_{i=1}^{L} \mathbf{U}_i$. Since tasks' required computation units and processing times are considered, $\mathbf{U}_i$ can be defined as $\mathbf{U}_i = \mathbf{U}_i^f \times \mathbf{U}_i^t$, where $\mathbf{U}_i^f$ and $\mathbf{U}_i^t$ are the required computation units and the required processing time for i-th entry of the queue, respectively. Similarly, $\mathbf{N}$ can also be expressed as $\mathbf{N} = \mathbf{N}^f \times \mathbf{N}^t$.

2) Action: The agent chooses action a to offload or process locally. We define the action space $\mathbf{A} = \{0, 1\}$, where 0 represents offload and 1 represents local processing. For offloading, the task is sent to the core network if the channel is good, otherwise it is currupted. For local processing, the task is sent to the MEC server.

3) Reward Function: The reward function is designed to maximize computational utilization of both cloud and MEC servers. It's defined as the computing units of completed tasks, with a weight applied to cloud processing to account for potential congestion and delays. The reward is calculated as:

$$R_t = \begin{cases} w * q_i^f, & \forall q_i^t = 0 \quad \text{with} \quad a = 0, \\ q_i^f, & \forall q_i^t = 0 \quad \text{with} \quad a = 1, \end{cases} \tag{2}$$

where $w(< 1)$ is the priorized weight when the task is processed at the cloud server.

III. Simulation Results

For performance analysis, we conduct extensive simulations with Python 3.10 along with Gymnasium, a standard API for reinforcement learning and PyTorch, an open source machine learning framework. We inherit the given gymnasium environment and create a custom environment to fit our scenario.

Fig. 2 shows histograms of the number of processed units for each algorithm. Before testing, the model is trained for 5000 episodes for each PORTO-MEC and OTO-MEC. PORTO-MEC outperforms the other comparative algorithms and the superior performance of PORTO-MEC can be attributed to its ability to handle the hidden channel conditions between the UAV and the core network. Unlike OTO-MEC, PORTO-MEC employs a DRQN to infer the hidden channel state from the sequence of observations and actions. This allows PORTO-MEC to make more informed decisions about task offloading, even when the exact channel quality is unknown.

Simulation results show that PORTO-MEC processes 58.82% higher rewards than OTO-MEC, 92.39% higher than local-only, and 213.51% higher than offload-only algorithms. This work contributes to more robust and efficient task offloading strategies for UAV-assisted MEC systems in 5G and beyond networks.

IV. Conclusion and Future Works

In this paper, we proposed PORTO-MEC, a partially observable recurrent task offloading algorithm for UAV-assisted MEC systems with hidden channel states. We formulated a POMDP considering available computing units, inferred channel conditions, and task information, and developed a DRQN-based approach to solve it. Performance analysis showed PORTO-MEC outperforms comparative algorithms in scenarios with hidden channel states. Future work will expand to multi-UAV scenarios, consider UAV power consumption, and explore federated learning applications in these partially observable environments.

References

[1] B. Ali, M. A. Gregory, and S. Li, "Multi-access edge computing architecture, data security and privacy: A review," *IEEE Access*, vol. 9, pp. 18 706–18 721, 2021.

[2] Z. Ning, H. Hu, X. Wang, L. Guo, S. Guo, G. Wang, and X. Gao, "Mobile Edge Computing and Machine Learning in The Internet of Unmanned Aerial Vehicles: A Survey," *ACM Computing Surveys*, 2023.

[3] M. H. Adnan, Z. A. Zukarnain, and O. A. Amodu, "Fundamental design aspects of UAV-enabled MEC systems: A review on models, challenges, and future opportunities," *Computer Science Review*, vol. 51, p. 100615, 2024.

[4] Q. Luo, T. H. Luan, W. Shi, and P. Fan, "Deep Reinforcement Learning Based Computation Offloading and Trajectory Planning for Multi-UAV Cooperative Target Search," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 2, pp. 504–520, 2023.

[5] Z. Gao, J. Fu, Z. Jing, Y. Dai, and L. Yang, "MOIPC-MAAC: Communication-Assisted Multi-Objective MARL for Trajectory Planning and Task Offloading in Multi-UAV Assisted MEC," *IEEE Internet of Things Journal*, 2024.

[6] W. Liu, B. Li, W. Xie, Y. Dai, and Z. Fei, "Energy efficient computation offloading in aerial edge networks with multi-agent cooperation," *IEEE Transactions on Wireless Communications*, vol. 22, no. 9, pp. 5725–5739, 2023.

[7] H. Guo, Y. Wang, J. Liu, and C. Liu, "Multi-UAV cooperative task offloading and resource allocation in 5G advanced and beyond," *IEEE Transactions on Wireless Communications*, vol. 23, no. 1, pp. 347–359, 2023.

[8] H. Lu, C. Gu, F. Luo, W. Ding, and X. Liu, "Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning," *Future Generation Computer Systems*, vol. 102, pp. 847–861, 2020.

[9] T. Song, "Opportunistic Task Offloading in UAV-assisted Mobile Edge Computing: A Deep Reinforcement Learning Approach," in *ICTC 2023*, 2023, pp. 881–884.

[10] Q. Zhang and S. Kassam, "Finite-state markov model for rayleigh fading channels," *IEEE Transactions on Communications*, vol. 47, pp. 1688–1692, 1999.

[11] H. Boujemaa, M. B. Said, and M. Siala, "Throughput Performance of ARQ and HARQ I Schemes over a Two-states Markov Channel Model," *2005 12th IEEE International Conference on Electronics, Circuits and Systems*, pp. 1–4, 12 2005.