2016 IEEE 8th International Conference on Cloud Computing Technology and Science

SEGUE: Quality of Service Aware Edge Cloud Service Migration

Wuyang Zhang, Yi Hu, Yanyong Zhang, Dipankar Raychaudhuri
WINLAB, Rutgers University, North Brunswick, NJ, USA
{wuyang, yihu, yyzhang, ray}@winlab.rutgers.edu

***Abstract*—Edge cloud computing moves cloud services to the edge of the network, thereby allowing clients to access services with a significantly reduced network delay. This service migration is intended to enable a range of latency sensitive mobile applications. In this paper, we propose to manage user QoS by actively migrating services to different edge clouds in response to degraded server or network performance. Previous studies have proposed a distance-based Markov Decision Process (MDP) for optimizing migration decisions. These models provide the feasibility of applying MDP to edge cloud service migration decisions. However, these models fail to consider dynamic network and server states in migration decisions. In this work, we address these limitations by designing a comprehensive edge cloud migration decision system, which we call SEGUE. SEGUE achieves optimal migration decisions by providing a long-term optimal QoS to mobile users in the presence of link quality and server load variation. The basis of SEGUE is in its QoS-aware service migration and its state based MDP model which effectively incorporates the two dominant factors in making migration decisions: 1) network state, and 2) server state. An evaluation of SEGUE performance is given through an augmented reality application. Our results demonstrate that SEGUE reduces the response time of this application by 27.21% and 53.70% compared to the lowest load migration model and the least hop migration model, respectively.**

I. INTRODUCTION

The rapid increase in demand for mobile devices within the realm of augmented reality and gaming applications motivate the need for real-time mobile cloud services. These real-time, mobile applications necessarily require low latencies to provide seamless end-user interaction, as well as intensive computation spanning large numbers of datasets. Augmented reality applications that use head-tracked systems, for example, require end-to-end latencies to be less than 16 ms [1]. Cloud-based virtual desktop applications require end-to-end latency below 60 ms if they are to match QoS of local execution [2]. Remotely rendered video conference, on the other hand, demand end-to-end latency below 150 ms [3]. Limited battery life, computation and storage capacity constraints inherent to mobile devices mean that application executions must be offloaded to cloud servers, which then return processed results to the mobile devices through the Internet. When cloud servers reside in remote data centers, end-to-end communication may translate into long delays characteristic of multi-hops transmissions over the Internet. Moving cloud computing to the edge of a network has helped to lessen these otherwise unacceptable delays while leveraging the benefits of a high-performance cloud, e.g., Fog Computing [4], Cloudlets, [5] and Follow Me Cloud [6]. While this improvement is not trivial, delivering cloud services from the edge of the network is not, by itself, sufficient to meet latency QoS. When a mobile client initially secures a one-hop away edge cloud server to ensure the shortest network delay, client mobility may cause the server to be be multi-hops away. The increased network distance, and the potential bottleneck bandwidth that might be introduced by the intermediate links may result in poor connectivity to the cloud service. Even when a mobile user moves around the originally connected edge cloud, service latency may increase because of unexpected crowds of mobile clients seeking to connect to the same edge cloud simultaneously. Thus, increased network or server processing delays may violate acceptable latency QoS constraints.

Cloud service migration may effectively provide expected QoS with respect to user mobility, dynamic networks and varying edge cloud states. To date, previous CloudNet [7] and VM Handoff [8] studies introduced virtual machine (VM) migration in real time under the assumption that the all-important variables of when and where to migrate were known. These assumptions cannot be made in the real world for two reasons. First of all, conditions that may or may not trigger migration of a cloud service may vary widely. One central consideration that must be accounted for is the tradeoff between the cost of migration and any real QoS improvement. Secondly, we need to quantify long-term performance of cloud servers with respect to any requested service migration to ensure that the best server is chosen, wherein that best choice is realized by the maximization of promised QoS for any mobile user over time.

Previous work [9], [10] proposed a static distance-based MDP model that solved the problem of where to migrate by defining each edge cloud migration possibility according to hop counts between it and a mobile user. A cost/reward function in MDP may be used to measure the trade-off between migration costs and performance gains, and it may be used to calibrate the long-term performance improvements by each edge server with respect to any prospective mobile client. Therefore, these studies did work to prove the feasibility of applying MDP with respect to the where to migrate decision. But the static distance-based MDP models did not fully support real-time mobile applications due to its inherent limitation to reflect the two ruling factors in any where to migration decision: 1)network state, and 2) server state.

The inherent limitation of previous distance-based models to reflect the two ruling factors in any where to migration decision can be illustrated in the fairly common instance of two edge clouds deemed identical because they both share an equal

2330-2186/16 $31.00 © 2016 IEEE
DOI 10.1109/CloudCom.2016.58

hop count to a mobile user. Yet, no two edge clouds are ever precisely identical because they are characterized by different network delays and different server processing delays. These differences provide a real difference in the choice between one edge cloud and another of equal hop count. For example, one edge cloud may become heavily loaded and congested because it has the smallest hop count for the client, and, accordingly, be the chosen destination for a specific migration. Another unaddressed problem exists, because a previous MDP edge cloud service migration model recalculates optimal edge cloud migration for mobile users without specifying an optimal interval period for such recalculation. Since running MDP is a computing intensive task, short recalculation intervals introduce the heavy overhead to the server. Conversely, longer recalculation intervals may translate into lazy migration resulting in periods of transgression of QoS guarantees. Finally, of course, operating MDP for edge cloud migration requires real-time feeds of pertinent parameters into the MDP model, while previous MDP models assume the parameters as static. Such assumptions make these models impractical, if not impossible, to apply to dynamic applications, network states or server states in the real world of real time.

