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/TPDS.2014.2316834, IEEE Transactions on Parallel and Distributed Systems

Decentralized Computation Offloading Game For Mobile Cloud Computing

Xu Chen, *Member, IEEE*

Abstract—Mobile cloud computing is envisioned as a promising approach to augment computation capabilities of mobile devices for emerging resource-hungry mobile applications. In this paper, we propose a game theoretic approach for achieving efficient computation offloading for mobile cloud computing. We formulate the decentralized computation offloading decision making problem among mobile device users as a decentralized computation offloading game. We analyze the structural property of the game and show that the game always admits a Nash equilibrium. We then design a decentralized computation offloading mechanism that can achieve a Nash equilibrium of the game and quantify its efficiency ratio over the centralized optimal solution. Numerical results demonstrate that the proposed mechanism can achieve efficient computation offloading performance and scale well as the system size increases.

Index Terms—Mobile cloud computing, decentralized computation offloading, game theory.

1 Introduction

As smart-phones are gaining enormous popularity, more and more new mobile applications such as face recognition, natural language processing, interactive gaming, and augmented reality are emerging and attract great attention [1], [2]. This kind of mobile applications are typically resource-hungry, demanding intensive computation and high energy consumption. Due to the physical size constraint, however, mobile devices are in general resource-constrained, having limited computation resources and limited battery life. The tension between resource-hungry applications and resource-constrained mobile devices hence poses a significant challenge for the future mobile platform development [3].

Mobile cloud computing is envisioned as a promising approach to address such a challenge. As illustrated in Figure 1, mobile cloud computing can augment the capabilities of mobile devices for resource-hungry applications, by offloading the computation via wireless access to the resource-rich cloud infrastructure such as Amazon Elastic Compute Cloud (EC2) and Windows Azure Services Platform. In the cloud, each mobile device is associated with a cloud clone, which runs on a virtual machine (VM) that can execute mobile applications on behalf of the mobile device[1] [5], [6].

Although the cloud based approach can significantly augment computation capability of mobile device users, the task of developing a comprehensive and reliable mobile cloud computing system remains challenging. A key challenge is how to achieve an efficient computation offloading coordination among mobile device users. One critical factor of affecting the performance of mobile cloud computing is the wireless access efficiency [7]. If too many mobile device users choose to offload the computation to the cloud via wireless access simultaneously, they may generate severe interference to each other, which would reduce the data rates for computation data transmission. This hence can lead to low energy efficiency for computation offloading and long data transmission time. In this case, it would not be beneficial for the mobile device users to offload computation to the cloud.

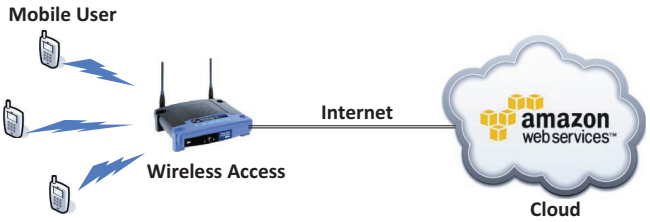


Fig. 1. An illustration of mobile cloud computing

In this paper, we adopt a game theoretic approach to address such a challenge. Game theory is a useful framework for designing decentralized mechanisms, such that the mobile device users in the system can self-organize into the mutually satisfactory computation offloading decisions. The self-organizing feature can add autonomics into mobile cloud computing system and help to ease the heavy burden of complex centralized management (e.g., information collection from massive mobile device users and computation offloading scheduling) by the cloud. Moreover, as different mobile devices are usually owned by different individuals and they may pursue different interests, game theory is a powerful tool to analyze the interactions among multiple mobile device users who act in their own interests

The author is with School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, USA. Email: xchen179@asu.edu.

1. In this study we focus on the mobile application services (e.g., remote application execution) of the cloud. However, the cloud can also provide a number of other services [4], such as platform services (e.g., storage and file backup services).

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/TPDS.2014.2316834, IEEE Transactions on Parallel and Distributed Systems

and devise incentive compatible computation offloading mechanisms such that no mobile user has the incentive to deviate unilaterally.

