Cost Aware Cloudlet Placement for Big Data Processing at the Edge

Qiang Fan, *Student Member, IEEE*, and Nirwan Ansari, *Fellow, IEEE*
Advanced Networking Laboratory
Department of Electrical and Computer Engineering
New Jersey Institute of Technology, Newark, NJ, 07102, USA
Email: {qf4, nirwan.ansari}@njit.edu

***Abstract*—As accessing computing resources from the remote cloud for big data processing inherently incurs high end-to-end (E2E) delay for mobile users, cloudlets, which are deployed at the edge of networks, can potentially mitigate this problem. Although load offloading in cloudlet networks has been proposed, placing the cloudlets to minimize the deployment cost of cloudlet providers and E2E delay of user requests has not been addressed so far. The locations and number of cloudlets and their servers have a crucial impact on both the deployment cost and E2E delay of user requests. Therefore, in this paper, we propose the Cost Aware cloudlet PlAcement in moBiLe Edge computing strategy (CAPABLE) to optimize the tradeoff between the deployment cost and E2E delay. When cloudlets are already placed in the network, we also design a load allocation scheme to minimize the E2E delay of user requests by assigning the workload of each region to the suitable cloudlets. The performance of CAPABLE is demonstrated by extensive simulation results.**

I. Introduction

Recent mobile applications, such as augmented reality, on-line gaming, and image processing, are becoming computation-intensive while the resource of battery powered mobile devices remains limited. Mobile Cloud Computing (MCC) [1] is introduced to offload user tasks to a centralized data center [2], [3] via Internet and thus reduces the task execution time and energy consumption of users. However, the cloud is usually remotely located and far away from its users, and thus inherently incurs a long E2E delay between a user and the cloud. Although this E2E delay may meet the demands of some applications such as web browsing, it is unbearable for many delay sensitive applications such as augmented reality and on-line gaming. Hence, the concept of cloudlet is employed to reduce the user E2E delay by moving the remote cloud resources to the edge of network. Since cloudlets, which are tiny versions of data centers, are generally placed at access points in the network that are close to users, users can access the computing resources with a lower E2E delay.

Mobile users frequently communicate with their cloudlets by transmitting their mobile data (i.e., application loads and related mobile data streams) over time. Mobile devices, embedded with various sensors [4], have become data stream generators producing different types of user information, e.g., locations, activities, motion and health information. However, most of mobile data are time sensitive, i.e., their potential value decreases as time passes. Therefore, cloudlets located close to users can significantly facilitate the mobile big data analysis in real-time. A typical example to utilize distributed cloudlets to analyze mobile big data is the traffic detection, i.e., when a user wants to get the traffic information on the road, it sends a request and its own motion data to a nearby cloudlet. After processing the big data aggregated from the huge number of users, cloudlets can calculate the best routing information for different users in real-time.

Although the cloudlet concept is a promising technique to reduce the user E2E delay, how to place cloudlets to minimize the E2E delay as well as the deployment cost has not been addressed. The budget of a cloudlet provider is always limited. The deployment cost of a cloudlet mainly comes from renting a site and installing a certain number of servers for the cloudlet. As we know, the site rentals are geographically dynamic, and thus the location of a cloudlet poses a significant impact on the deployment cost. Meanwhile, once the location of a cloudlet is decided, the cloudlet provider still needs to determine how many servers (i.e., the amount of computing resources) to be installed in the cloudlet, according to user density (i.e., workload density) near the cloudlet. Thus, the total deployment cost of a cloudlet provider depends on the locations and amount of cloudlets and their servers. In addition, the cloudlet providers should ensure a low E2E delay to improve the performance of mobile big data analysis. Users can achieve lower E2E delay from cloudlets in their physical proximity than from cloudlets far away. If cloudlets are deployed in the region with high user density, i.e., more users are able to access the computing resources in their proximity, the total E2E delay in the network will be reduced. Furthermore, the number of cloudlets does affect the total E2E delay of user requests. If more cloudlets are placed in the network, each user is more likely to access to a closer cloudlet, thus incurring the lower E2E delay between the user and its cloudlet. In an extreme case, when cloudlets are placed at every BS, the cloudlet network provisions the lowest E2E delay between users and their cloudlets.

