Service Migration in Mobile Edge Computing Based on Reinforcement Learning

Chen Fan$^{1a}$ and Li Li$^{1b}$

$^1$State Key Laboratory of Networking and Switching Technology, Beijing Laboratory of Advanced Information Networks, Beijing University of Posts and Telecommunications, Institute of Sensing Technology and Business, BUPT (Wuxi), China.

E-mail: $^{a}$fanchen@bupt.edu.cn; $^{b}$lili66@bupt.edu.cn

Abstract. Mobile edge computing (MEC) provides users with cloud computing capabilities at the edge of the mobile network, which effectively reduces network latency and improves the experience of end-users. User mobility in MEC is a factor that cannot be ignored, and mobility management is an urgent problem to be solved. Service migration is an effective way to manage user mobility. However, it is not appropriate to perform migration too frequently due to expensive migration overhead. In this paper, we propose a service migration decision algorithm to decide whether to migrate or not when the user moves out of the coverage of the offloaded MEC server. Markov decision process (MDP) is used to model the service migration decision problem. We comprehensively consider the distance between users and services, resource requirements of the services, and resource availability of the MEC servers. On the premise of considering both migration costs and resource conditions, aiming at maximizing quality of service (QoS), the reward function of MDP is defined, and the migration decision strategy is solved by Q-learning algorithm. Finally, our proposed migration decision algorithm is validated by simulation.

1. Introduction

The emergence of new services, such as Internet of Things, has brought higher and higher requirements on the network. 5G needs to simultaneously meet the requirements of high bandwidth, low latency, and massive connections. Mobile edge computing (MEC) [1], a key technology of 5G, provides users with cloud computing capabilities at the edge of the mobile network, which effectively reduces latency and improves user experience. User mobility cannot be ignored in MEC, and thus mobility management problem needs to be solved, which refers to how to guarantee quality of service (QoS) when users move.

There have been several solutions to deal with mobility management, such as backhaul network communication [2], power control [3], and service migration [3]. By performing service migration, the communication distance is always kept within a small range, which is beneficial for delay-sensitive services. Ideally, migration will be operated as soon as the user moves out of coverage of the current offloaded MEC server. However, migration overhead is expensive and cannot be ignored in reality, so too frequent migration is not desirable. It is necessary to design a proper method to decide whether to migrate.

The contributions of this paper are as follows:
1) We adopt reinforcement learning to model the service migration problem and propose an algorithm based on Q-learning to solve the problem.

2) Different from previous work, we consider the resource demand of the service and resource availability in the target MEC server when making migration decisions.

3) Simulations have been finished to validate our proposed algorithm.

2. Related work
Ksentini et al.[4] formulated the migration policy as a continuous-time MDP and found an optimal threshold policy. Wang et al. [5] further used the "constant + exponential" function to materialize the cost function. They used the equilibrium equation to find the closed solution, while improving the strategy iteration method to obtain a more accurate optimal strategy. Nadembega et al. [6] adopted mobility prediction to improve the migration process. The method is composed of three parts: (1) estimate the throughput user can obtain from each MEC server on each road segment; (2) estimate the time window in which the user performs the handover; (3) divide the whole service, and each micro data center is responsible for transmitting one part to the user during his movement. Wang et al. [7] proposed an algorithm that enhances the migration decision by predicting future migration costs. The algorithm assumes that the start and end of service are known, and the predicted migration cost can be obtained through the time window. The work was extended in [8], in which the multi-user and multi-task scenario was further considered. Yao et al. [9] combined service migration with path selection, and the migration decision was designed based on the shortest path so that the service can follow the vehicle and ensure QoS.

