IEEE TRANSACTIONS ON
EMERGING TOPICS
IN COMPUTING

Received 20 March 2015; revised 1 July 2015; accepted 7 September 2015. Date of publication 15 September 2015; date of current version 2 March 2018.

Digital Object Identifier 10.1109/TETC.2015.2479093

Decentralized and Optimal Resource Cooperation in Geo-Distributed Mobile Cloud Computing

RONG YU[1], (Member, IEEE), JIEFEI DING[1], (Student Member, IEEE), SABITA MAHARJAN[2], (Member, IEEE), STEIN GJESSING[2], (Senior Member, IEEE), YAN ZHANG[2], (Senior Member, IEEE), AND DANNY H. K. TSANG[3], (Fellow, IEEE)

[1]School of Automation, Guangdong University of Technology, Guangzhou 510006, China
[2]Simula Research Laboratory, Fornebu 1364, Norway
[3]Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong

CORRESPONDING AUTHOR: Y. ZHANG (yanzhang@simula.no)

This work was supported in part by the National Natural Science Foundation of China under Grant 61422201, Grant 61370159, Grant U1201253, and Grant U1301255, in part by the Guangdong Province Natural Science Foundation under Grant S2011030002886, in part by the Special-Support Project of Guangdong Province under Grant 2014TQ01X100, in part by the High Education Excellent Young Teacher Program of Guangdong Province under Grant YQ2013057, in part by the Science and Technology Program of Guangzhou under Grant 2014J2200097 through the Zhujiang New Star Program, in part by the Research Council of Norway under Project 240079/F20, and in part by the European Commission FP7 Project CROWN under Grant PIRSES-GA-2013-627490.

ABSTRACT Mobile cloud computing is a key enabling technology in the era of the Internet of Things. Geo-distributed mobile cloud computing (GMCC) is a new scenario that adds geography consideration in mobile cloud computing. In GMCC, users are able to access cloud resources that are geographically close to their mobile devices. This is expected to reduce the communication delay and the service providers' cost compared with the traditional centralized approach. In this paper, we focus on resource sharing through the cooperation among the service providers in geo-distributed mobile cloud computing. Then, we propose two different strategies for efficient resource cooperation in geographically distributed data centers. Furthermore, we present a coalition game theoretical approach to deal with the competition and the cooperation among the service providers. Utility functions have been specifically considered to incorporate the cost related to virtual machine migration and resource utilization. Illustrative results indicate that our proposed schemes are able to efficiently utilize limited resources with quality-of-service consideration.

INDEX TERMS Mobile cloud computing, resource management, cooperation, game theory.

I. INTRODUCTION

Geo-distributed mobile cloud computing (GMCC) is an emerging paradigm that integrates location information in mobile cloud computing [1]. In GMCC, users are able to access cloud storage and computation resource that are geographically close to their mobile devices [2], [3]. However, vehicles' high mobility poses a significant challenge for maintaining a stable network topology as well as providing reliable resource. In addition, various applications have different resource requirements [4]. Running mobile applications in GMCC needs to meet multiple resource requirement and latency requirement. Therefore, considering users' behavior and geographic information will make management strategy of cloud computing resource more resource-efficient and cost-efficient.

GMCC makes cloud computing concept more than a cluster of computer devices with unified features. This concept results in cloud that combines with cloudlet with a number of benefits: closer to user, lower communications delay [5] and lower data transmission flow [6]. It keeps the superiority of traditional cloud computing: providing users to run computation-intensive applications which are not easily performed on a resource-constrained mobile device [7], [8]. GMCC is in particular beneficial for high-mobility vehicle network where vehicles commonly have position information at any time. In case of fast moving vehicles, cost-efficient resource allocation scheme is very important for location based service, navigation service, and accident alert [9]. Therefore, it makes resource allocation more complicated and crucial.

Centralized infrastructure of mobile cloud computing schemes are well studied in many researches. However, such centralized data centers may bring several disadvantages, including limited resource sharing, high bandwidth for communications, and long distance to users [10]–[12]. It is envisioned that mobile cloud computing will gradually develop into distributed infrastructures, which can satisfy ever-increasing computing requirements for mobile users in a large area [13]. In this scenario, each data center is mainly responsible for users nearby. Service providers (SPs) can also take advantage of geo-diversity to increase revenue and enhance performance [14]. Network topology which includes both geographic information and connectivity is very significant for cloudlet distribution and connection. By employing the topology information, cloud resource allocation can reach global optimization.

Resource refers to a cluster of physical resource and bandwidth resource in the traditional cloud framework [15], [16]. Several resource allocation studies employ the centralized architecture of a data center. Barbarossa *et al.* [16] proposed a joint radio and computation resource optimization scheme in a single base station to minimize the transmission power of mobile devices. Kusic *et al.* [17] developed a dynamic resource provisioning framework for a multi-objective optimization in a virtualized computing environment. Computation resource is virtualized in the server environment which allows resource to be shared among multiple Virtual Machines (VMs). Khanna *et al.* [18] proposed a new online method to place VMs into physical hosts while minimizing the number of hosts to reduce cost. However, when data centers are far from mobile terminals, the users may experience high response delay and low QoS. Zhang *et al.* [19] investigated the cloud-based mobile application platform and developed an energy-efficient scheduling policy for collaborative tasks execution. Liu *et al.* [20] presented a new server selection strategy in mobile clouds in order to reduce services delay and improve services quality in mobile environments. The results demonstrated that short distance to data centers results in small bandwidth consumption, short startup delay, and satisfactory service quality.

In this paper, we focus on SPs' cooperation by resource sharing in resource allocation problem. In geo-distributed mobile cloud computing, the cloud resource and data centers are geographically distributed over a wide-area network. Thus, the cooperation can be further classified into two schemes: the local resource sharing and the remote resource sharing. To tackle this cooperation problem, we propose a coalition game theoretical approach based on the resource trading model. As a consequence, resource sharing and SPs' cooperation not only improve the resource utilization to achieve more revenues, but also increase the VM allocation rate to users. QoS is largely improved as more and more users that can access the services. The major contributions of this paper can be summarized as follows.

- We propose to exploit the cooperation among SPs in GMCC in order to significant increase resource utilization. The resource-rich SPs are encouraged to lease a portion of their resource to the resource-deficient SPs.
- We formulate the resource cooperation among SPs in the coalition game theory framework and then we leverage pricing mechanism and users demand to stimulate the resource cooperation.
- We optimize the virtual machines migration and resource allocation to deal with the vehicle mobility. We employ the graph theory to find the global optimization with high QoS and satisfying revenues.

The rest of this paper is organized as follows. Geo-distributed mobile cloud computing and resource cooperation are described in Section II. In Section III, we present problem formulation and analysis. In Section IV, we describe the proposed coalition game theoretical approach. The coalition formation scheme and the theoretical proof are given in Section V. Illustration results are given Section VI. Section VII concludes the paper.

II. GEO-DISTRIBUTED MOBILE CLOUD COMPUTING AND RESOURCE COOPERATION

In this section, we first describe the GMCC network architecture and then we present the resource cooperation cases among SPs.

A. GMCC NETWORK

Fig. 1 shows the GMCC network architecture where the data centers are distributed in each region. The data centers are mainly responsible for the local applications from mobile devices (MDs), e.g., vehicles and mobile phones. In traditional approach, applications highly relay on the capability of mobile device as application running on a mobile device. However, the Internet and cloud computing

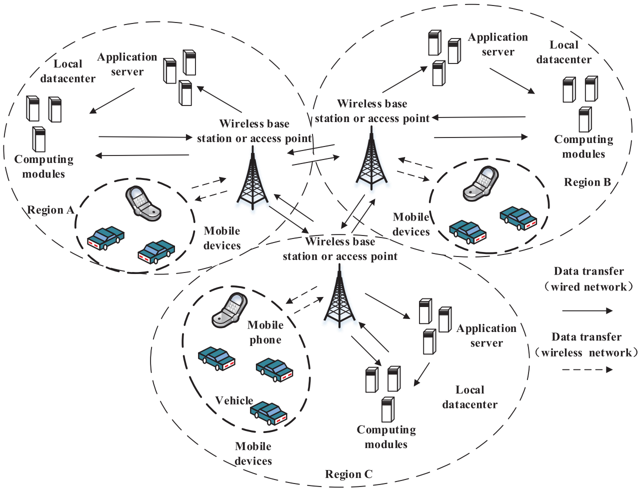


FIGURE 1. The geo-distributed mobile cloud computing environment.

technology break through this bottleneck and bring more opportunities to portable devices. By communicating with the roadside units, a vehicle can access services in the infrastructure-based mobile cloud computing network through wireless network access, e.g. Wireless Mesh Networks [21], [22]. People can finish the job on cloud by MDs as long as they can access to cloud resource. Data center is the core of cloud that it provides computing module to the application server. Service provider (SP) as the manager of cloud resource will allocate resource by VM to each user to running applications. Each data center is responsible for a certain area to provide timely service response.