A cloudlet provider aims to minimize the deployment cost while improving the quality of experience (QoE) for its users, in terms of the E2E delay of user requests. In this case, only optimizing the deployment cost or E2E delay cannot meet the cloudlet provider's objective. Therefore, we propose the Cost Aware cloudlet PlAcement in moBiLe Edge computing

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

(CAPABLE) to minimize the total cost of a cloudlet network (i.e., consisting of both the deployment cost and E2E delay cost of user requests). In other words, we optimize the tradeoff between the deployment cost and E2E delay of user requests. On one hand, we need to choose strategic locations for cloudlets while reducing the number of cloudlets and their servers to reduce the deployment cost of cloudlets. To the best of our knowledge, this work is the first to arrange flexible computing resources (i.e., servers) to cloudlets while placing cloudlets at the network edge. On the other hand, we need to minimize the E2E delay between users and their cloudlets by placing more cloudlets close to the regions with high user density.

The remainder of this paper is organized as follows. In Section II, we briefly review related works. In Section III, we describe the system model. In Section IV, we formulate the CAPABLE strategy, design the load allocation scheme in the cloudlet network, and analyze the problem. Section V shows the simulation results, and concluding remarks are presented in Section VI.

II. Related Works

Most existing works focus on analyzing the big data in the remote clouds, owing to their abundant and flexible resources. However, transmitting big mobile data from users to the remote cloud suffers from prohibitively long latency and increases the burden of the network. Thus, rather than transferring the big mobile data to the remote cloud for further processing, the cloudlet concept is proposed to move the computing resources to the vicinity of users. Tawalbeh *et al.* [5] proposed a mobile cloud computing model to run the big data applications in cloudlets rather than remote clouds. Satyanarayanan *et al.* [6] proposed the GigaSight architecture to perform the video processing in the local cloudlets to reduce the latency while saving the bandwidth of core networks. Sun and Ansari [7] proposed EdgeIoT by leveraging mobile edge computing to provision internet of things, and they [8] also proposed a profit maximization Avatar placement strategy for mobile edge computing, referred to as PRIMAL, which makes a tradeoff between the E2E delay reduction and migration overheads by selectively migrating vritual machines (VMs) to their optimal cloudlets. Furthermore, Sun *et al.* [9] proposed a green energy aware Avatar migration strategy, referred to as GEAR, to migrate VMs to cloudlets with more green energy to reduce the on-grid energy consumption while ensuring low E2E delay for users.

Although the research on cloudlet has recently received much attention, few has addressed the cloudlet placement problem, which poses a crucial impact on the E2E delay. Xu *et al.* [10] formulated a capacitated cloudlet placement problem and placed K capacitated cloudlets to some strategic locations to minimize the average E2E delay between mobile users and their cloudlets. Jia *et al.* [11] proposed a model to place K cloudlets in the network and realize the load balancing among the cloudlets. However, the existing works only aim to minimize the E2E delay for users by placing a certain number of cloudlets in the network, without considering the deployment cost of cloudlets paid by cloudlet providers. Moreover, existing works assume that the capacity of each cloudlet in terms of user requests is given before the cloudlet placement, Different from the existing works, we make a tradeoff between the deployment cost and the E2E delay between users and their cloudlets. In other words, we need to minimize the E2E delay of user requests as well as the deployment cost of cloudlet providers. When placing the cloudlets, we not only choose strategic locations for cloudlets, but also determine the optimal number of deployed cloudlets. Furthermore, when the number and locations of cloudlets are decided, we also need to select the optimal number of servers for different cloudlets based on the geographical workload density around different cloudlets.