3. System model

3.1. User mobility model
In the one-dimensional scenario, assuming that a user can only move between adjacent areas within a limited period. The set of user movement modes is denoted as M = {-1, 0, 1}, and the user movement mode at timeslot t is $m_t$. The initial position of user is recorded as the origin, and the one-dimensional scene can be regarded as moving along a one-dimensional coordinate axis. -1 and 1 indicate the user moving along the negative and positive direction of the coordinate axis, respectively, and 0 indicates the user remaining in the previous area. The user 's movement obeys the rule as follows:

$$p(m_t) = \begin{cases} p, & m_t \neq 0 \\ 1 - 2p, & m_t = 0 \end{cases}$$

(1)

To ensure QoS, we set the maximum distance through the backhaul network to be $thr$. When the distance exceeds $thr$, migration will be performed directly without any policy judgment.

Assuming that service is directly migrated to the nearest MEC server, and state is set to be the "hop" distance, and then the state transition can be obtained as shown in figure 1, where solid lines indicate no service migration, i.e., $a = 0$; and dotted lines indicate performing migration, i.e., $a = 1$. State transition probability can be deduced from figure 1, as shown in equation (2), where s and $s'$ represent the current state and the next state, respectively.

$$p(s'|s, a) = \begin{cases} p, & s' = s + 1, s \neq 1, a = 0 \\ 2p, & s = 0, s' = 1, a = 0 \\ 1 - 2p, & s' = s, a = 0 \\ 1, & s' = 0 \\ 0, & \text{others} \end{cases}$$

(2)
3.2. MDP model

3.2.1. Action. The action at timeslot $t$ is denoted as $a_t \in A = \{0, 1\}$, where $A$ is the action set. 0 indicates no migration, and 1 indicates migration.

3.2.2. State. We denote the distance between the user and the service by $d_t$, the computing resource requirement of the service by $C_d$ and the available computing resource of the target server by $C_a$. The state at timeslot $t$ is denoted by $s_t = \{d_t, C_a, C_d\}$.

3.2.3. Reward. The quality at timeslot $t$ is denoted by $q_t$, denoted as equation (3), where $q_0$ is the maximal quality, $\alpha$ is a constant coefficient.

$$q_t(d_t) = q_0 - \alpha \cdot d_t, \quad d_t \geq 0$$

Migration operation introduces migration costs. We denote the costs as equation (4), where $\beta$ is a constant coefficient, which means one-hop migration cost, and $\delta$ is a constant coefficient, which means the conversion cost of starting a virtual machine on the new MEC server.

$$c_t(d_t, a_t) = \begin{cases} 0, & \text{if } a_t = 0 \\ \delta + \beta \cdot d_t, & \text{if } a_t = 1 \end{cases}$$

The goal of the strategy is to maximize QoS, considering migration costs. Furthermore, storage and computing resources on MEC servers are far fewer than those on cloud servers [10]. Therefore, resource requirement of the service and load on the target server need to be considered, which can be realized in the reward function. If $a_t = 1$ and $C_a < C_d$, we add a big punishment $M$ to the reward to prevent the user from performing migration. Besides, $\omega$ is constant, which represents the ratio between the cost and maximum quality $q_0$. The reward function is defined as follows.

$$r_t(s_t, a_t) = \begin{cases} q(s_t), & \text{if } a_t = 0 \\ q_0 - \omega \cdot c_t(s_t, a_t), & \text{if } a_t = 1 \text{ and } C_a \geq C_d \\ q_0 - \omega \cdot c_t(s_t, a_t) - M, & \text{if } a_t = 1 \text{ and } C_a < C_d \end{cases}$$

4. Service migration algorithm

Based on the MDP model defined in subsection 3.2, the migration strategy is solved by Q-learning algorithm, which adopts $Q$ function to find the best action. $Q$ function can be expressed as $Q(s, a)$, which is a function related to both the state and action and regarded as a long-term reward. The update method of the $Q$ function is as follows in equation (6), where $s$ represents the current state, $s'$ represents the next state, $a$ represents the current action, $r$ represents the immediate reward, $\eta$ is
learning rate, and \( \gamma \) is a discount factor which ranges from zero to one, indicating the importance of future returns. The larger \( \gamma \) is, the more important future reward is.

\[
Q(s, a) = (1 - \eta) * Q(s, a) + \eta [r + \gamma \max Q(s', a)]
\]  

(6)

The result of using the Q-learning algorithm for the service migration decision problem is to obtain a Q-table. At the beginning of each timeslot, we observe the state. According to the optional actions, the action with the highest value in the row with the state in the Q-table can be selected. The method for solving the migration decision in MEC is shown in Algorithm 1.