Specifically, we model the decentralized computation offloading decision making problem among mobile device users for mobile cloud computing as a decentralized computation offloading game. We then propose a decentralized computation offloading mechanism that can achieve the Nash equilibrium of the game. The main results and contributions of this paper are as follows:

- *Decentralized computation offloading game formulation*: We formulate the decentralized computation offloading decision making problem among multiple mobile device users as a decentralized computation offloading game, by taking into account both communication and computation aspects of mobile cloud computing.
- *Analysis of Game Structure*: We analyze the decentralized computation offloading game in both homogenous and heterogeneous wireless access cases. For the homogenous case, we show that the game admits the beneficial cloud computing group structure, which guarantees the existence of Nash equilibrium. For the more general heterogeneous case, we show that the game is a potential game, and hence admits the finite improvement property and possesses a Nash equilibrium.
- *Decentralized mechanism for achieving Nash equilibrium*: We devise a decentralized computation offloading mechanism such that mobile device users make decisions locally, which can significantly reduce the controlling and signaling overhead of the cloud. We show that the mechanism can achieve a Nash equilibrium of the decentralized computation offloading game. We further quantify the price of anarchy, i.e., the efficiency ratio of the mechanism over the centralized optimal solution. Numerical results demonstrate that the proposed mechanism can achieve efficient computation offloading performance and scale well as the system size increases.

The rest of the paper is organized as follows. We first discuss related work in Section 2, and introduce the system model in Section 3. We then propose the decentralized computation offloading game and develop the decentralized computation offloading mechanism in Sections 4 and 5, respectively. We present the numerical results in Section 6 and finally conclude in Section 7.

2 Related Work

Most previous work has investigated the efficient computation offloading mechanism design from the perspective of a single mobile device user [6]–[15]. Rudenko *et al.* in [12] demonstrated by experiments that significant energy can be saved by computation offloading. Gonzalo *et al.* in [13] developed an adaptive offloading algorithm based on both the execution history of applications and the current system conditions. Xian *et al.* in [14] introduced an efficient timeout scheme for computation offloading to increase the energy efficiency on mobile devices. Rahimi *et al.* in [16] proposed a 2-tier cloud architecture to improve both performance and scalability of mobile cloud computing. Huang *et al.* in [11] proposed a Lyapunov optimization based dynamic offloading algorithm to improve the mobile cloud computing performance while meeting the application execution time. Barbera *et al.* in [7] showed by realistic measurements that the wireless access plays a key role in affecting the performance of mobile cloud computing. Wolski *et al.* in [15] proposed a prediction based decision making framework for determining when an offloaded computation will outperform local execution on the mobile device. Wen *et al.* in [6] presented an efficient offloading policy by jointly configuring the clock frequency in the mobile device and scheduling the data transmission to minimize the energy consumption.

To the best of our knowledge, only a few works have addressed the computation offloading problem under the setting of multiple mobile device users [17]–[19]. Yang *et al.* in [17] studied the scenario that multiple users share the wireless network bandwidth, and solved the problem of maximizing the mobile cloud computing performance by a centralized heuristic genetic algorithm. Rahimi *et al.* in [18] took into consideration user mobility information and proposed a centralized greedy scheme to solve the computation offloading problem with multiple mobile users. Barbarossa *et al.* in [19] proposed a centralized scheduling algorithm to jointly optimize the communication and computation resource allocations among multiple users with the latency requirements. The centralized computation offloading schemes above requires that all the mobile device users submit their own information (e.g., wireless channel gain and the size of computation tasks) to a centralized entity (e.g., the cloud), which will determine the offloading schedule accordingly. Along a different line, in this paper we adopt the game theoretic approach and devise a decentralized mechanism wherein each mobile device user makes the computation offloading decision locally. This can help to reduce the controlling and signaling overhead of the cloud.

3 System Model

In this section, we introduce the system model of mobile cloud computing. We consider a set of $\mathcal{N} = \{1, 2, ..., N\}$ collocated mobile device users and each of which has a computationally intensive and delay sensitive task to be completed. There exists a wireless access base-station s, through which the mobile device users can offload the computation to the cloud (e.g., Amazon EC2 or Microsoft Azure). Similar to many previous studies in mobile cloud computing [6], [17], [19] and mobile networking [20]–[22], to enable tractable analysis and get useful insights, we consider a quasi-static scenario where the set of mobile device users $\mathcal{N}$ remains unchanged

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/TPDS.2014.2316834, IEEE Transactions on Parallel and Distributed Systems

during a computation offloading period (e.g., within several seconds), while may change across different periods[2]. The general case that mobile users may depart and leave dynamically within a computation offloading period will be considered in a future work. Since both the communication and computation aspects play a key role in mobile cloud computing, we next introduce the communication and computation models in details.