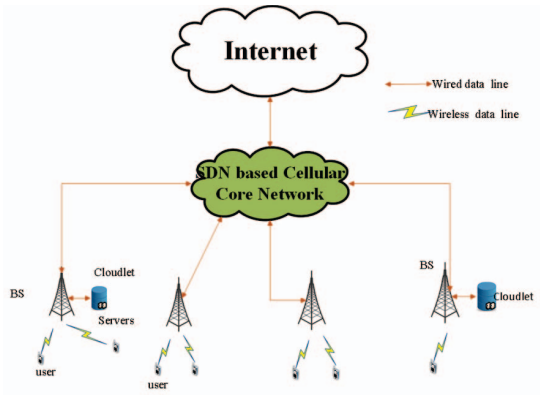

Fig. 1. Network architecture

III. System Model

A cloudlet network architecture is illustrated in Fig. 1, where cloudlets are collocated with several selected base stations (BSs). Meanwhile, the software defined network (SDN) based cellular network is employed as the cellular core network to provide efficient and flexible communications paths between BSs. Therefore, a mobile user can access its BS and then connect to a nearby cloudlet where its requests and mobile data streams can be offloaded. Based on the geographical distribution of user workloads (i.e., requests), the numbers of servers inside different cloudlets are different. The aggregated workload of a region (i.e., we define a region in the network as the area covered by a BS) can be dispatched to geographically distributed cloudlets with different capacities to minimize the total E2E delay of user requests.

Denote $\mathcal{I}$ as the set of possible positions of cloudlets (i.e., BSs) and $\mathcal{J}$ as the set of regions covered by different BSs. To indicate the locations of cloudlets in the network, we introduce a binary variable y_i, which represents whether a cloudlet is placed at BS i (i.e., $y_i = 1$) or not (i.e., $y_i = 0$). To show the assignment of each region's workloads, we introduce the continuous variable $x_{i,j}$ (i.e., $0 \leq x_{i,j} \leq 1$), where i is the index of BSs and j is the index of regions. Here, $x_{i,j}$ indicates the proportion of region j's workload that is assigned to the cloudlet at BS i. Moreover, we denote β_i (i.e., a non-negative

integer variable) as the number of servers installed in the cloudlet at BS i. Note that if no cloudlet is deployed at BS i, β_i will always be zero.

A. Deployment Cost Model

When cloudlet providers decide to deploy a cloudlet at a BS, they have to rent a facility (site), whose cost only depends on the geographical location, and then install the basic equipment. Thus, we consider this part of the deployment cost as the fixed cost, which is decided by the location of a cloudlet. In addition, cloudlet providers also need to install servers into a cloudlet to process the requests from different regions. Given the price of a server, the cost of servers in a cloudlet, which is considered as the dynamic cost, just depends on the number of servers in the cloudlet (i.e., β_i). Note that β_i should not exceed the maximum number of servers in the cloudlet at BS i, which is denoted as c_i. Therefore, we define the deployment cost of a cloudlet as the summation of the fixed cost and the dynamic cost, which can be expressed as

$$P_i = \sum_{i \in \mathcal{I}} f_i y_i + \sum_{i \in \mathcal{I}} g_i \beta_i. \tag{1}$$

Here, f_i is the fixed cost of a cloudlet at the BS i and g_i is the price of a server.

B. E2E Delay

When a request of a mobile user is sent to a cloudlet, the request goes through its BS and the SDN-based cellular core networks. Therefore, the E2E delay of the request consists of two parts: first, the E2E delay between its user and the BS, i.e., the wireless delay; second, the E2E delay between the BS and cloudlet. However, changing the locations of cloudlets does not affect the first part, which only depends on the user's service plan and the mobile provider's bandwidth allocation strategy [12]. Thus, we just consider the E2E delay between the BS and cloudlet in this paper (i.e., the E2E delay of a request is defined as the E2E delay between its BS and its cloudlet). By taking advantage of the SDN network, the SDN controller is used to measure the E2E delay between the BS in region j and the cloudlet at BS i, denoted as $d_{i,j}$ [13], [14].