The SEGUE system is designed to reliably answer those two critical migration questions of when and where. The state based MDP proposed in SEGUE separately tracks each edge cloud on its dynamic network states and edge cloud server states, the two states most prominently affecting response time to users, to arrive at target server determination for any service migration. Specifically, the state based MDP in SEGUE devises a novel reward function to accurately assess the real time network states and server states in the calculation of the migration cost and the long-term performance improvements from each server, when that serve is designated as the target server of the migration. Consequently, SEGUE is able to choose the optimal server for a mobile clients serve migration considering the network and server states. SEGUE adopts a QoS aware scheme to activate the MDP model when a QoS violation is predicted to solve for the when to migrate variable. Two components of SEGUE work together to achieve this. A state collection module monitors and collects key parameters continuously, parameters representing network states, server workloads and user mobility. The other component of the SEGUE QoS aware scheme to determine the When to challenge is a QoS prediction module. The QoS prediction module uses state collection module inputs to predict QoS. When a possible QoS violation is detected, the QoS prediction module compels the SEGUE system to run the state based MDP which then selects a target server, accordingly. This allows SEGUE to avoid unnecessary migration costs and bypass any possible QoS violations. We evaluated SEGUE to demonstrate its practicality through a real-time application with a real mobility trace from 320 taxis in Rome [11].

The remainder of this paper is structured as follows: Section II provides a SEGUE system overview. Section III present a detailed design of the SEGUE system; a performance evaluation is found in Section IV. Section V introduces background and reviews related works. The conclusions that follow are presented in Section VI.

II. System Overview

SEGUE's primary objective is the consistent meeting of reliable QoS standards for mobile users utilizing optimal service migration in real time scenarios of dynamic network and edge cloud server states. Four functional modules shown in Figure 1 collaborate in synchronized fashion within SEGUE to make this happen. These four modules, in order of initiation and timing, are, first, **the State Collection Module**, which gathers and stores real time network states, edge cloud workloads, and client mobility patterns, variables required as known inputs into the other system modules to provide reliable QoS predictions as well as the optimization of migration decision making, central to achieving SEGUEs objective. Since the response time of any application services is an implicit measure of dynamic network states as well as edge cloud server workloads, we first capture a current response time for each client, and, through SEGUE processing, then forecast expected response times given possible client movement and connection to potential edge clouds. We refine a hybrid push/probe technique [12] to capture the current service time for each application at each server and the network latency for users in each geographical region with respect to every accessible edge cloud in that region. The math behind the collection technique is thoroughly discussed in Section III. A mobile trace of each user is drawn and an individual mobility pattern extrapolated. The State Collection Module also records average down time of an application in service migration, as well as the service request frequency for the required input parameters in the calculation of the state based MDP model.

Once the network and service latency are known, SEGUE estimates the expected response time for any user moving around a geographical region with access to an edge cloud under dynamic network states and varying edge cloud workloads. **The QoS Prediction Module** monitors and predicts QoS whenever a user moves, or there are changes in server or network workloads. The QoS Prediction Module determines when to migrate. This occurs by detecting possible QoS violations. Specifically, if predicted QoS exceeds a given QoS threshold, the QoS Prediction Module triggers a search for a targeted server providing optimal QoS from another edge cloud through service migration. The detailed technique for QoS prediction is discussed in Section III.

As implied by its name, **the Edge Cloud Selection Module** uses outputs provided by the State Collection Module and the QoS Prediction Module to solve the problem of finding an optimal edge cloud to migrate to. Importantly, the Edge Cloud Selection Module provides a long-term best QoS to a mobile user from the selected edge cloud. In short, the Edge Cloud Selection Module runs MDP and returns an optimal edge cloud. The detailed procedure of the manner in which MDP provides that information is discussed in Section III.