3.1 Communication Model

We first introduce the communication model for wireless access. The wireless access base-station s can be either a WiFi access point, or a Femtocell network access point [23], or a macrocell base-station in cellular networks that manages the uplink/downlink communications of mobile device users. We denote $a_n \in \{0,1\}$ as the computation offloading decision of mobile device user n. Specifically, we have $a_n = 1$ if user n chooses to offload the computation to the cloud via wireless access. We have $a_n = 0$ if user n decides to compute its task locally on the mobile device. Given the decision profile $\boldsymbol{a} = (a_1, a_2, ..., a_N)$ of all the mobile device users, we can compute the uplink data rate for computation offloading of mobile device user n as [24]

$$R_n(\boldsymbol{a}) = W \log_2 \left(1 + \frac{P_n H_{n,s}}{\omega_n + \sum_{m \in \mathcal{N} \setminus \{n\}: a_m = 1} P_m H_{m,s}}\right). \quad (1)$$

Here W is the channel bandwidth and P_n is user n's transmission power which is determined by the wireless access base-station according to some power control algorithms such as [25], [26]. Further, $H_{n,s}$ denotes the channel gain between the mobile device user n and the base-station, and $\omega_n = \omega_n^0 + \omega_n^1$ denotes the background interference power including the noise power ω_n^0 and the interference power ω_n^1 from other mobile device users who carry out wireless transmission but do not involve in the mobile cloud computing.

From the communication model in (1), we see that if too many mobile device users choose to offload the computation via wireless access simultaneously, they may incur severe interference, leading to low data rates. As we discuss latter, this would negatively affect the performance of mobile cloud computing.

3.2 Computation Model

We then introduce the computation model. We consider that each mobile device user n has a computation task $\mathcal{I}_n \triangleq (B_n, D_n)$ that can be computed either locally on the mobile device or remotely on the cloud via computation offloading. Here B_n denotes the size of computation input data (e.g., the program codes and input parameters) involving in the computation task $\mathcal{I}_n$ and D_n denotes the total number of CPU cycles required to accomplish the computation task $\mathcal{I}_n$. A mobile device user n can apply the methods in [3], [5], [17] to obtain the information of B_n and D_n. We next discuss the computation overhead in terms of both energy consumption and processing time for both local and cloud computing approaches.

3.2.1 Local Computing

For the local computing approach, a mobile device user n executes its computation task $\mathcal{I}_n$ locally on the mobile device. Let F_n^l be the computation capability (i.e., CPU cycles per second) of mobile device user n. Here we allow that different mobile devices may have different computation capability. The computation execution time of the task $\mathcal{I}_n$ by local computing is then given as

$$T_n^l = \frac{D_n}{F_n^l}. \quad (2)$$

For the computational energy, we have that

$$E_n^l = \nu_n D_n, \quad (3)$$

where ν_n is the coefficient denoting the consumed energy per CPU cycle. According to the realistic measurements in [6], [27], we can set $\nu_n = 10^{-11}(F_n^l)^2$.

According to (2) and (3), we can then compute the overhead of the local computing approach in terms of computational time and energy as

$$Z_n^l = \gamma_n^T T_n^l + \gamma_n^E E_n^l, \quad (4)$$

where $0 \leq \gamma_n^T, \gamma_n^E \leq 1$ denote the weights of computational time and energy for mobile device user n's decision making, respectively. To provide rich modeling flexibility and meet user-specific demands, we allow that different users can choose different weighting parameters in the decision making. For example, when a user is at a low battery state, the user would like to put more weight on energy consumption (i.e., a larger γ_n^E) in the decision making, in order to save more energy. When a user is running some application that is sensitive to the delay (e.g., video streaming), then the user can put more weight on the processing time (i.e., a larger γ_n^T), in order to reduce the delay. Note that the weights could be dynamic if a user runs different applications or has different policies/demands at different computation offloading periods. For ease of exposition, in this paper we assume that the weights of a user are fixed within one computation offloading period, while can be changed in different periods.

3.2.2 Cloud Computing

For the cloud computing approach, a mobile device user n will offload its computation task $\mathcal{I}_n$ to the cloud and the cloud will execute the computation task on behalf of the mobile device user.