As we know, user mobility incurs the spatial and temporal dynamics of workloads among different regions. In this case, to determine the optimal locations and number of cloudlets and their servers, we need to estimate the average workloads of different regions based on the historical data of user movements. Mobile user movement often follows the repetitive pattern, i.e., a user usually commutes among several places such as home, workplace and gym for most of the time of one day [11], [15]. Thus, the average workload of a region can be expressed as

$$\lambda_j = \sum_k p_{k,j} u_k, \tag{2}$$

where u_k is the request arrival rate of user k and $p_{k,j}$ is the proportion of time when user k stays in region j.

Since user requests of region j are dispatched to different cloudlets, the total E2E delay of region j's requests, denoted as D_j, can be expressed as

$$D_j = \sum_{i \in \mathcal{I}} \lambda_j x_{i,j} d_{i,j}. \tag{3}$$

IV. Problem Formulation and Analysis

In order to achieve the minimum E2E delay of user requests, a cloudlet provider should assign the requests to the closest cloudlets. If the number of cloudlets and their servers is increasing, the E2E delay of user requests will be reduced accordingly. Consequently, the cloudlet provider has to invest more on deploying the cloudlets. However, from the cloudlet provider's perspective, we aim to use the minimum deployment cost to achieve the lowest E2E delay of user requests. Thus, only optimizing the deployment cost or E2E delay cannot meet the provider's objective. Therefore, we need to design an optimal cloudlet placement strategy to optimize the tradeoff between the deployment cost and the total E2E delay of user requests for the cloudlet network. Denote ρ as the total cost of a cloudlet network, which is defined as the sum of the deployment cost and the delay cost.

$$\rho = \sum_{i \in \mathcal{I}} f_i y_i + \sum_{i \in \mathcal{I}} g_i \beta_i + \gamma \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \lambda_j x_{i,j} d_{i,j}. \tag{4}$$

Note that γ is the cost coefficient that maps the E2E delay to the delay cost. It is modeled as follows:

$$\gamma = \frac{\sum_{i=1}^{p} f_i^{\max} + g_i c}{\sum_j \lambda_j d_j^{\max}} \times \frac{\eta_2}{\eta_1}. \tag{5}$$

Here, $d_j^{\max}$ indicates the E2E delay between the BS of region j and its farthest cloudlet (i.e., the maximum E2E delay for requests of region j), p is the maximum number of cloudlets that can be deployed in the network, and c is the maximum number of servers in a cloudlet. Furthermore, $\sum_{i=1}^{p} f_i^{\max} + g_i c$ represents the highest deployment cost, i.e., the maximum number of cloudlets and servers are placed at BSs with the highest land prices. Meanwhile, $\sum_j \lambda_j d_j^{\max}$ represents the maximum E2E delay of user requests (i.e., when all requests of each region are served by the farthest cloudlets). In order to set different tradeoff relations between the deployment cost and E2E delay of user requests, we introduce two tradeoff coefficients: η_1 and η_2 for the deployment cost and E2E delay, respectively, where $\eta_1 + \eta_2 = 1$ and $\eta_1, \eta_2 \in (0, 1)$. Increasing the value of η_1 would increase the ratio of the deployment cost to the E2E delay, and encourage the cloudlet provider to place fewer cloudlets. Thus, altering η_1 or η_2 can adjust the tradeoff between the deployment cost and E2E delay.

The objective of the CAPABLE strategy is to minimize the total cost of deploying a cloudlet network. Consequently, we formulate CAPABLE as follows:

$$P1: \min_{x_{i,j} y_i \beta_i} \sum_{i \in \mathcal{I}} f_i y_i + \sum_{i \in \mathcal{I}} g_i \beta_i + \gamma \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \lambda_j x_{i,j} d_{i,j} \quad (6)$$

$$s.t. \quad \sum_{i \in \mathcal{I}} x_{i,j} = 1, \forall j \in \mathcal{J}, \quad (7)$$

$$\sum_{j \in \mathcal{J}} \lambda_j x_{i,j} \leq s\beta_i, \forall i \in \mathcal{I}, \quad (8)$$