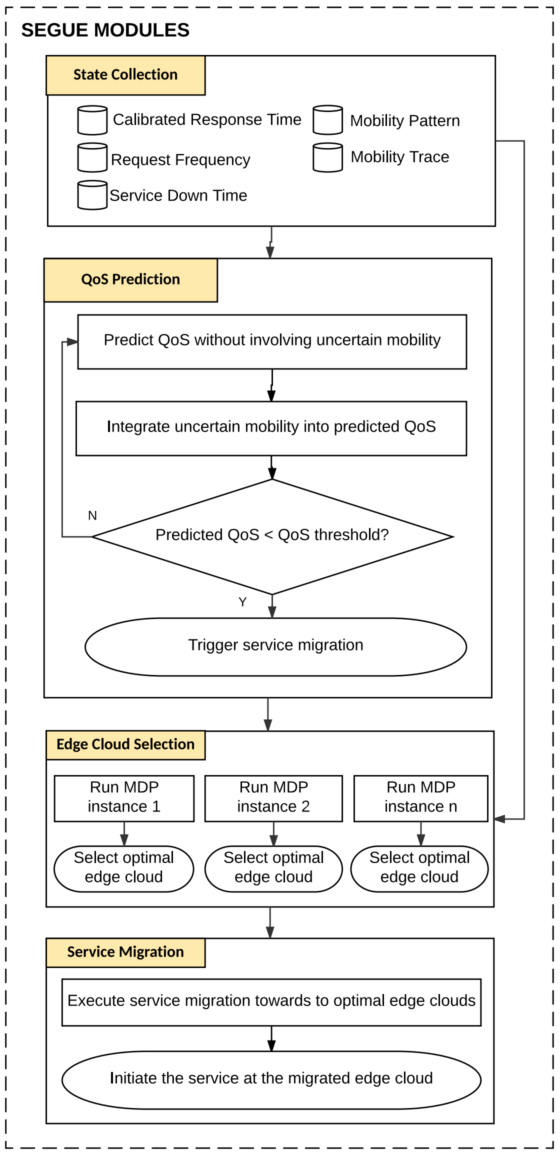

SEGUE MODULES
State Collection
Calibrated Response Time
Mobility Pattern
Request Frequency
Mobility Trace
Service Down Time
QoS Prediction
Predict QoS without involving uncertain mobility
Integrate uncertain mobility into predicted QoS
N
Predicted QoS < QoS threshold?
Y
Trigger service migration
Edge Cloud Selection
Run MDP instance 1
Run MDP instance 2
Run MDP instance n
Select optimal edge cloud
Select optimal edge cloud
Select optimal edge cloud
Service Migration
Execute service migration towards to optimal edge clouds
Initiate the service at the migrated edge cloud

Fig. 1: SEGUE system modules

The Service Migration Module actually performs the migration of service applications to the new, targeted edge cloud server. This module transfers VM states from the source edge cloud server to the selected target. The application service in the targeted edge cloud server is then initialized. Note that in the exceptional case in which a mobile user initially joins the network and then requests an application service, this module prepares the application environment and initializes the application VM on the edge cloud server. In this case alone, the best-case edge cloud is assumed to be the one of minimal hop count to the mobile user.

III. System Module Specification

This section provides a qualitative, detailed analysis of each module as they interact within SEGUE to timely find a best-case server migration to an optimal edge cloud.

A. How To Realize Network And Edge Cloud Server States?

In this section, we discuss the realization of the network and edge cloud server states through refining a hybrid approaches of server push and client proxy probe [12]. The response time of any application services is an implicit measure of dynamic network states with respects to edge cloud server workloads. Since a dynamic state directly leads to varying response times, SEGUE collects the response time to present the dynamic states. Hybrid approaches of server push and client proxy probe are refined to provide a calibrated response time to accurately reveal the dynamic states. The server push [13] refers to the server constantly monitoring and reporting response time to SEGUE's **State Collection Module**, whenever there is a response time shift that exceeds a predefined threshold. The client proxy probe corresponds to a client proxy periodically sending forth probing requests to all potential edge cloud servers and thereupon reports the response time of each prospective server to SEGUE's **State Collection Module**. The server push focuses on the application level and the service level, it clusters response time from anywhere in the network. Thereby, it reflects the server state, but cannot distinguish the network link state. This is why the client proxy probe records the response time to reflect the linked state from a client proxy to an edge cloud serve. Yet, doing so generates significant overhead with respect to both the network and edge cloud servers.

The hybrid approach, meanwhile, combines the server push and the client proxy probe through calibrating the response time reported by the server with respect to the response time reported by the client proxy probe. We assume the network is divided into N geographical regions and any mobile resides at a region $n \in [0, N-1]$ at time i. The edge cloud server records the most recent response time, R_n, reported by client proxy probe at the time i, where the corresponding response time, S, is reported by the server push at the time i. This is where we employ a calibration factor of $A_n = R_n/S$ to reflect the network condition from the geographical region, n, to this edge cloud server. Herein, then, the i-th response time, $S(i)$ reported by the server push for period I, the time duration between the client proxy probe, is multiplied as $S(i)$ with the calibration factor A_n to estimate the response time $E_n(i) = A_n \cdot S(i)$. $E_n(i)$, then, effectively evaluates the response time from the the geographical region, n, to the edge cloud server. Importantly, this hybrid approach brings the frequency of client proxy probes down thereby reducing overhead, while providing an accurate assessment of response time in the form of the calibrated response time to the linked state of the network. This calibrated response time comprises the performance metric needed to evaluate both the network state and the edge cloud server workload.

B. How To Predict QoS To Trigger Service Migration?

The QoS Prediction Module allows SEGUE to use an accurate QoS prediction to trigger service migration, since precision timing of migration is an important feature of the cost/benefit trade-off of any edge cloud migration. There are two possible approaches of finding an optimal time to migrate, known, figuratively, as the Always migration and the Periodic Check migration. A proposed QoS-aware approach decides when to migrate service, which amounts to a measure of the trade-off between the transmission cost and the migration cost of edge cloud migration. It is, of course, possible to continuously migrate services to edge cloud providing any QoS improvement whatsoever. However, this approach burdens the user with frequent service interruptions and incurs the significant cost of network resources . To prevent the shortcomings of always migration, the previous work [9] proposed to check whether to trigger the migration with a regular period or whenever a mobile user changed locations. However, this approach prevents the system from realizing the degraded QoS and migrating the service to an optimal edge cloud before the next regular period of checking whether to migrate. To resolve this issue, SEGUE provides a QoS aware approach to decide when to migrate service.

