Mobility-Aware Service Migration for Seamless Provision: A Reinforcement Learning Approach

Yuan Miao[†], Feng Lyu[†], Fan Wu[‡], Huaqing Wu[§], Ju Ren[‡], Yaoxue Zhang[‡], and Xuemin (Sherman) Shen[§]
[†]School of Computer Science and Engineering, Central South University, Changsha, China
[‡] Department of Computer Science and Technology, Tsinghua University, Beijing, China
[§]Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada
Email:{miaoyuancsu, fenglyu}@csu.edu.cn,{renju, wufancs, zhangyx}@tsinghua.edu.cn, {h272wu, sshen}@uwaterloo.ca

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9839226

***Abstract*—Mobile Edge Computing (MEC) is a promising paradigm to support high-quality time-sensitive applications. In this paper, we investigate the service migration (i.e., whether, when, and where to migrate the services) to seamlessly serve mobile users in small-cell MEC systems. The service migration is formulated as an optimization problem to minimize the long-term system average delay that consists of queuing, communication, and migration delays. Considering the dynamic user mobility and network conditions, the formulated problem is non-convex and difficult to solve in real time. To this end, we propose a Mobility-aware Service Migration scheme, named *MSM*, to make real-time decisions on service migrations by utilizing reinforcement learning (RL) approaches. Specifically, we first design a user classification mechanism based on users' mobility patterns to reduce the complexity of decision-making. We then formulate the service migration as a Markov decision process and devise an RL-based framework to make service migration decisions in real time in the dynamic MEC environment. Extensive data-driven experiments demonstrate the efficacy of *MSM* in reducing the system average delay.**

***Index Terms*—Mobility patterns, small-cell mobile edge computing, service migration, reinforcement learning**

I. Introduction

With the rapid development of IoT devices, mobile edge computing (MEC) has become a promising solution to meet the requirements of emerging delay-sensitive applications [1], such as autonomous driving [2]–[4] and intelligent video acceleration. Specifically, it sinks the computing resources to network edges [5], thus providing users with a low-latency and satisfactory service experience because of the shortened network distance. In addition, it also benefits data safety and privacy protection. However, due to user mobility, service migration has become a focusing challenge in MEC systems. For example, as shown in Fig. 1, there is a system where each edge sever can serve multiple cells (APs). During period t, a mobile user moves from area A to area B. If we still place the user's service in the MEC node near area A, its perceived latency will greatly deteriorate because of the increasing network distance.

In general, service migration incurs additional migration delay as well as potential handover failure. Moreover, when services are placed just according to locations, popular edges will be overloaded, resulting in a great increase in queuing delay. Therefore, the optimal service migration decision depends on multi-dimensional factors, such as users' location, communication channel condition, available resources on MEC nodes, etc [6]–[8]. In addition, the long-term service migration optimization strategy is a sequential decision-making problem, where the decision made at present also has a great impact on the performance of the future system. With limited knowledge of user future mobility, it is difficult to minimize the system delay in the long term. Furthermore, service migration will become more challenging with an increasing number of users.

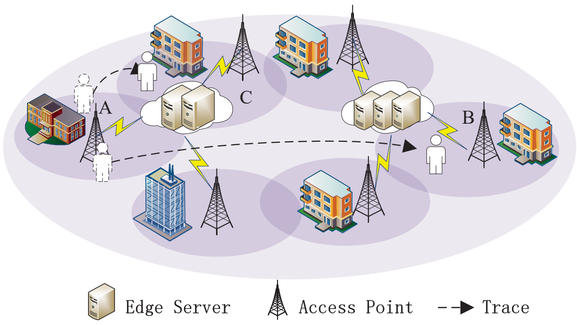


Fig. 1. Service migration for mobile users in a small-cell MEC system.

Previous works were to optimize the service migration by predicting user mobility [9] [10]. In [11], Lyu *et. al.* have collected the association records from an operational Wi-Fi system and achieved several key insights in user mobility patterns. Based on that, we further study the mobility characteristics of users, and plot the cumulative distribution functions (CDFs) of the number of associations per user per day and the corresponding association durations in Figs. 2(a) and 2(b), respectively. The frequent access point (AP) transitions and short durations indicate the high dynamics of user associations. In addition, there are some quite long durations in Fig. 2(b), i.e., users spend most of the association time on several APs, indicating the AP preference of users.

