Optimal Placement of Virtual Machines in Mobile Edge Computing

Lei Zhao*, Jiajia Liu*†, Yongpeng Shi*, Wen Sun*, and Hongzhi Guo*
*School of Cyber Engineering, Xidian University, China
†Email: liujiajia@xidian.edu.cn

***Abstract*—Mobile edge computing (MEC), as an extension of the cloud computing paradigm to the edge network, is a promising solution to provide resource-intensive and time-critical applications to mobile users. It overcomes some obstacles of traditional mobile cloud computing by offering ultra-short latency and less core network traffic. This paper proposes a new framework based on the architecture of MEC to deliver cloud services to the edge. We introduce enumeration based optimal placement algorithm (EOPA) and divide-and-conquer based near-optimal placement algorithm (DCNOPA) to attain minimal data traffic by distributing virtual machine replica copies (VRCs) of applications to the edge network. Simulation results show that compared to the famous K-medians clustering algorithm (KMCA), the performance of DCNOPA is much closer to that of EOPA with lower computational complexity. Furthermore, we investigate the optimal number of VRCs within a given limitation of benefit-to-cost ratio.**

I. Introduction

Over the last decades, the number of mobile devices for users, such as smartphones and smartwatches, has been growing with astonishing speed [1]. Correspondingly, there are more and more resource-intensive applications provided by various service providers running in the cloud for the increasing mobile users [2]. By the concept of mobile cloud computing [3] [4], mobile applications are running in the cloud data center, which can take advantages of the tremendous resources of the farm of servers to hold diverse applications. However, the WAN round trip time between mobile users and remote cloud is a fatal limitation to the time critical applications [5]. Recently, mobile edge computing (MEC), as an alternative for latency-intolerant applications, is proposed by European Telecommunications Standards Institute (ETSI) [6], to be an extension of the cloud computing paradigm from the core of the data center to the edge of the network based on virtual machine (VM) technology.

With the increasing provision of real-time interactive services such as virtual reality and augmented reality, the mobile data traffic in the edge network grows rapidly leading to severe network resource consumption. What's more, heavy traffic will also cause exceptionally long latency to access services which is unacceptable for mobile users to get the delay-sensitive and interaction-heavy services. As reliable and low latency communications have been considered as one of the critical issues in the edge network for providing of emerging services, efficient control of data traffic is crucial to meet the strict-delay demand and improve the QoE of mobile users.

There have been a number of studies on MEC. Taleb *et al.* [7] proposed a solution for reducing core network traffic and ensuring ultra-short latency through a smart MEC architecture, which is based on migration of the light-weight Linux-based application containers to follow users. Zeng *et al.* [8] investigated the request completion time minimization problem with joint consideration of task image placement and load balancing for a hybrid computing environment. They formulated this problem as a mixed-integer nonlinear programming problem and proposed a low-complexity three-stage algorithm. A novel proxy VM migration scheme was proposed and evaluated by Sun *et al.* [9] to minimize the traffic in the core network. They assumed that each cellular base station is connected to a fog node to provide computing resource locally and each user's device is associated with a proxy VM located in a fog node where the application VM is in the remote cloud. Yin *et al.* [10] studied the challenges of edge server placement considering cost budgets and expectations of users. For a comprehensive survey, please refer to Liu *et al.* [11], in which diverse MEC applications are classified based on different criteria.

Few work considers the data traffic in the edge network. When the applications are not only latency-sensitive but also compute-intensive and memory-intensive [12] [13], launching one application replica for each user and migrating the replicas to follow users are undesirable for the edge network capacity. In this paper, we propose a new framework to deploy virtual machine replica copies (VRCs) of one application to the edge network with the objective of minimizing the average data traffic considering the total cost of deploying k VRCs. By controlling the number of VRCs of the resource intensive application and finding the optimal placement of these VRCs, the data traffic in the edge network can be significantly decreased.