SEGUE monitors and predicts QoS whenever a mobile user accesses the edge cloud service, triggering service migration whenever a possible QoS violation is detected. The state collection module maintains a group of calibrated response time $E_{m|n}(i)$ for any edge cloud, m. $E_{m|n}(i)$, therefore, represents QoS of edge cloud m for a mobile user located in the geographical region n. To predict QoS based on the user mobility, we integrate the linearity of expectation into the Autoregressive Integrated Moving Average(ARIMA) forecaster [14]. With a QoS $E_{m|n}(i)$, the ARIMA forecast provides a predicted QoS $R_{m|n}(i)$ after time T. Note, T defines a period that is longer than the maximum service migration time to ensure sufficient time to complete a service migration should a QoS violation be detected in the meantime. We then weight the QoS prediction by a factor of mobility uncertainty. The QoS prediction module assumes that mobile location in the geographical region, n , follows the one dimensional mobility pattern given by $(p, 1-p-q, q)$, where $p, (1-p-q)$ in which q denotes the possibility of moving back to a previous region signified by $n1$ with respect to remaining at a current region of n and moving forward to a region of $n+1$, respectively. The predicted QoS of $P_{m|n}(i)$ for a mobile user that locates in the geographical region n and connects to the edge cloud m integrated with the uncertain mobility is given by

$$P_{m|n}(i) = p \cdot R_{m|n-1}(i) + (1-p-q) \cdot R_{m|n}(i) + q \cdot R_{m|n+1}(i) \tag{1}$$

where $R_{m|n-1}(i), R_{m|n}(i), R_{m|n+1}(i)$ denotes the predicted QoS for a mobile user located in the geographical region encompassed by $n-1, n, n+1$ and connects to the edge cloud m with integrating the uncertain mobility. When the predict QoS $P_{m|n}(i)$ violates the QoS threshold of $P_{m|n}(i) > threshold$, the service migration is triggered. This QoS-aware approach to trigger the service migration maintains that SEGUE constantly

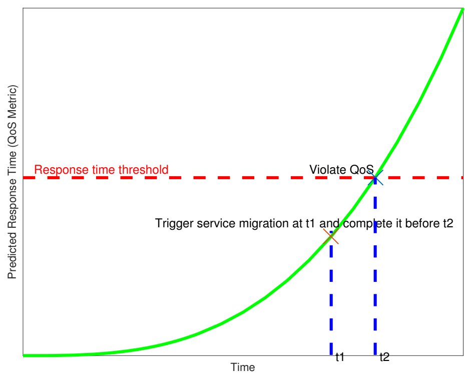


Fig. 2: Trigger service migration based on QoS prediction

monitors and predicts QoS, thereby preventing QoS violations.

C. Where To Migrate?

For optimization purposes the ”where to migrate” question is, basically, as important as the ”when to migrate” question. A state based MDP model provides that answer.

1) State based MDP formulation of edge cloud service migration: State based MDP formulation of edge cloud service migration assumes a total of M edge clouds in the network providing the service with in a range of a number of different geographical regions in which a mobile user can move, denoted by N. A mobile user moves through a geographical region, $n \in [0, N-1]$ and accesses the service from the edge cloud, $m \in [0, M-1]$ at the time t. We define this state as $S_{m|n}(t)$. When the QoS prediction module triggers the migration, we consider a migration action $a_\pi \in [0, M-1]$ to avoid the QoS violation. The action a_π upon the state $S_{m|n}(t)$ migrates the service from the edge cloud m to the optimal edge cloud m', incurring a transit state, represented by $S_{m'|n}(t)$.

The selected edge cloud m' is anticipated to provide the optimal QoS in long-term when the mobile user moves to the geographical region n'. For the value of n', we assume a mobile user is in the geographical region n at the time t, able to move to neighbor geographical regions of $n-1$, $n+1$ or choose to remain in n at the time $t+1$. This assumption falls within a mobility possibility of $p, q, 1-p-q$, respectively. Therefore, considering an uncertain mobility based on migrating the service to the edge cloud m', the final transit state at the time $t+1$ is $S_{m'|n-1}(t+1)$, $S_{m'|n+1}(t+1)$ and $S_{m'|n}(t+1)$ with the transition possibility of $p, q, 1-p-q$. The complete state transition for the state $S_{m|n}(t)$ with the action a_π is given by

$$S_{m|n}(t) \xrightarrow{a_\pi} S_{m'|n}(t) \longrightarrow \begin{cases} S_{m'|n-1}(t+1) & p \\ S_{m'|n+1}(t+1) & q \\ S_{m'|n}(t+1) & 1-p-q \end{cases} \tag{2}$$

2) Performance Objective: Given a state $S_{m|n}(t_0)$ at an initial time t_0, we search for an optimal action a_π that migrates the service to the edge cloud m' in order to provide long-term optimal QoS. To accomplish this, we introduce a novel reward function to measure the trade-off between the accumulative QoS improvement to determine when to migrate the service to the edge cloud m' and the migration cost(service down time). The reward function $G_{a_\pi}(t_1)$ is given by

$$G_{a_\pi}(t_1) = \left[R_{m|n'}(t_1) - R_{m'|n'}(t_1)\right] \cdot f - d_{avg} \tag{3}$$

where $t_1 = t_0 + 1$, t_1 denotes a constant period after the initial time t_0. Significantly, $R_{m|n'}(t_1)$ denotes the predicted QoS for a mobile user in the geographical region n' accessing the service from the original edge cloud m, $R_{m'|n'}(t_1)$ denotes the predicted QoS for a mobile user in the geographical region n' who accesses the service from the migrated edge cloud m'. The variables f denotes the request frequency, while d_{avg} denotes the average service down time. Note that f and d_{avg} are imported from **the State Collection Module**, while $R_{m'|n'}(t_1)$ and $R_{m'|n}(t_1)$ are imported from the **the QoS Prediction Module**. In Equation 3, $\left[R_{m'|n'}(t_1) - R_{m'|n}(t_1)\right] \cdot f$ measures the accumulative QoS improvement at the time t_1 of when to migrate the service from the source edge cloud m to the target edge cloud m', and the d_{avg} measures the service down time duration that mobile user suffers due to the service migration. Thus, the reward function quantifies the trade-off in the service migration at the time t_1. The performance objective is achieved by finding an optimal action that maximizes the reward in the long-term, not merely at the time t_1. The long-term reward function $V(S_{m|n}(t_0))$ is given by