Running application in cloud will consume resource from the application server [23], [24]. Therefore, both radio resource and computing resource are requested from data centers. Resource allocation can be further classified into two kinds. The local resource allocation is suitable for applications which has a small traveling radius or high latency requirement service. The remote resource allocation requires high bandwidth for data transmission. Thus, it can be employed to the situation when application requests exceed its capability or users approach to leaving. VM migration bridges the two kinds of resource allocation by migrating service to the region it moves toward. Users can enjoy the local service even when they step to other region. Resource cooperation tends to balance the extreme unbalance of SPs resource utilization. In this framework, the resource cooperation can be performed in the same data center or among different data centers.

B. RESOURCE COOPERATION IN GMCC NETWORK

Fig.2 shows the details of resource cooperation. SPs can provide a certain kind of service and have resource in different data centers in GMCC network. The three data centers are located in different regions. More specifically, SP_1 rents resource from data center A and data center B. SP_2 rents resource from data center A and data center C, and SP_3 has resource in data center B and data center C. Actually, there are many SPs that rent resource in the same data center, and each SP may have resource from more than two data centers. It is noticed that the same SPs in different data centers may have different coefficient of resource utilization. For instance, SP_1 in data center B has a higher coefficient than it has in data center A. In the same data center, the coefficient of resource utilization is also different among different SPs. For instance, SP_2 has higher resource utilization than SP_3 in data center C.

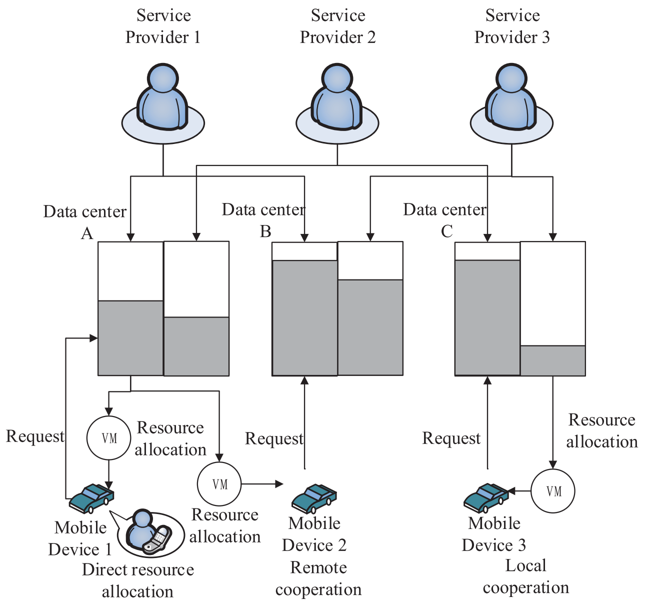


FIGURE 2. **Resource sharing and cooperation.**

In the case of non-cooperation, SPs will directly allocate the resource to user when it has enough resource. Take mobile device 1 (MD_1) for example. The application request from MD_1 can be directly allocated with VM on local SP_1. VM is a software platform that resembles the underlying hardware of physical machine through hardware resource virtualization. With enough resource, the host physical machine could simultaneously operate multiple VMs form different application requirements independently. We consider two kinds of resource sharing: the local cooperation and the remote cooperation.

1) REMOTE COOPERATION

In the remote cooperation, the resource sharing happens between different data centers. For instance, MD_2 sends the application request to the local operator SP_1 in data center B. Since the local operator runs at high level of utilization and has few resource to operate new applications, SP_1 may ask for resource sharing from a remote cooperator, such as SP_1 in data center A. The remote cooperator will evaluate the revenue before it agrees to establish the remote cooperation.

2) LOCAL COOPERATION

In the local cooperation, the resource sharing is between different SPs in the same data center. For example, SP_3 receives the application request from MD_3 and ask SP_2 for cooperation to extent the available resource. If SP_2 and SP_3 can both obtain benefits through working together, the local coalition is formed by running the application on device of SP_3.

III. PROBLEM FORMULATION

In this section, we first describe the virtualized framework of GMCC network and give the definition of 2-layer network graph. Then, we build the economic model. Finally, we discuss VM resource allocation.

A. WIRELESS NETWORK AND DATA CENTER

We consider M geographically distributed clouds in different regions. The resource in each cloud can be rented by N different SPs. One SP can rent long-term reserving resource from more than one region. SP k in region l is denoted by

SP_k^l ($k = 1, 2, \cdots, N$, $l = 1, 2, \cdots, M$). The long-term reserving resource is fixed asset of SP_k^l, and denoted by the maximum capacity of each kind of resource: the CPU resource ($\max \mathrm{C}_k^l$), the memory resource ($\max \mathrm{M}_k^l$) and the bandwidth resource ($\max \mathrm{B}_k^l$). The bandwidth resource in mobile network is radio resource for wireless access. The occupied resource should satisfied the constrains. SP can extend the capability by renting short-term resource from other SPs as an on-demand basis method. The resource requirement from user j can be denoted as $R_j = (k, l, C_j, M_j, B_j, T_j)$, which includes the information of SP k in region l, the required CPU resource C_j, the required memory resource M_j, the required bandwidth B_j, and the maximum latency time T_j. Every application has a specific maximum latency T_j which should not be exceeded to provide a satisfactory service. The model in this paper is proposed to take advantages of both long-term reserving resource and short-term demanding resource. Resource cooperation provides a approach for enhancing service capability and making full use of cloud resource.

The directed 2-layer graph $g^i = \{V, E^i, w\}$ ($i = 1, 2$) is employed to describe the resource distribution and relationship of SPs, as shown in Fig. 3. The set of vertex V denotes all SPs in the graph [25]. We encode SP_k^l from S_1 to S_{12} in the graph to form a matrix and distinguish them. SPs in the graph are connected through wired communication. The connected edge E^i is proposed for the capability to form an coalition. Weight w on each edge denotes the difference of resource utilization. For example, $w_{i,j}$ is the difference of resource utilization of S_i and S_j. If S_i rent resource from S_j and $w_{i,j} > 0$, the utilization of S_i and S_j will come to balance. Therefore, weight can be a director on balancing the resource allocation in network. The direction of network may change over time. A 2-layer graph framework is proposed for different resource sharing. The first layer represents the CPU resource and memory resource which can be shared between different data centers. The second layer is related to bandwidth resource which can only be shared in the same data center. Thus, the connected edge may be different in different layers.

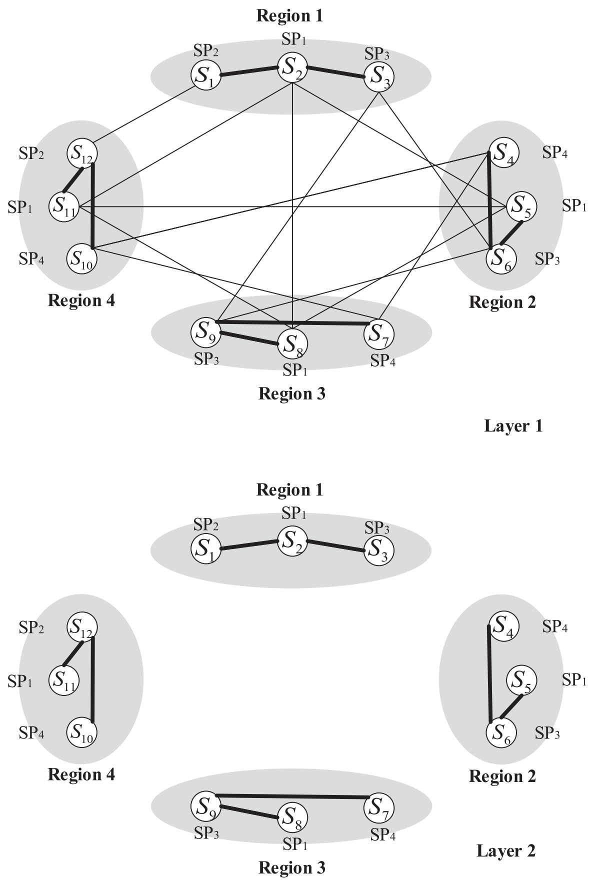


FIGURE 3. **The graph of the GMCC network.**

The resource sharing in the same data center is denoted by bold lines in Fig. 3. The local cooperation is based on the agreement of sharing both physical resource (e.g. CPU resource or memory resource) and bandwidth resource. For example, $e^1_{\mathrm{S}_1,\mathrm{S}_2} = e^2_{\mathrm{S}_1,\mathrm{S}_2} = 1(e^1_{\mathrm{S}_1,\mathrm{S}_2} \in E^1$, $e^2_{\mathrm{S}_1,\mathrm{S}_2} \in E^2)$ means that SP_1^1 and SP_1^2 can form cooperation and share the CPU resource, memory resource and bandwidth resource. If SPs on the competitive relations or on a high-security conditions, its will not take the risk of information leakage by running applications on other SPs devices. $e^2_{\mathrm{S}_1,\mathrm{S}_3} = e^1_{\mathrm{S}_1,\mathrm{S}_3} = 0$ shows that SP_2^1 and SP_3^1 can not form the resource cooperation. The lines between different regions show that remote resource sharing only connects the same SPs, e.g., line $e^1_{\mathrm{S}_3,\mathrm{S}_6} = 1$ between SP_3^1 and SP_3^2 means that SP_3^1 allows the application running on its remote server SP_3^2. The set of SPs V and w are same in each layer.