Our main contributions are given as follows:

- Based on the mobile edge computing architecture, we first study the scenario of how to deploy k VRCs of one application from the central cloud into $|V|$ MEC servers to minimize the average data traffic in the edge network, which has been rarely studied.
- To address this problem, first, we propose EOPA which enumerates all placements of k VRCs and evaluates the average data traffic for each placement case. EOPA can get the optimal case that minimizes the average data traffic with computational complexity $O(k \cdot |V| \cdot C^k_{|V|})$. To reduce the computational complexity, we develop DCNOPA which divides the original k VRCs placement problem into k subproblems. Each subproblem will find the optimal placement for one VRC. As a result, it can get a near-optimal placement

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

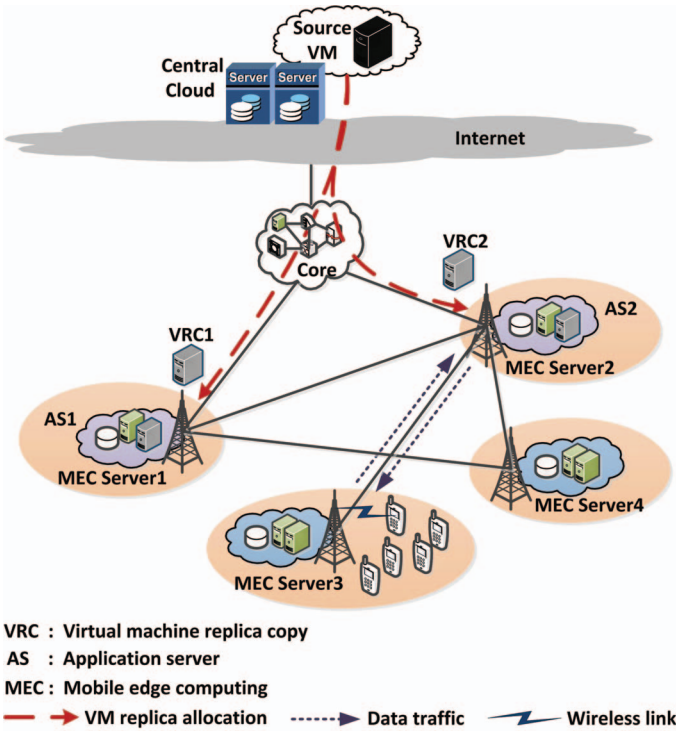


Fig. 1: The proposed design scenario of MEC. The central cloud replicates the source application VM and distributes VRC1 and VRC2 to MEC server1 and MEC server2 respectively which turns the MEC servers into AS1 and AS2 to provide application service in the edge network.

for k VRCs with $O(|V| \cdot \log |V|)$ much lower than that of the famous KMCA.

- We demonstrate through extensive simulations to verify the performances of the proposed algorithms. DCNOPA can get a near-optimal placement with the lower computational complexity than that of KMCA. We also investigate the efficiency on reducing data traffic by increasing the number of VRCs, and get the optimal number of VRCs within a given limitation of benefit-to-cost ratio.

The rest of the paper is organized as follows. Section II illustrates the network model and formulates the placement problem of k VRCs. Two different placement solutions are elaborated and analyzed in Section III. The simulation results are presented in Section IV to validate the two placement algorithms, and illustrate the efficiency on reducing data traffic of different number of VRCs. Finally, we conclude this paper in Section V.

II. Network Model and Problem Formulation

In this section, we propose a framework based on the architecture of MEC which is illustrated in Fig. 1. We assume that the resource-intensive application is encapsulated in the source VM which runs in the central cloud. As illustrated in Fig. 1, the central cloud creates multiple replica copies (VRC1, VRC2) of the source VM holding the target application and then distributes these VRCs to the edge network. Over the Internet and core network, the VRCs finally come to the edge and are assigned to MEC servers1 and MEC server2 respectively. The MEC server1 and MEC server2 who are chosen to hold VRC1 and VRC2 play the role of application servers (ASs) of the concerned application to provide specific service. Every MEC server has its own service region which has no overlapping parts among each other.