In this paper, we propose a Mobility-aware Service Migration scheme (*MSM*) for small-cell MEC scenarios. In *MSM*, we aim to minimize the long-term system average delay, which consists of three major components, i.e., average queuing delay, average communication delay, and average migration delay. *MSM* mainly takes two steps to optimize the service migration (i.e., whether, when, and where to migrate the services). Firstly, to decrease the complexity of multi-user service

Corresponding author: Feng Lyu

migration and reduce unnecessary migrations, we propose a user classification mechanism based on the observed mobility patterns (i.e., high dynamics of user associations, and AP preference of users). Secondly, to deal with the unavailability of future information, we formulate the service migration as a Markov decision process (MDP) and design a reinforcement learning (RL) based framework to explore the dynamic MEC environment. The framework learns from historical experience and further assists the system to make real-time migration decisions.

The main contributions of this paper are summarized as follows:

- We model the multi-user service migration as an optimization problem to minimize the system average delay including queuing delay, communication delay, and migration delay in the long term.
- We propose an *MSM* scheme in small-cell MEC scenarios, which classifies the user mobility and utilizes an RL-based framework to learn the optimal service migration policy from historical experience.
- We validate the performance of *MSM* with extensive simulations, driven by real traces. Results show that *MSM* can effectively reduce the service delay.

II. Related Work

Service placement has been studied in cloud computing [12] [13], which aimed to reduce service response time, improve resource utilization, balance network loads, etc. Different from cloud computing, in MEC scenarios, the dynamic service migration across edges should be considered because of user mobility. Therefore, the methods proposed in cloud computing cannot be applied to MEC scenarios.

A vital challenge of service migration in MEC systems is the limited knowledge of user mobility. To solve it, some researches were devoted to predicting future information, such as the cost [14], the distribution of requests [9], the user-centric locations [10], and so on. Some researches also work without the request of user mobility knowledge. The authors in [15]–[18] optimized the dynamic service migration by Lyapunov technology, where the long-term optimization problem was decomposed into a series of real-time optimization problems. Then, the solutions to optimize service placement in one-shot time were proposed.

Badri *et. al.* [19] proposed a parallel sample average approximation algorithm for MEC to solve the service migration with energy budget limitation of edge servers. Xu *et. al.* [20] designed a path-selection algorithm in vehicular edge computing systems to jointly optimize the cost of network providers and the quality of experience of users. Ma et. al. [21] designed an edge computing platform architecture for seamless live migration, which leveraged container layered storage.

The authors of [22] considered the service migration problem in MEC scenarios without queuing and formulated the problem as a distance-based MDP. They proposed a modified policy-iteration approach and a mathematical framework to find the optimal solution. Different from [22], we consider the impact of mobility patterns in small-cell MEC systems and service queuing on the average system delay. To minimize the long-term system average delay, we design a mobility-aware service migration scheme for seamless provision.

III. System Model And Problem Formulation

A. System Model

The small-cell MEC system that we consider in this paper consists of N APs and E edge nodes, where the users may handoff among the network frequently. The sets of APs, edge nodes, and online users at period t are denoted by $\mathcal{N}$, $\mathcal{E}$, and $\mathcal{U}(t)$, respectively. Besides, we consider that there is a network operator running these edge nodes to serve users, and the system runs in a discrete time-slotted style. In particular, the large time intervals are divided into many small periodic intervals of the same size (called time slots). Here, we utilize $\mathcal{T}$ to denote the set of time slots. At the beginning of each time slot $t \in \mathcal{T}$, every mobile user would associate with a certain AP and send the service request to the network operator. Correspondingly, the network operator gathers all the request information, and then determines the optimal edges to run services according to current global system information. In addition, the users' locations and network environment do not change during a time slot, but they can vary between two adjacent time slots.