$$\beta_i \leq c y_i, \forall i \in \mathcal{I}, \quad (9)$$

$$\sum_{i \in \mathcal{I}} y_i \leq p, \quad (10)$$

$$x_{i,j} \in [0,1], \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \quad (11)$$

$$y_i \in \{0,1\}, \forall i \in \mathcal{I}, \quad (12)$$

$$\beta_i \geq 0 \ \text{ integer}, \ \forall i \in \mathcal{I}. \quad (13)$$

Here, λ_j is the average workload of region j, s is the workload capacity of a server in terms of requests, and c is the maximum number of servers in a cloudlet. Constraint (7) ensures that the workload of a region is assigned to different cloudlets. Constraint (8) imposes the workload of a cloudlet not to be more than the capacity of the cloudlet. Constraint (9) imposes the number of installed servers in a cloudlet to be less than the maximum number of servers in the cloudlet. Constraint (10) means that the deployed cloudlets should be less than the maximum number of cloudlets in the network.

It can be shown that P1 is NP-hard, and thus we can use the Mixed-Integer Programming (MIP) tool in the CPLEX solver to find the sub-optimal solution.

Note that users moving in the network result in the geographical and temporal workload dynamics of different regions. When the locations and number of cloudlets and their servers are already decided by the CAPABLE strategy, we also need to design a load allocation scheme to assign the dynamic workload of each region (i.e., λ_j^t) to suitable cloudlets in time slot t. The objective of the load allocation scheme is to minimize the total E2E delay of requests in each time slot, and is thus formulated as follows:

$$P2: \min_{x_{i,j}^t} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} d_{i,j} x_{i,j}^t \quad (14)$$

$$s.t. \quad \sum_{i \in \mathcal{I}} x_{i,j}^t = 1, \forall j \in \mathcal{J}, \quad (15)$$

$$\sum_{j \in \mathcal{J}} \lambda_j^t x_{i,j}^t \leq s\beta_i y_i, \forall i \in \mathcal{I}, \quad (16)$$

$$x_{i,j}^t \in [0,1], \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \quad (17)$$

where both y_i and β_i are already given by the CAPABLE strategy. Since Problem P2 is a linear programming problem, we can achieve the optimal solution by using the linear programming tool (LP) of CPLEX, which ensures that requests generated from each region can be assigned to optimal cloudlets to minimize the total E2E delay of user requests in each time slot.

V. Simulation Results

In this section, we simulate the proposed CAPABLE strategy in the cloudlet network. For comparison, we select other two cloudlet placement strategies, i.e., Heaviest-AP First Placement (HAF) strategy and the K-median algorithm [11]. The idea of HAF is to place cloudlets at the BSs having the heaviest workloads. Meanwhile, the K-median algorithm is to select some strategic positions for cloudlets to minimize the E2E delay between users' BSs and their cloudlets. Note that both HAF and the K-median algorithm are designed to place a fixed number of cloudlets (i.e., 8 cloudlets in this simulation).

TABLE I
System Parameters

Parameter	Value
The length of time slot	10 mins
Number of users	1000
Number of BSs	20
Number of available cloudlets	8
Fixed cost for a cloudlet at BS i, f_i	$N(500, 200)$
Price of a server, g_i	100
Maximum number of servers in a cloudlet, c	10
Workload capacity of a server, s	50

We set up a network of the topology that has 20 BSs in an area of 80 km^2. Each BS has a coverage area of 4 km^2. There are 1000 users distributed in the network. The request arrival rate for each user is determined by the normal distribution with an average of 2, and a variance of 0.5. The maximum number of cloudlets in the network is 8, while each cloudlet has 10 servers at most. The capacity of a server in terms of requests is 50. Meanwhile, the E2E delay between the BS in region j and the cloudlet at BS i is estimated to be proportional to the distance between them, i.e., $d_{i,j} = \epsilon L_{i,j}$ $(i \in \mathcal{I}, j \in \mathcal{J})$, where ϵ is the coefficient that maps the distance to the E2E delay, and $L_{i,j}$ is the distance between the BS in region j and the cloudlet at BS i. The time domain is divided into discrete time slots with each being 10 mins. Mobile users usually stay at several places, e.g., home and workplace, for most of the time of one day. Thus, we assume that each user randomly changes its position within its specific 5 regions covered by 5 BSs. The detailed simulation parameters are shown in Table 1.