Network Model. The physical edge network is modeled by a graph $G(V, E)$, where $V = \{v, v = 1, 2, ..., |V|\}$ is the set of MEC servers, and E is the set of links between MEC servers. We assume that all mobile users are randomly distributed in the edge network. And k VRCs, $S = \{s_i,\ i = 1, 2, ..., k\}$, are assigned to the MEC servers in the edge network. The assignment of VRCs should follow two constrains: 1) each VRC of an application only can be assigned to one MEC server, and 2) each MEC server also only holds one VRC of an application as illustrated in Eqn. (3) and Eqn. (4) respectively, where an indicator variable $I_{v,s}$ is used to indicate whether VRC s of an application deployed in MEC server v or not, where $v \in V$ and $s \in S$, i.e., if VRC s deployed in MEC server v, $I_{v,s} = 1$, otherwise, $I_{v,s} = 0$.

We assume that the positions of mobile users are randomly located into the edge network which is divided into different regions of MEC servers. We use $R = \{R_v,\ \forall v \in V\}$ to denote the set of the number of requests from the mobile users within the region of each MEC server and the network bandwidth requirement for accessing the application is B. All requests for the application will be transmitted to the nearest MEC servers holding the VRCs to get service. The resource capacity provided to one VRC by the MEC server is limited. If one VRC is overloaded, the extra requests will be retransmitted to the next appropriate VRC, where C_{max} represents the maximum capacity assigned to one VRC. The percentage of the requests from the region of MEC server v that should be served by MEC server u is denoted by $\delta_{v,u}$. We use an indicator variable $P_{v,u}(m, n)$ to indicate whether the edge (m, n) in the routing path of requests from the region of MEC server v to MEC server u. The overall data traffic for serving the requests of the mobile users within the region of MEC server v by MEC server u as

$$f_{v,u} = R_v \cdot \delta_{v,u} \cdot B \cdot \sum_{(m,n)\in E} P_{v,u}(m, n),\ \forall v, u \in V. \quad (1)$$

Problem Formulation of k-VMRP (VM replica placement). For a given edge network topology, the remote cloud needs to decide how to deploy k VRCs into different MEC servers to minimize the average data traffic in the edge network. This allocation determination would influence the data traffic and the cost of deploying services into the edge. We define this problem as the placement problem of k VRCs (k-VMRP) which is to find a placement for k VRCs to minimize the average data traffic for each request in the edge network as Eqn. (2) under the constraints of Eqn. (3,4,5).

$$\min \frac{1}{|V|} \sum_{v,u\in V} \sum_{s\in S} \frac{I_{u,s} \cdot f_{v,u}}{R_v}, \quad (2)$$

$$s.t. \quad \sum_{u\in V} I_{u,s} = 1, \forall s \in S, \quad (3)$$

$$\sum_{s\in S} I_{u,s} = 1, \forall u \in V, \quad (4)$$

$$\delta_{v,u} \cdot R_v \le C_{max}, \forall u, v \in V. \quad (5)$$

Cost Model for Deploying k VRCs. When the remote cloud decides to deploy k VRCs of an application to the edge network, it has to consider the total cost of the deployment strategy [14]. We formulate this total cost into two parts, i.e., the deployment cost and the traffic cost. We consider the synchronization between VRCs and resource demands of VRCs as the deployment cost, which is formulated as the direct proportion of the number of VRCs, i.e., $\omega \cdot k$, where $\omega > 0$. The traffic cost presents the average data traffic generated in the edge network with k VRCs of an application providing service, i.e., D_k evaluated by Eqn. (2). After normalization of the deployment cost and traffic cost, we define the total cost of deploying k VRCs as

$$\rho_k = N(D_k) + N(\omega \cdot k), \tag{6}$$

where $N(\omega \cdot k)$ and $N(D_k)$ represent the normalized deployment cost and traffic cost for deploying k VRCs respectively.

III. Algorithmic Solutions of k-VMRP

A. Enumeration Based Optimal Placement Algorithm

The goal of the optimal placement solution is to find the placement case $S^{'} = \{I_{u,s}, \forall u \in V, \forall s \in S\}$ from the set of all possible placements of k VRCs, i.e., A, to get the minimal average data traffic for each request. This solution is composed by two parts: 1) enumerating all possible placement cases for k VRCs, and 2) evaluating the average data traffic for each placement case. All requests of mobile users in the edge network selects the AS which can minimize the average data traffic to provide service under the constrains of Eqn. (3,4,5).

