This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2508382, IEEE Transactions on Emerging Topics in Computing

Cost Efficient Resource Management in Fog Computing Supported Medical Cyber-Physical System

Lin Gu, *Member, IEEE*, Deze Zeng, *Member, IEEE*, Song Guo, *Senior Member, IEEE*, Ahmed Barnawi, *Member, IEEE*, and Yong Xiang, *Senior Member, IEEE*

Abstract—With the recent development in information and communication technology (ICT), more and more smart devices penetrate into people's daily life to promote the life quality. As a growing healthcare trend, Medical Cyber-Physical Systems (MCPSs) enables seamless and intelligent interaction between the computational elements and medical devices. To support MCPSs, cloud resources are usually explored to process the sensing data from medical devices. However, the high quality-of-service (QoS) of MCPS challenges the unstable and long-delay links between cloud data center and medical devices. To combat this issue, mobile edge cloud computing, or fog computing, which pushes the computation resources onto the network edge (e.g., cellular base stations), emerges as a promising solution. We are thus motivated to integrate fog computation and MCPS to build fog computing supported MCPS (FC-MCPS). In particular, we jointly investigate base station association, task distribution and virtual machine placement towards cost efficient FC-MCPS. We first formulate the problem into a mixed-integer non-linear linear program (MINLP) and then linearize it into a mixed integer linear programming (MILP). To address the computation complexity, we further propose an linear programming (LP) based two-phase heuristic algorithm. Extensive experiment results validate the high cost efficiency of our algorithm by the fact that it produces near optimal solution and significantly outperforms a greedy algorithm.

Index Terms—Mobile Edge Computing, Fog Computing, Medical Cyber Physical System, Cost Efficiency

1 Introduction

Cyber-Physical Systems (CPSs) emerge as engineered systems that offer integrations of computation, networking, and physical processes, enabling seamless interaction between cyber services and physical components [1]. Building on the discipline of computing, sensing, communication and embedded system technologies, CPS is a natural evolution resulting from the fast development of information and communication technology (ICT) in the past decades and will transform the way people interact with the physical world. CPS can be applied in various areas such as transportation, manufacturing, agriculture, energy, healthcare and so on. Its unarguable huge economic and societal potential has attracted worldwide attentions to develop the technology for promoting the interaction between the cyber space and the physical world.

On the other hand, we have also witnessed a booming development in healthcare industry for the past decades. Large enterprises like Nike, Samsung and Apple, have all released their wearable devices and health care applications. Especially, Apple has been working with the Mayo Clinic, Radboud University Medical Center, the University Hospital in Nijmegen to develop their health kit which aims to provide a central platform for health information[1]. Berg Insight[2] reports that there were 3 million patients using connected home medical monitoring devices worldwide at the end of 2013. They also estimate the number of connected medical devices will grow at a yearly rate 18% between 2010 and 2016 to reach around 5 million connections all over the world. With the vast number of connected medical devices and the trend towards Cloud Computing [2], Medical Cyber-Physical System (MCPS) [3]–[5] is proposed as one important branch of CPS.

MCPS integrates medical devices, including both monitoring devices (e.g., heart-rate monitors) and delivery devices (e.g., medication infusion pumps, ventilators), with software applications. By coordinating the operation of the *delivery devices* based on the data from

This research was partially supported by Strategic International Collaborative Research Program (SICORP) Japanese (JST) - US (NSF) Joint Research on Big Data and Disaster Research, the NSF of China (Grant No. 61402425, 61501412), the Fundamental Research Funds for National University, China University of Geosciences, Wuhan, China (Grant No. CUG14065, CUGL150829).

- *L. Gu is with Service Computing Technology and System Lab, Cluster and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, China. E-mail: linny.gu@gmail.com*
- *S. Guo is with University of Aizu, Aizu-wakamatsu, Fukushima, Japan. E-mail: sguo@u-aizu.ac.jp*
- *D. Zeng is with China University of Geosciences, Wuhan, Hubei, China. E-mail: deze@cug.edu.cn*
- *A. Barnawi is with King Abdulaziz University, Jeddah, Saudi Arabia. E-mail: ambarnawi@kau.edu.sa*
- *Y. Xiang is with the School of Information Technology, Deakin University, Burwood, Australia. E-mail: yxiang@deakin.edu.au*

1. http://www.applehealthkit.com/
2. http://www.reportlinker.com/p0356852/mHealth-and-Home-Monitoring-Edition.html

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2508382, IEEE Transactions on Emerging Topics in Computing

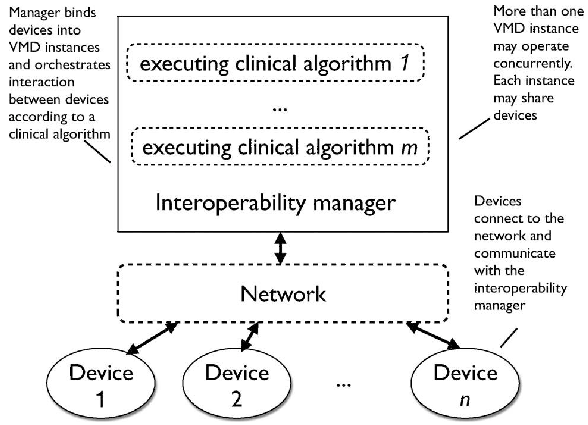


Fig. 1. VMD based on physical medical devices and cyber clinical algorithms [5]

the *monitoring devices*, MCPS provides more intelligent information to the care-giver, detects failures of individual devices, and enables automatic controlling of the delivery devices. This essentially improves patient safety and treatment effectiveness. It is a clear trend that MCPSs will be increasingly used not only in clinics but also in our daily lives to provide high-quality healthcare services.

In MCPS, some low-capability sensors/actuators require interaction with external software applications for full functionality. For example, Apple's health kit falls into such paradigm that the sensing data generated at the front-end (e.g., apple watch, iPhone, etc.) are actually sent to a central platform for storage and analysis. Such paradigm is conceptually described by Lee et al. [5] as virtual medical devices (VMDs) shown in Fig. 1, where a VMD instance binds a clinical algorithm and a set of related physical devices. The clinical algorithms, i.e., VMD applications, define how these connected devices shall interact with each other in a given clinical scenario. To host these VMD applications, resource-rich cloud data center is intuitively regarded as an ideal platform. However, the continuous growing number of connected medical devices have imposed a critical challenge as the bulk data produced from these devices must be timely transferred and analyzed by the VMD applications to provide fast and accurate feedbacks. Besides the low latency requirement, MCPSs may also ask for mobility support and location-awareness.

We notice that these demands can be well satisfied by exploring mobile edge cloud computing [6], [7] or fog computing [8], which can share the burden by hosting the VMD applications in the network edge, e.g., base stations (BSs). Bonomi et al. [8] from Cisco argue that fog computing platform is critical to support future cloud services and applications, especially to the Internet-of-Things (IoT) applications featured by geo-distribution, latency-sensitivity and high-resilience. Fog computing, as "clouds at the edge", allocates services near the devices to improve the quality-of-service (QoS) and other non-functional properties (NFPs) [9]. Fig. 2 gives an overview of a fog computing supported MCPS (FC-MCPS), where BSs provide not only the communication services to the medical devices but also the computation and storage resources to host the VMD applications as virtual machines (VMs). By such means, medical data streams can be uploaded and processed directly in a BS residing with the corresponding VMD application VM, instead of sending to the centralized data center.

From the perspective of MCPS service providers, certain operational expenditure (OPEX) must be paid to infrastructure service providers (ISPs) for using the communication and computation services. Firstly, data must be uploaded to the associated BS then routed to the BS with the processing VM. Note that there is no restriction on the choices of data uploading BS and the data processing BS. They can refer to either different BSs or the same one. For example, in Fig. 2, we can see that the data for application 1 can be uploaded and processed at BS 2 while application 3's data is uploaded at BS 2 but processed by BS 4. Secondly, to process data stream for a specific application, a corresponding VM should be deployed in BS. Therefore, BS 2 and BS 4 are deployed with the VMs for applications 1 and 3, respectively. Similar to existing data centers, deploying VM is not free and certain server rental cost will be charged. As BSs may be provided by different ISPs with different charging policy, cost-diversity exists among these BSs. Given that QoS is guaranteed, it is essential to explore such cost-diversity for cost minimization. Therefore, in this paper, we are motivated to investigate the QoS guaranteed minimum cost resource management problem in FC-MCPS. The main contributions are as follows.

