Mobility-aware Service Migration in MEC System

1st Tuan Phong Tran
Department of Information Communication Convergence Technology
Soongsil University
Seoul, Korea
phongtt@soongsil.ac.kr

2nd Myungsik Yoo
School of Electronic Engineering
Soongsil University
Seoul, Korea
myoo@ssu.ac.kr

***Abstract*—Multi-access edge computing (MEC) has emerged as an effective approach for enhancing system quality. Nevertheless, the movement of users and variations in demand for the service might lead to an increase in system delays. This article investigates the issue of service migration, with a particular focus on the factors of mobility and service availability. Specifically, we model the Markov Decision Process (MDP) problem. To make effective service migration decisions, we propose a deep reinforcement learning (DRL) model. Furthermore, a recurrent neural network (RNN) is implemented in order to enhance model performance by predicting user movement. The experimental results demonstrate the effectiveness of the proposed method in reducing the system delay.**

***Index Terms*—MEC, service migration, Markov decision process, Deep reinforcement learning, Recurrent neural network.**

I. Introduction

Multi-access Edge Computing (MEC) is proposed to provide services to resource-constrained devices. The deployment of services in close proximity to end users offers a potential solution to address system latency issues. However, due to the constantly changing demands of users, the requested services may not be implemented on the local MEC server. Hence, many studies [1], [2] have been conducted to enhance the efficiency of service migration inside the MEC system. Nevertheless, optimal models are formulated with the assumption that customers neither move nor neglect the variety of service types. This results in the approaches being challenging to implement in real-world environments.

In this study, a service migration framework is proposed for the MEC system. Specifically, we formulate an optimal model with the purpose of minimizing system latency. In contrast to other studies, the optimal model takes user mobility and the variety of service categories into account. In addition, we utilize a recurrent neural network (RNN) that predicts the future location using historical movement data of mobile users. The optimization problem is then transformed into an MDP model and solved using proposed DRL model. Finally, comprehensive experiments are performed to investigate the performance of the framework.

II. Problem formulation

A. System Model

The system model is illustrated in Figure 1. The system comprises a central cloud server and M MEC servers, represented as $\{1, ..., M\}$. The cloud server covers a comprehensive range of services, while each MEC server is constrained by resource limits and can only deploy a certain number of services. We denote the set of N users as $\{1, ..., N\}$, who are making requests to services that are deployed on MEC servers. It is noteworthy that these users can move between various coverage zones across multiple MEC servers. In situations where users attempt to change locations or request services that are inaccessible on MEC servers, the system is required to facilitate the migration of the requested service to the nearby MEC server. The cloud server has the responsibility of gathering comprehensive system information and making decisions on service migration.

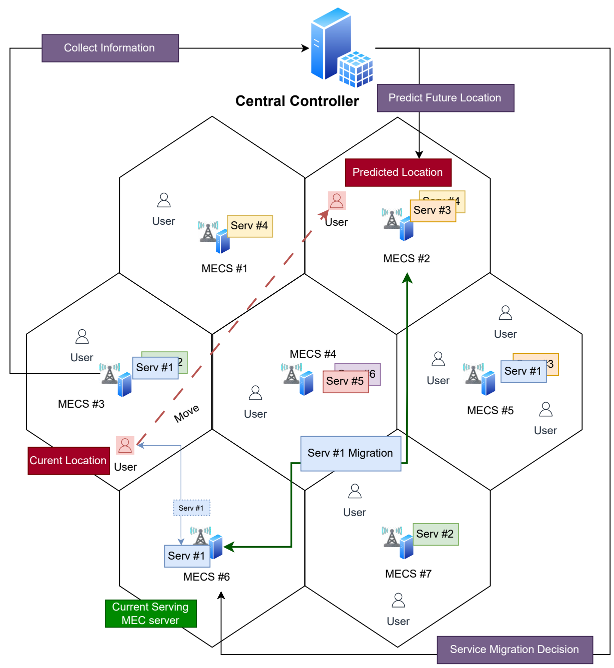

Fig. 1. Proposed MEC system

We investigate T time slots in this study. The set of K available services of MEC server m in time slot t is denoted as $S_m(t) = \{s_1^m(t), s_2^m(t), ..., s_K^m(t)\}$. In particular, service $s_k^m(t)$ is represented as a tuple

$$s_k^m(t) = \{a_k^m(t), b_k^m(t)\} \tag{1}$$

where $a_k^m(t)$ and $b_k^m(t)$ denote the type and size of service

s_k^m, respectively. On the other hand, user n generates only one request which is denoted as follows.

$$r_n(t) = \{s_n(t), d_n(t), c_n(t)\} \tag{2}$$

where $s_n(t)$ is the type of service, whereas $d_n(t)$ and $c_n(t)$ represent the size of data and computational requirement, respectively.