Definition 1: **Average data traffic for placement case $S^{'}$.** The average data traffic for placement case $S^{'}$ is calculated by taking the summation of the minimal average data traffic for serving the requests from each MEC server's region, and evaluate this summation in the average of $|V|$.

$$D_{avg}(S^{'}) = \frac{1}{|V|} \sum_{v \in V} \min_{s \in S} \sum_{u \in V} \frac{I_{u,s} \cdot f_{v,u}}{R_v}. \tag{7}$$

The details of EOPA are described in *Algorithm 1*. The performance and efficiency of this solution are analyzed in *Proposition 1*.

Proposition 1: EOPA in Algorithm 1 can find an optimal solution to the placement problem of k VRCs in mobile edge networks.

Proof: Since EOPA in *Algorithm 1* traverses the whole solution space and evaluates the average data traffic for each placement case of k VRCs. It is obvious that EOPA can find an optimal placement to k-VMRP. ■

Algorithm 1 Enumeration Based Optimal Placement Algorithm.

Input: $G,\ R,\ k$
Output: S_{opt}
1: Initialize A denotes the set of all placement cases of k replicas, $A = \Phi$
2: Enumerate all combinations of k replicas in node set V, and record into set A
3: $D_{min} = \infty$
4: **for** $S^{'} \subseteq A$ **do**
5: Evaluate the average data traffic D_{avg} for placement case $S^{'}$ by Eqn. (3)
6: **if** $D_{avg} \leq D_{min}$ **then**
7: $D_{min} = D_{avg}$
8: $S_{opt} = S^{'}$
9: **end if**
10: **end for**
11: **return** S_{opt}

B. Divide-and-Conquer Based Near-Optimal Placement Algorithm.

To solve k-VMRP, we divide all MEC servers into k clusters. It only needs to deploy one VRC for each cluster. Thus, the original k-VMRP is reduced to the problem of finding an optimal placement for one VRC in each cluster, which will significantly reduce the complexity. The main steps of this solution are listed as follows.

Step 1, we sort all MEC servers by R_v and select k MEC servers whose regions cover the maximum requests, and distribute them into k clusters $C = \{c_i,\ i = 1, 2, ..., k\}$ respectively to be the initial ASs.

Step 2, we define the diameter of each cluster as $\lambda_c = \lceil \frac{\Lambda}{k} \rceil$, where Λ presents the largest distance between any pair nodes in graph G, and divide all the MEC servers into k clusters. The distance is measured by hops. All MEC servers within λ_c hops from the initial AS in $c_i \in C$ will be assigned into cluster c_i, under the constrain that every MEC server only can be partitioned into one cluster.

Step 3, we use the notion of neighbors $\Omega = \{\omega_v^i,\ \forall v \in V,\ i = 1, \ldots, \Lambda\}$ to balance the clustering procedure, where ω_v^i is the set of all MEC servers which are at a distance of i hops from MEC server v. Before each initial AS selecting its serving region of MEC servers into cluster c_i, we sort all clusters in ascending order by the number of MEC servers already held in that cluster. The cluster $c_i \in C$ which holds fewer MEC servers will get higher priority to select MEC servers from the remain MEC servers.

Step 4, each of the remain MEC servers which are still out of the k clusters will be assigned into the relevant cluster $c_i \in C$ which provides service for the requests from the region of that MEC server with the minimal data traffic.

Step 5, evaluating the performance of each MEC server in cluster $c_i \in C$ with one VRC deployed. We select the MEC server, which can minimize the average data traffic when serving requests from all other regions of MEC servers in cluster c_i, as the location of the VRC.

Definition 2: **Average data traffic for one VM replica placement in each cluster.** When deploying the VRC into MEC server u in cluster c_i to providing services to the mobile users in this cluster, the average data traffic for requests in cluster c_i is defined as

$$D_{avg}^{c_i}(u) = \frac{1}{|c_i|} \sum_{v \in c_i} \frac{f_{v,u}}{R_v},\ \forall u \in c_i \tag{8}$$

The details of DCNOPA are illustrated in *Algorithm 2*. The

performance and efficiency of this solution are analyzed in *Proposition 2*.