We set the tradeoff coefficient $\eta_1 = 0.3$, and run the simulation. Fig. 2 shows the total cost of the cloudlet network, consisting of the deployment cost and the E2E delay cost of all user requests, for three different cloudlet placement strategies. CAPABLE achieves the lowest total cost as compared to other two strategies because it minimizes both the deployment cost and the E2E delay cost rather than just considering the E2E delay.

Fig. 3 shows the deployment cost of cloudlets for different strategies. The result indicates that CAPABLE decreases 47.5 $\%$ deployment cost as compared to the K-median algorithm.

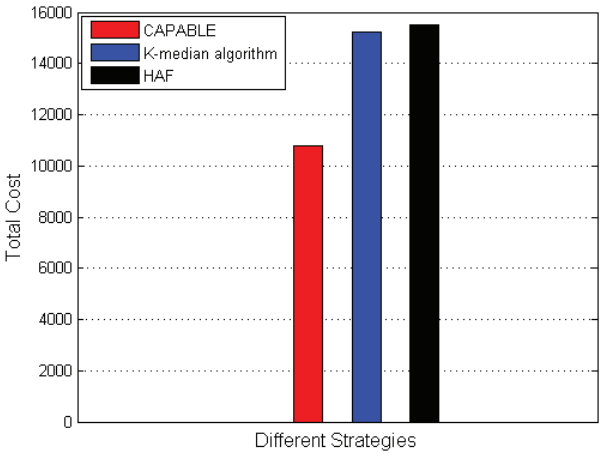

Fig. 2. Total cost comparison

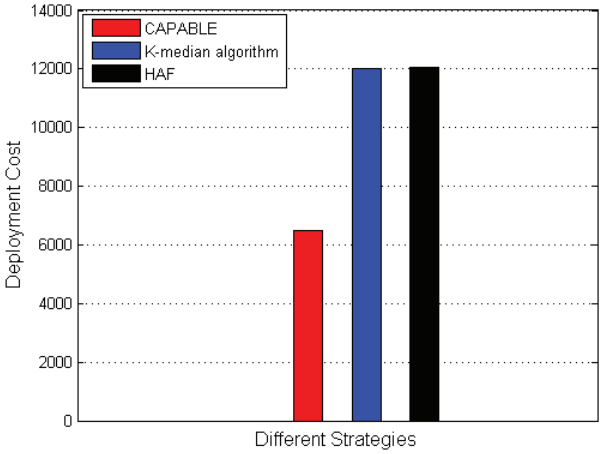

Fig. 3. Deployment cost comparison

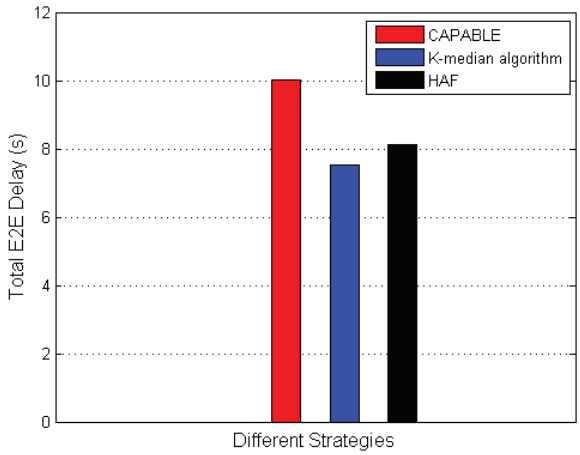

Fig. 4. E2E delay comparison

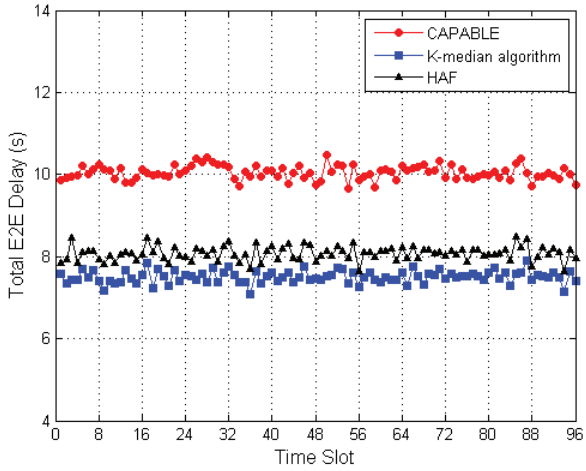

Fig. 5. Total E2E delay of user requests in each time slot

This is because CAPABLE considers the geographical dynamics of land price, and selects the suitable number of cloudlets and their servers to minimize the deployment cost.

Fig. 4 shows the total E2E delay of user requests by applying different cloudlet placement strategies. We can see that the total E2E delay of CAPABLE is increased by 29% as compared to that of the K-median algorithm. It is attributed to the fact that both K-median algorithm and HAF employ more cloudlets and servers and deploy them to suitable positions to reduce the E2E delay. In