B. Computation Delay

The following equation is used to calculate the latency in servicing MEC server s caused by the processing request $r_n(t)$:

$$T_n^s(t) = \frac{d_n(t)c_n(t)}{f_n^s(t)}, \tag{3}$$

where $f_n^s(t)$ represents the allocated computing resources for the execution of request $r_n(t)$.

C. Communication Delay

We consider the transmission latency from both transmitting requests and migrating services between servers. Due to the negligible size of the response data, the delay induced by obtaining the request result is neglected.

MEC servers are accessed by users through a wireless connection. The wireless bandwidth between user n and local MEC server l are denoted as $B_{n,l}(t)$. The transmission power for user n and Gaussian noise power spectrum density are denoted as $P_n(t)$ and N_0, respectively. In time slot t, the wireless data rate $R_{n,l}(t)$ between the user n and local MEC servers l are formulated as follows [3]:

$$R_{n,l}(t) = B_{n,l}(t)\log_2\big(1 + \frac{P_n(t)G_{n,l}}{N_0 B_{n,l}(t)}\big), \tag{4}$$

where $G_{n,l}$ is the wireless channel gain [4].

In wired network, the link bandwidth is proportional to the data transmit rate. Therefore, for time slot t, the data rate $R_{l,s}(t)$ of the wired MEC-MEC connection is as follows:

$$R_{l,s}(t) = B_{l,s}(t), \tag{5}$$

where $B_{l,s}(t)$ denotes the allocated channel bandwidths for the MEC-MEC connections.

The communication delay from user n to local MEC server l is denoted by $T_{n,l}(t)$, while the wired communication delay between MEC servers is denoted by $T_{l,s}(t)$. The delay are determined in accordance with equations (4) and (5) as follows:

$$T_{n,l}(t) = \frac{d_n(t)}{R_{n,l}(t)}, \tag{6}$$

$$T_{l,s}(t) = \frac{d_n(t)}{R_{l,s}(t)}. \tag{7}$$

D. Migration Delay

When a user changes location, the system autonomously determines whether to move the service in order to minimize the duration of the service interruption. Nevertheless, moving services constantly has the potential to result in an increase in system latency. Hence, through the utilization of the services available from MEC servers, the system can effectively mitigate the redundant service migration. Hence, the latency resulting from the procedure of migrating a service from one MEC server to another is defined as follows:

$$T_{p,s}^m(t) = \begin{cases} 0 & \text{for} \quad d_n(t) \in S_s(t), \\ \frac{b_k^m(t)}{R_{p,s}(t)} & \text{otherwise}, \end{cases} \tag{8}$$

where $b_k^m(t)$ and $R_{p,s}(t)$ denote service size and the data rate of wired connection between the previous serving MEC server p to the new serving MEC s in time slot ts, respectively.

E. Problem formulation

The main goal of this research is to reduce the latency of the entire system, taking into account constraints on resources and time-sensitive requirements. The objective function is as follows:

$$\min \sum_{t=1}^{T}\sum_{n=1}^{N}\frac{1}{T}\frac{1}{N}\left[T_n^s(t) + T_{n,l}(t) + T_{l,s}(t) + T_{p,s}^m(t)\right]. \tag{9}$$

Additionally, the following constraints are presented:

$$\begin{aligned} C1: &\quad 1 \le s \le M, \\ C2: &\quad \sum_{d=1}^{N} B_{n,l} \le B_{max}, \\ C3: &\quad FA^s(t) \ge \sum_{n=1}^{N} f_n^s(t)\ , f_n^s(t) \ge 0, \end{aligned}$$

Constraint C1 ensures that only one of the MEC servers may be chosen by the migration decision. Constraint C2 states that the aggregate bandwidths between every user and the local MEC server l must not reach the utmost allowed. The computing resources that are currently available on the MEC server s are denoted as $FA^s(t)$. As mentioned in Constraint C3 [3], the total of all computing resources allocated to each task cannot surpass the current resource availability.

III. Methodology

This section presents a proposed methodology for tackling the service migration issue. First, an RNN model is used to predict the forthcoming user trajectory. Subsequently, the optimization issue is formulated as an MDP. Finally, the proximal policy optimization (PPO) framework [5] is introduced to make optimal decisions on service migration.