Algorithm 2 Divide-and-Conquer Based Near-Optimal Placement Algorithm.

Input: $G,\ R,\ k,\ \Omega,\ \Lambda,\ C$
Output: $S^{'}$

Initialize λ_c denotes the diameter of a initial cluster, $\lambda_c = \lceil \frac{\Lambda}{k} \rceil$, $c_i = \Phi$, $c_i \in C$
Sort R in a decreasing order, select k MEC servers covering the maximum requests to be the initial ASs in each c_i respectively
$L = \Phi$
for d = 1 to λ_c **do**
  Sort C by the number of MEC servers already gathered in each cluster in ascending order
  for $c_i \in C$ **do**
    for $v \in c_i$ **do**
      for $b \in \omega_v^d$ **do**
        if $b \notin L$ **then**
          $c_i = c_i \cup b$
          $L = L \cup b$
        end if
      end for
    end for
  end for
end for
Assign each remain MEC server into cluster $c_i \in C$ that leads to minimal traffic
for $c_i \in C$ **do**
  for $v \in c_i$ **do**
    Evaluate $D_{avg}^{c_i}(v)$ by Eqn. (8)
  end for
  Select $v^{'} \in c_i$ that minimizes $D_{avg}^{c_i}$
  Update $S^{'}$ by $S^{'} = S^{'} \cup \{v^{'}\}$
end for
return $S^{'}$

Proposition 2: DCNOPA in Algorithm 2 can achieve a near-optimal solution to k-VMRP in mobile edge networks.

Proof: It breaks the original problem of placing k VRCs to all the MEC servers into k subproblems. Each subproblem is to find an optimal placement for one VRC in cluster $c_i \in C$. By combining the optimal one VCR placement in each cluster, we can get a near-optimal placement of k VRCs for k-VMRP. ■

C. Analysis and Discussions

EOPA in *Algorithm 1* is very simple and effective, but it has a very high computational complexity since EOPA has to enumerate all combinations of k VRCs from $|V|$ MEC servers so as to find the minimum average data traffic. More details on the computational complexity of EOPA are discussed in *Proposition 3*.

Proposition 3: The computational complexity of EOPA in *Algorithm 1* is $O(k \cdot |V| \cdot C_{|V|}^k)$.

Proof: The computational complexity of *Algorithm 1* is comprised of two parts, 1) enumerating all combinations of the placement of k VRCs in set of $|V|$ MEC servers and 2) evaluating D_{avg} for each placement case $S^{'}$. In particular, the running time of enumerating all possible placements is $O(C_{|V|}^k)$. The computational complexity of evaluate D_{avg} for each placement case $S^{'}$ is $O(k \cdot |V|)$. By adding up the evaluation running time of all placements, the computational complexity of *Algorithm 1* is $O(k \cdot |V| \cdot C_{|V|}^k)$. ■

Compared to EOPA in *Algorithm 1*, DCNOPA has much more advantage on computational efficiency which is analyzed in *Proposition 4*.

Proposition 4: The computational complexity of DCNOPA in *Algorithm 2* is $O(|V| \cdot \log |V|)$.

Proof: In the beginning of this scheme, R is sorted with $|V|$ elements in a decreasing order to get k MEC servers which cover the maximum requests locally. This procedure will totally require $O(|V| \cdot \log |V|)$ running time. Next, we divide the set of all MEC servers V into k clusters. This dividing procedure will cost $O(k \cdot \log k + k \cdot |V|) = O(k \cdot |V|)$ running time. Then, the running time of evaluating $D_{avg}^{c_i}$ of each MEC server in cluster c_i and selecting MEC server $u \in c_i$ which can minimize $D_{avg}^{c_i}$ as the location of the VRC is $O(k \cdot |V|)$. Finally, the running time of *Algorithm 2* can be calculated by adding them up, i.e., $O(|V| \cdot \log |V| + k \cdot |V| + k \cdot |V|) = O(|V| \cdot \log |V|)$. ■