- We propose the concept of FC-MCPS by forging fog computing into MCPS to host the VMD applications. In particular, we study a cost-efficient resource management problem with guaranteed QoS in FC-MCPS.
- We formulate the cost minimization problem in a form of mixed-integer nonlinear programming (MINLP) with joint consideration of communication BS association, subcarrier allocation, computation BS association, VM deployment and task distribution.
- To deal with the high computational complexity of solving MINLP, we linearize it as a mixed-integer linear programming (MILP) problem. We further propose a low-complexity two-phase LP-based heuristic algorithm and validate its high efficiency through extensive experiments.

The rest of the paper proceeds as follows. Section 2 gives a toy example to illustrate the motivation of our work. Section 3 introduces our system model. The cost optimization is formulated as an MINLP problem in Section 4 and then a two-phase heuristic algorithm is proposed in Section 5. The theoretical findings are verified by experiments in Section 6. Related work is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2508382, IEEE Transactions on Emerging Topics in Computing

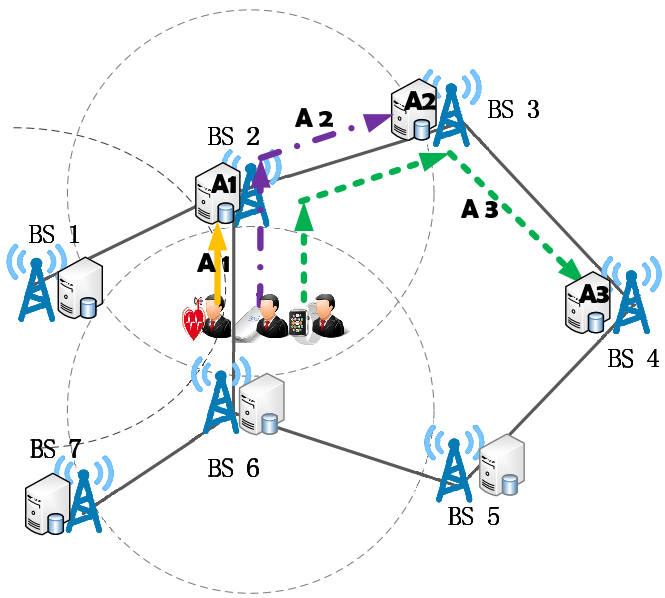


Fig. 2. An Overview of Fog-supported MCPS

TABLE 1
Notations

Constants	
I	The user set
J	The BS set
A	The application set
S	The subcarrier set
C_j^u	The uplink cost at BS j
C_{jk}	The communication cost between BS j and k
D^a	The delay constraint of application a
Λ_i^a	The arrival rate for application a from user i
L_i^a	The data length for application a from user i
Z_{ij}	A binary variable indicating if user i is in the coverage of BS j
R	The data rate of one subcarrier
H_k	The storage capacity of BS k
U_k	The computation capacity of BS k
Variables	
x_{ij}	A binary variable indicating if BS j is associated to user i or not
x_{ij}^s	A binary variable indicating if subcarrier s in BS j is allocated to user $i$or not
y_{jk}^a	A binary variable indicating if data from BS j for application a is processed in BS k or not
y_k^a	A binary variable indicating if a VM for application a is placed in BS k or not
λ_{jk}^a	The request rate for application a from BS j to BS k
μ_k^a	The processing rate for application a in BS k

summarized in Section 7. Finally, Section 8 concludes our work.

2 Motivation

To illustrate the motivation of our work, let us first consider a toy example shown in Fig. 2. There are 7 BSs (i.e., BS 1 to BS 7) and 3 users with 3 different applications (i.e., A 1, 2 and 3). All three users are within the transmission range of BS 2 and BS 6, with the uplink costs of two and one, respectively. User 1 is also in the transmission range of BS 1 with an uplink cost of one. For each BS, we assume there are three subcarriers. Each subcarrier can contribute an uploading data rate of one per time unit. Generally, the communication cost between each pair of BSs is determined by ISP providers and locations. For easy understanding, we use the number of hops to indicate the inter-BS communication cost in this example, e.g., the communication cost is one from BS 2 to BS 3 and two from BS 2 to BS 4. We aim at minimizing the total unit cost including the total VM deployment cost, uploading cost and inter-BS communication cost.

In the case shown in Fig. 2, BSs 2, 3 and 4 are deployed with VMs for applications 1, 2 and 3, respectively. Each BS charges 1 per time unit for hosting one VM. All three users are associated with BS 2 and all three data streams are uploaded through BS 2 and then processed in BSs 2, 3 and 4, respectively. The uploading cost can be calculated as $2+2+2=6$, while the VM deployment cost and the inter-BS communication cost are $1+1+1=3$ and $0+1+2=3$, respectively. In this case, the total cost is $6+3+3=12$ and the uploading delay experienced by each user is 1 because each application can use only 1 subcarrier of BS 2. To minimize the total cost without violating the QoS requirement (e.g., uploading delay in this example), we consider another solution that BSs 1, 2 and 6 are used to host the VMs for applications 1, 2 and 3, respectively. Meanwhile, users 1, 2 and 3 are associated with BSs 1, 2 and 6, respectively. Each BS allocates all its 3 subcarriers to its associated user. We can see that the uploading cost, the VM deployment cost and the inter-BS communication cost become $1+2+1=4$, $1+1+1=3$ and 0, respectively. The total cost is therefore reduced to $4+3+0=7$. Meanwhile, the uploading delay of all users is decreased to $\frac{1}{3}$ thanks to full occupation of all 3 subcarriers of corresponding BS.

From this example, we can see that a better solution of BS association, subcarrier allocation, VM deployment and task distribution not only contributes to lower total cost but also improves QoS of applications (e.g., the response time). It is significant to investigate these factors for QoS-guaranteed minimum-cost MCPS applications in fog computing environments.

3 System Model and Problem Statement

In this section, we introduce our FC-MCPS system model. For the convenience of the readers, the major notations used in this paper are listed in Table 1.

3.1 Network Model

We consider a connected cellular networks, e.g., Fig. 2, as an undirected graph $G=(N,E)$, where node set N includes both user set I and BS set J, i.e., $N=I\cup J$ while edge set E represents these links between nodes. Both users and BSs are randomly located in a two-dimensional area. There is a link e_{ij} between user

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2508382, IEEE Transactions on Emerging Topics in Computing

$i \in I$ and BS $j \in J$ if and only if they are within the communication range of each other. More accurately, e_{ij} exists if j is reachable from i as the BS usually has longer transmission distance than the medical devices. Let binary Z_{ij} indicate whether BS j is reachable from user i or not. Without loss of generality, we assume that all BSs are reachable from each other. However, the BSs may be owned by different ISPs or locate with different distances (e.g., number of hops) with each other. For BS $j \in J$, let C_j^u denote the uplink cost at BS j and C_{jk} be the inter-BS communication cost between BSs j and k. The values of C_j^u and C_{jk} vary on the ISPs and inter-BS distances. On the other hand, the link capacity between BSs is usually sufficient and the transmission delay is highly related with the network topology. In this paper, we assume the transmission delay D_{jk} on e_{jk} is given in advance.

For uplink communications, we assume that a subcarrier set S is available at each BS. Without loss of generality, all the BSs are with the same number of subcarriers and all the subcarriers are with the same bandwidth. For each subcarrier $s \in S$, the same data rate R can be achieved. Different from existing BSs in 3G or 4G cellular networks, a BS in FC-MCPS is not only equipped with antennas for wireless communication but also with a server that can accommodate VMs for different uses. We assume BS j in the network is associated with storage and computation resources with the capacity of H_j and U_j, respectively. To host a VM for application a at BS j, a rental cost C_j^a is charged per time unit.

A user $i \in I$ who carries several medical devices randomly roaming in the area may have a number of different applications running at the same time. We denote the application set in the whole system as A. For each application, the sensing data from involved medical devices shall be periodically uploaded to associated BS. The uploading events happen randomly as a Poisson process [10], [11]. In particular, let Λ_i^a be the average data arrival rate of user i for application a. Each time, a data stream with length of L_i^a is uploaded by user i. The uploaded data can be processed at any BS $k, k \in J$ as long as there exists a corresponding VM. The healthcare applications are usually delay-sensitive and ask for certain QoS in terms of delay. We denote the maximum tolerable delay for application a as D^a.

