LEO Satellite MEC Networks: Joint Offloading and Resource Allocation using Deep Reinforcement Learning

Thanh Phung Truong, 김재민, 허동현, 오준석, 원동욱, 경연웅* and 조성래
중앙대학교, *국립공주대학교
{tptruong, jmkim, dhhur, jsoh, dwwon}@uclab.re.kr, ywkyung@kongju.ac.kr, srcho@cau.ac.kr

Abstract

Low Earth Orbit (LEO) satellite networks integrated with Multi-Access Edge Computing (MEC) hold significant promise for meeting the diverse demands of resource-hungry applications and resource-constrained devices. This paper proposes an LEO-MEC system in which ground users partially offload their tasks to the satellites for execution. Here, we design a deep reinforcement learning framework for optimizing the offloading ratio, computing, and communication resources to reduce the task execution time in the system.

Ⅰ. Introduction

Due to the surge in IoT applications, especially those requiring low latency and high computation, meeting the needs of remote, disaster-stricken, airborne, and maritime users has become critical. Low Earth Orbit (LEO) satellite networks are emerging as vital complements to terrestrial networks, offering broad, reliable coverage globally [1]. Meanwhile, Multi-Access Edge Computing (MEC) has gained traction as a solution to enhance the capabilities of ground IoT devices [2]. Leveraging the proximity of LEO satellites, the propagation delay to visible satellites can be reduced, enabling efficient offloading of computations to MEC servers on satellites.

Effective resource management is vital to address the conflict between resource-demanding applications and resource-limited IoT devices. However, few studies have comprehensively considered communication, computing, and caching resources in LEO satellite MEC networks. The interdependence of communication, computing, and caching resource allocation makes joint optimization challenging. Moreover, optimizing the association between IoT devices and satellites requires careful consideration of computing capacity constraints and cache deployment stategies.

Motivated by this observation, this study investigates an LEO-MEC system. Here, tasks with large size from ground users are offloaded to the LEO satellite network equipped with a strong MEC server with enough capacity. In such a system, we formulate a joint offloading and resource allocation problem to minimize the total delay of ground users. To solve the problem, a deep reinforcement learning (DRL) approach is proposed to jointly design the offloading and resource allocation strategies.

Ⅱ. System Model

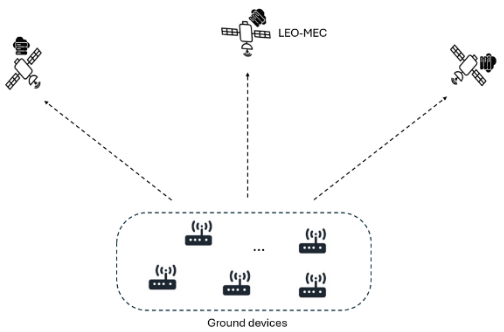

Figure 1. The LEO-MEC System Model

The system model considered in Fig. 1 consists of K ground users and S LEO-MECs. The LEO satellites fly around the earth on their otbit and the coverage of each satellite is seperated. The transmission rate from user k to satellite s can be calculated as

$$r_{k,s} = x_{k,s}\alpha_k B log_2\left(1 + \frac{p_k g_{k,s}}{\sigma^2}\right), \qquad (1)$$

where $x_{k,s}$ is the association variable, which is equal to 1 if user k is connected to the satellite s, otherwise, $x_{k,s} = 0$; $\alpha_{k,s}$ is the communication resource allocated to user k, B, p_k, $g_{k,s}$, and σ^2 denote the transmisison bandwidth, transmit power at user k, channel element between user k and satellite s, and Gaussian noise power, respectively.

At each time slot, user k has a task $\tau_k = (l_k, c_k)$ to be executed, where l_k and c_k are the data size in bits and the required computing resource of the task. Each task can be processed locally or partially offloaded to the LEO-MEC with the offloading ratio $o_k \in [0.1]$. Accordingly, the transmission time for offloading the task of user k to the satellites is calculated as

$$t_k^o = \frac{o_k l_k}{\sum_s r_{k,s}}. \qquad (2)$$

Accordingly, the time for executing the offloaded part at the LEO-MEC can be calculated as

$$t_k^m = \frac{o_k l_k}{f_{s,k}}, \tag{3}$$

where $f_{s,k}$ denotes the computing resource that satellite s allocated to user k. At the users, the time for executing the task locally can be calculated as

$$t_k^l = \frac{(1-o_k) l_k}{f_k}, \tag{4}$$

where f_k is the computation capacity of user k.