B. Service Migration Model

Let $x_i^j(t)$ and $y_i^e(t)$ be the dynamic AP association and service placement binary indicators of user i at time slot t, respectively. In particular, if user $i \in \mathcal{U}(t)$ associates with AP $j \in \mathcal{N}$ at time slot t, then $x_i^j(t) = 1$, otherwise, $x_i^j(t) = 0$. Similarly, if the tasks of user i are executed on the edge node $e \in \mathcal{E}$, then $y_i^e(t) = 1$, otherwise, $y_i^e(t) = 0$. For convenience of description, we further define two indicator vectors, i.e., $X(t)$ and $Y(t)$, in which $X(t) = \left[x_1^1(t), x_1^2(t), ..., x_{|\mathcal{U}(t)|}^{|\mathcal{N}|}(t)\right]$ and $Y(t) = \left[y_1^1(t), y_1^2(t), ..., y_{|\mathcal{U}(t)|}^{|\mathcal{E}|}(t)\right]$.

At a time slot $t \in \mathcal{T}$, each mobile user in the system can only associate with one AP. Meanwhile, he/she can only be served by one edge node. Therefore, we have following the constraints:

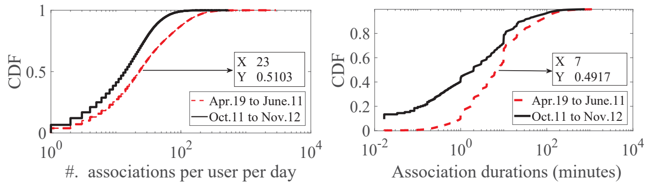
$$\sum_{j=1}^{N} x_i^j(t) = 1, \forall i, t, \tag{1a}$$

$$\sum_{e=1}^{E} y_i^e(t) = 1, \forall i, t. \tag{1b}$$

Moreover, if the service of user i is migrated at time slot t, we have $\sum_{e=1}^{E} y_i^e(t) y_i^e(t-1) = 0$, where $t > 1$.

C. QoS Model

In MEC scenarios, the QoS is jointly affected by three major factors, i.e., queuing delay, communication delay, and migration delay.



(a) the number of associations (b) the duration of associations

Fig. 2. CDFs of user associations.

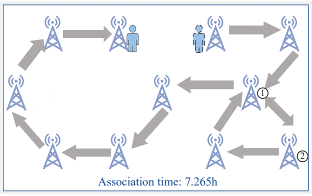


Fig. 3. Alice's trace.

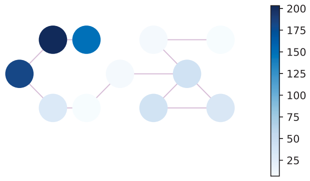


Fig. 4. Duration.

1) Queuing Delay: To compute the queuing delay, we model each edge node as an M/M/1 queue. Specifically, we consider that each user $i \in \mathcal{U}(t)$ begins by offloading its task stream to the edge node with task arrival rate $\lambda_i(t)$, and each edge node $e \in \mathcal{E}$ works with service rate μ_e and maximum workload ζ_e. For a time slot t, the workload limitation of edge nodes should satisfy:

$$\sum_{i=1}^{|\mathcal{U}(t)|} y_i^e(t)\lambda_i(t) \leq \zeta_e, \forall t. \tag{2}$$

Therefore, the total queuing delay of all tasks in the edge set $\mathcal{E}$ at time slot t is

$$Q(Y(t)) = \sum_{i=1}^{|\mathcal{U}(t)|} \sum_{e=1}^{|\mathcal{E}(t)|} \frac{y_i^e(t)\lambda_i(t)}{\mu_e - \sum_{u=1}^{|\mathcal{U}(t)|} y_u^e(t)\lambda_u(t)}. \tag{3}$$

2) Communication Delay: Without loss of generality, the communication delay in our model mainly includes the data transmission delay, which depends on the size of transmitted data and the bandwidth. Therefore, given the users' AP association vector $X(t)$ and service placement vector $Y(t)$, we can compute the overall communication delay for all services at time slot t as follows:

$$C(Y(t)) = \sum_{i=1}^{|\mathcal{U}(t)|} \sum_{j=1}^{N} \sum_{e=1}^{E} \frac{B_i(t)x_i^j(t)y_i^e(t)}{r_j^e(t)}, \tag{4}$$

where $B_i(t)$ and $r_j^e(t)$ denote the transferred data size of user i and the data transmission rate from source AP j to destination edge e at time slot t, respectively.