To compare with the two algorithms proposed before, i.e., EOPA, DCNOPA, we also introduce random placement algorithm and k-medians clustering algorithm [15], i.e., RPA, KMCA, to make comparisons. RPA as a general placement solution, which randomly chooses one feasible placement case from the entire solution space of VRC placement with the computational complexity $O(1)$. We apply KMCA into our model to iteratively finds k clusters in the set of all MEC servers, each with a median location at its center to be assigned with one VRC providing service for the cluster to minimize the within-cluster average data traffic. At the end of each iteration, if the average data traffic is converged with the current median locations for VRCs, then the algorithm exits, otherwise, update the clusters and enter the next iteration. The computational complexity of KMCA is $O(k^3 \cdot |V|^2 \cdot t)$, where t is the iteration times.

IV. Numerical Results

A. Simulation Settings

We have developed a simulator in Python to simulate the impact of different placement of VRCs on data traffic in the edge network. Without loss of generality, we took the real world online network topologies from the Topology Zoo [16], a collection of annotated network graphs derived from public network maps, for implementing VRCs deployment into edge networks. The network bandwidth requirement B is set to 800 KB referred from the Cisco documents [17], We assume that 1000 users are scattered around the whole edge network and the distribution of launching requests follows Poisson-process.

Furthermore, we investigate the running time of the proposed algorithms where all the algorithms run on a desktop with Debian 64 bit operating system in an Intel i5 core computer with 4 GB RAM.

B. Comparisons of Different Solutions for k-VMRP

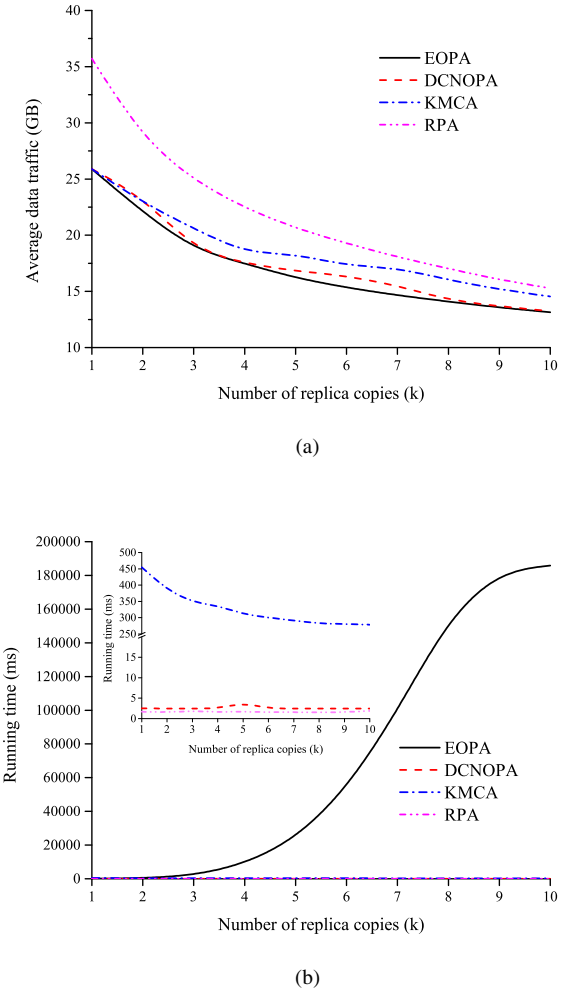


Fig. 2: Performance illustration of EOPA, DCNOPA, KMCA and RPA. (a) Comparisons of average data traffic under different number of VRCs. (b) Comparisons of the overall running time.