For the computation offloading, a mobile device user n would incur the extra overhead in terms of time and

2. This assumption holds for many applications such as face recognition and natural language processing, in which the size of computation input data is not large and hence the computation offloading can be finished in a smaller time scale (e.g., within several seconds) than the time scale of users' mobility.

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/TPDS.2014.2316834, IEEE Transactions on Parallel and Distributed Systems

energy for transmitting the computation input data to the cloud via wireless access. According to the communication model in Section 3.1, we can compute the transmission time and energy of mobile device user n for offloading the input data of size B_n as, respectively,

$$T^c_{n,off}(\boldsymbol{a}) = \frac{B_n}{R_n(\boldsymbol{a})}, \tag{5}$$

and

$$E^c_n(\boldsymbol{a}) = \frac{P_n B_n}{R_n(\boldsymbol{a})}. \tag{6}$$

After the offloading, the cloud will execute the computation task $\mathcal{I}_n$. Let F^c_n be the computation capability (i.e., CPU cycles per second) assigned to user n by the cloud. The execution time of the task $\mathcal{I}_n$ of mobile device user n on the cloud can be then given as

$$T^c_{n,exe} = \frac{D_n}{F^c_n}. \tag{7}$$

According to (5), (6), and (7), we can compute the overhead of the cloud computing approach in terms of processing time and energy as

$$Z^c_n(\boldsymbol{a}) = \gamma^T_n \left(T^c_{n,off}(\boldsymbol{a}) + T^c_{n,exe}\right) + \gamma^E_n E^c_n(\boldsymbol{a}). \tag{8}$$

Similar to many studies such as [10]–[14], we neglect the time overhead for the cloud to send the computation outcome back to the mobile device user, due to the fact that for many applications (e.g., face recognition), the size of the computation outcome in general is much smaller than the size of computation input data including the mobile system settings, program codes and input parameters.

According to the communication and computation models above, we see that the computation offloading decisions $\boldsymbol{a}$ among the mobile device users are coupled. If too many mobile device users simultaneously choose to offload the computation task to the cloud via wireless access, they may incur severe interference and this would lead to a low data rate. When the data rate $R_n(\boldsymbol{a})$ of a mobile device user n is low, it would consume high energy in the wireless access for offloading the computation input data to cloud and incur long transmission time as well. In this case, it would be more beneficial for the user to compute the task locally on the mobile device to avoid the long processing time and high energy consumption by the cloud computing approach. In the following sections, we will adopt a game theoretic approach to address the issue of how to achieve efficient computation offloading decision makings among the mobile device users.

4 Decentralized Computation Offloading Game

In this section, we develop a game theoretic approach for achieving efficient computation offloading decision makings among the mobile device users. The primary rationale of adopting the game theoretic approach is that the mobile devices are owned by different individuals and they may pursue different interests. Game theory is a powerful framework to analyze the interactions among multiple mobile device users who act in their own interests and devise incentive compatible computation offloading mechanisms such that no user has the incentive to deviate unilaterally. Moreover, by leveraging the intelligence of each individual mobile device user, game theory is a useful tool for devising decentralized mechanisms with low complexity, such that the users can self-organize into a mutually satisfactory solution. This can help to ease the heavy burden of complex centralized management by the cloud and reduce the controlling and signaling overhead between the cloud and mobile device users.

4.1 Game Formulation

We consider the decentralized computation offloading decision making problem among the mobile device users within a computation offloading period. Let $a_{-n} = (a_1, ..., a_{n-1}, a_{n+1}, ..., a_N)$ be computation offloading decisions by all other users except user n. Given other users' decisions a_{-n}, user n would like to select a proper decision $a_n \in \{0, 1\}$ (i.e., local computing or cloud computing) to minimize its computation overhead in terms of energy consumption and processing time, i.e.,

$$\min_{a_n \in \{0,1\}} V_n(a_n, a_{-n}), \forall n \in \mathcal{N}.$$

According to (4) and (8), we can obtain the overhead function of mobile device user n as

$$V_n(a_n, a_{-n}) = \begin{cases} Z^l_n, & \text{if } a_n = 0, \\ Z^c_n(\boldsymbol{a}), & \text{if } a_n = 1. \end{cases} \tag{9}$$

We then formulate the problem above as a strategic game $\Gamma = (\mathcal{N}, \{\mathcal{A}_n\}_{n\in\mathcal{N}}, \{V_n\}_{n\in\mathcal{N}})$, where the set of mobile device users $\mathcal{N}$ is the set of players, $\mathcal{A}_n \triangleq \{0, 1\}$ is the set of strategies for user n, and the overhead function $V_n(a_n, a_{-n})$ of each user n is the cost function to be minimized by player n. In the sequel, we call the game Γ as the decentralized computation offloading game. We now introduce the concept of Nash equilibrium [28].