3.2 Problem Statement

Our objective is to minimize the overall unit cost for deploying FC-MCPS in a given BS infrastructure to guarantee the required QoS of all applications from users. As mentioned in Section 3, cost-diversity exists among BSs. It is desirable to associate all users to the BS with the smallest uplink communication cost. But unfortunately, the number of connections is limited by the number of subcarriers. On the other hand, placing all VMs in the associated BS helps reduce the inter-BS communication cost. But it is not efficient since the VM deployment cost of the associated BS might be significantly higher and therefore is not ignorable. Moreover, a VM usually requires certain resources to ensure the QoS while the resource capacities of a BS are limited. Finding out the BS association solution and an appropriate set of BSs to host the VMs for each application is a key issue to cost minimization. Because sensing data from medical devices shall be routed from user-associated BS to the BS with the corresponding VM, it is indispensable to investigate user association, task distribution and VM deployment towards cost-efficient FC-MCPS.

4 Problem Formulation

In this section, we present an MINLP formulation on the minimum cost problem with the joint consideration of user association, task distribution and VM deployment.

4.1 User Association Constraints

We first define binary variables $x_{ij}, \forall i \in I, j \in J_i$ to denote whether user i is associated with BS j or not, i.e.,

$$x_{ij} = \begin{cases} 1, \text{ if user } i \text{ is associated with BS } j, \\ 0, \text{ otherwise.} \end{cases} \tag{1}$$

Note that a user $i \in I$ can only be associated with a BS $j \in J$ if and only if j is reachable from user i, i.e., $Z_{ij} = 1$. In our toy example in Fig. 2, Z_{12}, Z_{22}, Z_{32}, Z_{16}, Z_{26} and Z_{36} are all equal to 1. That means, all the three users can be associated with either BS 2 or 6. In general, this can be expressed as

$$\sum_{s \in S} x_{ij}^s \leq Z_{ij}, \forall i \in I, j \in J. \tag{2}$$

To granulate the QoS of all applications from all users, one premise is that every user must be associated with one and only one BS, i.e.,

$$\sum_{j \in J} x_{ij} = 1, \forall i \in I. \tag{3}$$

To associate a user to a BS, subcarriers must be allocated for the communications between them. Let $x_{ij}^s \in \{0, 1\}$ represent whether subcarrier s at BS j is allocated to user i or not, i.e.,

$$x_{ij}^s = \begin{cases} 1, \text{ if subcarrier } s \text{ at BS } j \text{ is allocated to user } i, \\ 0, \text{ otherwise.} \end{cases} \tag{4}$$

A user i can be associated with BS j when and only when one or more subcarriers at BS j are allocated, i.e.,

$$\frac{\sum_{s \in S} x_{ij}^s}{|S|} \leq x_{ij} \leq \sum_{s \in S} x_{ij}^s, \forall i \in I, j \in J, s \in S. \tag{5}$$

Although there is no restriction on the number of subcarriers allocated to a user from a BS, one subcarrier can be only allocated to at most one user. For example, in Fig. 2, 3 users are associated with BS 2 which has

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2508382, IEEE Transactions on Emerging Topics in Computing

3 subcarriers. Therefore, each user can use only one subcarrier. That is

$$\sum_{i \in I} x_{ij}^{s} \leq 1, \forall j \in J, s \in S. \tag{6}$$

4.2 Task Distribution Constraints

The data uploaded at the user-associated BS can be processed at any BS in the system. As shown in Fig. 2, the data for applications 1 and 2 are uploaded at BS 2 while processed in BSs 2 and 3, respectively. Let $y_{jk}^{a}, \forall a \in A, j \in J, k \in J$ represent whether data for application a uploaded through BS j is then processed in BSs k, i.e.,

$$y_{jk}^{a} = \begin{cases} 1, \text{ if data for } a \text{ from BS } j \text{ is processed in } k \text{ ,} \\ 0, \text{ otherwise,} \end{cases} \tag{7}$$

where k can be equal to j if the data is processed directly at the user-associated BS like BS 2 in Fig. 2.

Since medical data is uploaded through BS j and then processed in BS k, y_{jk}^{a} and x_{ij} have the following relationship

$$y_{jk}^{a} \leq \sum_{i \in I} x_{ij}, \forall a \in A, j, k \in J, \tag{8}$$

which indicates that BS j can distribute application data to BS k only when it is associated with application users. In Fig. 2, BS 1 is not associated with any users, therefore no data is distributed to any other BS from it.

Using λ_{jk}^{a} to denote the data rate of application a routed from BS j to BS k for processing, we have the relationship between λ_{jk}^{a} and y_{jk}^{a} as follows

$$\frac{\lambda_{jk}^{a}}{N} \leq y_{jk}^{a} \leq \lambda_{jk}^{a} \cdot N, \forall a \in A, j, k \in J, \tag{9}$$

where N is an arbitrarily large number.

To ensure that all uploaded data are completely processed, for each application the total data received from users through BS j shall be equal to data finally processed at all BS k, i.e.,

$$\sum_{i \in I} \Lambda_{i}^{a} x_{ij} = \sum_{k \in J} \lambda_{jk}^{a}, \forall j \in J, a \in A. \tag{10}$$

As shown in our toy example (Fig. 2), all data uploaded at BS 2 are either processed locally or distributed to other BSs.

4.3 VM Placement Constraints

Let binary variable y_{k}^{a} indicate whether a VM of application a is hosted by BS k or not. Obviously, if data of application a is processed in BS k, i.e., $y_{jk}^{a} = 1$, a corresponding VM must be deployed in k. That is,

$$y_{jk}^{a} \leq y_{k}^{a}, \forall a \in A, j, k \in J. \tag{11}$$

A BS can process medical data stream of application a if and only if a VM for a is deployed on it. Like in Fig. 2, data of application 2 can only be processed in BS 3 because $y_{3}^{2} = 1$. Therefore, we have

$$\frac{\mu_{k}^{a}}{N} \leq y_{k}^{a} \leq N \mu_{k}^{a}, \forall a \in A, k \in J. \tag{12}$$

Since resources (i.e., hard disk and computational unit) of a BS are limited, the total VM resource requirements at BS k shall not exceed its capacity. Therefore, we have

$$\sum_{a \in A} y_{k}^{a} H^{a} \leq H_{k}, \forall k \in J, \tag{13}$$

and

$$\sum_{a \in A} \mu_{k}^{a} \Upsilon^{a} \leq U_{k}, \forall k \in J, \tag{14}$$

where μ_{k}^{a} is the processing speed for application a and Υ^{a} is a scaling factor to indicate the relationship between precessing speed and allocated computation resource.

4.4 QoS Constraints

According to the data analysis procedure in FC-MCPS, the medical sensing data shall go through three stages: 1) uploading to the user-associated BS j, 2) transferring from j to processing BS k and 3) processing at k. They all influence the overall delay and hence the QoS.

Let us first consider the uplink delay, which is mainly determined by the achievable data rate for a given application. The uplink data rate r_{ij} at BS j for user i is related with the number of allocated subcarriers and thus can be calculated as

$$r_{ij} = \sum_{s \in S} x_{ij}^{s} R, \forall i \in I, j \in J, \tag{15}$$

from which we can observe that the more subcarriers are allocated to a user, the higher data rate can be achieved.

