DRL based Dynamic Service Mobility for Marginal Downtime in Multi-access Edge Computing

Lusungu Josh Mwasinga[1], Syed Muhammad Raza [2], 추현승 [3]
[1] Dept. of Computer Science and Engineering, Sungkyunkwan University
[2, 3] Dept. of Electrical and Computer Engineering, Sungkyunkwan University

Abstract

The advent of the Multi-access Edge Computing (MEC) paradigm allows mobile users to offload resource-intensive and delay-stringent services to nearby servers, thereby significantly enhancing the quality of experience. Due to erratic roaming of mobile users in the network environment, maintaining maximum quality of experience becomes challenging as they move farther away from the serving edge server, particularly due to the increased latency resulting from the extended distance. The services could be migrated, under policies obtained using Deep Reinforcement Learning (DRL) techniques, to an optimal edge server, however, this operation incurs significant costs in terms of service downtime, thereby adversely affecting service quality of experience. Thus, this study addresses the service mobility problem of deciding whether to migrate and where to migrate the service instance for maximized migration benefits and marginal service downtime.

1. Introduction

Multi-Access Edge Computing (MEC) has shown the potential to enable service providers to deliver enhanced Ultra-Reliable Low-Latency Communication (URLLC) services to high-mobility users. Under the MEC paradigm, cloud computing resources are distributed at the network edge in proximity to mobile users. This approach enables mobile users to offload services that demand massive computing resources and stringent delay requirements to Edge Servers for to meet the demands. By offloading tasks to edge servers, mobile devices can save battery life as most computation-intensive tasks are processed on the resource rich edge servers. Processing tasks on the resource-rich ES reduces service delay as requests do not traverse the backhaul network for processing at remote cloud. The MEC paradigm relies on service mobility to ensure users moving further away from the service edge server continues to access their services with expected service quality of experience.

Offloaded services should be relocated to optimal edge servers following user mobility to satisfy expected service quality of experience. A major issue with service mobility is extended service downtime. With respect to service migration, service downtime can be defined as a period during which a mobile user cannot access its service because it is still being migrated. Extended service downtime adversely affects quality of experience for mobile users as it increases the waiting time before service access resumes.

This study focuses on the service mobility problem to enhance QoE for mobile users as they roam within the network environment. Specifically, we address issues surrounding the decision on when to migrate and the selection of optimal Edge Server where a migrated service instance could be deployed to provide maximum QoE benefits for target mobile user. The target issues are of critical relevance in the sense that service mobility yields two conflicting results. On the one hand, transferring services to a nearby edge server ensures mobile users continue accessing services with enhance QoE. On the other hand, the process incurs significant costs in terms of service downtime as well as bandwidth consumption, among others. This work exploits Deep Reinforcement Learning (DRL) to learn an optimal policy for service mobility such that the process achieves its primary objective of enhancing QoE for mobile users as marginal service downtime.

2. Related Work

The service mobility decision problem has been widely studied by various scholars, with varying optimization objectives and approaches, ranging from heuristic techniques to those employing machine learning (ML). For instance, the greedy approach in [1] compares service migration cost and perceived QoE in the new location, and depending on the outcome, the service is either relocated or continues to executing on the current ES. Ordinarily, this approach initiates service mobility if perceived QoE benefits outweigh service migration time as costs. Considering the dynamic nature of mobile network environments, this approach may result in transferring the service instance to sub-optimal edge server which was initially determined as optimal.

Improving on heuristic-based approaches, other studies have proposed to employ ML-based approaches such as reinforcement learning. For instance, the work of [2] presents a Deep Reinforcement Learning based approach, where the objective is to learn a policy for optimal service migration that ensures high-mobility users continue to exploit services with acceptable QoE upon migrating service instances at acceptable migration costs.

Improving on existing works, we exploit the Deep Q-learning (DQN) [3] algorithm to design and implement a service mobility decision approach in MEC. The objective is to ensure service instances for mobile users are migrated to the optimal destination MEC servers with minimal service migration cost.

3. DRL based Service Mobility in MEC

In this section we present the DRL based service mobility approach. Its main aim is to enhance service performance with marginal service downtime.