The remote cooperation has to consider many factors, such as the trip of mobile devices, the communications cost and the resource utilization of local SPs. The difference between the local and remote resource sharing is significant. If SP_1^1 rents the resource from the remote SPs, such as SP_1^2, the revenue sharing is between the same SP in different regions. In the local resource sharing, SP_1^1 has to pay for renting resource from cooperator SP_3^1. QoS is used to evaluate the service from SPs. If SP has enough resource according to application request R_j and finishes tasks within the maximum responsive time T_j, SP will allocate VM to user j. Otherwise, SP will refuse the application from user j and drop the application request R_j. In this paper, QoS Q is evaluated by the ratio of accepted request n_{acc} and total number of request n_{req}.

$$Q = \frac{n_{acc}}{n_{req}}. \tag{1}$$

B. ECONOMIC MODEL

The payment P^j_{user} from user j at time t is based on the resource it requests: the CPU resource C_j, the memory resource M_j, and the bandwidth resource B_j. Thus,

$$P^j_{user} = (cC_j + aM_j + bB_j)Pr. \tag{2}$$

Here, Pr is the unit price. c, a, b are the fixed coefficients of three kinds of resource (CPU resource, memory resource, bandwidth resource) and satisfy the normalization relationship $c + a + b = 1$. The information of traveling, e.g., the traveling destination, the road conditions, and driving behavior, can be detected and collected by the global positioning system and recommender systems [26]. Therefore, the information like, user j departure from $region1$ to $region2$, spending time Δ_j^1 in $region1$ and Δ_j^2 in $region2$, can be estimated by the GMCC system.

VM live migration provides a significant benefit for users to run applications in a mobile environment without disrupting service. The cost of migration is mainly affected by the size of memory M_j [27]. In the process of live migration, the physical memory image is pushed across network to the new destination while the source VM keeps running. Therefore, the data of memory will be transmitted several rounds before VM completes the migration. The total rounds of pre-copying iteration can be denoted by

$$n = \left\lceil \log_{\lambda} \frac{M_{thd}}{M_j} \right\rceil < \log_{\lambda} \frac{M_{thd}}{M_j} + 1, \tag{3}$$

$$\lambda = \frac{r_{tra}}{r_{mem}}. \tag{4}$$

Here, M_{thd} is the threshold value of the remaining dirty memory. r_{tra} is memory transmission rate during migration, and r_{mem} is memory dirtying rate. Let λ denote the ratio of r_{tra} to r_{mem}. According to [27], the migration latency T_m can be denoted by

$$T_m = \frac{M_j}{r_{tra}} \cdot \frac{1 - \lambda^{n+1}}{1 - \lambda} < \frac{1}{r_{tra}} \cdot \frac{M_j - \lambda^2 V_{thd}}{1 - \lambda}. \tag{5}$$

The energy for migration can be represented as

$$E_{mig} = E_b T_m r_{tra} = E_b \cdot \frac{M_j - \lambda^2 V_{thd}}{1 - \lambda}, \tag{6}$$

where E_b is unit energy for migration. The cost for migration C_{mig}^j is charged by the consumed energy and data transmission.

$$\mathrm{C}_{mig}^j = E_{mig}\kappa + C_{bdw} d_j, \tag{7}$$

where κ is a constant for energy payment. C_{bdw} is the unit cost for wired transmission. d_j is the distance of wired transmission for application R_j. The cost of wireless transmission is not included in the migration progress. The cost for running application R_j is represented as

$$C_{app}^j = (cC_j + aM_j + bB_j)\mathrm{C}_{ope} + C_{bdw} d_j, \tag{8}$$

where C_{ope} is the unit cost for running application. We have $Pr > C_{ope}$, because the payment from user need to cover the cost for running application. The cost for renting resource w from other SPs can be denoted as

$$\mathrm{C}_{rent}{}^j(w) = (cC_j + aM_j + bB_j)P_{rent}(w), \tag{9}$$

$$P_{rent}(w) = \Omega(1 + \mu w). \tag{10}$$

$P_{rent}(w)$ is the payment for renting resource to cooperator. Some SPs who have a plenty of unoccupied resource, can improve utilization by setting an appropriately leasing price. Leasing price is closely related to the resource utilization by w. Ω and μ are constants. Moreover, the payment would fulfill the constrain of $C_{ope} < P_{rent} < Pr$. We assume that the GMCC network will not charge any extra for cooperation. Thus, the revenue of the cooperator can be denoted by $R_j(w) = C_{rent}{}^j(w)$.

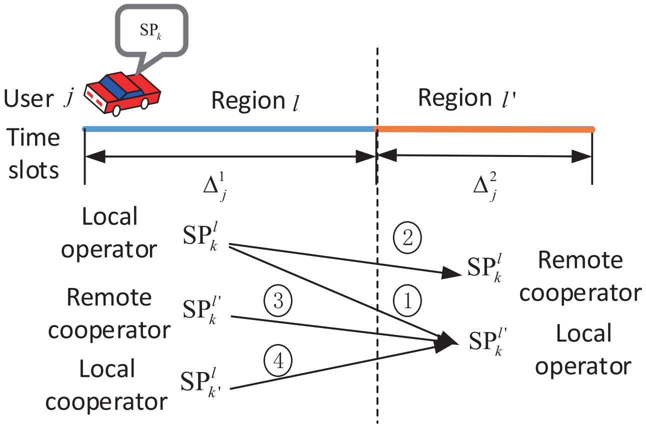


FIGURE 4. **The four cases of VM allocation.**

C. SEVERAL CASES REVENUE MODEL

The local cooperation and remote cooperation will have many possibilities if involves multiple tasks situation. In this paper, we assume that one application from user will be allocated one VM. Thus, a resource cooperation group for one application only involves two SPs to finish the task. According to Fig. 4, VM allocation has three choices: on the local operator, on the local cooperator, or on the remote cooperator. Each choice exists many possible cooperation groups. We classify this possibility into four different cases, and its revenues are $\gamma_j \in \{\gamma_j^1, \gamma_j^2, \gamma_j^3, \gamma_j^4\}$ independently. γ_j is the revenue of the choosing case in this four cases. SP will make decision right after receiving a request. The decision will take consideration of vehicle j traveling trace and two time slots at most. The time slot is the duration that spending on one region in a trip. If vehicle will go through two region and drive to the third region, it can make a new decision in the second region. Then, combining with current state of network, we aim to find the case with the highest revenue as the decision.

User j sends the request $R_j = (k, l, C_j, M_j, B_j, T_j)$ to local operator SP$_k^l$ at the beginning of the trip in region l. Therefore, SP$_k^l$ will allocate VM to user j if resource is enough. Otherwise, SP$_k^l$ can rent resource from the remote cooperator SP$_k^{l'}$ or from the local cooperator SP$_{k'}^l$ by employing the cooperator's computing resource to allocate VM. When it arrives region l' after Δ_j^1, SP$_k^{l'}$ becomes the local operator, and SP$_k^l$ is the remote cooperator during Δ_j^2. The traveling time duration Δ_j is the sum of time slots Δ_j^1 and Δ_j^2.

$$\Delta_j = \Delta_j^1 + \Delta_j^2. \tag{11}$$

Case 1: SP_k^l firstly allocates VM to user j. When user j comes to region l', this VM will be migrated to region l' and running on $SP_k^{l'}$ during Δ_j^2. Therefore, the revenue of SP_1^1 can be denoted by

$$\gamma_j^1 = P_{user}^j \Delta_j^1 - C_{app}^j \Delta_j^1 - C_{mig}^j T_m. \tag{12}$$

Case 2: Running application j in the local operator SP_k^l for the traveling duration Δ_j. The revenue of SP_k^l is γ_j^2.

$$\gamma_j^2 = P_{user}^j \Delta_j - C_{app}^1 \Delta_j - C_{bdw} d_j \Delta_j^2. \tag{13}$$

Case 1 and *Case 2* are denoted on the directed line in Fig. 4, are both allocating resource on the local operator SP_k^l in Δ_j^1. The difference is that *Case* 1 will migrate the VM to the local operator $\mathrm{SP}_k^{l'}$ when the location changes, while *Case* 2 will not.