$$V(S_{m|n}(t_0)) = \sum_{t=t_0}^{\infty} \gamma^t \cdot G_{a_\pi}(t_1) \tag{4}$$

where $\gamma(0 \leq \gamma < 1)$ is the discount factor that converges into Equation 4. Equation 4 measures the sum of the reward in the long-term beginning at the state $S_{m|n}(t_0)$ based upon the action a_π. We convert the cumulative sum definition of the long-term reward given by Equation 4 into the recursive definition given by

$$V^*(S_0) = max\left[\sum_{S_1} P_{a^*_\pi(S_0)\to S_1} \cdot G_{a^*_\pi(S_1)} + \gamma \cdot V^*(S_1)\right] \tag{5}$$

where $P_{a^*_\pi(S_0)\to S_1}$ denotes the transition probability from the initial state $S_0 = S_{m|n}(t_0)$ to the final transit state $S_1 = S_{m'|n'}(t_1)$. $\sum_{S_1} P_{a^*_\pi(S_0)\to S_1} \cdot G_{a^*_\pi(S_1)}$ measures the weighted sum of the QoS improvements with the different final transit states to enable consideration of uncertain mobility. Equation 5 recursively defines the maximal long-term reward $V^*(S_0)$ from the beginning state $S_0 = S_{m|n}(t_0)$ based on the optimal action a^*_π. Equation 5 can be calculated by Bellman's value iteration [15] and outputs the optimal action $a^*_\pi = m' \in [0, M-1]$ for each state $S_0 = S_{m|n}(t_0)$. Therefore, with Bellman's value iteration process, we can obtain an optimal action set A^*_π that involves $M \cdot N$ different optimal actions a^*_π for $M \cdot N$ different states. Each optimal action is mapped to a corresponding state, represented by $S_{m|n}(t_0), m \in [0, M-1], n \in [0, N-1]$. When a mobile user's the predicted QoS violates the QoS threshold at the time t_0, **the Edge Cloud Selection Module** runs MDP by incorporating the real-time QoS prediction $R_{m|n}(t_1)$ into the reward function, through inputting user's connected edge cloud m and the current geographical region n to MDP. MDP finds the corresponding optimal action $a^*_\pi = m'$ in the optimal action set A^*_π for this state $S_{m|n}(t_0)$.

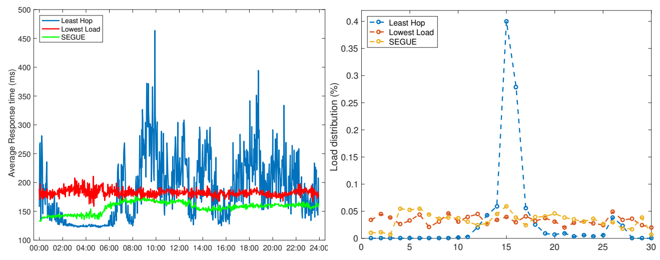
Then **the Edge Cloud Selection Module** passes this optimal action into **the Service Migration Module** that migrates the service from the source edge cloud m to the optimal edge cloud m', accordingly. Note the optimal action set A^*_π is not static, since the reward function in MDP involves the real-time predicted QoS that captures the dynamic link states and the edge cloud server loads. The optimal action set A^*_π will be updated whenever SEGUE runs MDP with the input of the real-time predicted QoS. Our proposed MDP model reflects the dynamic link states and the edge cloud server loads. Therefore, the dynamic optimal actions generated by MDP maintains the long-term best QoS in real-time.

IV. Performance Evaluation

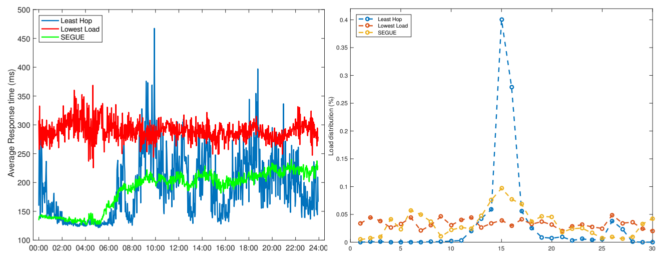
As we have provided a solid overview and detailed understanding of SEGUE, Section IV evaluates SEGUE performance using an augmented reality application applied to a real world scenario. The augmented reality application is a representative mobile application that has the two-fold stringent criterion of low latency and high bandwidth. Our evaluations reliability is assured through the real mobility trace of 320 taxis in Rome. Our evaluation itself is realized through an end-to-end response time of the augmented application and workload distribution among edge clouds. We also compare the performance of SEGUE with two other edge cloud service migration decision models: lowest load edge cloud migration, and least hop edge cloud migration, which are designed to provide optimal QoS when to specifically consider the server load states and the network states.