A. Mobility Prediction

A multi-input, multi-output recurrent neural network is utilized for the purpose of predicting movement of users. This approach not only leverages data from all users concurrently but also eliminates the need of constructing distinct prediction models for each users. The Convolutional Long Short-term Memory (ConvLSTM) [6] model shown in Fig. 2 is used in this investigation. The effectiveness of ConvLSTM in extracting spatiotemporal information is enhanced by its combination with convolutional blocks.

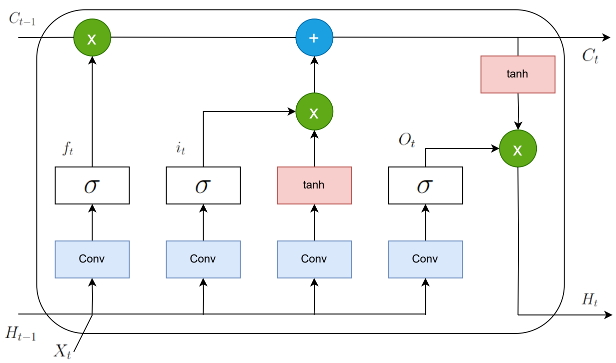

Fig. 2. Convolutional Long Short-term Memory block

The input to the model consists of the spatial locations of all users at t_p previous time steps. Upon completion of the training process, the model provides the location of all users at a certain time t_f, which is a specified number of time steps into the future. Figure 2 illustrates the architectural framework of the mobility prediction model.

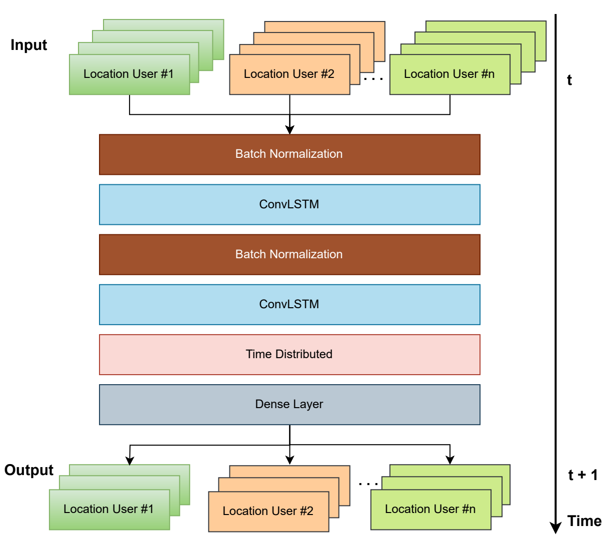

Fig. 3. Mobility Prediction

B. MDP-based Service Migration

In order to use DRL for the purpose of addressing the optimization issue, it is required to define five key components of the MDP. These components are presented as follows.

- State Space:

$$s_t = \{M(t), N(t), B(t)\},$$

- Action Space: $a_t = S(t)$ where $S(t)$ represents the MEC servers that are available to handle user requests.
- State transition probability: $P = p(s_{t+1}|s_t, a_t)$.
- Reward function: The reward r_t indicates the instantaneous incentive obtained for executing action a_t while in state s_t. Minimizing latency is the goal of the equation (9).

$$r_t = -\sum_{n=1}^{N} \frac{1}{N} \left[T_n^s(t) + T_{n,l}(t) + T_{l,s}(t) + T_{p,s}^m(t)\right] \tag{10}$$

- Discount factor: $\gamma \in [0, 1]$.

C. PPO algorithm

This research adopts the PPO framework, which is based on the Actor-Critic approach. The primary advantage of a PPO algorithm is its capacity to effectively manage the trade-off between exploration and exploitation, while also guaranteeing stability and consistent enhancements to the policy. The architecture of PPO is shown in Fig. 4.