Then, for application data with length L_{i}^{a}, we can calculate its uplink delay as

$$d_{ij}^{a} = \frac{L_{i}^{a}}{r_{ij}}. \tag{16}$$

We assume that the data received by user-associated BS from users will be distributed to BSs with corresponding VMs with a fixed probability. Thus, the data arrival in each BS can be also regarded as a Poisson process. We denote the average arrival rate of data from user i to BS j and the one from BS j to BS k as $\lambda_{ij} (\lambda_{ij} \leq 1)$ and $\lambda_{jk} (\lambda_{jk} \leq 1)$, respectively. The data processing procedure in BS k for application a can be viewed as a queuing system with the average arrival rate $\sum_{j \in J} \lambda_{jk}$ and service rate μ_{k}^{a}. To ensure the steady state, we first have

$$\sum_{j \in J} \lambda_{jk}^{a} < \mu_{k}^{a}. \tag{17}$$

Under the existence of steady state, the expected processing delay d_{k}^{a} at each BS can be written as:

$$d_{k}^{a} = \frac{1}{\mu_{k}^{a} - \sum_{j \in J} \lambda_{jk}^{a}}. \tag{18}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2508382, IEEE Transactions on Emerging Topics in Computing

Therefore, the overall expected delay for application a from user i going through j and k is

$$d_{ijk}^a = d_{ij}^a x_{ij} + D_{jk} y_{jk}^a + d_k^a y_k^a. \tag{19}$$

To satisfy the QoS requirement, the overall expected delay d_{ijk}^a for any application and any user shall not exceed the application delay constraint D^a. This leads to

$$d_{ij}^a x_{ij} + D_{jk} y_{jk}^a + d_k^a y_k^a \leq D^a, \forall a \in A, i \in I, j, k \in J. \tag{20}$$

Taking d_{ij} and d_k^a into (20), we further have

$$\begin{aligned} &L_i^a(x_{ij}\mu_k^a - x_{ij}\sum_{j\in J}\lambda_{jk}^a) + D_{jk}R(\sum_{s\in S} x_{ij}^s y_{jk}^a \mu_k^a - \\ &\sum_{s\in S} x_{ij}^s y_{jk}^a \sum_{j\in J}\lambda_{jk}^a) + R\sum_{s\in S} x_{ij}^s y_k^a \leq D^a R(\sum_{s\in S} x_{ij}^s \mu_k^a - \\ &\sum_{s\in S} x_{ij}^s \sum_{j\in J}\lambda_{jk}^a), \forall a \in A, i \in I, j, k \in J. \end{aligned} \tag{21}$$

4.5 An MINLP Formulation

When the subcarriers are exclusively allocated to a user, the uplink cost is determined by the uplink rate r_{ij}, regardless of the data volume. On the other hand, the inter-BS communication cost is determined by the actual network traffic as $\lambda_{jk}^a L_i^a$. Hence, the total communication cost can be calculated as

$$C_{com} = \sum_{i\in I}\sum_{j\in J} C_j^u r_{ij} + \sum_{j\in J}\sum_{k\in J} C_{jk}\lambda_{jk}^a L_i^a. \tag{22}$$

Besides the communication cost, the total cost shall also take the VM deployment cost into consideration, i.e.,

$$C_{total} = \sum_{k\in J} y_k^a C_j^a + C_{com}, \forall a \in A, k \in J. \tag{23}$$

Our goal is to minimize the overall cost by choosing the best settings of $x_{ij}, x_{ij}^s, y_{jk}^a, y_k^a, z_{ij}, \lambda_{jk}^a$, and μ_k^a. By summarizing all constraints discussed above, we can formulate this cost optimization as a mixed-integer non-linear programming (MINLP) problem.

MINLP:

$$\begin{aligned} &\min: (23), \\ &\text{s.t.}: (2), (3), (5), (6), (9)-(14), (17), (21) \\ &\quad x_{ij}, x_{ij}^s, y_{jk}^a, y_k^a \in \{0,1\}, \\ &\quad \forall a \in A, i \in I, j, k \in J, s \in S. \end{aligned}$$

4.6 Linearization

It is difficult to solve the MINLP problem directly due to its high computational complexity. Fortunately, we notice that (21) is a cubic inequation. The two cubic terms are products of two binary variables and one floating-point variable. It is possible to linearize them to lower the computation complexity. To this end, we first define a new auxiliary variable α_{jk} as follows:

$$\alpha_{ijk}^{as} = x_{ij}^s y_{jk}^a, \forall a \in A, j, k \in J, s \in S, \tag{24}$$

which can be equivalently replaced by the following linear constraints:

$$x_{ij}^s + y_{jk}^a - 1 \leq \alpha_{ijk}^{as} \leq x_{ij}^s, \forall a \in A, j, k \in J, s \in S, \tag{25}$$

$$\alpha_{ijk}^{as} \leq y_{jk}^a, \forall a \in A, j \in J, \forall k \in J, s \in S. \tag{26}$$

Constraint (21) then becomes a quadratic inequation with product terms of one binary variable and one floating-point variable as follows:

$$\begin{aligned} &L_i^a(x_{ij}\mu_k^a - x_{ij}\sum_{j\in J}\lambda_{jk}^a) + D_{jk}R\sum_{s\in S}\alpha_{ijk}^{as}(\mu_k^a - \sum_{j\in J}\lambda_{jk}^a) \\ &+ R\sum_{s\in S}\alpha_{ijk}^{as} \leq D^a R(\sum_{s\in S} x_{ij}^s \mu_k^a - \sum_{s\in S} x_{ij}^s \sum_{j\in J}\lambda_{jk}^a), \\ &\forall a \in A, i \in I, j, k \in J. \end{aligned} \tag{27}$$

Let $\gamma_{ijk}^a = x_{ij}\mu_k^a$, which can be equivalently rewritten into the following linear constraints:

$$0 \leq \gamma_{ijk}^a \leq \mu_k^a, \forall a \in A, i \in I, j, k \in J, \tag{28}$$

$$\mu_k^a + x_{ij} - 1 \leq \gamma_{ijk}^a \leq x_{ij}, \forall a \in A, i \in I, j, k \in J. \tag{29}$$

Similarly, letting $\delta_{ijk}^a = x_{ij}\lambda_{jk}^a$, $\epsilon_{ijk}^{as} = \alpha_{ijk}^{as}\mu_k^a$, $\eta_{ijk}^{as} = \alpha_{ijk}^{as}\lambda_{jk}^a$, $\theta_{ijk}^{as} = x_{ij}^s\mu_k^a$, and $\xi_{ijk}^{as} = x_{ij}^s\lambda_{jk}^a$, we have the following new linear constraints:

$$0 \leq \delta_{ijk}^a \leq \lambda_{jk}^a, \forall a \in A, i \in I, j, k \in J, \tag{30}$$

$$x_{ij} + \lambda_{jk}^a - 1 \leq \delta_{ijk}^a \leq x_{ij}, \forall a \in A, i \in I, j, k \in J. \tag{31}$$

$$0 \leq \epsilon_{ijk}^{as} \leq \mu_k^a, \forall a \in A, i \in I, j, k \in J, s \in S, \tag{32}$$

$$\begin{aligned} &\alpha_{ijk}^{as} + \mu_k^a - 1 \leq \epsilon_{ijk}^{as} \leq \alpha_{ijk}^{as}, \\ &\forall a \in A, i \in I, j, k \in J, s \in S. \end{aligned} \tag{33}$$

$$0 \leq \eta_{ijk}^{as} \leq \lambda_{jk}^a, \forall a \in A, i \in I, j, k \in J, s \in S, \tag{34}$$

$$\begin{aligned} &\alpha_{ijk}^{as} + \lambda_{jk}^a - 1 \leq \eta_{ijk}^{as} \leq \alpha_{ijk}^{as}, \\ &\forall a \in A, i \in I, j, k \in J, s \in S. \end{aligned} \tag{35}$$

$$0 \leq \theta_{ijk}^{as} \leq \mu_k^a, \forall a \in A, i \in I, j, k \in J, s \in S, \tag{36}$$

$$\begin{aligned} &x_{ij}^s + \mu_k^a - 1 \leq \theta_{ijk}^{as} \leq x_{ij}^s, \\ &\forall a \in A, i \in I, j, k \in J, s \in S. \end{aligned} \tag{37}$$

$$0 \leq \xi_{ijk}^{as} \leq \lambda_{jk}^a, \forall a \in A, i \in I, j, k \in J, s \in S, \tag{38}$$

$$\begin{aligned} &x_{ij}^s + \lambda_{jk}^a - 1 \leq \xi_{ijk}^{as} \leq x_{ij}^s, \\ &\forall a \in A, i \in I, j, k \in J, s \in S. \end{aligned} \tag{39}$$

Hence, (21) can be written in a linear form as:

$$\begin{aligned} &L_i^a(\gamma_{ijk}^a - \sum_{j\in J}\delta_{ijk}^a) + D_{jk}R(\sum_{s\in S}\epsilon_{ijk}^{as} - \\ &\sum_{s\in S}\sum_{j\in J}\eta_{ijk}^{as}) + R\sum_{s\in S}\alpha_{ijk}^a \leq D^a R(\sum_{s\in S}\theta_{ijk}^{as} - \\ &\sum_{s\in S}\sum_{j\in J}\xi_{ijk}^{as}), \forall a \in A, i \in I, j, k \in J. \end{aligned} \tag{40}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2508382, IEEE Transactions on Emerging Topics in Computing

Finally, our **MINLP** problem can be linearized into a mixed-integer linear programming (**MILP**) below.