A. Experiment Set Up

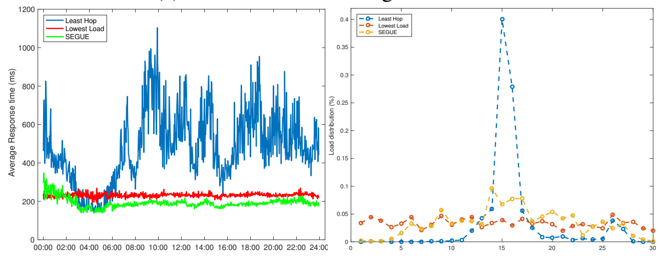
1) Simulation Of Edge Cloud: 30 edge clouds with the identical computation and storage capacity were deployed to provide the service of augmented reality for 320 taxis in Rome. Taxis' activity areas were partitioned into 30 geographical regions. Accordingly, one edge cloud has been placed in each geographical region. Implementation of this application on the Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT) worked to provide our QoS estimation of the augmented reality application supported by SEGUE under both dynamic network and edge cloud states. The deployed node on ORBIT is equipped with Intel(R) Core(TM) i7-3930K CPU 3.20GHz, 4GB memory and 500GB disk. To simulate the



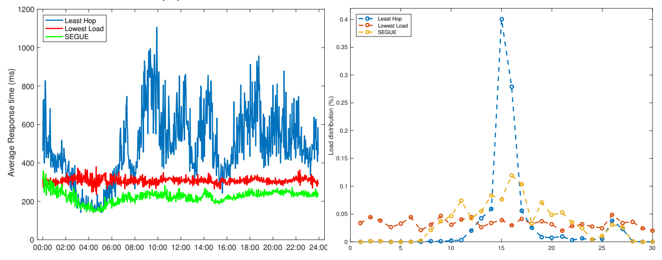
(a) low workload and high bandwidth



(b) low workload and low bandwidth



(c) high workload and high bandwidth



(d) high workload and low bandwidth

Fig. 3: performance evaluation of the average response time and the workload distribution

dynamic states, we varied the network states through software-based shaping the link bandwidth B, and shifted the workload L through modifying the available computing resources used for the augmented reality service. A database that records the corresponding response time r (QoS metric) with a specific link bandwidth B and workload L was built. In the simulation of the augmented reality application in SEGUE itself, edge cloud servers input the link bandwidth B and the workload L into the database, which then returns the corresponding response time r according to the inputs.

2) Simulation Of Mobile User: A mobile user requests the service of an augmented reality application from the edge cloud assigned by SEGUE. That user is located and mobile in one of 30 geographical regions, changing geographical regions in 2 minute intervals with a specific mobility pattern. To simulate movement with a real trace, we adopt the mobility trace of 320 taxis in Rome that records the GPS location every 7 seconds. To incorporate this real trace into the MDP model and to reduce the computing complexity of the MDP model, we process the trace data to obtain the one-dimension mobility trace and, thus, the mobility pattern for each taxi. We map GPS locations into geographical region indexes to collect the one-dimension mobility trace. Every GPS location (x, y) initiates a distance calculation with respect to edge clouds, where d equals the distance between it and a corresponding edge cloud with an origin of point (x_0, y_0). Accordingly, $d = \sqrt{(x - x_0)^2 + (y - y_0)^2}$. Then we calculate the distance unit $d_{unit} = (d_{max} - d_{min})/30$, where d_{max} and d_{min} are the maximal and the minimal distance. GPS location (x, y) is mapped into a geographical region index $n \in [0, 29]$ by $n = d/d_{unit}$. An one-dimension mobile pattern $(p, q, 1 - p - q)$ is then extracted for the 320 taxis, where $p = C_{left}/C, q = C_{right}/C$, C denotes the total movement times, C_{left} denotes the left movement times and C_{right} denotes the right movement times for each taxi.

3) Simulation Of SEGUE: The SEGUE system constantly monitors the current QoS of a mobile user to predicts future QoS. When the predicted QoS violates the QoS threshold, SEGUE runs a migration decision algorithm to migrate the service. When the mobile user initially requests for the service, SEGUE assigns the edge cloud with the minimal hop count to the mobile user. Then, SEGUE migrates the service to an edge cloud according to the migration decision in operation at the time. To compare the performance of the state based MDP in SEGUE, we also implement another two algorithms: 1)A lowest-load edge cloud migration and 2) A least hop edge cloud migration. The lowest workload edge cloud migration obviously works under the assumption that service will be migrated to the edge cloud with the lowest workload, while the least hop edge cloud migration indicates that service will be migrated to the edge cloud with the minimal hop count to the mobile user.

4) Parameters Configuration: The QoS threshold is set at 200 ms, the service down time duration as 2 seconds, and the request frequency as 300 requests in every 2 minutes.

B. Experimental Results

To evaluate the performance of the SEGUE system in varying states to demonstrate its adaptability to available computing resources and network bandwidths, we created different scenarios by varying the link bandwidth B and the edge cloud server workload L. Two aspects of the performance were evaluated: 1) the average response time of all mobile users calculated in 2 minute intervals, and 2) the workload distribution of all edge clouds calculated for a whole day. The states of link bandwidth B are divided into two categories: high bandwidth–with the normal distribution of 60 to 100 Mbps; and low bandwidth– with the normal distribution of 10 to 30 Mbps. The states of edge cloud server workload L

are divided into two categories: high workload– wherein 40% to 70% computing resources are available to the augmented reality service, low workload wherein 80% to 100% computing resources are available to the augmented reality service.