3) Migration Delay: In small-cell MEC scenarios, frequent migrations greatly increase the additional migration delay, i.e., the delay of transferring service profiles across edges. Let $f_i^{ke}(t)$ denote the size of user i's migrated data from source edge k to destination edge e at time slot t. The total migration delay at time slot t can be computed as:

$$M(Y(t)) = \sum_{i=1}^{|\mathcal{U}(t)|} \sum_{k=1}^{E} \sum_{e=1}^{E} \left(\frac{S_i^{ke}(t)f_i^{ke}(t)}{r_k^e(t)} \right), \tag{5}$$

where $S_i^{ke}(t)$ is a binary indicator representing whether user i migrates service from edge k to edge e at time slot t.

D. Problem Formulation

Given the AP associations $X(t)$, we aim to find the optimal policy $Y(t)$ to minimize the system average delay in the long term, including average queuing delay, average communication delay, and average migration delay. Hence, the service migration optimization problem (SMOP) can be formulated as follows:

$$\min_{Y(t)} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \frac{Q(Y(t)) + C(Y(t)) + M(Y(t))}{\lambda_{sum}(t)} |_{X(t)} \tag{6}$$
$$\text{s.t.} \quad (1a)(1b)(2),$$

where $\lambda_{sum}(t)$ is the overall task arrival rate at time slot t.

The optimal long-term policy of SMOP requires complete future system information (i.e., request distribution or users' mobility), which is difficult to obtain in advance. Moreover, SMOP is a sequential decision making problem, which means the policy made at present will have a profound impact on the future system performance. To address this problem, we propose the *MSM* scheme to learn the optimal service migration policy for the SMOP problem.

IV. Empirical Study on Data

In this section, we explain the user association patterns with collected traces from the operational Wi-Fi system introduced in work [11].

1) High Dynamics of Associations: Fig. 3 demonstrates the trace of a randomly chosen user (called Alice) on May 20. The user associations are highly dynamic with frequent AP transitions. During the day, the user's total association time reaches about 7.265h. Meanwhile, she associates to 11 different APs and there are multiple AP transitions between AP 1 and AP 2.

2) AP Preference of Users: The accumulated duration of Alice in association with each AP on May 20 is shown in Fig. 4, labeled by the node colors. We can clearly observe the AP preference of users. Particularly, the average accumulated association duration of the user spends on the three dark-colored AP nodes, reaches about 1.788 hours, while the average accumulated duration on other APs is close to 0.238 hours. This duration distribution is analogous with that in Fig. 2(b). Based on these phenomena, we can classify APs into preferrable APs and less-preferrable APs for each user. Generally, a user spends most of the association time and data traffic on its preferable APs.

V. User Mobility Patterns Based Service Migration

With the above observations, we propose the *MSM* scheme for the SMOP problem. As shown in Fig. 5, *MSM* utilizes user mobility patterns and an RL approach to select suitable edges for mobile users. Specifically, to efficiently learn the optimal service migration strategy, *MSM* mainly consists of

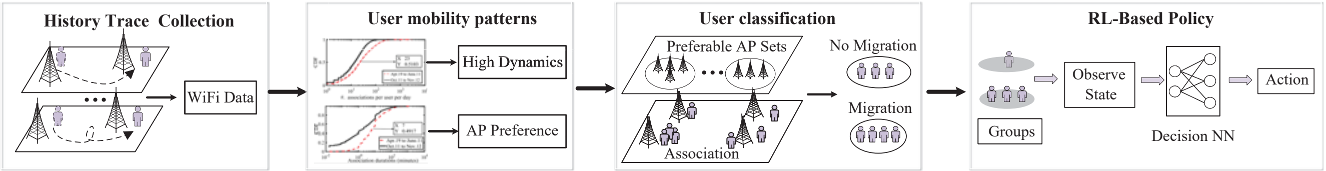


Fig. 5. *MSM* system architecture.

two steps: user classification and RL-based service migration. In the user classification step, we only consider migration for users who move to preferable APs to decrease the unnecessary migrations. In the RL-based service migration step, we further group the selected users and formulate the service migration as an MDP to overcome the unavailability of future information. Then we use an RL approach to learn the optimal service migration policy from historical experience.

A. *User Classification*

Generally, users move dynamically with frequent AP transitions and short association durations. Hence the migration delay caused by following user mobility will greatly affect the system performance. Considering the knowledge of user mobility patterns, i.e., the AP preference of users, we propose a user classification mechanism in *MSM* to avoid unnecessary migrations.