MILP:

$$\begin{aligned}
&\min: (23),\\
&\text{s.t.}: (2),(3),(5),(6),(9)-(14),(17),\\
&\quad (25)-(29),(30)-(39),(40),\\
&\quad x_{ij}, x^s_{ij}, y^a_{jk}, y^a_k, z_{ij}, \alpha^a_{ijk}, \beta^{as}_{ijk} \in \{0,1\},\\
&\quad \forall a \in A, i \in I, j,k \in J, s \in S.
\end{aligned}$$

5 Algorithm Design

After linearizing MINLP to MILP, we are able to solve the optimal programming model using solvers such as CPLEX and Gurobi. However, it is still time consuming due to the existence of a large number of integer variables. To this end, we propose a two-phase heuristic algorithm based on our formulation in this section.

5.1 Uplink Communication Cost Minimization

In this phase, we intend to obtain the optimal user association solution towards minimum uplink communication cost, without the consideration of task distribution. In this case, we only need to consider user association constraints (2), (3), (5) and (6) with the objective of minimizing $\sum_{i\in I}\sum_{j\in J} C_{ij} r_{ij}$, formulated as:

MILP-Uplink:

$$\begin{aligned}
&\min: \sum_{i\in I}\sum_{j\in J} C_{ij} r_{ij},\\
&\text{s.t.}: (2),(3),(5) \text{ and } (6)\\
&\quad x_{ij}, x^s_{ij} \in \{0,1\},\\
&\quad \forall i \in I, j \in J, s \in S.
\end{aligned}$$

5.2 Joint Optimization

After obtaining the user association solution, we try to find out the task distribution and VM deployment solution which minimizes both inter-BS communication cost and VM deployment cost. As this phase is based on the user association solution, the values of $x_{ij}, x^s_{ij}, \forall i \in I, j \in J, s \in S$ are already known as $x'_{ij}, x^{s'}_{ij}, \forall i \in I, j \in J, s \in S$. With uplink delay calculated by $x^{s'}_{ij}$, we can only consider the feasible task distribution relationship between $i \in I$, $j \in J$ and $k \in J$ that definitely guaratees the QoS constraints, i.e., $x'_{ij} = 1$ and $d^a_{ij} + D_{jk} \le D^a$. The task distribution that does not satisfy such condition is directly regarded as infeasible by forcing $y^a_{jk} \equiv 0$. For those feasible ones, if and only if the request received at $j \in J$ is distributed to $k \in J$, i.e., $y^a_{jk} = 1$, the processing delay at k shall be considered. Thus, the QoS constraints for those feasible task distribution can be written as

$$\begin{aligned}
d^a_{ij} + D_{jk} + d^a_k y^a_{jk} \le D^a, \forall a \in A, i \in I, j,k \in J,\\
\text{if } x'_{ij} = 1 \text{ and } d^a_{ij} + D_{jk} \le D^a.
\end{aligned} \tag{41}$$

Taking the definitions of d^a_{ij} and d^a_k in (16) and (18), respectively, we further have

$$\begin{aligned}
&L^a_i(\mu^a_k - \sum_{j\in J}\lambda^a_{jk}) + D_{jk} R \sum_{s\in S} x^{s'}_{ij}(\mu^a_k - \sum_{j\in J}\lambda^a_{jk})\\
&+ R\sum_{s\in S} x^{s'}_{ij} y^a_{jk} \le D^a R \sum_{s\in S} x^{s'}_{ij}(\mu^a_k - \sum_{j\in J}\lambda^a_{jk}),\\
&\forall a \in A, i \in I, j,k \in J.
\end{aligned} \tag{42}$$

Thus, we can obtain the MILP formulation for minimizing the inter-BS communication cost and VM deployment cost as

MILP-Joint:

$$\begin{aligned}
&\min: \sum_{j\in J}\sum_{k\in J} C_{jk}\lambda^a_{jk} L^a_i + \sum_{k\in J} y^a_k C^a_j,\\
&\text{s.t.}: (9)-(14),(17),(42)\\
&\quad y^a_k, y^a_{jk} \in \{0,1\},\\
&\quad \forall a \in A, j \in J.
\end{aligned}$$

5.3 Two-Phase LP-based Heuristic Algorithm

Based on the above two sub-problem formulations, we present our two-phase linear programming (LP) based heuristic algorithm in this section. The algorithm is briefly summarized in Algorithm 1. In this first phase, we try to find out the user association with minimum uplink communication cost. To this end, we first relax all the binaries in **MILP-Uplink** and solve the resulting **LP-Uplink** problem (line 1). Note that all the solutions after solving LP-Uplink are floating-point values. We next try to find a user association and subcarrier allocation solution by rounding the corresponding floating-point variables. Intuitively, the one with the largest value shall be converted with the highest priority. Therefore, for each user $i \in I$, we first sort values of $x'_{ij}, \forall j \in J$ obtained by solving the liearized MILP-Uplink in an ascending order in line 3 and then try to sequentially round up the value of x'_{ij} until one feasible user-association is found, i.e., with vacant subcarrier, in line 4.

By taking the values of $x'_{ij}, x'^s_{ij}, \forall i \in I, j \in J, s \in S$ and relaxing $y^a_k, y^a_{jk}, a \in A, j,k \in J$ into floating-point variables ranging from 0 to 1, we obtain an LP-relaxation of MILP-Joint as **LP-Joint** (line 6). After solving LP-Joint, we will get floating-point solutions for $y^a_k, y^a_{jk}, a \in A, j,k \in J$ (line 7), which provides us the indications on the VM placement and task distribution. That is, VM $a \in A$ shall be placed on BS $k \in K$ with non-zero value of y^a_k. Similarly, the requests received at BS $j \in J$ are with high probability to be distributed to $k \in J$. However, this does not always guarantee the QoS of all users. To this end, we first try to round up the values of any non-zero $y^a_k, y^a_{jk}, a \in A, j,k \in J$ and take them into a variant of LP-Joint, **LP-Joint-Strict**, where non-zero and zero $y^a_k, y^a_{jk}, a \in A, j,k \in J$ are forced to be 1 and 0, respectively (line 10). By solving LP-Joint-Strict, we can check whether the solution after rounding up the floating-point values can guarantee the QoS or not.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2508382, IEEE Transactions on Emerging Topics in Computing

Algorithm 1 A Two-Phase LP-based Algorithm

1: Relax all the binaries in the MILP-Uplink, and solve the resulting LP-Uplink programming
2: **for all** user $i \in I$ **do**
3: Sort $x'_{ij}, j \in J$ decreasingly into set X_i
4: Sequentially try to round the variables in X_i if vacant subscarrier on the corresponding BS is available
5: **end for**
6: Construct an LP problem by integrating fixed $x'_{ij}, x'^{s}_{ij}, i \in I, j \in J, s \in S$ as

LP-Joint:

$$\min : \sum_{j \in J} \sum_{k \in J} C_{jk} \lambda^a_{jk} L^a_i + \sum_{k \in J} y^a_k C^a_j,$$
$$\text{s.t. :}(9) - (14), (17), (42)$$

7: Solve above LP-Joint, and get the solution of $y^a_k, y^a_{jk}, a \in A, j, k \in J$
8: $QoS\_guaranteed \leftarrow$ False
9: **while** not $QoS\_guaranteed$ **do**
10: Solve the **LP-Joint-Strict** by forcing the values of non-zero $y^a_k, y^a_{jk}, a \in A, j, k \in J$ to be 1 and the others as 0, and get the corresponding $cost$
11: **if LP-Joint-Strict** is feasible **then**
12: QoS_guaranteed $\leftarrow$ True
13: **else**
14: Solve **LP-Joint-Relax** to get new floating-point values of $y^a_k, y^a_{jk}, a \in A, j, k \in J$
15: **end if**
16: **end while**
17: Sort the values of μ^a_k after solving **LP-Joint-Strict** into M ascendingly
18: **for all** $m \in M$ **do**
19: Try not to place a on k by forcing $y^a_k \equiv 0$
20: Solve **LP-Joint-Strict** to check feasibility
21: **if** new solution is feasible and $new\_cost < cost$ **then**
22: $cost \leftarrow new\_cost$
23: Redistribute the tasks accordingly
24: **else**
25: Recovery y^a_k back to be 1
26: **end if**
27: **end for**

If not, we try to solve another variant of LP-Joint, **LP-Joint-Relax**, where non-zero $y^a_k, y^a_{jk}, a \in A, j, k \in J$ are forced to be 1 while the others are still treated as floating-point variables ranging from 0 to 1 (line 14). By solving LP-Joint-Relax, we may get more non-zero values $y^a_k, y^a_{jk}, a \in A, j, k \in J$, which will be then rounded up and taken into LP-Joint-Strict to check whether QoS is guaranteed. Such routine continues until the QoS gets satisfied, i.e., LP-Joint-Strict is feasible (line 12).