Case 3: When SP_k^l is on a high level of utility, it will rent resource from $\mathrm{SP}_{k'}^l$ in the same data center to run application j. In local cooperation, the revenue of SP_k^l, and $\mathrm{SP}_{k'}^l$ can be denoted as γ_j^3 and $\gamma_j^{3'}$ respectively.

$$\gamma_j^3 = P_{user}^j \Delta_j^1 - C_{app}^j \Delta_j^2 - P_{rent}^j(w) \Delta_j^1 - C_{mig}^j T_m, \tag{14}$$

$$\gamma_j^{3'} = \mathrm{C}_{rent}^j(w) \Delta_j^1 - C_{app}^j \Delta_j^1. \tag{15}$$

Case 4: In the remote cooperation, we suppose that the next station of user j is region l'. Therefore, running application j on $\mathrm{SP}_k^{l'}$ will reduce the cost for migration but increase the cost for data transmission. The revenue of SP_k^l and $\mathrm{SP}_k^{l'}$ are γ_j^4 and $\gamma_j^{4'}$ respectively.

$$\gamma_j^4 = P_{user}^{j'} \Delta_j - C_{app}^j \Delta_j^2 - P_{rent}^j(w) \Delta_j^1 - C_{bdw} d_j \Delta_j^1, \tag{16}$$

$$\gamma_j^{4'} = P_{rent}^j(w) \Delta_j^1 - C_{app}^j \Delta_j^1. \tag{17}$$

Moreover, the line from $\mathrm{SP}_k^{l'}$ to SP_k^l or from $\mathrm{SP}_{k'}^l$ to SP_k^l are not exist. Because SP_k^l will only form the cooperation in SP_k^l busy hours, there is no need for VM to migrate back to the original cooperator when user j comes to the second region. If the remote cooperator is a different SP, e.g., $SP_{k'}^{l'}$, SP_k^l has to pay for the migration cost. The revenue of SP_k^l is $\gamma_j^{4''}$, given by

$$\begin{aligned} \gamma_j^{4''} = P_{user}^j \Delta_j^1 - C_{app}^j \Delta_j^2 - P_{rent}^j(w) \Delta_j^1 \\ - C_{bdw} d_j \Delta_j^1 - C_{mig}^j T_m. \end{aligned} \tag{18}$$

By comparing (16) with (18), or (14) with (18), $\mathrm{SP}_{k'}^{l'}$ is not included in a possible choice since the cooperation will be costly for data transmission and VM migration.

IV. COALITION GAME THEORETICAL APPROACH

A. UTILITY FUNCTION

SP intends to improve the revenue through the cooperation by increasing the utility of resource. However, resource over-utilization may reduce QoS. In this paper, the utility function of SPs involves two parts: revenue and penalty. Over-utilization means that the utilization of resource is at a very high level, and has few left for newcomers. In this case, SP will refuse applications from newcomers until it has enough resource. The penalty is to avoid resource over-utilization. Furthermore, SP will both lose revenues and decrease QoS. The penalty function ϑ_k^l of SP_k^l is an upward curve.

$$\vartheta_k^l = \upsilon(\rho_k^l)^o, \tag{19}$$

$$\rho_k{}^l = \frac{1}{3}\left(\frac{C_k{}^l}{\mathrm{maxC}_k{}^l} + \frac{M_k{}^l}{\max \mathrm{M}_k^l} + \frac{B_k{}^l}{\mathrm{maxB}_k{}^l}\right), \tag{20}$$

where ρ_k^l is the utilization of SP_k^l. υ and o are the indexes of penalty, and $o \geq 2$ [28], [29]. Here, we employ $o = 3$. The weight of edge w_{ij} denotes the average difference of utilization.

$$w_{ij} = \begin{cases} -\dfrac{\rho_i - \rho_j}{\overline{\rho}}, & \mathit{if}\ w_{ij} \geq 0, \\ 0, & \mathit{if}\ w_{ij} < 0. \end{cases} \tag{21}$$

$$\overline{\rho} = \frac{1}{m}\sum_{i=1}^{m} \rho_i, \tag{22}$$

where $\overline{\rho}^l$ is average coefficient of utilization. If w_{ij} is the weight of edge between different regions, m is the number of regions where S_i has resource. If w_{ij} is the weight of edge in the same region, m is the number of SPs in region l. If weight is less than zero, resource sharing will violate the market rule and break the balance of the GMCC network. Because, SP with small utilization will lease resource to SP with larger utilization.

The utility function of SP_k^l (U_k^l) is evaluated by the revenues from all applications. We firstly calculate the utility of application j, $U_{k,j}^l$, by making the difference between the utility before and after SP_k^l accepting application j.

$$U_{k,j}^l = \gamma_j - \upsilon[(\rho_k^{l'})^o - (\rho_k^l)^o], \tag{23}$$

$$\begin{aligned} U_k^l &= \sum_{j=1}^{m} U_{k,j}^l \\ &= \sum_{j=1}^{m} \gamma_j - \upsilon(\rho_k^l)^o. \end{aligned} \tag{24}$$

Here, $\rho_k^{l'}$ is the utilization of SP_k^l before accepting application j, and ρ_k^l is the utilization of SP_k^l after accepting application j. All SPs are trying to maximize utility in order to maintain high QoS while improving the revenue. Therefore, in cloud market, the objective function of SP_k^l can be presented as

$$\begin{aligned} \mathit{max.}\ U_k^l &= \sum_{j=1}^{m} \gamma_j - \upsilon(\rho_k^l)^o \\ \mathit{s.t.}\ &\sum_{j=1}^{m} C_j \leq \mathrm{maxC}_k^l, \\ &\sum_{j=1}^{m} M_j \leq \mathrm{maxM}_k^l, \\ &\sum_{j=1}^{m} B_j \leq \mathrm{maxB}_k^l. \end{aligned} \tag{25}$$

B. COALITION GAME

To increase the available resource for mobile applications and maximize the revenue, SPs can share their resource by cooperation in the coalition game. Such as SP in a different region is an independent player, thus, the GMCC environment will be partitioned by $\mathcal{G} = \{G_1, G_2, \cdots, G_l\}$. Before the coalition is created, we have $\mathcal{G} = \mathcal{SP} = \{\mathrm{S}_1, \ldots, \mathrm{S}_N\}$, which means that all SPs work alone and have no cooperation at the beginning. When application $R_j = (k, l, C_j, M_j, B_j, T_j)$ comes, user j will send request to local SP_k^l and wait for answer. From the section of case study, there are three possibilities to run application. *Case* 1 and *Case* 2 are both the way that running application without coalition (the case of non-cooperation) since the resource is allocated by local operator SP_k^l. we calculate the revenues in (12) and (13). Then, the added utility γ_j^* of SP_k^l is the lager one in this two cases. The utility of SP_k^l after accepting application j is U_k^l.

$$\gamma_j^* = \begin{cases} \gamma_j^1 & if,\ \Delta_j^2 < \Delta_j^1, \\ \gamma_j^2 & if,\ \Delta_j^2 > \Delta_j^1, \mathrm{C}_{mig}^j T_m > \mathrm{C}_{bdw} d_j \Delta_j^2. \end{cases} \tag{26}$$

$$U_k^l = \sum_{i=1, i\neq j}^{m} \gamma_i + \gamma_j^* - \upsilon(\rho_k^l)^o, \tag{27}$$

here, γ_i is the revenue from other applications running on SP_k^l. The utility of SP_k^l before application j coming can be represented as

$$U_k^{l'} = \sum_{i=1, i\neq j}^{m} \gamma_i - \upsilon(\rho_k^{l'})^o, \tag{28}$$

$$\rho_k^{l'} = \rho_k^l - \delta, \tag{29}$$

where $\rho_k^{l'}$ and ρ_k^l represent the resource utilization before and after the coalition respectively. δ is the added workload from application R_j. If $U_k^{l'} > U_k^l$, SP_k^l can improve the utility through coalition.

$$\gamma_j^* - \upsilon(\rho_k^l)^o < -\upsilon(\rho_k^{l'})^o, \tag{30}$$

$$\gamma_j^* < \upsilon[(\rho_k^l)^o - (\rho_k^{l'})^o]. \tag{31}$$

Therefore, (31) is the condition that SP_k^l will agree to merge into coalition with SPs in remote cloud.

We assume that the remote cooperator in *Case* 4 is SP_k^r. According to (15), the utility of SP_k^r after accepting application R_j is U_k^r.

$$U_k^r = \sum_{i=1, i\neq j}^{m} \gamma_i + \gamma_j^4 - \upsilon(\rho_k^r)^o, \tag{32}$$

where, ρ_k^r is the resource utilization of SP_k^r after joining coalition. γ_i is the revenue from other applications which running on SP_k^r. Therefore, the utility of $SP_k^{r'}$ before coalition is

$$U_k^{r'} = \sum_{i=1, i\neq j}^{m} \gamma_i - \upsilon(\rho_k^{r'})^o, \tag{33}$$

$$\rho_k^{r'} = \rho_k^r - \delta. \tag{34}$$

For SP_k^r, the condition of joining the coalition is based on it's utility improvement. Therefore, the condition is $U_k^{r'} < U_k^r$, and given by

$$\gamma_j^4 > \upsilon[(\rho_k^r)^o - (\rho_k^{r'})^o]. \tag{35}$$

In *Case* 3, SP_k^l will rent resource from a local cooperator to run applications. The utility of SP_k^l before local cooperation U_k^l is given in (27). The utility after local coalition $U_k^{l'}$ is obtained by (14).

$$U_k^l = \sum_{i=1, i\neq j}^{m} \gamma_i + \gamma_j^* - \upsilon(\rho_k^l)^o, \tag{36}$$

$$U_k^{l'} = \sum_{i=1, i\neq j}^{m} \gamma_i + \gamma_j^3 - \upsilon(\rho_k^{l'})^o. \tag{37}$$

If SP_k^l can improve the utility through local cooperation and $U_k^{l'} > U_k^l$, SP_k^l will send cooperation request to a local SP.

$$\gamma_j^* - \upsilon(\rho_k^l)^o < \gamma_j^3 - \upsilon(\rho_k^{l'})^o, \tag{38}$$

$$\gamma_j^* - \gamma_j^3 < \upsilon[(\rho_k^l)^o - (\rho_k^{l'})^o]. \tag{39}$$

If the local cooperator is SP_q^l, the utility before and after local coalition are denoted as U_q^l and $U_q^{l'}$ respectively. According to (15), we have

$$U_q^l = \sum_{i=1, i\neq j}^{m} \gamma_i - \upsilon(\rho_q^l)^o, \tag{40}$$

$$U_q^{l'} = \sum_{i=1, i\neq j}^{m} \gamma_i + \gamma_j^{3'} - \upsilon(\rho_q^{l'})^o, \tag{41}$$

$$\rho_q^{l'} = \rho_q^l - \delta. \tag{42}$$

The condition that SP_q^l is willing to cooperate with SP_k^l is $U_q^{l'} > U_q^l$. Thus, we have

$$\upsilon(\rho_q^l)^o < \gamma_j^{3'} - \upsilon(\rho_q^{l'})^o, \tag{43}$$

$$\gamma_j^{3'} > \upsilon[(\rho_q^{l'})^o - (\rho_q^l)^o]. \tag{44}$$

Therefore, (39) and (44) are the conditions of local coalition.