Firstly, to capture the AP preference, we extract a preferable AP set for each user. Let $\theta_i(j)$ be the ratio of the association time user i spends on AP j to its overall association time. We can define the preferable AP set for user i as follows:

$$P_i = \{x | x \in \mathcal{N}, \theta_i(x) > \xi\}, \tag{7}$$

where ξ is a constant representing the threshold.

With the preferable AP sets, we classify users into two categories, i.e., one needs to consider service migration, and the other does not need service migration. Specifically, we select the users who move to preferable APs to consider service migration, because they may stay there for a long time. For users moving to less-preferable APs, service migration is undesirable due to the short association time and frequent AP transitions. These services should be placed on the original location.

B. *RL-Based Service Migration Policy*

As a branch of machine learning, RL is an effective solution to find the optimal policy to maximize the cumulative rewards in a long time. It means that RL cares about the impact of current decisions on the future rather than immediate rewards, which is consistent with the requirements of SMOP. In this section, we present the proposed RL-based service migration policy with details, for the users moving to preferable APs.

1) User Grouping: For practical MEC scenarios, the number of associated users varies over time, and the large and erratic number of users is an issue to RL's state space and action space. To address the challenge, we group users to reduce the dimension of the multi-user information.

Generally, each AP has its own geographic attributes. For the APs in the same department building, their association variations are similar; for the APs in the different department buildings, their association variations are significantly different [11]. These phenomena indicate that users in the same department share similar mobility patterns. Therefore, we group the selected users according to the departments of associated APs.

2) MDP-based Offloading Problem Formulation: The SMOP problem can be described as an MDP. We further define the state space, action space, and reward function as follows.

a) State Space: For the ease of reading, we utilize $\hat{\boldsymbol{\mu}}(t)$, $\hat{\boldsymbol{r}}(t)$ and $\dot{\boldsymbol{\lambda}}(t)$ to denote the vectors of the corresponding states:

$$\hat{\boldsymbol{\mu}}(t) = [\hat{\mu}_1(t), ..., \hat{\mu}_E(t)], \tag{8}$$

$$\hat{\boldsymbol{r}}(t) = [\hat{r}_{1,1}(t), ..., \hat{r}_{K,E}(t)], \tag{9}$$

$$\dot{\boldsymbol{\lambda}}(t) = [\dot{\boldsymbol{\lambda}}_1(t), ..., \dot{\boldsymbol{\lambda}}_K(t)], \tag{10}$$

where $\hat{\boldsymbol{\mu}}(t)$, $\hat{\boldsymbol{r}}(t)$, $\dot{\boldsymbol{\lambda}}(t)$, and K denote the available service resource of edges, the average transmission data rates between groups and edges, the task arrival distribution of groups, and the number of groups, respectively. The task arrival distribution of group $k \in \{1, ..., K\}$ can be expressed as:

$$\dot{\boldsymbol{\lambda}}_k(t) = [\dot{\lambda}_{k,0}(t), ..., \dot{\lambda}_{k,E}(t)], \tag{11}$$

where $\dot{\lambda}_{k,0}(t)$ denotes the overall task arrival rate of new users and $\dot{\lambda}_{k,e}(t)$ denotes the overall task arrival rate of users who ran tasks on edge e at time slot $t-1$. Note that $\hat{\boldsymbol{\mu}}(t)$, $\hat{\boldsymbol{r}}(t)$, $\dot{\boldsymbol{\lambda}}(t)$ can be easily calculated based on the information on $\lambda(t), r(t), \mu(t)$. The state $\boldsymbol{s}(t)$ at time slot t is described as:

$$\boldsymbol{s}(t) = [\hat{\boldsymbol{\mu}}(t), \dot{\boldsymbol{\lambda}}(t), \hat{\boldsymbol{r}}(t)]. \tag{12}$$

b) Action Space: The action $\boldsymbol{a}(t)$ performed at processing period t can be defined as:

$$\boldsymbol{a}(t) = [a_1(t), ..., a_K(t)]. \tag{13}$$

Note that $\boldsymbol{a}(t)$ denotes the locations of the group service placement. Hence, action space $\mathcal{A}$ is expressed as:

$$\mathcal{A} = \{a_k(t) \in \{1, 2, \ldots, E\}, \forall k\}. \tag{14}$$

c) Reward: Since our scheme aims to decrease the long-term average system delay, combined with Eq. (6) in the SMOP problem, the reward in our case is equal to the negative value of system average delay.