For a better solution without violating the QoS constraints, we next try to lower the VM placement cost and reschedule the task distribution accordingly (line 17). The philosophy behind such design is that the VM with small task load could be potentially released and the tasks allocated onto it can be migrated to others, provided that the QoS is guaranteed. Therefore, we first sort $\mu^a_k, \forall a \in A, k \in J$ into set M ascendingly. For each element in M, we try to release its corresponding VM a on k by forcing y^a_k to be 0 (line 19). Taking the new values of $y^a_k, a \in A, k \in J$ into **LP-Joint-Strict**, we first check whether VM a on BS k can be released or not (line 20). If it is feasible, we can further check whether the resulting new cost is less than current cost (line 21). If it is, we treat the new solution as current best solution and set the cost value correspondingly (lines 22 and 23); otherwise, we continue to check the next VM placement (line 25). This procedure proceeds until all elements in M get checked (lines 18-27). Finally, we get our VM placement and task distribution solution.

Remark: Algorithm 1 is with polynomial time complexity. Obviously, all the loops in Algorithm 1 are with limited number of iterations. For example, in either the phase to find out one feasible solution in lines 9-16 or the one trying to lower the VM placement cost in lines 18-27, at most $|A| * |J|$ number of iterations are needed ($|\cdot|$ denotes the cardinality function). While, in each iteration, only LP problem is involved. It has been well known that we are able to get the optimal solution of LP problem in polynomial time [12]. Therefore, the whole algorithm can be solved in polynomial time.

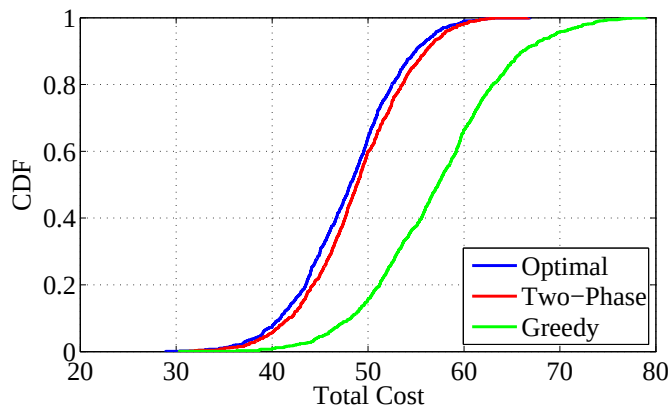


Fig. 3. CDF of the total cost

6 Performance Evaluation

In this section, we present the performance results of our two-phase LP-based algorithm by comparing it against a greedy algorithm, which goes as follows. It first greedily associates each user to the BS with the smallest uploading cost provided that there is vacant subcarrier on the BS. After that, it checks over all the BSs as potential processing BS to find out the one with the minimum incremental cost, including both the VM deployment cost and the inter-BS communication cost.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2508382, IEEE Transactions on Emerging Topics in Computing

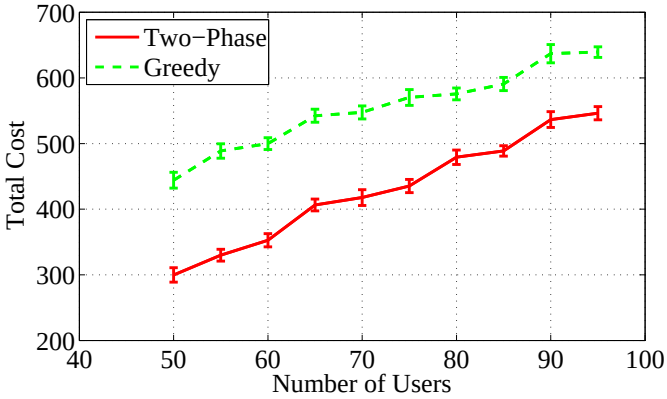


Fig. 4. On the effect of the number of users

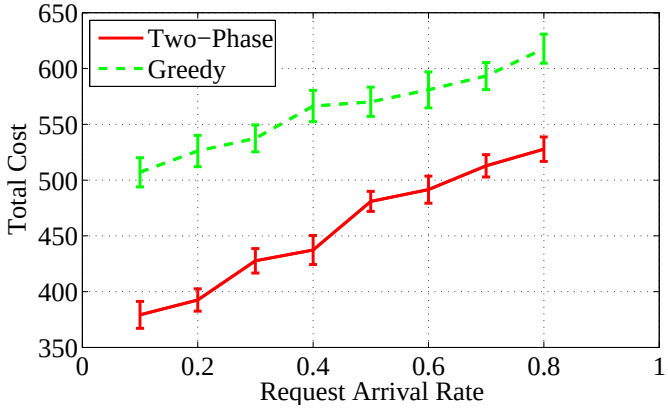


Fig. 5. On the effect of arrival rate

Let us first check the performance comparison of all three algorithms, i.e., optimal, two-phase and greedy, in a small-scale network in size 100×100, under the settings of 8 users and 5 BSs with the communication range of 10. Each user has 2 applications with arrival rate uniformly distributed in the range of $[0.1, 0.5]$ and the application data lengths are all set as 5. The maximum tolerable delay of each application is 10. There are 2 subcarriers available at each BS. The capacity of each resource in a BS is normalized to 1 and the communication cost, VM cost, link delays are randomly generated in the range of $[1, 3]$, $[3, 6]$ and $[1, 3]$, respectively. We run 1000 simulation instances with different random seeds and plot the cumulatively distribution function (CDF) of the total cost for all the three algorithms in Fig. 3. To solve the MILP problem for the optimal solution and the LP in the heuristic two-phase algorithm, commercial solver Gurobi[3] is used.

We can see that "Two-Phase" closely achieves the performance of "Optimal" and significantly outperforms "Greedy". This validates the correctness and high efficiency of our two-phase algorithm. It is quite time consuming to get an optimal solution for larger topology. As we have validated the near-optimality of our two-phase algorithm, we omit the optimal solution hereafter to investigate how various parameters, e.g., QoS requirement, the number of subcarriers, request rate, etc., affect the total cost in a larger network.

The default settings of our large-scale network experiments are as follows. We consider an area in size 300×300 with $|I| = 80$ users and $|J| = 50$ BSs with 5 subcarriers each. The communication ranges of BS are all set as 20. In each simulation, the communication cost, VM cost, link delays are randomly generated in the range of $[1, 3]$, $[3, 6]$ and $[1, 3]$, respectively. While the task arrival rate is uniformly distributed in the range of $[0.1, 0.5]$ with the application data lengths are all set as 5. The maximum tolerable delay for each application task is set as 8. The We run each simulation 200 times and obtain the mean value with associated standard deviation.

Fig. 4 shows the total cost under different number of users varying from 50 to 95. We can see that the total costs of both algorithm rise with the number of users. This is because more users bring in more user requests, leading to more traffic between BSs and more task VM deployments hence a higher total cost. An interesting observation from Fig. 4 is that when the number of users is large, e.g., 90, the results of greedy algorithm approach to our two-phase algorithm. When we have only 50 users, the constraints of storage, link capacity or delay are not so strict since all resources are sufficient. Both algorithms have enough optimization space to lower the total cost. With the increase of users, the requests for all kinds of resources are also increasing. In this case, more resources in BSs are used, leaving less space for optimization. Therefore, the gap between two algorithms decreases.