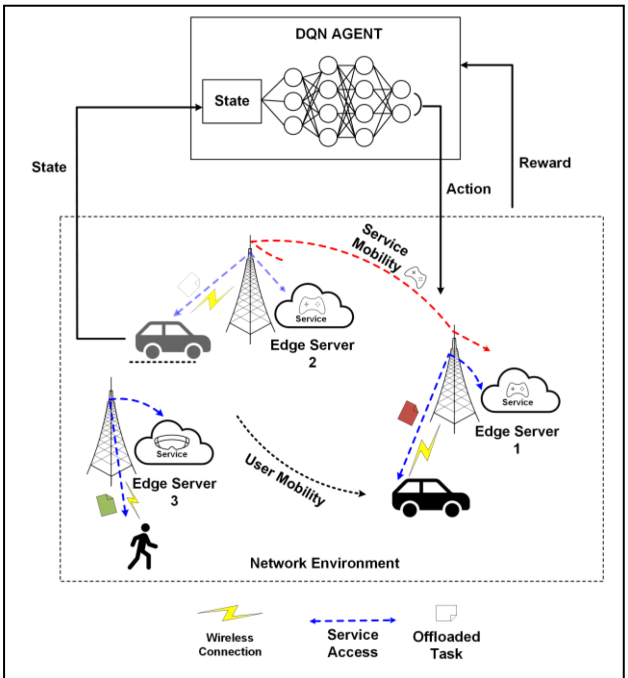

Figure 1: DQN for Service Mobility in MEC

As depicted in Figure 1, services for mobile users are deployed on a selected edge cloud to leverage the computing resources. As users move farther away from the current service Edge server, for instance, Edge Server 2, the system must decide whether to relocate the service instance to another Edge Server or continue executing in the current edge server. This decision is made by the agent, which is basically a DQN algorithm that we describe in this section. It can be observed in Figure 1 that the DQN agent decides to migrate the game service instance to Edge Server 1 where it continues to execute.

With DQN algorithm, there is an agent that focuses on learning the optimal service mobility policy by continuously interacting with the network environment. During each time step, the agent observes the current environment state and performs an action, which corresponds to the decision of whether to relocate the service instance or not. The state provided by environment captures two pieces of vital information that constitute the decision criteria. First is the distance between current location of mobile user and its service edge server. The state also contains information on available computing resources of edge servers. Upon executing an action, the agent receives a reward, which is a feedback mechanism to reinforce good actions performed by agent while penalizing actions yielding undesirable results. The reward received by the agent can be defined as:

$$\text{Reward} = \text{QoE} - \text{service\_delay}$$

4. Performance Evaluation

This section presents simulation results of the DQN based service mobility approach. We simulate the approach in a network environment that we model as a grid consisting of 36 base stations. During simulations, the number of edge servers in the network environment is varied from 6 to 36 and are randomly collocated with base stations. The number of mobile users in the network environment is fixed to 60, with each user having a single service instance on one of the edge servers. During each time slot, the user moves to a different service area where it connects to the previous edge server. Upon moving, the DQN agent decides whether to initiate service mobility or not.

Following the performed simulation experiments, we quantify the impact of varying the number of edge servers on the service downtime. As depicted in Figure 2, varying the number of edge servers in the network environment bears a positive impact on service QoE. This is because increasing

Figure 2: Impact of varying the number of Edge Servers on service downtime during service mobility.

the number of Edge Servers reduces the service downtime, which is a major contributing factor to QoE deterioration during service mobility. Deploying additional edge servers in the MEC environment increases the aggregate amount of computing resources that could be used to accommodate migrated service instances. Furthermore, as service downtime is largely affected by migration distance, increasing the number of edge servers increases the density of edge servers in the environment. This enables the agent to migrate services at shorter distances, thereby reducing the service downtime.

5. Conclusion

In this paper, dynamic service mobility with is discussed with an aim of enhancing user service quality of experience while incurring marginal service downtime. Owing to its recent advances, Deep Reinforcement Learning is exploited to learn the optimal service mobility policies. Furthermore, we present our service downtime analysis with respect to service migration in MEC environment comprising varying number of MEC servers. Through simulation experiments, it has been observed that deploying additional edge servers in the network environment reduces the downtime as the service instance is relocated to the optimal edge server.

Acknowledgement

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2020R1A2C2008447), and by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2022-2015-0-00742), and the MSIT (Ministry of Science and ICT), Korea, under the ICT Creative Consilience program (IITP-2022-2020-0-01821) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

References

[1] T. Taleb and A. Ksentini, "Follow me cloud: interworking federated clouds and distributed mobile networks," *IEEE Netw.*, vol. 27, no. 5, pp. 12–19, 2013.

[2] C. Zhang and Z. Zheng, "Task migration for mobile edge computing using deep reinforcement learning," *Futur. Gener. Comput. Syst.*, vol. 96, pp. 111–118, 2019, doi: 10.1016/j.future.2019.01.059.

[3] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015, doi: 10.1038/nature14236.