V. PARETO OPTIMALITY AND STABILITY

A. PARETO OPTIMALITY

In a coalition game, SPs prefer to run applications with high utility. There are several possible operations: i) an individual SP would like to join a coalition if its utility can be improved in the coalition; ii) If SP k in coalition A finds out that join coalition B will obtain more utility than before, SP k would like to leave A and join B. iii) If SP k in coalition A find out that leaving coalition will achieve more utility, SP k will leave the coalition and work alone. We employ an effective mechanism, namely merge-and-split, to derive the coalition game stable formation. In the merge-and-split mechanism, *Pareto optimality* is used as the criterion of the operation of the players.

Definition 1: Consider two sets of coalitions $\mathcal{G}_1 = \{G_1^1, G_2^1, \cdots, G_l^1\}$ and $\mathcal{G}_2 = \{G_1^2, G_2^2, \cdots, G_m^2\}$, which are two different partitions of a same set $G \subset \mathcal{S}$. For a player SP_i, let $u_k(SP_i)$ denote the utility of SP_i in the coalition $\mathcal{G}_k$ ($k = 1, 2$). The coalition $\mathcal{G}_1$ is preferred over $\mathcal{G}_2$ by *Pareto order*, denoted by $\mathcal{G}_1 \triangleright \mathcal{G}_2$, if and only if

$$u_1(SP_i) \geq u_2(SP_i), \quad \forall SP_i \in \mathcal{S}', \tag{45}$$

with at least an inequality for a player SP_k.

Following the criterion of Pareto order, the players will be reorganized so that the coalitions are reformed for improving the utilities. This procedure usually takes many rounds. In each round, all the coalitions would be involved so that their utilities can increase. This shows that the reorganization of coalitions is naturally a global operation. In order to facilitate the procedure, we decouple the global operation by a number of distributed operations using the following two fundamental rules.

- *Merge:* For any set of coalitions $\{G_1, \cdots, G_l\}$, if $\{\bigcup_{j=1}^{l} G_j\} \triangleright \{G_1, \cdots, G_l\}$, then merge $\{G_1, \cdots, G_l\}$ to $\{\bigcup_{j=1}^{l} G_j\}$, denoted by $\{G_1, \cdots, G_l\} \rightarrow \{\bigcup_{j=1}^{l} G_j\}$.
- *Split:* For any coalitions $U_{j=1}^{l} G_j$, if $\{G_1, \cdots, G_l\} \triangleright \{\bigcup_{j=1}^{l} G_j\}$, then split $\{\bigcup_{j=1}^{l} G_j\}$ into $\{G_1, \cdots, G_k\}$, denoted by $\{\bigcup_{j=1}^{l} G_j\} \rightarrow \{G_1, \cdots, G_k\}$.

By using these rules of merge-and-split, SPs are allowed to negotiate and constitute the coalitions. The globally Pareto-optimal collection of coalitions can be consolidated gradually.

B. STRATEGY FOR COALITION FORMATION

For any application, the decision of merge and split is a distributed operation. It will not be affected by time and place unless the decision is canceled by SPs. For decision making, a control system will evaluate all the potential cases and choose the best one as the result. The result should follow the maximum latency constraint. In order to narrow the searching space of a coalition game and improve computation efficiency, we propose the two main stages to find out the solution for user j.

1) STAGE 1

After receiving application requests from user j, the GMCC system will refresh the network parameters, such as the weight of each edge, utilization coefficient. A base station will calculate the revenue in four cases and get the utility of three states: non-cooperation, remote coalition, local coalition.

2) STAGE 2

The entire procedure of coalition formation has four main steps.

- *Step 1:* Decide whether SP_k^l needs to rent resource to run application R_j by (31). If (31) is false, the application R_j will be running on SP_k^l. It is the state of non-cooperation. Otherwise, go to step 2.
- *Step 2:* SP_k^l will send the cooperation request to SP in the next station of user j, such as SP_k^r. SP_k^r will make the cooperation decision according to (35). If condition is fulfilled and the latency constraint is satisfied, the remote coalition between SP_k^l and SP_k^r is established. Otherwise, SP_k^l will remain in it state. Go to step 3.
- *Step 3:* Find the connected SPs in the same region with maximized weight of edge, e.g., SP_q^l. If (39) and (44) are simultaneously satisfied and the latency constraint is fulfilled, go to step 4. Otherwise, go to step 5.
- *Step 4:* If SP_k^l has already in a coalition, it will split from the former coalition. Then, SP_k^l will form the local coalition with SP_q^l.
- *Step 5:* The GMCC system will refuse application R_j. Then, user j will resend the application after a time interval.

C. SCHEME FOR RESOURCE ALLOCATION AND SCHEDULING

In this section, we mainly discuss the stability and convergence of the proposed strategy of coalition formation. We use Pareto-optimal $\mathbb{D}_c$-stable partition to demonstrate the stability of coalition according to [30].

Definition 2: A collection of coalitions $S := \{S_1, \ldots, S_k\}$ is said to be $\mathbb{D}_c$-stable if it satisfies two conditions:

(a) $i \in \{1, \ldots, k\}$ and for each partition $\{P_1, \ldots, P_l\}$ of the coalition G_i: $u(G_i) \geq \sum_{j=1}^{l} u(SP_j)$.

(b) $S \subseteq \{1, \ldots, k\}$: $\sum_{j \in S} u(G_i) \geq u(\cup_{i \in T} G_i)$.

Theorem 1: The final coalition formation under the proposed strategy can be $\mathbb{D}_c$-stable [31].

Proof: We first consider condition (a). In the final coalition set $\mathcal{G} = \{G_1^1, G_2^1, \cdots, G_l^1\}$, we assume that SP_i is included in the coalition G_k, $\mathrm{SP}_i \in G_k$. However, if SP_i can obtain a higher utility by working alone, or joining other coalitions G_l, condition (a) will be violated. Therefore, according to merge-and-split rules, SP_i will leave from the current coalition G_k. Thus, coalition G_k will not exist. The coalition formations in $\mathcal{G}$ are unstable and can not be the final coalition set. Therefore, condition (a) must be satisfied for any stable coalition generated under the proposed strategy.

For condition (b), we consider the situation in the same final coalition set $\mathcal{G} = \{G_1^1, G_2^1, \cdots, G_l^1\}$. If coalition G_k can obtain a higher utility when it combines with other SPs and come into a larger coalition $G_k'(G_k \subseteq G_k')$, The G_k will merge into G_k', such that $u(G_k) < u(G_k')$. $\mathcal{G}$ can not be the final coalition set for the same reason. Thus, for stable formation of the final coalition set, condition (b) needs to be satisfied.

In this scenario, we only consider resource cooperation of one VM at one time. Therefore, one coalition only has two members and it can not further merge into a larger coalition. Different VMs are independent at the coalition formation progress. In summary, conditions (a) and (b) will both involve into the final coalition set in order to ensure the stability of the final result.

Theorem 2: In *Theorem*1, the final coalition formation is $\mathbb{D}_c$-stable. Therefore, if this partition exists and is stable on the final coalition set, the Pareto optimal solution will be the only one stable solution.

Proof: We assume that there are exist two different optimal solutions simultaneously. In the first solution, SP$_i$ cooperates with SP$_j$ into a coalition G_k, SP$_i$, SP$_j$ $\subseteq G_k$, and SP$_l$ in the coalition of working alone. According to the rule of merge-and-split, SP$_i$ can improve utility through cooperating with SP$_j$ more than with SP$_l$. In the second solution, SP$_i$ merges with SP$_l$ and SP$_j$ works alone. According to the rule of merge-and-split, SP$_i$ will finally work with someone whom can improve its utility at most. Therefore, this two solution are violated. In other words, the Pareto optimal one is only stable situation.

Theorem 3: In the remote coalition, we only need to consider SP in the next station.

Proof: If SPs in the next station is available for cooperation, it will save the cost for migration than other remote SPs in elsewhere, according to (16) and (18). Otherwise, the local coalition can obtain more revenue than the remote by comparing (14) with (18). Therefore, in the step 2 of second stage, we only consider SP in the next station as the remote cooperator.