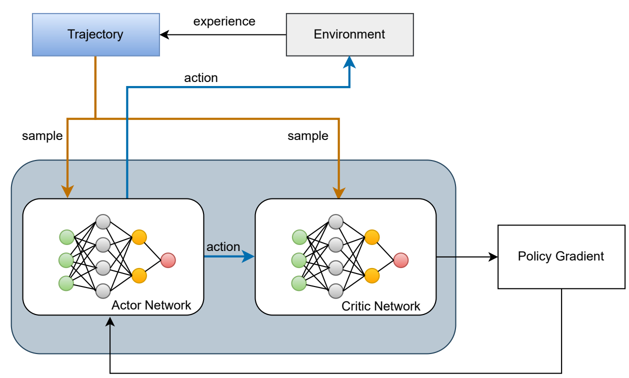

Fig. 4. Proximal Policy Optimization architecture

This method includes 3 steps as follows:

- Environment Interaction: interact with the environment to collect of trajectories. These trajectories are used for both policy evaluation and policy optimization.
- Policy evaluation: train the critic network with the objective of minimizing the error between the estimated and actual returns. The primary aim is to enhance the accuracy of the value function in approximating the actual expected result.
- Policy optimization: optimizing the actor network by maximizing the goal function.

IV. Experiments

Experiments is conducted to evaluate the performance of the proposed framework in making decisions in service migration.

A. Experimental Settings

Experiments simulate the MEC system deployed on an area of $(4km, 4km)$. The cloud server is placed in centralized location, while MEC servers are distributed more widely over the given region. To evaluate the efficiency of the proposed system, real mobility data sourced from San Francisco, USA is used. Furthermore, Table I presents additional experimental parameters.

TABLE I
ENVIRONMENT PARAMETERS

Parameters	Value
The amount of MEC servers	16
The amount of mobile users	100
Range of MEC server computing resources (GHz)	$[0, 25]$
Size of task data (MB)	1
Computational requirement (cycles/bit)	737.5
Channel bandwidth (MHz)	20
Wired transmission rate (MBps)	$[0, 150]$
Noise power spectrum N_0 (dBm/Hz)	-174

B. Comparison Experiments

Fig. 5 presents experimental results examining the latency of various service migration strategies. No Migration, which always maintains the service on a specific MEC server, will increase service latency when users move away from it. In contrast, the average latency for Deep Q-Network (DQN) and PPO is 1067 and 805, respectively. This demonstrates PPO's better capacity to balance exploration and exploitation. Mobility-aware PPO achieves the lowest latency of 562 seconds. This result demonstrates that accurate prediction of the user's next location enhances the effectiveness of optimal decision-making.

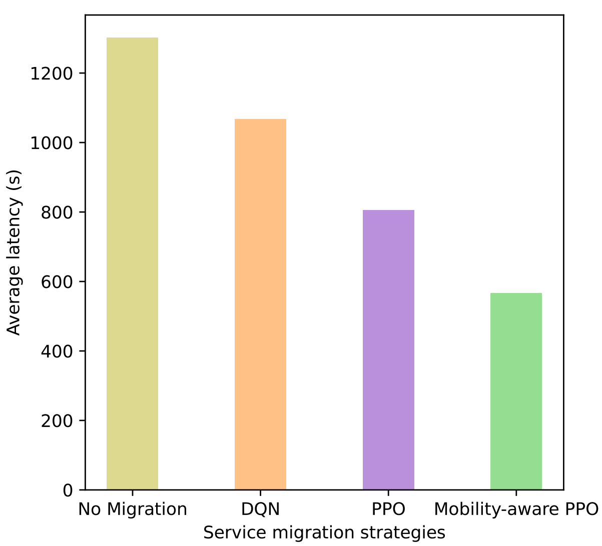

Fig. 5. Performance Comparision

V. Conclusion

The study focuses on addressing the issue of mobility-aware service migration in the MEC system. A recurrent neural network is used for the purpose of forecasting the user's future location. Consequently, the optimization issue is formulated as an MDP and then addressed using the PPO model. In the future, our objective is to investigate the optimization of energy consumption inside this system.

Acknowledgment

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) funded by the Korean Government [Ministry of Science and ICT (MSIT)], South Korea, through the Development of Candidate Element Technology for Intelligent 6G Mobile Core Network under Grant 2022-0-01015; and in part by MSIT under the Information Technology Research Center (ITRC) Support Program Supervised by IITP under Grant IITP-2023-2021-0-02046.

References

[1] S. Lu, J. Wu, J. Shi, P. Lu, J. Fang, and H. Liu, "A dynamic service placement based on deep reinforcement learning in mobile edge computing," *Network*, vol. 2, no. 1, pp. 106–122, 2022.

[2] F. Tang, C. Liu, K. Li, Z. Tang, and K. Li, "Task migration optimization for guaranteeing delay deadline with mobility consideration in mobile edge computing," *Journal of Systems Architecture*, vol. 112, p. 101849, 2021.

[3] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4925–4934, 2021.

[4] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, pp. 587–597, 2018.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[6] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.