3) Asynchronous Advantage Actor-Critic (A3C): In this paper, we utilize the A3C algorithm to deal with the corresponding MDP problem. It utilizes a general asynchronous and concurrent RL framework to accelerate convergence efficiency.

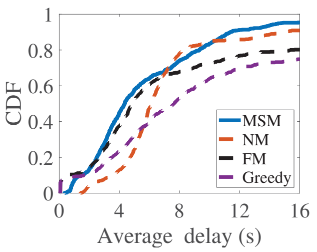


Fig. 7. System average delay.

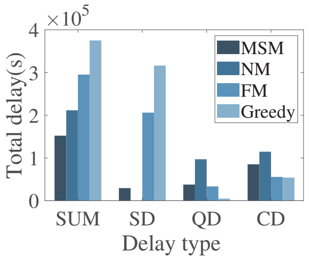


Fig. 8. Accumulated delay in a week.

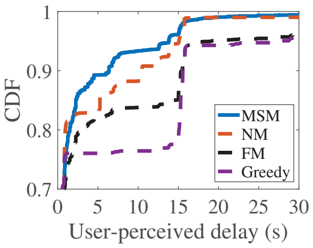


Fig. 9. User-perceived delay.

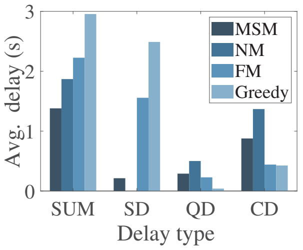


Fig. 10. Average user-perceived delay.

VI. PERFORMANCE EVALUATION

A. Simulation Setup

1) APs: We consider an MEC system consisting of $N = 50$ APs, which are chosen randomly from user traces. Moreover, to obtain the preferable AP set for each user, we analyze the historical associations from April 20th to June 3rd in 2019.

2) MEC nodes: We select $E = 10$ most popular APs from the previous 50 APs and let them act as edge nodes. Moreover, we set the total service rate of all edge nodes to be slightly larger than the overall required service rate of users. The service rate of every edge node is set to be proportional to the number of associated users.

3) Users: In our simulations, we classify the user tasks into two categories: heavy and light computational applications. The user task arrival rate for heavy computational applications is distributed in $\{0.5, 0.55, 0.6\}$ per slot and the data size of each task is 50M. Similarly, the user task arrival rate for light computational applications is distributed in $\{0.1, 0.2, 0.3\}$ per slot and the data size of each task is 5M. Without loss of generality, a mobile user can arbitrarily generate different types of tasks.

B. Performance Benchmark

1) No Migration (NM): This scheme always keeps the initial service placement policy unchanged for each user.

2) Follow mobility (FM): This scheme always migrates the services to the nearest MEC nodes.

3) Greedy: This scheme always minimizes the current queuing delay and communication delay while ignoring the migration delay in the network.

C. Effectiveness of MSM

1) Convergence: Fig. 6 shows the sum of system average delay obtained in each episode, which becomes stable when the episode number is higher than 800. It means that the *MSM* scheme converges after 800 learning episodes.

2) System delay optimality: To test the effectiveness of *MSM*, we utilize the association records from June 3 to June 10 in 2019, which lasts for a full week. We first plot the system delay of *MSM* and benchmarks in Figs. 7 and 8. We can observe that *MSM* scheme outperforms other baseline schemes. For example, the median average delay is 4.3s, 4.8s, 6.1s, and 7.6s in the *MSM*, FM, NM, and Greedy schemes, respectively.

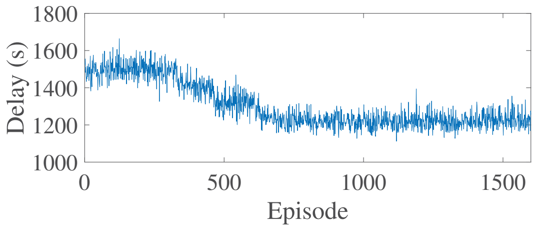


Fig. 6. Learning curve.