Definition 1. *A strategy profile $\boldsymbol{a}^* = (a^*_1, ..., a^*_N)$ is a Nash equilibrium of the decentralized computation offloading game if at the equilibrium $\boldsymbol{a}^*$, no player can further reduce its overhead by unilaterally changing its strategy, i.e.,*

$$V_n(a^*_n, a^*_{-n}) \leq V_n(a_n, a^*_{-n}), \forall a_n \in \mathcal{A}_n, n \in \mathcal{N}. \tag{10}$$

The Nash equilibrium has the nice self-stability property such that the users at the equilibrium can achieve a mutually satisfactory solution and no user has the incentive to deviate. This property is very important to the decentralized computation offloading problem, since the mobile devices are owned by different individuals and they may act in their own interests.

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/TPDS.2014.2316834, IEEE Transactions on Parallel and Distributed Systems

4.2 Game Property

We then study the existence of Nash equilibrium of the decentralized computation offloading game. To proceed, we first introduce an important concept of best response [28].

Definition 2. *Given the strategies a_{-n} of the other players, player n's strategy $a_n^* \in \mathcal{A}_n$ is a best response if*

$$V_n(a_n^*, a_{-n}) \leq V_n(a_n, a_{-n}), \forall a_n \in \mathcal{A}_n. \quad (11)$$

According to (10) and (11), we see that at the Nash equilibrium all the users play the best response strategies towards each other. Based on the concept of best response, we have the following observation for the decentralized computation offloading game.

Lemma 1. *Given the strategies a_{-n} of other mobile device users in the decentralized computation offloading game, the best response of a user n is given as the following threshold strategy*

$$a_n^* = \begin{cases} 1, & \text{if } \sum_{m \in \mathcal{N}\setminus\{n\}: a_m = 1} P_m H_{m,s} \leq L_n, \\ 0, & \text{otherwise,} \end{cases}$$

where the threshold

$$L_n = \frac{P_n H_{n,s}}{2^{\frac{(\gamma_n^T + \gamma_n^E P_n) B_n}{W\left(\gamma_n^T T_n^l + \gamma_n^E E_n^l - \gamma_n^T T_{n,exe}^c\right)}} - 1} - \omega_n.$$

The proof is given in Section 8.1 of the separate supplementary file. According to Lemma 1, we see that when the received interference $\sum_{m \in \mathcal{N}\setminus\{n\}: a_m = 1} P_m H_{m,s}$ is lower enough, it is beneficial for user n to offload the computation to the cloud. Otherwise, the user n should compute the task on the mobile device locally. Since the wireless access plays a critical role in mobile cloud computing, we next discuss the existence of Nash equilibrium of the the decentralized computation offloading game in both homogeneous and heterogeneous wireless access cases.

4.2.1 Homogeneous Wireless Access Case

We first consider the case that users' wireless access is homogenous, i.e., $P_m H_{m,s} = P_n H_{n,s} = K$, for any $n, m \in \mathcal{N}$. This can correspond to the scenario that all the mobile device users experience the similar channel condition and are assigned with the same transmission power by the base-station. However, different users may have different thresholds L_n, i.e., they are heterogeneous in terms of computation capabilities and tasks.

For the homogenous wireless access case, without loss of generality, we can order the set $\mathcal{N}$ of mobile device users so that $\frac{L_1}{K} \geq \frac{L_2}{K} \geq ... \geq \frac{L_N}{K}$. Based on this, we have the following useful observation.

Lemma 2. *For the decentralized computation offloading game with homogenous wireless access, if there exists a non-empty beneficial cloud computing group of mobile device users $\mathcal{S} \subseteq \mathcal{N}$ such that*

$$|\mathcal{S}| \quad \leq \quad \frac{L_i}{K} + 1, \forall i \in \mathcal{S}, \quad (12)$$

and further if $\mathcal{S} \subset \mathcal{N}$,

$$|\mathcal{S}| > \frac{L_j}{K}, \forall j \in \mathcal{N}\setminus\mathcal{S}, \quad (13)$$

then the strategy profile wherein users $i \in \mathcal{S}$ play the strategy $a_i = 1$ and the other users $j \in \mathcal{N}\setminus\mathcal{S}$ play the strategy $a_j = 0$ is a Nash equilibrium.