Similar phenomena can be also observed from Fig. 5. When the maximum request rate is small, e.g., 0.1, our algorithm shows a significant advantage over the greedy one. However, after the maximum request rate reaches 0.6, the gap between two algorithms again decreases. The reason is that the resources of most BSs (subcarriers, storage and computation units) are occupied for data uploading and computation due to the high data loads from users, leaving less space for optimization. Besides, we can see that the total cost shows as an increasing function of the task arrival rates for both algorithms. To process more requests with guaranteed QoS, more subcarriers and computation resource are needed. This leads to a larger communication traffic and hence a higher communication cost between users and BSs. Meanwhile, more VMs are required to be placed in BSs and this will also lead to a higher VM cost. Therefore, the results of both algorithms increase. Nevertheless, under all settings our algorithm outperforms the greedy one.

Then, Fig. 6 illustrates the total cost as a decreasing function of the number of BSs from 50 to 77. Larger number of BSs leads to more resources and better options during BS association, task distribution and VM

3. www.gurobi.com

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2508382, IEEE Transactions on Emerging Topics in Computing

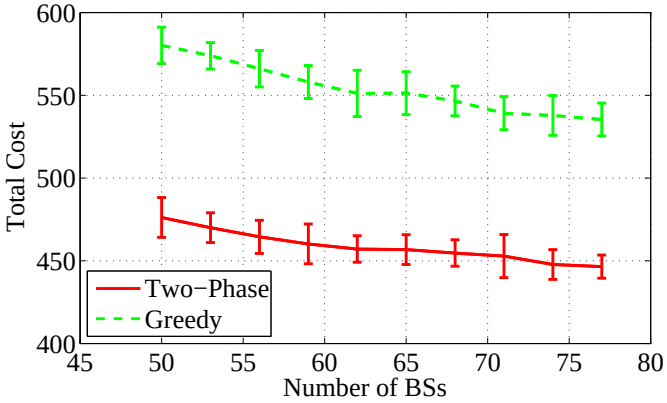


Fig. 6. On the effect of the number of BSs

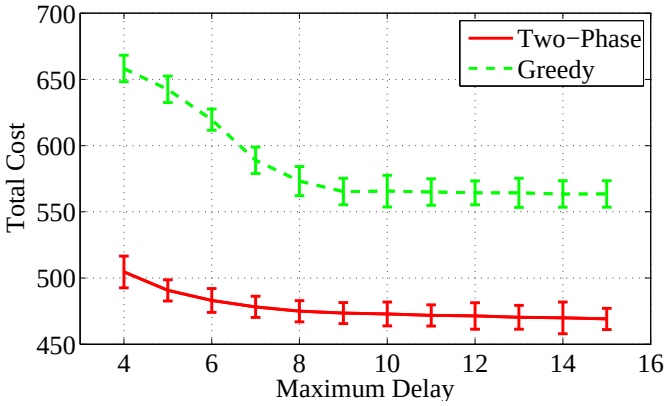


Fig. 7. On the effect of maximum delay

deployment. Hence, the total costs of both algorithms decrease. As observed from Fig. 6, when the number of BSs reaches 65, the total costs of both algorithms converge. This is because when the network is large enough most data and their corresponding VMs can always be placed in BSs with minimum communication and VM costs. Further increasing the number of BSs (e.g., from 65 to 77) will no longer change the distributions of tasks or VM deployment or reduce the total costs significantly.

Next, we show in Fig. 7 the results for evaluating the effect of the maximum tolerable delay for each application D^a from 4 to 15. The advantage of our two phase algorithm over greedy can still be observed. We notice that the total cost is a non-increasing function of D^a. The reason is that when D^a is larger, less VMs are needed in BSs to guarantee the QoS. Therefore, the VM costs of both algorithms shall decrease as the maximum delay increases. Moreover, a looser QoS requirement also helps find cost-efficient routing strategies for data uploading because cheaper links with larger delay can be selected. However, as the maximum delay increases, the results of both algorithms converge. This is because with a larger maximum delay, most users and their corresponding data can be associated to and processed at BSs with the cheapest communication and VM cost. Hence, increasing

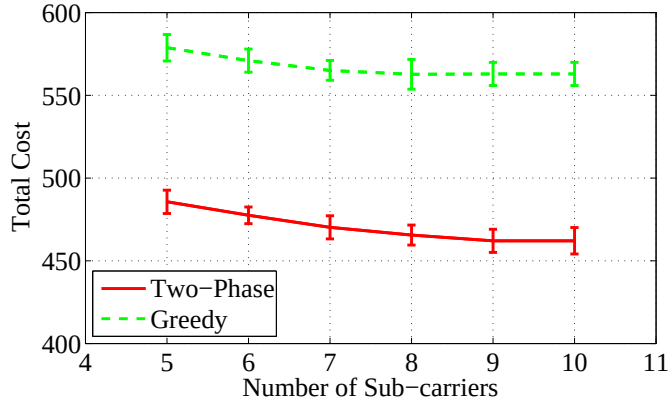


Fig. 8. On the effect of maximum delay

the maximum tolerable delay will not effect the total cost any more.

Finally, Fig. 8 investigates the effect of the number of subcarriers in each BS varying from 5 to 10. An interesting observation from Fig. 8 is that the total cost first decreases and then converge with the increasing number of subcarriers. More subcarriers in each BS result in a smaller uploading delay and allow more user connections at the same time. This gives more optimization space for communication cost via BS association and data routing. However to process data requests with a fixed average arrival rate, certain number of VMs must be placed in BS. This leads to a minimum VM cost, and hence the total costs shall not continuously decrease with the number of subcarriers.

7 Related Work

7.1 Medical Cyber Physical System

The advantages of MCPS are increasingly attracting the attentions from both academic and industry. Lee et al. [4] discuss the developing trends in MCPS, including increased reliance on software to deliver new functionality, wider use of network connectivity in MCPS, demand for continuous patient monitoring, and new opportunities for research and development. Later they discuss the design of complex MCPS with both security and effectiveness consideration. They further summarize the challenges and research directions in MCPS and emphasize the significance of software platform for MCPS in [5]. Recently, Don et al. [13] describe a three-layer (i.e., physical layer, service layer and application layer) big data processing framework for MCPS with dynamic provisioning and fully elastic system for decision making in health care applications. Our work is complementary to existing MCPS studies that we integrate the emerging fog computing paradigm to support MCPS in the consideration of scalability, awareness and latency.

7.2 Base Station Association

There have been many studies [14]–[17] on base station association policies under different models and assump-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2508382, IEEE Transactions on Emerging Topics in Computing

tions. Son et al. [14] develop a theoretical framework for BS energy saving that encompasses dynamic BS operation and the related problem of user association together. They formulate a cost minimization that allows for a flexible tradeoff between flow-level performance and energy consumption. Lees et al. [15] propose a novel BS association scheme to tackle the challenges when temporal fairness is not guaranteed, or deterministic BS association may make wrong decisions, e.g., under the dynamics of mobility or flow arrivals/departures. Kim et al. [16] develop a framework for user association in infrastructure-based wireless networks, specifically focusing on flow-level cell load balancing under spatially inhomogeneous traffic distributions. Sung et al. [17] formulate a predictive association game, which is designed to determine an optimal base station for maximum throughput and reduced handover frequency. Ibach et al. [18] propose consensus protocols and task distribution in swarm robots environments. Our work by the first time envisions a new scenario that the BSs are augmented with computation resources to host cyber services. In this case, base station association is related not only to the wireless communications, but also to the computations. The two issues are highly correlated and jointly investigated.

7.3 Fog Computing

Cisco provisions that there will be around 50 billion network connected devices by 2020 [19]. Fog computing is thus proposed [8], with the concept of putting the cloud services to the network edge. Right after its emergence, many efforts have been devoted to this area. Zhu et al. [20] discover that it is possible to adapt the user conditions (e.g., network status, device computing load, etc.) to improve the web site performance by edge servers in fog computing architecture. Vaquero et al. [21] discuss several key technologies related fog computing such as network function virtulization, peer-to-peer, IoT and outline several research challenges. Although they mention the importance of management in fog computing, no actual solution is proposed. Mukherjee et al. [22] provide an experience report on the use of a grid computing framework called Condor to realize data-parallel "black-box" style execution of analysis for real time or near real-time IoT data processing. They also envision that it is possible to utilize the edge devices for computation, thereby extending he computation to the edges of a compute cloud rather than restricting it to the core of servers. Recently, Stantchev et al. [23] exemplify the benefit of cloud computing and fog computing to healthcare and elderly-care by an application scenario developed in OpSIT-Project in Germany. Furthermore, in a special issue Colomo-Palacios et al. [24] assess the state of the art in semantic technologies and linked data in grid and cloud environments. Their study highlights the importance of fog computing to healthcare application. Inspired by such concept, our work further investigates the cost-efficient resource management issues in FC-MCPS and provides many theoretical results to guide the practical FC-MCPS deployment.