Algorithm 1: Service Migration Decision Algorithm

| Input: state set S, action set A |
| Output: Q-table |
| 1. Initialize Q-table |
| 2. for each episode: |
| 3. Initialize state s |
| 4. for each step of episode do |
| 5. Get Q value from Q-table according to the state \( s \) and select an action according to \( \epsilon - \) greedy strategy. |
| 6. Observe new state \( s' \), and get the immediate reward according to equation (5) |
| 7. Update Q-table according to equation (6) |
| 8. Update state: \( s \leftarrow s' \) |
| 9. end for |
| 10. end for |

5. Performance evaluation

In this section, a numerical experiment is carried out to evaluate the performance of our proposed scheme by comparison with the following two schemes: (1) always migrate scheme: once the user moves out of the coverage of current offloaded MEC server, migration will be initiated. (2) MDP scheme [4]: Based on MDP, only the distance between the user and the offloaded MEC server is considered.

5.1. Experimental settings.

The parameters in the experiment are shown in table 1.

| Parameters | Meaning | Value |
|------------|---------|-------|
| thr        | maximum distance       | 10    |
| \( q_0 \)  | maximum quality of service | 36    |
| \( \alpha \) | coefficient of quality of service | 4     |
| \( \beta \) | coefficient of migration cost  | 2     |
| \( \delta \) | constant of the migration cost | 10    |
| \( \omega \) | coefficient of migration cost in reward function | 1     |
| \( C_a \)  | MEC server’s available resource | [0, 10] |
| \( \gamma \) | learning rate          | 0.5   |

5.2. Numerical results

5.2.1. Impact of migration cost on the migration strategy. By changing \( \omega \) in the reward function, the
migration strategy can be obtained as shown in table 2. ω indicates the proportion of migration cost in reward function. The larger ω is, the more important the migration cost is when making decisions. d represents the communication distance. The value in the table represents the action to be taken, where N means no migration, and Y means migration operation.

| ω/d | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1.8 | N   | N   | N   | N   | Y   | Y   | Y   | Y   | Y   |
| 1.6 | N   | N   | N   | Y   | Y   | Y   | Y   | Y   | Y   |
| 1.4 | N   | N   | N   | Y   | Y   | Y   | Y   | Y   | Y   |
| 1.2 | N   | N   | Y   | Y   | Y   | Y   | Y   | Y   | Y   |
| 1.0 | N   | N   | Y   | Y   | Y   | Y   | Y   | Y   | Y   |
| 0.8 | N   | Y   | Y   | Y   | Y   | Y   | Y   | Y   | Y   |
| 0.6 | N   | Y   | Y   | Y   | Y   | Y   | Y   | Y   | Y   |
| 0.4 | Y   | Y   | Y   | Y   | Y   | Y   | Y   | Y   | Y   |

As is shown in table 2, if ω is fixed, the migration strategy is a threshold strategy. When the distance is bigger than the threshold, service migration needs to be performed. On the one hand, if ω is relatively small, the threshold is small, that is to say, service migration tends to be performed when the distance is small. In particular, if ω is smaller than a certain value, it means that migration cost is negligible compared with migration benefit. Therefore, the migration strategy degenerates into the always migrate strategy. On the other hand, if ω is relatively large, service migration will be performed only when the user is relatively far from the MEC server. If ω is very large, within the distance smaller than the maximum communication distance, i.e., thr, service migration will never be performed. That is to say, the migration strategy degenerates into the never migration strategy.

Figure 2. The impact of ω on the total reward.

Figure 3. The impact of \( C_d \) on the total reward.