contrast, CAPABLE has fewer cloudlets to serve the geographical distributed workloads, and thus its E2E delay is higher than other two strategies. After cloudlets are placed in the network, we use the proposed load allocation scheme to assign user requests to different cloudlets to minimize the E2E delay in each time slot. Fig. 5 shows the dynamic E2E delay in each time slot for different strategies, when users move among different regions. We can see that the E2E delay of CAPABLE is higher than other two strategies. Although CAPABLE sacrifices a bearable amount of E2E delay, it significantly reduces the deployment cost for cloudlet providers.

Fig. 6 and Fig. 7 illustrate the deployment cost of cloudlets and the total E2E delay of user requests for the CAPABLE strategy with different η_1, respectively. A smaller η_1 indicates that the cloudlet placement is more delay sensitive. As a result, CAPABLE focuses on achieving the lower E2E delay by deploying more cloudlets and servers in the network, which incur higher deployment cost. When η_1 increases, CAPABLE starts to pay more attention to the deployment cost. As a result, it takes the locations and number of cloudlets and servers into account to reduce the deployment cost while sacrificing a small amount of E2E delay. From the perspective of a cloudlet provider, we can adjust the tradeoff relation between the deployment cost and E2E delay of user requests by selecting a suitable tradeoff coefficient η_1, based on the requirement of the network.