As illustrated in Fig. 2, we compare the performances of EOPA, DCNOPA, KMCA and RPA among different number of VRCs using the network topology of Noel communications network of Washington State, USA with 19 nodes and 25 edges. In Fig. 2(a), it shows that the four algorithms all bring the average data traffic down with the number of VRCs increasing from 1 to 10. More VRCs can reduce the average distance between users and application servers which lead to less traffic in the edge network. It is very clearly that EOPA can get the optimal performance. Apparently the performance of RPA is far from the optimal result especially when the number of VRCs is small. While DCNOPA and KMCA can obtain a near-optimal solution. What's more, from Fig. 2(a), we know that the performance of DCNOPA on minimizing the average data traffic is better than that of KCMA.

Fig. 2(b) presents the comparison of the running time among these algorithms. From Fig. 2(b), we can observe that with the number of VRCs increasing, the computational complexity of OEPA increases sharply which is much higher than that of the other schemes. Although lower than that of OEPA, the running time of KMCA is still much higher than that of DCNOPA and RPA. DCNOPA can get much lower complexity compared with both OEPA and KMCA. Actually, the running time of DCNOPA is even close to that of RPA.

TABLE I: Topology Settings

	Number of nodes	Number of links
TLex	12	16
Shentel	28	35
IRIS	51	64
Missouri	67	83

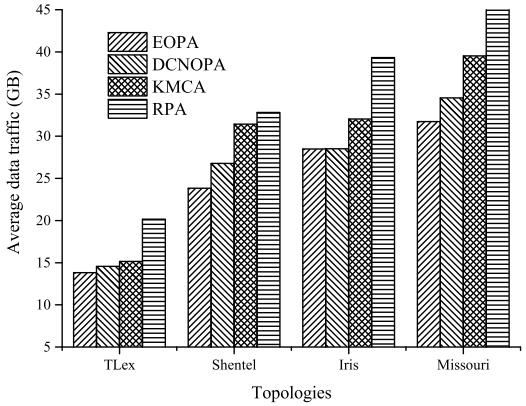


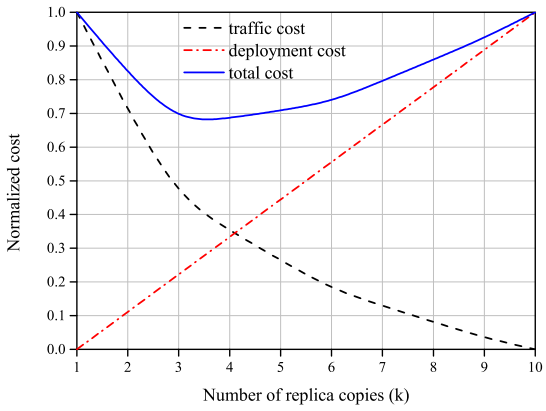
Fig. 3: Performance comparisons of EOPA, DCNOPA, KMCA and RPA under different network topologies where the number of VRCs $k = 3$.

Fig. 3 illustrates the comparison results of average data traffic in edge networks among EOPA, DCNOPA, KMCA and RPA with different network topologies listed in Table I, and the number of VRCs $k = 3$. From Fig. 3, we can see that for all the four algorithms, the average data traffic grows with the increasing scale of networks. Moreover, it shows that DCNOPA is able to obtain a near-optimal solution. Compared with the performance of KMCA, DCNOPA can get much less average data traffic which is much closer to that of EOPA in all the four topologies. However, the performance of RPA becomes much poorer with the increasing scale of networks.

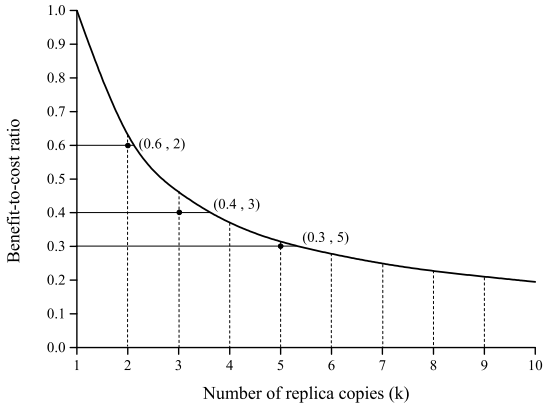
C. Optimal Setting of the Number of VRCs

As shown in Fig. 4(a), deploying more VRCs to the edge network will reduce the traffic cost, while the deployment cost is increased. By adding up the normalized traffic cost and deployment cost, we get the total cost with the increasing number of VRCs as illustrated in Fig. 4(a). It shows that when the number of VRCs is no more than 4, the total cost decreases with the increasing number of VRCs. However, the total cost will increase with the added VRCs when the number of VRCs is more than 4. For the topology of Noel communication network, deploying 4 VRCs of an application can get the minimal total cost.