Algorithm 1 Algorithm for finding beneficial cloud computing group

1: **Input**: the set of ordered mobile device users with $\frac{L_1}{K} \geq \frac{L_2}{K} \geq ... \geq \frac{L_N}{K}$ and $\frac{L_1}{K} \geq 0$.
2: **Output**: a beneficial cloud computing group $\mathcal{S}$.
3: **set** $\mathcal{S} = \{1\}$.
4: **for** $t = 2$ to N **do**
5: **set** $\tilde{\mathcal{S}} = \mathcal{S} \cup \{t\}$
6: **if** $|\tilde{\mathcal{S}}| > \frac{L_t}{K} + 1$ **then**
7: **stop** and go to **return**.
8: **else set** $\mathcal{S} = \tilde{\mathcal{S}}$.
9: **end if**
10: **end for**
11: **return** $\mathcal{S}$.

The proof is given in Section 8.2 of the separate supplementary file. For example, for a set of 4 users with $\left(\frac{L_1}{K}, \frac{L_2}{K}, \frac{L_3}{K}, \frac{L_4}{K}\right) = (5, 4, 3, 2)$, the beneficial cloud computing group is $\mathcal{S} = \{1, 2, 3\}$. In general, when $\frac{L_1}{K} \geq 0$, we can construct the beneficial cloud computing group by using Algorithm 1. Thus, we have the following result.

Theorem 1. *The decentralized computation offloading game with homogenous wireless access always has a Nash equilibrium. More specifically, when $\frac{L_1}{K} < 0$, all users $n \in \mathcal{N}$ playing the strategy $a_n = 0$ is a Nash equilibrium. When $\frac{L_1}{K} \geq 0$, we can construct a beneficial cloud computing group $\mathcal{S} \neq \varnothing$ by Algorithm 1 such that the strategy profile wherein users $i \in \mathcal{S}$ play the strategy $a_i = 1$ and the other users $j \in \mathcal{N}\setminus\mathcal{S}$ play the strategy $a_j = 0$ is a Nash equilibrium.*

The proof is given in Section 8.3 of the separate supplementary file. Since the computational complexity of ordering operation (e.g., quicksort algorithm) is typically $\mathcal{O}(N \log N)$ and the construction procedure in Algorithm 1 involves at most N operations (with each operation of the complexity of $\mathcal{O}(1)$), the beneficial cloud computing group construction algorithm has a low computational complexity of $\mathcal{O}(N \log N)$. This implies that we can compute the Nash equilibrium of the decentralized computation offloading game in the homogenous wireless access case in a fast manner.

4.2.2 General Wireless Access Case

We next consider the general case including the case that users' wireless access can be heterogeneous, i.e.,

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/TPDS.2014.2316834, IEEE Transactions on Parallel and Distributed Systems

$P_m H_{m,s} \neq P_n H_{n,s}$. Since mobile device users may have different transmission power P_n, channel gain $H_{n,s}$ and thresholds L_n, the analysis based on the beneficial cloud computing group in the homogenous case can not apply here. We hence resort to a power tool of potential game [29].

Definition 3. *A game is called a potential game if it admits a potential function $\Phi(\boldsymbol{a})$ such that for every $n \in \mathcal{N}$, $a_{-n} \in \prod_{i \neq n} \mathcal{A}_i$, and $a_n^{'}, a_n \in \mathcal{A}_n$, if*

$$V_n(a_n^{'}, a_{-n}) < V_n(a_n, a_{-n}), \tag{14}$$

we have

$$\Phi(a_n^{'}, a_{-n}) < \Phi(a_n, a_{-n}). \tag{15}$$

Definition 4. *The event where a player n changes to an action $a_n^{'}$ from the action a_n is a better response update if and only if its cost function is decreased, i.e.,*

$$V_n(a_n^{'}, a_{-n}) < V_n(a_n, a_{-n}). \tag{16}$$

An appealing property of the potential game is that it admits the finite improvement property, such that any asynchronous better response update process (i.e., no more than one player updates the strategy at any given time) must be finite and leads to a Nash equilibrium [29]. Here the potential function to a game has the same spirit as the Lyapunov function to a dynamical system. If a dynamic system is shown to have a Lyapunov function, then the system has a stable point. Similarly, if a game admits a potential function, the game must have a Nash equilibrium.