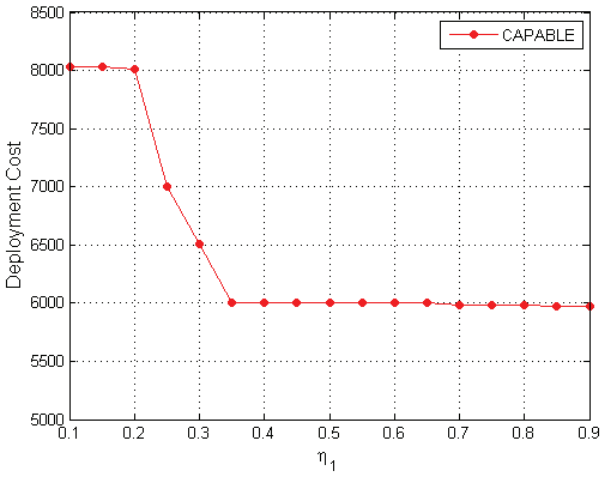

Fig. 6. Deployment cost v.s. η_1

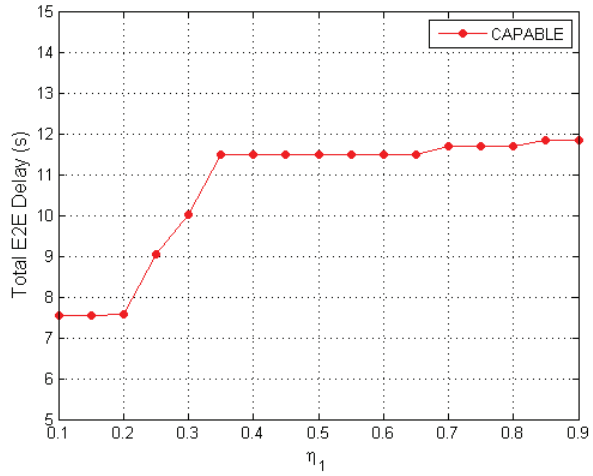

Fig. 7. E2E delay v.s. η_1

VI. Conclusion

In this paper, we have proposed CAPABLE, which minimizes the total cost of the cloudlet network by optimize the tradeoff between the deployment cost of cloudlets and E2E delay of user requests. Moreover, since users are moving in the network, i.e., incurring the spatial and temporal workload dynamics among different regions, we also design a load allocation scheme to minimize the E2E delay in each time slot, given the already deployed cloudlets. We have demonstrated that CAPABLE achieves the lowest total cost of the cloudlet network consisting of both the deployment cost and E2E delay cost as compared to other two cloudlet placement strategies. Meanwhile, the simulation results also show that CAPABLE can save a significant deployment cost by sacrificing a bearable amount of E2E delay that can be determined by setting a parameter at the provider's disposal.

References

[1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[2] Y. Zhang and N. Ansari, "On architecture design, congestion notification, tcp incast and power consumption in data centers," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 1, pp. 39–64, First Quarter, 2013.

[3] X. Sun, N. Ansari, and R. Wang, "Optimizing resource utilization of a data center," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2822–2846, 2016.

[4] Q. Fan, J. Fan, J. Li, and X. Wang, "A multi-hop energy-efficient sleeping MAC protocol based on TDMA scheduling for wireless mesh sensor networks," *Journal of Networks*, vol. 7, no. 9, pp. 1355–1361, 2012.

[5] L. A. Tawalbeh, W. Bakheder, and H. Song, "A mobile cloud computing model using the cloudlet scheme for big data applications," in *2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, Washington, DC, 2016, pp. 73–77.

[6] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, and B. Amos, "Edge analytics in the internet of things," *IEEE Pervasive Computing*, vol. 14, no. 2, pp. 24–31, 2015.

[7] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the internet of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016.

[8] ——, "PRIMAL: PRofIt Maximization Avatar pLacement for Mobile Edge Computing," in *Proc. of IEEE International Conference on Communications (ICC'2016)*, Kuala Lumpur, Malaysia, May 2016.

[9] X. Sun, N. Ansari, and Q. Fan, "Green energy aware avatar migration strategy in green cloudlet networks," in *Proceedings - IEEE 7th International Conference on Cloud Computing Technology and Science, (CloudCom' 2015)*, Vancouver, Canada, Nov. 2015.

[10] Z. Xu, S. Member, W. Liang, and S. Member, "Efficient Algorithms for Capacitated Cloudlet Placements," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 10, pp. 2866–2880, Oct. 2016.

[11] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Transactions on Cloud Computing*, DOI: 10.1109/TCC.2015.2449834 2015.

[12] Q. Fan and N. Ansari, "Green energy aware user association in heterogeneous networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC'2016)*, Doha, Qatar, Apr. 2016.

[13] N. L. Van Adrichem, C. Doerr, and F. A. Kuipers, "Opennetmon: Network monitoring in openflow software-defined networks," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, Krakow, Poland, May 2014, pp. 1–8.

[14] C. Yu, C. Lumezanu, A. Sharma, Q. Xu, G. Jiang, and H. V. Madhyastha, "Software-defined latency monitoring in data center networks," in *International Conference on Passive and Active Network Measurement*, vol. 8995, Mar. 2015, pp. 360–372.

[15] J. Ghosh, S. J. Philip, and C. Qiao, "Sociological orbit aware location approximation and routing in manet," in *2nd International Conference on Broadband Networks*, Boston, MA, Oct. 2005, pp. 641–650.