In this study, we aim to minimize the total delay for tasks execution by optimizing the task offloading ratios, communication and computing resources. Therefore, we formulate the following problem

$$\min_{\alpha_k,\ o_k,\ f_{s,k}} \textstyle\sum_k \left(t_k^o + t_k^m + t_k^l\right) \tag{5a}$$

$$\text{s.t.} \quad o_k \in [0,1], k \in \{1,2,..,K\}, \tag{5b}$$

$$\textstyle\sum_k \alpha_k = 1, k \in \{1,2,..,K\}, \tag{5c}$$

$$\textstyle\sum_k x_{k,s} f_{s,k} \le F_{max}, \tag{5d}$$

where constraint (5b) is the value range of the offloading ratios, (5c) indicates the amount of communication resource in the system, and (5d) is to ensure the computing capacity of the satellites.

Ⅲ. Proposed Method

To solve the problem, we propose a DRL framework based on the deep deterministic policy gradient (DDPG) algorithm. First, we transform the problem into a reinforcement learning model, which includes:

- *State*: The state space contains the task information, association status, and the channel gain between users and satellites, expressed as

$$s = \{\tau_k, x_{k,s}, g_{k,s} |_{k=\{1,2,\ldots,K\}}\}. \tag{6}$$

- *Action:* The action space contains the optimize variables, expressed as

$$a = \{\alpha_k, o_k, f_{s,k} |_{k=\{1,2,\ldots,K\}}\}. \tag{7}$$

- *Reward:* The reward function is defined as the negative value of the total time, expressed as

$$r = -\textstyle\sum_k \left(t_k^o + t_k^m + t_k^l\right). \tag{8}$$

The DDPG is applied to train the agent that decides action to interact with environment for maximizing the obtained reward. The algorithm has an actor network to decide actions, which is trained by using a gradient ascent function, expressed as

$$\nabla_{\theta^\mu} J = \nabla_{\theta^\mu} Q^{\theta^Q}\left(s, \mu^{\theta^\mu}(s)\right), \tag{9}$$

where $\mu^{\theta^\mu}()$ denotes the actor network with parameter θ^μ, $Q^{\theta^Q}()$ denotes the critic network with parameter θ^Q aims in evaluating the actions. The critic network is trained by minimizing the loss function expressed as

$$L = Q^{\theta^Q}(s,a) - y, \tag{10}$$

where $y = r + \gamma Q'^{\theta^{Q'}}(s', \mu'^{\theta^{\mu'}}(s'))$ is the target function calculated according to the target networks $Q'^{\theta^{Q'}}()$ and $\mu'^{\theta^{\mu'}}$, s' is the next state. The target networks are updated by a soft-update function with the coefficient ϵ, expressed as

$$\theta^{\mu'} \leftarrow (1-\epsilon)\theta^{\mu'} + \epsilon\theta^{\mu},$$

$$\theta^{Q'} \leftarrow (1-\epsilon)\theta^{Q'} + \epsilon\theta^{Q}. \tag{11}$$

Ⅲ. Conclusion

In this work, we considered an LEO-MEC system in which ground users partially offload their tasks to LEO satellites equipped with MEC servers. We formulated a problem of minimizing the total time for completing the users' tasks by optimizing the offloading ratios, communication resources for users, and computing resource allocation at satellites. To solve this problem, we proposed a DRL framework that applies the DDPG to train the agent to decide actions at each observation state.

ACKNOWLEDGMENT

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support programs(IITP-2024-RS-2022-00156353 and IITP-2024-RS-2023-00258639) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).

REFERENCES

[1] Jung, S., Lee, M. S., Kim, J., Yun, M. Y., Kim, J., & Kim, J. H., "Trustworthy handover in LEO satellite mobile networks," ICT Express 8.3 (2022): 432-437.

[2] Ranaweera, P., De Alwis, C., Jurcut, A. D., & Liyanage, M, "Realizing contact-less applications with multi-access edge computing," ICT Express 8.4 (2022): 575-587.