In each state set, we analyze the average response time of all mobile users in every 2 minutes and workload distribution among 30 edge cloud servers in a whole day, using the SEGUE system for edge cloud migration purposes, as well as the lowest workload migration, the least hop migration. Figure 3a represents the standard QoS provided by these three edge cloud service migration models with the low workload and the high bandwidth. The average response time provided by SEGUE, the lowest workload migration and the least hop migration is 156.93ms, 182.28ms and 187.81ms, respectively. SEGUE improves QoS by 13.90% and 16.44% compared to the other two migration models at the low workload and the high bandwidth.

The merit of SEGUE is self-evident in comparison to either the high workload or the low bandwidth states. Due to cluster effects of this real mobile trace, an edge cloud server will be in high workload, making QoS significantly and adversely impacted, when the migration models such as the least hop migration model are unable to balance the workload among edge clouds. Such clustering is a fairly common scenario for end-point mobility; i.e., when crowds of taxis may either depart from a taxi company simultaneously or pick up at approximately the same time to deliver clients to popular scenic areas. Figure 3c, which illustrates performance when bandwidths are kept high and the workload of edge cloud servers are boosted, proves this point. The average response time provided by SEGUE, the lowest workload migration and the least hop migration is 200.82ms, 239.37ms and 481.40ms. SEGUE improves QoS by 58.28% compared to the least hop migration models during a time of high workload and high bandwidth. The result proves SEGUE's capability of adapting to the dynamic edge cloud server states. We then considered the impact of dynamic network states on QoS by keeping the workload low, while reducing the available bandwidth. Figure 3b shows the average response time provided by SEGUE, the lowest workload migration and the least hop migration: 188.84ms, 289.71ms and 190.71ms. SEGUE improves QoS by 33.76% compared to the lowest load migration model. QoS provided by SEGUE and the least hop migration model maintains stability due to the consideration of the network state. The result proves SEGUE's capability of adapting to the dynamic network states.

The lowest load migration model and the least hop migration model perform well at the restricted states, states for which these two models were designed to provide optimal QoS. However, both the edge cloud server and the network states are dynamic. With considering the both two dynamic states, SEGUE presents its merits from the above scenarios. Importantly, regardless of the specific scenario under observation, SEGUE outperforms its closest competitor by tightening QoS in all cases. Figure3d shows the performance that sets the states as the high workload and the low bandwidth, where two other migration models except SEGUE fails to consider both two dynamic states. The average response time provided by SEGUE, the lowest workload migration and the least hop migration: 224.04ms, 307.80ms and 483.86ms, respectively. SEGUE improves QoS by 27.21% and 53.70% at this condition. SEGUE recognizes, by design, that both the edge cloud server and network states are dynamic. Exclusive consideration of one state invariably fails to maintain the optimal QoS in the other state, both of which comprise two independent dynamic real-time states. SEGUE proves its adaptive nature as it smoothly integrate both dynamic server and the network states into one best-case migration solution, thereby maintaining the optimal QoS of edge cloud service.

Besides considering the response time as the QoS metric, workload balance, from the perspective of edge cloud, is also critical to be paid attention. It ensures the fair and efficient distribution of computing resources. To observe the performance of load balance, the workload distribution of all edge clouds calculated for a whole day has been recorded. It is well-known that the lowest load migration model is specifically designed to guarantee the fair workload balance. Figure 3 shows in any conditions, the workload distribution provided by SEGUE performs closely with that provided by the lowest load migration model. Thus, SEGUE is capable of fairly balancing workload among edge clouds.

V. Background and related works

Mobile devices have already replaced fixed-hosts/servers as the primary Internet platforms. Constrained by the battery life, processor speed, memory size and storage capacity; however, the real-time mobility of mobile devices may be best served by leveraging remote high-performance clouds over the Internet to execute their sundry resource-intensive applications [16]. However, WAN latency and bandwidth-induced delay over the Internet present fundamental obstacles to QoS optimization of user-interactive applications. To overcome the high network delay, while leveraging a high-performance cloud, [4], [5], [6] moving cloud computing to the edge of the network has been proposed with proven benefits. Edge cloud computing alleviates otherwise high network delays, while delivers resource-rich computing with a reasonable physical proximity to mobile user as well as continuing, in most case, one-hop network access. Nevertheless, despite its immediate benefits, edge cloud is the small-scale solution supporting nearby mobile devices, while QoS becomes more sensitive to the load of any particular edge cloud. This is where SEGUE comes in as a guarantor of QoS through edge cloud service migration.

When excessive network congestion caused by a large number of mobile clients connecting to a nearby edge cloud, computing and storage resources of the edge cloud are effectively exhausted when it comes to satisfying a promised QoS. Moreover, over the course of the client movements, an original one hop edge cloud server turns out to be multi-hops away and the ensuing intermediate links may introduce a bottleneck bandwidth that results in poor network conditions [8]. The increased network delay incurred by the user mobility catalyzes the failure of meeting a promised QoS. Cloud service migration cleanly resolves these two challenges. Previous works [17], [8], [7]propose real-time service migration techniques under the assumption that the critical questions of when and where to migrate are known conditions. In real-time applications, however, the instant and the destination of service migration is not straightforward to perceive. Up until now, the traditional cloud migration decision models [18], applied in data centers regarding preventing cloud bursting, reducing economic costs, maximizing the usage of computing resources(such as CPU, memory, and storage capacity) and bandwidth resources have been the primary migration variables in a migration decision-making model. However, user mobility turns out to be a new critical factor considering in any edge cloud service migration decision model. Traditional cloud migration decision models inadequately integrate into edge cloud service migration. Recent works [9], [10] propose a distance-based MDP migration decision model to incorporate user mobility as one of migration decision factors. However, a strictly distance-based MDP model fails to consider both dynamic network and edge cloud load conditions. In this paper, we propose a comprehensive edge cloud migration decision system, SEGUE, to address the aforementioned problem.