To compare the accumulated delay for all users in the week in detail, we plot the results of all types of delays in Fig. 8. migration delay, and the sum of all types of delays, respectively. We have three major observations as follows: 1) The accumulated delay of *MSM* is higher with around 28% improvement over NM, 48% improvement over FM, and 59% improvement over Greedy; 2) FM optimizes communication time and the Greedy scheme minimizes the sum of communication time and queuing time, while large migration delay reduces the effectiveness of these schemes; and 3) *MSM* has advantages over benchmarks through reducing unnecessary migrations.

3) User-perceived delay optimality: Unlike system delay, the user-perceived delay is a vital metric to measure the users' QoS. Therefore, we plot the CDFs of user-perceived delay and user average delay results for *MSM* and benchmarks in Figs. 9 and 10, respectively. We can observe that *MSM* outperforms other baseline schemes.

4) Delay varies over time: As shown in Fig. 11, the number of associated users varies periodically by days. For instance, at each night, the system is idle and the number of associated users is very low, less than 100; yet in the daytime, the system becomes busy and the number is large, up to 400. This regular change is consistent with our living habits. To test the effectiveness of *MSM*, we further study the performance of *MSM* and benchmarks on each day and at different work states. We plot the average user-perceived delay with different user numbers in Fig. 12, and plot the overall system delay and the number of user movements of each day in Fig. 13, respectively. We have two major observations. Firstly, our scheme achieves better performance whether the system is idle or busy as shown in Fig. 12. Secondly, the overall delay on each day varies with the number of user movements and *MSM* achieves the best performance at most times in Fig. 13.

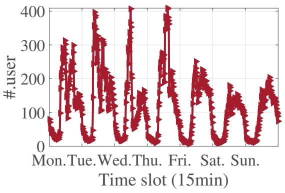


Fig. 11. Number of mobile users.

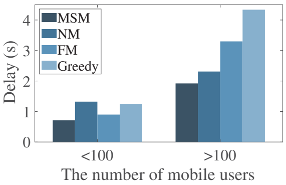


Fig. 12. Avg. user-perceived delay.

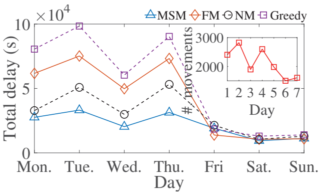


Fig. 13. System delay.

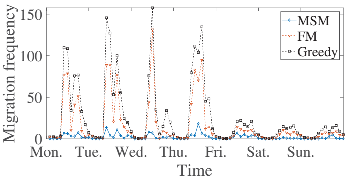


Fig. 14. Migration frequency.

5) Migration analysis: In Fig. 14, we plot the number of service migrations of *MSM* and benchmarks at each time slot. It is observed that FM and Greedy schemes migrate services frequently at most times.

VII. Conclusion

In this paper, we have investigated the service migration in MEC scenarios and proposed an RL-based *MSM* scheme to jointly optimize the system average delay in the long term based on the analysis of history mobility traces. *MSM* has three significant merits: 1) *MSM* reduces the complexity of multi-user service migration by studying user mobility patterns; 2) *MSM* can select the suitable service placement location in real-time to keep pace with the dynamic and complicated network scenarios by using the RL-based approach; and 3) *MSM* can optimize the long-term system average delay. For future work, we will investigate the task scheduling at servers to further improve the overall performance of the small-cell MEC system.

Acknowledgments

This research was supported in part by National Natural Science Foundation of China (Grants No.62102223, 62122095, 62072472, 62002389, U19A2067), in part Funded by China Postdoctoral Science Foundation (Grants No.2021M691785), 111 Project (Grant No.B1805), the Young Elite Scientist Sponsorship Program by CAST (Grant No.YESS20200238), Natural Science Foundation of Hunan Province of China (Grants No.2020JJ2050, 2021JJ20079), the Young Talents Plan of Hunan Province of China (Grants No.2021RC3004), and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

[1] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, “A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet,” *ACM Comput. Surv. (CSUR)*, vol. 52, no. 6, pp. 1–36, 2019.

[2] H. Wu, F. Lyu, C. Zhou, J. Chen, L. Wang, and X. Shen, “Optimal uav caching and trajectory in aerial-assisted vehicular networks: A learning-based approach,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2783–2797, 2020.