Theorem 4: In the local coalition, if SP$_q^l$ (the vertex with largest weight of edge) turns down the cooperation request from SP$_k^l$, other connected SPs will refuse it too. In other words, because if SP$_q^l$ can not improve the utility through joining the local coalition, SPs with smaller weights of edge will obtain an negative utility from coalition. Therefore, we narrow the searching range of cooperator by only asking the SP with the largest weight, such as SP$_q^l$.

Proof: We assume that SP$_q^l$ and SP$_s^l$ are the potential cooperators of SP$_k^l$. w and w' are the weigh of edges from SP$_k^l$ to SP$_q^l$ and from SP$_k^l$ to SP$_s^l$ respectively, $w > w'$. According to (18), we have $\rho_q^l < \rho_s^l$.

$$\rho_q^{l'} = \rho_q^l + \delta, \tag{46}$$

$$\rho_s^{l'} = \rho_s^l + \delta, \tag{47}$$

where δ is the added workload of running application R_j. The condition of SP$_q^l$ refusing cooperation request can be calculated from (44).

$$\gamma_j^{3'} < \upsilon[(\rho_q^{l'})^o - (\rho_q^l)^o], \tag{48}$$

$$\begin{aligned}\upsilon[(\rho_q^{l'})^o - (\rho_q^l)^o] &= (\rho_q^{l'} - \rho_q^l)\sum_{i=1}^{o}[(\rho_q^{l'})^{o-i}(\rho_q^l)^{i-1}]\\ &= \delta\sum_{i=1}^{o}[(\rho_q^{l'})^{o-i}(\rho_q^l)^{i-1}] \qquad (49)\\ &< \upsilon[(\rho_s^{l'})^o - (\rho_s^l)^o].\end{aligned}$$

$$\Omega(1+\mu w') < \Omega(1+\mu w). \tag{50}$$

Here, $\gamma_j^{3'}$ is the revenue of SP$_q^l$ which can be obtained from (15). Therefore, the revenue of SP$_s^l$ is $\gamma_j^{3''}$ and which can be obtained by

$$\gamma_j^{3''} < \upsilon[(\rho_s^{l'})^o - (\rho_s^l)^o]. \tag{51}$$

This shows that SP$_s^l$ will refuse the cooperation request from SP$_k^l$ and SP$_k^l$ dose not need to send any request to SPs with weight of edge lower than SP$_q^l$.

VI. NUMERICAL RESULTS

In this section, we evaluate the performance of our proposed scheme through simulations. The parameters setting is described, and the numerical results are as follows.

TABLE 1. Load data in 13 regions.

Application index	CPU resource	Memory resource	Bandwidth resource
1	3	3	2
2	2	2	2
3	2	1	1

We consider 4 SPs and 4 data centers in the GMCC network model. The capability of each data centers is the same, and shared by each SP equally. We suppose the applications from users is random from three sets of applications (Table 1). The resource in this table are transformed into the units of each kind of resource (e.g., one unit of CPU means 4000 MIPS, one unit of memory means 4000 MB). We adopt this approach to make calculation more simple and clear according to [32] and [33]. The capability of cloud resource is included the CPU resource (maxC$_k^l$ $= 4800$), the memory resource (maxM$_k^l$ $= 4040$) and the bandwidth resource (maxB$_k^l$ $= 3560$). The ratio of three kinds of resource are evaluated by the cost, $c = 0.5$, $a = 0.4$, $b = 0.1$. The unit price for revenue is $Pr = 6$ and for running application is $C_{ope} = 3$. Constant for energy payment is $\kappa = 0.01$. The unit cost for wired transmission is $C_{bdw} = 0.08$. We set constants $\Omega = 3$ and $\mu = 0.5$ respectively.

We focus on the mobile services in vehicles which applications will be used through out the trip. Therefore, the number and frequency of applications can be obtained from the statistics of vehicle migration. We adopt the real traces of vehicles from CRAWDAD [34]. This trace was recorded from 536 urban taxi cabs in San Francisco in a month. We use the duration of taking passengers as the time window of using mobile cloud applications. Fig. 5 shows the number of trips during a day. The time from 5 a.m. to 10 a.m. is the most busy time during one day. Our simulation is based on the statistics during this period. Furthermore, the parameters of live migration are referred to [27]. Therefore, we have memory transmission rate $r_{tra} = 350\ MB/s$ and memory dirtying rate $r_{mem} = 450\ MB/s$. The threshold value of the remaining dirty memory M_{thd} is 0.9;

1) Coalition Strategy Performance: Fig. 6 shows the average utilization of each SP. In Fig. 6 (a), the blue bars represent the average utility of all SPs which work without coalition to share resource in the whole observation duration. SPs can only occupy its own resource to run applications. The red bars denote the average utility of SPs who agree resource

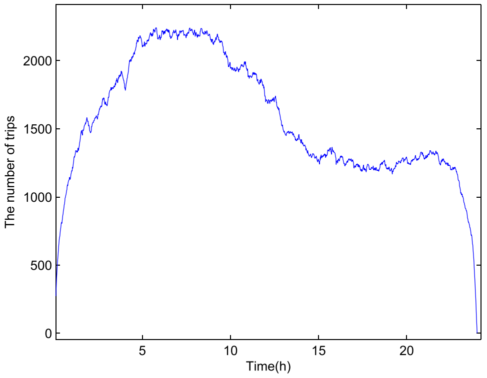
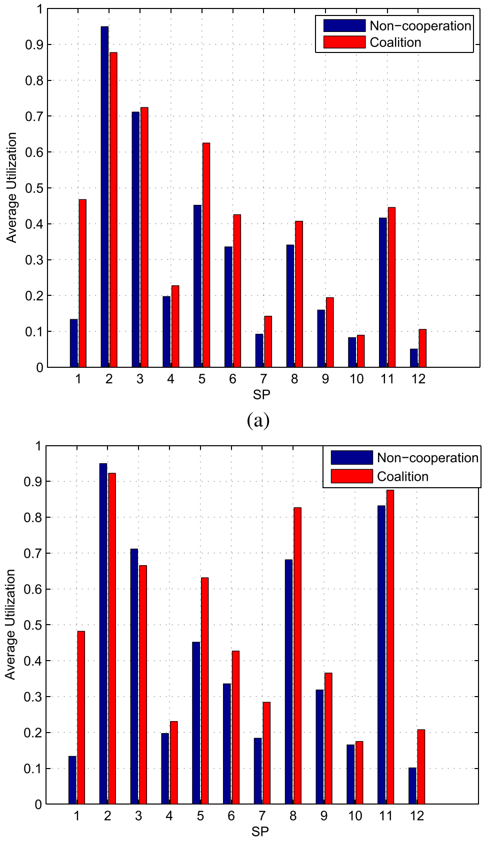


FIGURE 5. **The trip statistics during one day.**



(a)

(b)

FIGURE 6. **The average utility of each SP. (a) The average utility of SPs with same capability. (b) The average utility of SPs with different capability.**

sharing with admitted SPs and can form the coalition to run applications. We may notice that most SPs can improve their utilization through cooperation by renting or leasing resource. However, the utilization of SP_2 is slightly decreased. The reason is that cooperation can remove the burden from SP with a high level of utilization to the SPs with lower utilization. As the result, SP_2 can reserve a part of resource to new applications and achieve high QoS.

In Fig. 6 (b), we conduct the simulation on the case of SP has different capability. All the simulation condition are not changed except that the capability of SPs from 7 to 12 are reduced by half. The simulation shows us that the proposed approach will still work with satisfying performance in the case of heterogeneous data centers.

The advantage of employing network structure is that it can direct the coalition formation progress and make the unbalanced network toward an balanced one. Heterogeneity index is proposed to evaluate coalition strategy performance. The main result of coalition game is to balance the work load of SPs. It makes the SPs with low resource utilization that can leasing resource to other SPs which need more resource to running applications. As the result, the utilization of SPs will come to balance. We employ heterogeneity index to evaluate the balance of SP resource utilization of local resource sharing and remote resource sharing. Heterogeneity index of SP is based on Lorenz curve and Gini coefficient which is used to describe income inequality in microeconomics [35]. Here, we employ the heterogeneity index H to evaluate the unbalance degree of utility among local SPs and remote SPs respectively.

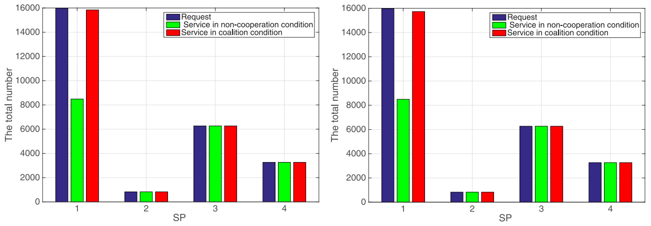
$$H = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} |\rho_i - \rho_j|}{2N^2\bar{\rho}}, \tag{52}$$

$$\bar{\rho} = \frac{1}{N}\sum_{i=1}^{N}\rho_i. \tag{53}$$

Here, $\bar{\rho}$ is the average value of utilization from the set of SPs with the same properties, such as, in the same region or belonging to the same SP. N is the number of SPs. For example, the heterogeneity index of region 1 can be calculated by the utilizations from SP_2^1, SP_1^1 and SP_3^1. Thus, $N = 3$. Whereas, if we want to evaluate the heterogeneity index of SP_1 in all regions, SPs includes SP_1^1, SP_1^2, SP_1^3 and SP_1^4, $N = 4$. We make a comparison between the non-cooperation case and coalition case. The average heterogeneity index of four regions decreases from 0.100 to 0.079. The average heterogeneity index of 4 distributed SPs changes from 0.333 to 0.093. The results reveal that the coalition in the GMCC network can both improve the utilization and balance the workload in network.