VI. CONCLUSIONS

Achieving optimal Edge Cloud Service Migration presents a multi-faceted problem with respect to both when and where to migrate service. The SEGUE system is designed to resolve this problem by using four individual processing modules. SEGUE's State Collection module collects edge cloud network states and server workload, and a reliable predicted QoS that embraces a degree of mobile uncertainty is achieved through the QoS Prediction Module. SEGUE's third critical module, the Edge Cloud Selection Module chooses a best Edge Cloud to migrate with using the aforementioned variables provided by the State Collection and QoS Prediction Module as inputs. Once a best Edge Cloud is selected, the Service Migration Module migrates the service from the source edge cloud to the optimal edge cloud, as previously discussed.

The basis of SEGUE is in its QoS-aware service migration and its state based MDP model which effectively incorporates the two dominant factors in making migration decisions: 1) network state, and 2) server state. We carefully evaluate the real-time performance of SEGUE through an augmented reality application, along with a real mobility trace of 320 taxis in Rome. SEGUEs performance was compared to the lowest load migration model and the least hop migration models. Our results demonstrate that SEGUE reduces the response time of this service application by 27.21% and 53.70% compared to the two service migration models. We believe that the proposed technique will enable real-time cloud applications in future mobile environments. Future work will consider experimental proof-of-concept validation of the simulation based results provided here and also evaluate the resource cost of this proposed system.

REFERENCES

[1] Stephen R Ellis, Katerina Mania, Bernard D Adelstein, and Michael I Hill. Generalizeability of latency detection in a variety of virtual environments. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 48, pages 2632–2636. SAGE Publications, 2004.

[2] Brandon Taylor, Yoshihisa Abe, Anind Dey, Mahadev Satyanarayanan, Dan Siewiorek, and Asim Smailagic. Virtual machines for remote computing: Measuring the user experience, 2015.

[3] Tim Szigeti and Christina Hattingh. *End-to-end qos network design*. Cisco press, 2005.

[4] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[5] Grace Lewis, Sebastián Echeverría, Soumya Simanta, Ben Bradshaw, and James Root. Tactical cloudlets: Moving cloud computing to the edge. In *2014 IEEE Military Communications Conference*, pages 1440–1446. IEEE, 2014.

[6] Tarik Taleb and Adlen Ksentini. Follow me cloud: interworking federated clouds and distributed mobile networks. *IEEE Network*, 27(5):12–19, 2013.

[7] Timothy Wood, KK Ramakrishnan, Prashant Shenoy, and Jacobus Van der Merwe. Cloudnet: dynamic pooling of cloud resources by live wan migration of virtual machines. In *ACM Sigplan Notices*, volume 46, pages 121–132. ACM, 2011.

[8] Kiryong Ha, Yoshihisa Abe, Zhuo Chen, Wenlu Hu, Brandon Amos, Padmanabhan Pillai, and Mahadev Satyanarayanan. Adaptive vm handoff across cloudlets. Technical report, Technical Report CMU-CS-15-113, CMU School of Computer Science, 2015.

[9] Shiqiang Wang, Rahul Urgaonkar, Murtaza Zafer, Ting He, Kevin Chan, and Kin K Leung. Dynamic service migration in mobile edge-clouds. In *IFIP Networking Conference (IFIP Networking), 2015*, pages 1–9. IEEE, 2015.

[10] Adlen Ksentini, Tarik Taleb, and Min Chen. A markov decision process-based service migration procedure for follow me cloud. In *2014 IEEE International Conference on Communications (ICC)*, pages 1350–1354. IEEE, 2014.

[11] Lorenzo Bracciale, Marco Bonola, Pierpaolo Loreti, Giuseppe Bianchi, Raul Amici, and Antonello Rabuffi. CRAWDAD dataset roma/taxi (v. 2014-07-17). Downloaded from http://crawdad.org/roma/taxi/20140717, July 2014.

[12] Z-M Fei, Samrat Bhattacharjee, Ellen W Zegura, and Mostafa H Ammar. A novel server selection technique for improving the response time of a replicated service. In *INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 783–791. IEEE, 1998.

[13] James S Gwertzman and Margo Seltzer. The case for geographical push-caching. In *Hot Topics in Operating Systems, 1995.(HotOS-V), Proceedings., Fifth Workshop on*, pages 51–55. IEEE, 1995.

[14] Sangsoo Lee and Daniel Fambro. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, (1678):179–188, 1999.

[15] David L Poole and Alan K Mackworth. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 2010.

[16] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing*, 8(4):14–23, 2009.

[17] KK Ramakrishnan, Prashant Shenoy, and Jacobus Van der Merwe. Live data center migration across wans: a robust cooperative context aware approach. In *Proceedings of the 2007 SIGCOMM workshop on Internet network management*, pages 262–267. ACM, 2007.

[18] Tian Guo, Upendra Sharma, Timothy Wood, Sambit Sahu, and Prashant Shenoy. Seagull: intelligent cloud bursting for enterprise applications. In *Presented as part of the 2012 USENIX Annual Technical Conference (USENIX ATC 12)*, pages 361–366, 2012.