To further quantify the tradeoff between the benefit and cost with increasing number of VRCs, we define the benefit-to-cost



(a)



(b)

Fig. 4: Optimal setting of k with the network topology of Noel communication. (a) Normalized total cost. (b) Getting the optimal k by setting floor of benefit-to-cost ratio α.

ratio as $\alpha = \frac{N(D_1)/N(D_k)}{N(\omega \cdot k)}$, where we simplify the deployment cost by setting $\omega = 1$. As shown in Fig. 4(b), the benefit-to-cost ratio α continues to descend with the increasing number of VRCs. If α holds a value 1.0, it indicates that the benefit gets even with the cost. For giving the minimal limitation of α, we could get the relevant optimal setting for VRC number k. In this topology, when $\alpha = 0.6$, the optimal setting for k is 2, since more VRCs will lead to less average data traffic, and when there are more than 2 VRCs, the value of α will less than 0.6.

V. Conclusions

In this paper we have proposed a new framework to push the resource-intensive application to the edge closer to mobile users based on the architecture of MEC. And we mainly take our view into the data traffic in the edge network which has been seldom taken into consideration. We proposed EOPA and DCNOPA as the solutions to solve the problem of optimal placement of k VRCs into the edge network. The experimental results showed that compared with EOPA, DCNOPA can get near-optimal performance with lower computation complexity, which is much better than that of KMCA and RPA. We also investigated the benefit-to-cost ratio of different number of VRCs to get the optimal number of VRCs in the constrain of a minimal limitation of benefit-to-cost ratio.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61372073, 61373043, 61202394, 61472367, 61432015, and 61601357), in part by China 111 Project (B16037), and in part by the Fundamental Research Funds for the Central Universities (JB171501, JB161502, and XJS16045).

References

[1] M. Satyanarayanan, "Mobile computing: the next decade," in *ACM MobiSys*, 2010.

[2] W.-T. Tsai, X. Sun, and J. Balasooriya, "Service-oriented cloud computing architecture," in *IEEE ITNG*, 2010.

[3] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[4] J. H. Christensen, "Using restful web-services and cloud computing to create next generation mobile applications," in *ACM OOPSLA*, 2009.

[5] A. ur Rehman Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 393–413, 2014.

[6] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing a key technology towards 5G," ETSI White Paper, 2015.

[7] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 38–43, 2017.

[8] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Transactions on Computers*, vol. 65, no. 12, pp. 3702–3712, 2016.

[9] 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.

[10] H. Yin, X. Zhang, H. H. Liu, Y. Luo, C. Tian, S. Zhao, and F. Li, "Edge provisioning with flexible server placement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1031–1045, 2017.

[11] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. de Foy, and Y. Zhang, "Mobile edge cloud system: Architectures, challenges, and approaches," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–14, 2017.

[12] M. Satyanarayanan, "Augmenting cognition," *IEEE Pervasive Computing*, vol. 3, no. 2, pp. 4–5, 2004.

[13] D. Zeng, P. Li, S. Guo, T. Miyazaki, J. Hu, and Y. Xiang, "Energy minimization in multi-task software-defined sensor networks," *IEEE Transactions on Computers*, vol. 64, no. 11, pp. 3128–3139, 2015.

[14] D. Zeng, L. Gu, L. Lian, S. Guo, H. Yao, and J. Hu, "On cost-efficient sensor placement for contaminant detection in water distribution systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2177–2185, 2016.

[15] 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*, vol. PP, no. 99, 2015.

[16] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

[17] "Vni service adoption forecast-services gauge," [Online]:http://www.cisco.com/c/en/us/solutions/service-provider/vni-service-adoption-forecast/vnisa_services_gauge.html.