2) Profit From Coalition: The ratio of accepted applications is an important QoS parameter. From Fig.7 (a), SP_1 only has 60.90% applications being accepted and 39.10% applications being refused. Thus, QoS is 60.90% which is evaluated by (1). If SP_1 working in the coalition case, QoS will be increased to 99.30%, which is immensely improved. However, QoS of other three SPs remain unchanged since they have sufficient resource to provide service to users. Fig.7 (b), shows the profit comparison when SPs has different capability. The QoS of SP_1 in coalition case is 98.47%. The slight decrease of QoS is due to the capability of SPs from S_7 to S_{12} cut in half. It lead to the total number of coalition decreased.

Fig. 8 shows the profit of PSs in the penalty index $\upsilon = 10000$. The blue bars represent the profit of each



(a) (b)

FIGURE 7. The total number of accepted application (penalty index $\upsilon = 220000$). (a) The accepted application for SPs with same capability. (b) The accepted application for SPs with different capability.

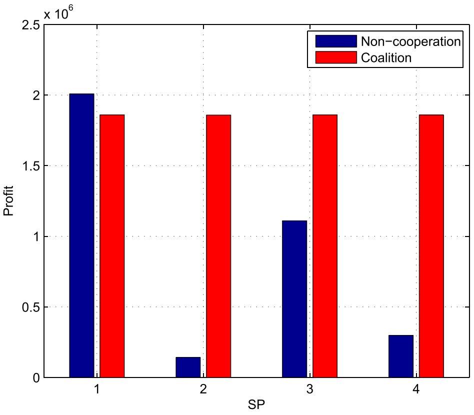
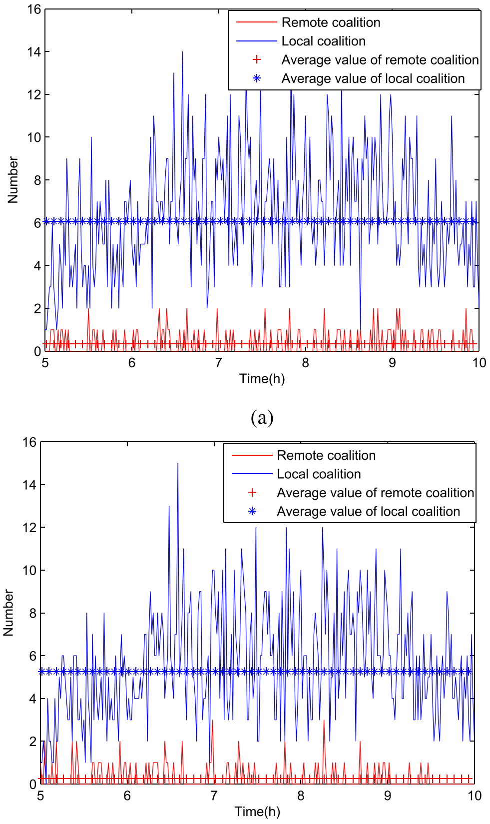


FIGURE 8. The profit of each PS (penalty index $\upsilon = 10000$).



(a)

(b)

FIGURE 9. The performance of coalition. (a) Coalition performance in the case of SPs with same capability. (b) Coalition performance in in the case of SPs with different capability.

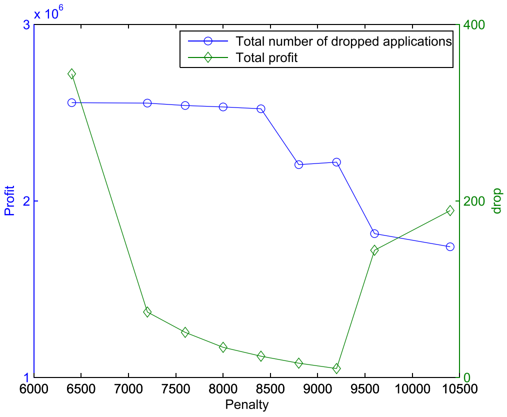


FIGURE 10. The total profit and total number of dropped applications on different penalty index υ.

SP which works alone. The red bars show the profit in the condition of coalition. Except SP_1, the profit of other SPs increases significantly. The profit of SP_1 slightly decreasing is due to SP_1 pays for resource renting to increase QoS. Therefore, SP_2, SP_3, andSP_4 obtain more opportunities for using resource and getting payment from cooperators. The total profit in the coalition situation increases 114% by comparing with non-cooperation scheme. It can be concluded that the coalition approach can increase the total profit of SPs.

3) Performance of Remote Coalition and Local Coalition: The number of remote coalition and local coalition varies every time. In Fig. 9 (a), the average number of local coalition is more than remote coalition. It is because the remote coalition is decided by the next station of vehicles, which may be costly for data transmission and limited by latency constraint. Fig. 9 (b) shows us that the average number of both coalition are decreased. Because the capability of SP has great influence on coalition.

4) Impact of Penalty: The index of penalty υ has a great impact on coalition formation. From (16) and (22), a large penalty index will make a system to be very sensitive to the resource utilization. In this case, an SP will prefer to join the coalition for resource sharing when it on the level of high resource utilization. However, in order to allocate VM to more applications, SP should pay for the renting resource from other SPs and obtain less revenue from users. On the contrary, a small penalty index will decrease probability of coalition formation. Fig. 10 shows the impact of penalty index υ on the GMCC network system. The blue line with circle is the total profit of all SPs with different penalty index υ. The total profit will decrease as υ increasing. As the number of cooperation increased, the cost for data transmission and migration will both increase. The green line with square represents the total number of dropped applications in the GMCC network. The number of dropped applications decreases as υ increases. However, at the end of green line, the number of dropped applications increases as υ continues to scale down. It is because SP with a large penalty will refuse to accept applications, and operate on the low level of resource utilization. Thus, revenue will continue to decrease.

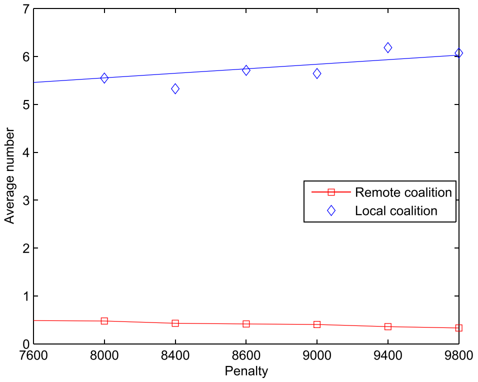


FIGURE 11. The average number of remote coalition and local coalition in different penalty index.

In order to further explain this phenomenon, we choose nodes from $\upsilon = 7600$ to $\upsilon = 9800$, as shown in Fig.11. As the penalty index increases, the average number of local coalition will increase, but the average number of remote coalition will decrease. Therefore, the expanding number of migration will reduce the profit of SPs.

VII. CONCLUSION

In this paper, we introduced a coalition game based model for resource management and sharing among the GMCC network. As the computing modules of mobile Internet applications can be offloaded to the powerful server in the cloud, the cloud service providers conform with a virtual resource network. It provides CPU, memory and bandwidth resource in order to support the mobile Internet applications. The coalition game in this GMCC network promotes resource cooperation either among the local SPs or remote SPs. It is a win-win strategy for SPs which can both improve the revenue by increasing the utility of resource appropriately, and largely enhance QoS by few cost for renting resource. Further, we have introduced and applied an improved coalition approach for which stability and uniqueness of result have been proved. Simulation results indicate that our scheme enhances resource utilization of SPs and improves QoS.

REFERENCES

[1] T. Xing, D. Huang, S. Ata, and D. Medhi, "MobiCloud: A geo-distributed mobile cloud computing platform," in *Proc. 8th Int. Conf. Netw. Service Manage.*, Oct. 2012, pp. 164–168.

[2] H. Li, D. Liu, Y. Dai, and T. H. Luan, "Engineering searchable encryption of mobile cloud networks: When QoE meets QoP," *IEEE Wireless Commun.*, vol. 22, no. 4, pp. 74–80, Aug. 2015.

[3] Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," *J. Internet Technol.*, vol. 16, no. 2, pp. 317–323, 2015.

[4] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2685–2700, Dec. 2013.

[5] J. Li, D. Li, J. Zheng, and Y. Quan, "Location-aware multi-user resource allocation in distributed clouds," in *Advanced Computer Architecture*. New York, NY, USA: Springer-Verlag, 2014, pp. 152–162.

[6] M. Quwaider and Y. Jararweh, "Cloudlet-based efficient data collection in wireless body area networks," *Simul. Model. Pract. Theory*, vol. 50, pp. 57–71, Jan. 2015.

[7] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[8] H. Li, Y. Yang, T. Luan, X. Liang, L. Zhou, and X. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Trans. Dependable Secure Comput.*, vol. PP, no. 99, p. 1, Feb. 2015.

[9] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency Comput., Pract. Exper.*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012.