8 Conclusion

Witnessing the vast potential of fog computing, we are inspired to forge fog computing into MCPS and propose FC-MCPS. To tackle the cost-efficiency problem in FC-MCPS, we argue that BS association, task distribution and VM deployment are all critical. Therefore, in this paper, we jointly study these three issues towards minimizing the overall cost while satisfying the QoS requirement. Specially, the problem is first fomulated into an MINLP problem. To tackle the high computational complexity of solving MINLP, we linearize it into an MILP problem and propose a two-phase LP-based heuristic algorithm accordingly. Through extensive experiments, we show that our algorithm has substantial advantage over the greedy one.

References

[1] S. Khaitan and J. McCalley, "Design techniques and applications of cyberphysical systems: A survey," *Systems Journal, IEEE*, vol. PP, no. 99, pp. 1–16, 2014.

[2] V. Stantchev, R. Colomo-Palacios, and M. Niedermayer, "Cloud Computing Based Systems for Healthcare," *The Scientific World Journal*, vol. 2014, 2014.

[3] A. Cheng, "Cyber-physical medical and medication systems," in *Distributed Computing Systems Workshops, 2008. ICDCS '08. 28th International Conference on*, June 2008, pp. 529–532.

[4] I. Lee and O. Sokolsky, "Medical cyber physical systems," in *Proceedings of the 47th Design Automation Conference*. ACM, 2010, pp. 743–748.

[5] I. Lee, O. Sokolsky, S. Chen, J. Hatcliff, E. Jee, B. Kim, A. King, M. Mullen-Fortino, S. Park, A. Roederer *et al.*, "Challenges and research directions in medical cyber–physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 75–90, 2012.

[6] "the case for mobile edge-clouds."

[7] M. Beck, S. Feld, A. Fichtner, C. Linnhoff-Popien, and T. Schimper, "Me-volte: Network functions for energy-efficient video transcoding at the mobile edge," in *Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on*, Feb 2015, pp. 38–44.

[8] F. , R. Milito, J. Zhu, and S. 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*. ACM, 2012, pp. 13–16.

[9] V. Stantchev and G. Tamm, "Addressing non-functional properties of services in it service management," *Non-functional properties in service oriented architecture: Requirements, models and methods*, pp. 324–334, 2011.

[10] I. Marshall and C. Roadknight, "Linking cache performance to user behaviour," *Computer Networks and ISDN Systems*, vol. 30, no. 223, pp. 2123 – 2130, 1998.

[11] S. Gunduz and M. Ozsu, "A poisson model for user accesses to web pages," in *Computer and Information Sciences - ISCIS 2003*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, vol. 2869, pp. 332–339.

[12] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*. ACM, 1984, pp. 302–311.

[13] S. Don and D. Min, "Medical cyber physical systems and big-data platforms," in *The Fourth Workshop on Medical Cyber-Physical Systems*, 2013.

[14] K. Son, H. Kim, Y. Yi, and B. Krishnamachari, "Base station operation and user association mechanisms for energy-delay tradeoffs in green cellular networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 8, pp. 1525–1536, September 2011.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2508382, IEEE Transactions on Emerging Topics in Computing

[15] S. Lee, K. Son, H. Gong, and Y. Yi, "Base station association in wireless cellular networks: An emulation based approach," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 8, pp. 2720–2729, 2012.

[16] H. Kim, G. de Veciana, X. Yang, and M. Venkatachalam, "Distributed α-optimal user association and cell load balancing in wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 1, pp. 177–190, Feb 2012.

[17] N. Sung, N.-T. Pham, H. Yoon, S. Lee, and W. Hwang, "Base station association schemes to reduce unnecessary handovers using location awareness in femtocell networks," *Wireless Networks*, vol. 19, no. 5, pp. 741–753, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11276-012-0498-0

[18] P. Ibach, N. Milanovic, J. Richling, V. Stantchev, A. Wiesner, and M. Malek, "Cero: Ce robots community," *IEE Proceedings-Software*, vol. 152, no. 5, pp. 210–214, 2005.

[19] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO white paper*, vol. 1, 2011.

[20] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving web sites performance using edge servers in fog computing architecture," in *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*. IEEE, 2013, pp. 320–323.

[21] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

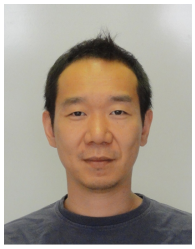
[22] A. Mukherjee, S. Dey, H. Paul, and B. Das, "Utilising condor for data parallel analytics in an iot context; an experience report," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on*, Oct 2013, pp. 325–331.

[23] V. Stantchev, A. Barnawi, S. Ghulam, J. Schubert, and G. Tamm, "Smart items, fog and cloud computing as enablers of servitization in healthcare," *Sensors & Transducers*, vol. 185, pp. 121–128, 2015.

[24] R. Colomo-Palacios, V. Stantchev, and A. Rodríguez-González, "Special issue on exploiting semantic technologies with particularization on linked data over grid and cloud architectures," *Future Generation Computer Systems*, vol. 32, pp. 260–262, 2014.

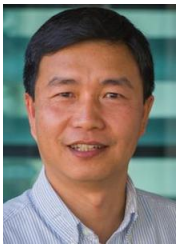
Lin Gu received her Ph.D. and M.S. degrees in computer science from University of Aizu, Fukushima, Japan, in 2015 and 2011, respectively. She received her B.S. degree from School of Computer Science and Technology, Huazhong University of Science and Technology, China in 2009. She is now with the Services Computing Technology and System Lab, Cluster and Grid Computing Lab in the School of Computer Science and Technology, Huazhong University of Science and Technology. Her current research interests include cloud computing, Big Data and Software-defined Networking.

Deze Zeng received his Ph.D. and M.S. degrees in computer science from University of Aizu, Aizu-Wakamatsu, Japan, in 2013 and 2009, respectively. He received his B.S. degree from School of Computer Science and Technology, Huazhong University of Science and Technology, China in 2007. He is currently an associate professor in School of Computer Science, China University of Geosciences, Wuhan, China. His current research interests include: cloud computing, software-defined sensor networks, data center networking, networking protocol design and analysis. He is a member of IEEE.



Song Guo received the PhD degree in computer science from the University of Ottawa, Canada. He is currently a Full Professor at the University of Aizu, Japan. His research interests are mainly in the areas of wireless communication and networking, cloud computing, big data, and cyber-physical systems. He has published over 300 papers in referred journals and conferences in these areas and received three IEEE/ACM best paper awards. Recently, he coauthored two monographs titled "Cloud Networking for Big Data" and "Cooperative Device-to-Device Communication in Cognitive Radio Cellular Networks", and coeidted a textbook "Big Data Concepts, Theories, and Applications" with Springer. He has served as Associate Editor of IEEE Transactions on Emerging Topics, IEEE Transactions on Parallel and Distributed Systems, Wireless Networks, and many other major journals. He has also been in organizing and technical committees of numerous international conferences. Dr. Guo is a senior member of the ACM, a senior member of the IEEE, and an IEEE Communications Society Distinguished Lecturer.

Ahmed Barnawi is an associate professor at the Faculty of Computing and IT, King Abdulaziz University, Jeddah, Saudi Arabia, where he works since 2007. He received PhD at the University of Bradford, UK in 2006. He was visiting professor at the University of Calgary in 2009. His research areas are cellular and mobile communications, mobile ad hoc and sensor networks, cognitive radio networks and security. He received three strategic research grants and registered two patents in the US. He is IEEE member.



Yong Xiang received the Ph.D. degree in Electrical and Electronic Engineering from The University of Melbourne, Australia. He is a Professor and the Director of the Artificial Intelligence and Image Processing Research Cluster, School of Information Technology, Deakin University, Australia. His research interests include signal and system estimation, information and network security, multimedia (speech/image/video) processing, and wireless sensor networks. He has served as Program Chair, TPC Chair, Symposium Chair, and Session Chair for a number of international conferences. He currently serves as Associate Editor of IEEE Access. Dr. Xiang is a senior member of the IEEE.