5.2.2. Impact of migration costs on the total reward. Figure 2 demonstrates the impact of ω on the total reward. We can see that when ω = 0.2 and ω = 0.4, the total reward of the MDP scheme is always the same as always migration scheme. The reason is that when ω is small, migration cost can be ignored, and the MDP scheme degenerates into always migration scheme. The total reward of our proposed scheme is higher than the others because the resource requirements of the service and available resource of the MEC server are considered. The proposed scheme rarely fails to migrate, while the other two schemes are not. With the increase of ω, the total reward of our proposed scheme and always migration scheme decrease to a certain extent, because larger ω indicates that the migration cost is more important. As shown in table 2, the bigger the ω is, the bigger the threshold for
performing migration is, that is to say, the number of migrations will decrease, so that the total reward will decrease. However, total reward of the MDP scheme changes slightly with the change of \( \omega \), because reducing the number of migrations not only reduces the total reward but also reduces the number of migration failures, and they strike a balance.

5.2.3. Impact of resource demand of service on the total reward. Figure 3 shows the impact of resource demand \( C_d \) on the total reward. As is shown in figure 3, if the resource demand is small, the total reward of the three schemes is positive. When \( C_d \) becomes larger, the reward of MDP scheme and always migration scheme both decrease sharply, because neither of these two schemes takes resource availability of the MEC server into account. When \( C_d \) becomes larger, it is likely that the MEC server’s remaining resources cannot support the migration, resulting in migration failures, and the penalty value for migration failure is large. On the whole, the total reward of our proposed method is always higher than the others.

6. Conclusion
In this paper, we formulate the service migration process using MDP and propose a migration decision algorithm to decide whether to migrate or not. Our proposed algorithm is based on reinforcement learning, aiming to maximize user’s quality of service. What’s more, we take limited resources in MEC servers into account. The simulation results prove the effectiveness of our proposed algorithm.

Acknowledgments
The research work is supported by the project of National Science and Technology Major Project of the Ministry of Science and Technology of China (No: 2018ZX03001014-003).

References
[1] ETSI. Mobile Edge Computing: A key technology towards 5G. https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key Technology_towards_5g.pdf. 2015
[2] Zhang F, Liu G, Fu X, Yahyapour R. A Survey on Virtual Machine Migration: Challenges, Techniques, and Open Issues. *IEEE Communications Surveys & Tutorials*. 2018;20(2):1206-43.
[3] Mach P, Becvar Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *IEEE Communications Surveys & Tutorials*. 2017;19(3):1628-56.
[4] Ksentini A, Taleb T, Chen M. A Markov decision process-based service migration procedure for follow me cloud. 2014 IEEE International Conference on Communications (ICC); 2014: IEEE.
[5] Wang S, Urgaonkar R, Zafer M, He T, Chan K, Leung KK, editors. Dynamic service migration in mobile edge-clouds. 2015 IFIP Networking Conference (IFIP Networking); 2015 20-22 May 2015.
[6] Ndehembega A, Hafid AS, Brisebois R, editors. Mobility prediction model-based service migration procedure for follow me cloud to support QoS and QoE. 2016 IEEE International Conference on Communications (ICC); 2016 22-27 May 2016.
[7] Wang S, Urgaonkar R, Chan K, He T, Zafer M, Leung KK, editors. Dynamic service placement for mobile micro-clouds with predicted future costs. 2015 IEEE International Conference on Communications (ICC); 2015 8-12 June 2015.
[8] Wang S, Urgaonkar R, He T, Chan K, Zafer M, Leung KK. Dynamic Service Placement for Mobile Micro-Clouds with Predicted Future Costs. *IEEE Transactions on Parallel and Distributed Systems*. 2017;28(4):1002-16.
[9] Yao H, Bai C, Zeng D, Liang Q, Fan Y. Migrate or not? Exploring virtual machine migration in roadside cloudlet - based vehicular cloud. *Concurrency and Computation: Practice and Experience*. 2015;27(18):5780-92.
[10] Mao Y, You C, Zhang J, Huang K, Letaief KB. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*. 2017;19(4):2322-58.