[10] M. A. Vouk, "Cloud computing—Issues, research and implementations," *J. Comput. Inf. Technol.*, vol. 16, no. 4, pp. 235–246, 2008.

[11] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: Towards a cloud definition," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50–55, 2008.

[12] M. Armbrust *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[13] S. S. Qureshi, T. Ahmad, K. Rafique, and Shuja-ul-islam, "Mobile cloud computing as future for mobile applications—Implementation methods and challenging issues," in *Proc. IEEE Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Sep. 2011, pp. 467–471.

[14] P. T. Endo *et al.*, "Resource allocation for distributed cloud: Concepts and research challenges," *IEEE Netw.*, vol. 25, no. 4, pp. 42–46, Jul./Aug. 2011.

[15] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generat. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, May 2012.

[16] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Joint allocation of computation and communication resources in multiuser mobile cloud computing," in *Proc. IEEE 14th Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2013, pp. 26–30.

[17] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Comput.*, vol. 12, no. 1, pp. 1–15, Mar. 2009.

[18] G. Khanna, K. Beaty, G. Kar, and A. Kochut, "Application performance management in virtualized server environments," in *Proc. 10th IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2006, pp. 373–381.

[19] W. Zhang, Y. Wen, and D. O. Wu, "Energy-efficient scheduling policy for collaborative execution in mobile cloud computing," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 190–194.

[20] R. Liu *et al.*, "A novel server selection approach for mobile cloud streaming service," *Simul. Model. Pract. Theory*, vol. 50, pp. 72–82, Jan. 2015.

[21] Z.-H. Zhang, J. Xue-Feng, J.-J. Li, and W. Jiang, "An identity-based authentication scheme in cloud computing," in *Proc. Int. Conf. Ind. Control Electron. Eng. (ICICEE)*, Aug. 2012, pp. 984–986.

[22] P. Guo, J. Wang, X. H. Geng, C. S. Kim, and J.-U. Kim, "A variable threshold-value authentication architecture for wireless mesh networks," *J. Internet Technol.*, vol. 15, no. 6, pp. 929–936, 2014.

[23] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. PP, no. 99, p. 1, Feb. 2015.

[24] Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Trans. Commun.*, vol. E98-B, no. 1, pp. 190–200, 2015.

[25] S. Xie and Y. Wang, "Construction of tree network with limited delivery latency in homogeneous wireless sensor networks," *Wireless Pers. Commun.*, vol. 78, no. 1, pp. 231–246, 2014.

[26] T. Ma *et al.*, "Social network and tag sources based augmenting collaborative recommender system," *IEICE Trans. Inf. Syst.*, vol. E98-D, no. 4, pp. 902–910, 2015.

[27] H. Liu, H. Jin, C.-Z. Xu, and X. Liao, ''Performance and energy modeling for live migration of virtual machines,'' *Cluster Comput.*, vol. 16, no. 2, pp. 249–264, 2013.

[28] C. Jiang, L. Duan, C. Liu, J. Wan, and L. Zhou, ''VRAA: Virtualized resource auction and allocation based on incentive and penalty,'' *Cluster Comput.*, vol. 16, no. 4, pp. 639–650, 2013.

[29] J. Carlstrom and R. Rom, ''Application-aware admission control and scheduling in Web servers,'' in *Proc. IEEE 21st Annu. Joint Conf. INFOCOM*, vol. 2. Jun. 2002, pp. 506–515.

[30] R. Yu, J. Ding, W. Zhong, Y. Liu, and S. Xie, ''PHEV charging and discharging cooperation in V2G networks: A coalition game approach,'' *IEEE Internet Things J.*, vol. 1, no. 6, pp. 578–589, Dec. 2014.

[31] W. Chao, Z. M. Fadlullah, N. Kato, and A. Takeuchi, ''GT-CFS: A game theoretic coalition formulation strategy for reducing power loss in micro grids,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2307–2317, Sep. 2013.

[32] T. Truong-Huu and C.-K. Tham, ''A game-theoretic model for dynamic pricing and competition among cloud providers,'' in *Proc. IEEE/ACM 6th Int. Conf. Utility Cloud Comput. (UCC)*, Dec. 2013, pp. 235–238.

[33] H. Liang, L. X. Cai, D. Huang, X. Shen, and D. Peng, ''An SMDP-based service model for interdomain resource allocation in mobile cloud networks,'' *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2222–2232, Jun. 2012.

[34] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo. (Sep. 2009). *CRAWDAD Dataset Dartmouth/Campus (v. 2009-09-09)*. [Online]. Available: http://crawdad.org/dartmouth/campus/20090909

[35] H.-B. Hu and X.-F. Wang, ''Unified index to quantifying heterogeneity of complex networks,'' *Phys. A, Statist. Mech. Appl.*, vol. 387, no. 14, pp. 3769–3780, 2008.

RONG YU (S'05–M'08) received the Ph.D. degree from Tsinghua University, China, in 2007. He was with the School of Electronic and Information Engineering, South China University of Technology. In 2010, he joined the Institute of Intelligent Information Processing, Guangdong University of Technology, where he is currently a Full Professor. He serves as the Deputy Secretary-General of the Internet of Things (IoT) Industry Alliance, Guangdong, China, and the Deputy Head of the IoT Engineering Center, Guangdong. He has authored or co-authored over 70 international journal and conference papers and holds over ten patents. His research interest mainly focuses on wireless communications and networking, including cognitive radio, wireless sensor networks, and home networking. He is a member of the Home Networking Standard Committee in China, where he leads the standardization work of three standards.

JIEFEI DING received the M.S. degree from the Guangdong University of Technology, Guangdong, China. She spent five months to study with the Hong Kong University of Science and Technology as a Post-Graduate Visiting Internship Student in 2015. Her research interests include cloud computing resource management, software-defined network, and demand response management in smart grid.

SABITA MAHARJAN (M'09) received the M.E. degree from the Antenna and Propagation Laboratory, Tokyo Institute of Technology, Tokyo, Japan, in 2008, and the Ph.D. degree in networks and distributed systems from the Simula Research Laboratory, University of Oslo, Oslo, Norway, in 2013. She is currently a Post-Doctoral Fellow with the Simula Research Laboratory, Fornebu, Norway. Her research interests include wireless networks, network security, smart grid communications, cyber-physical systems, and machine-to-machine communications.

STEIN GJESSING received the Dr.Philos. degree from the University of Oslo, in 1985. Gjessing acted as head of the Department of Informatics for 4 years from 1987. From 1996 to 2001, he was the Chairman of the National Research Program Distributed IT-System founded by the Research Council of Norway. He participated in three European funded projects, such as Macrame, Arches, and Ascissa. He is currently an Adjunct Researcher with the Simula Research Laboratory and a Professor of Computer Science with the Department of Informatics, University of Oslo. His current research interests are routing, transport protocols, and wireless networks, including cognitive radio and smart grid applications.

YAN ZHANG (SM'05) received the Ph.D. degree from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. He is currently the Head of the Department of Networks with the Simula Research Laboratory, Norway, and an Associate Professor (part-time) with the Department of Informatics, University of Oslo, Norway. His current research interests include wireless networks and reliable and secure cyber-physical systems (e.g., healthcare, transport, and smart grid). He is a Senior Member of the IEEE ComSoc and the IEEE VT Society. He has received seven best paper awards. He is an Associate Editor or on the Editorial Board of a number of well-established scientific international journals, e.g., *Wireless Communications and Mobile Computing* (Wiley). He serves as the Guest Editor of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the *IEEE Communications Magazine*, the IEEE WIRELESS COMMUNICATIONS, and the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. He holds chair positions in a number of conferences, including the IEEE PIMRC 2016, the IEEE CCNC 2016, WICON 2016, the IEEE SmartGridComm 2015, and the IEEE CloudCom 2015. He serves as a TPC Member for numerous international conferences, including the IEEE INFOCOM, the IEEE ICC, the IEEE GLOBECOM, and the IEEE WCNC.

DANNY H. K. TSANG (M'82–SM'00–F'12) received the Ph.D. degree in electrical engineering from the Moore School of Electrical Engineering, University of Pennsylvania, USA, in 1989. During his leave from the Hong Kong University of Science and Technology (HKUST) from 2000 to 2001, he assumed the role of Principal Architect with Sycamore Networks, USA. He was responsible for the network architecture design of Ethernet MAN/WAN over SONET/DWDM networks. He also contributed to the 64B/65B encoding proposal for Transparent GFP in the T1X1.5 standard which was advanced to become the ITU G.GFP standard. The coding scheme has now been adopted by International Telecommunication Union (ITU) Generic Framing Procedure recommendation GFP-T (ITU-T G.7041/Y.1303). He has been with the Department of Electronic and Computer Engineering, HKUST, since the summer of 1992, where he is currently a Professor with the Department of Electronic and Computer Engineering. His current research interests include cloud computing, cognitive radio networks, and smart grids. He became an HKIE Fellow in 2013. He was a Guest Editor of the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS of the Special Issue on Advances in P2P Streaming Systems, an Associate Editor of the *Journal of Optical Networking* (Optical Society of America), and a Guest Editor of the IEEE SYSTEMS JOURNAL. He serves as a Technical Editor of the *IEEE Communications Magazine*.