[3] F. Lyu, H. Zhu, N. Cheng, H. Zhou, W. Xu, M. Li, and X. Shen, “Characterizing urban vehicle-to-vehicle communications for reliable safety applications,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2586–2602, 2020.

[4] F. Wu, W. Yang, J. Lu, F. Lyu, J. Ren, and Y. Zhang, “RLSS: A Reinforcement Learning Scheme for HD Map Data Source Selection in Vehicular NDN,” *IEEE Internet Things J.*, pp. 1–1, 2021.

[5] Z. Fan, W. Yang, F. Wu, J. Cao, and W. Shi, “Serving at the Edge: An Edge Computing Service Architecture Based on ICN,” *ACM Trans. Internet Technol.*, vol. 22, no. 1, pp. 1–27, Oct. 2021.

[6] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, “Live service migration in mobile edge clouds,” *IEEE Wirel. Commun.*, vol. 25, no. 1, pp. 140–147, 2018.

[7] Y. Deng, F. Lyu, J. Ren, H. Wu, Y. Zhou, Y. Zhang, and X. Shen, “AUCTION: Automated and Quality-Aware Client Selection Framework for Efficient Federated Learning,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 8, pp. 1996–2009, 2022.

[8] F. Wu, W. Yang, J. Ren, F. Lyu, P. Yang, Y. Zhang, and X. Shen, “NDN-MMRA: Multi-Stage Multicast Rate Adaptation in Named Data Networking WLAN,” *IEEE Trans. Multimedia*, vol. 23, pp. 3250–3263, 2021.

[9] L. Yang, J. Cao, G. Liang, and X. Han, “Cost aware service placement and load dispatching in mobile cloud systems,” *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1440–1452, 2015.

[10] Q. Wu, X. Chen, Z. Zhou, and L. Chen, “Mobile social data learning for user-centric location prediction with application in mobile edge service migration,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7737–7747, 2019.

[11] F. Lyu, J. Ren, N. Cheng, P. Yang, M. Li, Y. Zhang, and X. Shen, “LEAD: Large-scale edge cache deployment based on spatio-temporal WiFi traffic statistics,” *IEEE Trans. Mob. Comput.*, vol. 20, no. 8, pp. 2607–2623, 2020.

[12] F. Liu, P. Shu, and J. C. Lui, “AppATP: An energy conserving adaptive mobile-cloud transmission protocol,” *IEEE Trans. Comput.*, vol. 64, no. 11, pp. 3051–3063, 2015.

[13] J. W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang, “Joint VM placement and routing for data center traffic engineering,” in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012, pp. 2876–2880.

[14] S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer, and K. K. Leung, “Dynamic service placement for mobile micro-clouds with predicted future costs,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 4, pp. 1002–1016, 2016.

[15] T. Ouyang, Z. Zhou, and X. Chen, “Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2333–2345, 2018.

[16] Z. Ning, P. Dong, X. Wang, S. Wang, X. Hu, S. Guo, T. Qiu, B. Hu, and R. Y. Kwok, “Distributed and dynamic service placement in pervasive edge computing networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 6, pp. 1277–1292, 2020.

[17] B. Gao, Z. Zhou, F. Liu, and F. Xu, “Winning at the starting line: Joint network selection and service placement for mobile edge computing,” in *Proc. IEEE INFOCOM*, Paris, France, 2019, pp. 1459–1467.

[18] Y. Sun, S. Zhou, and J. Xu, “EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2637–2646, 2017.

[19] H. Badri, T. Bahreini, D. Grosu, and K. Yang, “Energy-aware application placement in mobile edge computing: A stochastic optimization approach,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 4, pp. 909–922, 2019.

[20] J. Xu, X. Ma, A. Zhou, Q. Duan, and S. Wang, “Path selection for seamless service migration in vehicular edge computing,” *IEEE Internet Things J.*, vol. 7, no. 9, pp. 9040–9049, 2020.

[21] L. Ma, S. Yi, N. Carter, and Q. Li, “Efficient live migration of edge services leveraging container layered storage,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 9, pp. 2020–2033, 2018.

[22] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, “Dynamic service migration in mobile edge computing based on markov decision process,” *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1272–1288, 2019.