We now prove the existence of Nash equilibrium of the general decentralized computation offloading game by showing that the game is a potential game. Specifically, we define the potential function as

$$\begin{aligned}\Phi(\boldsymbol{a}) =& \frac{1}{2} \sum_{n=1}^{N} \sum_{m \neq n} P_n H_{n,s} P_m H_{m,s} I_{\{a_n=1\}} I_{\{a_m=1\}} \\ &+ \sum_{n=1}^{N} P_n H_{n,s} L_n I_{\{a_n=0\}},\end{aligned} \tag{17}$$

where $I_{\{A\}}$ is the indicator function such as $I_{\{A\}} = 1$ if the event A is true and $I_{\{A\}} = 0$ otherwise.

Theorem 2. *The general decentralized computation offloading game is a potential game with the potential function as given in (17), and hence always has a Nash equilibrium and the finite improvement property.*

The proof is given in Section 8.4 of the separate supplementary file. Theorem 2 implies that any asynchronous better response update process is guaranteed to reach a Nash equilibrium within a finite number of iterations. This motivates the algorithm design in following Section 5.

5 Decentralized Computation Offloading Mechanism

In this section we propose a decentralized computation offloading mechanism in Algorithm 2 for achieving the Nash equilibrium of the decentralized computation offloading game.

5.1 Mechanism Design

The motivation of using the decentralized computation offloading mechanism is to coordinate mobile device users to achieve a mutually satisfactory decision making, prior to the computation task execution. The key idea of the mechanism design is to utilize the finite improvement property of the decentralized computation offloading game and let one mobile device user improve its computation offloading decision at a time. Specifically, by using the clock signal from the wireless access base-station for synchronization, we consider a slotted time structure for the computation offloading decision update. Each decision slot t consists the following two parts:

- **Interference Measurement**: Each mobile device user n locally measures the received interference $\mu_n(t) = \sum_{m \in \mathcal{N} \setminus \{n\}: a_m(t)=1} P_m H_{m,s}$ generated by other users who currently choose the decisions of offloading the computation tasks to the cloud via wireless access. To facilitate the interference measurement, for example, the users m who choose decisions $a_m(t) = 1$ at the current slot will transmit some pilot signals to the base-station. And each mobile device user can then enquire its received interference $\mu_n(t)$ from the base-station.
- **Decision Update Contention**: We exploit the finite improvement property of the game by having one mobile device user carry out a decision update at each decision slot. We let users who can improve their computation performance compete for the decision update opportunity in a decentralized manner. More specifically, according to Lemma 1, each mobile device user n first computes its set of best response update based on the measured interference $\mu_n(t)$ as

$$\begin{aligned}\Delta_n(t) &\triangleq \{a_n^* : V_n(a_n^*, a_{-n}(t)) < V_n(a_n(t), a_{-n}(t))\} \\ &= \begin{cases} \{1\}, & \text{if } a_n(t) = 0 \text{ and } \mu_n(t) \leq L_n, \\ \{0\}, & \text{if } a_n(t) = 1 \text{ and } \mu_n(t) > L_n, \\ \varnothing, & \text{otherwise.} \end{cases}\end{aligned}$$

  The best response here is similar to the steepest descent direction selection to reduce user's overhead. Then, if $\Delta_n(t) \neq \varnothing$ (i.e., user n can improve), user n will contend for the decision update opportunity. Otherwise, user n will not contend and adhere to the current decision at next decision slot, i.e., $a_n(t+1) = a_n(t)$. For the decision update contention, for example, we can adopt the random backoff-based mechanism by setting the time length of decision

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/TPDS.2014.2316834, IEEE Transactions on Parallel and Distributed Systems

update contention as τ^*. Each contending user n first generates a backoff time value τ_n according to the uniform distribution over $[0, \tau^*]$ and countdown until the backoff timer expires. When the timer expires, if the user has not received any request-to-update (RTU) message from other mobile device users yet, the user will update its decision for the next slot as $a_n(t+1) \in \Delta_n(t)$ and then broadcast a RTU message to all users to indicate that it wins the decision update contention. For other users, on hearing the RTU message, they will not update their decisions and will choose the same decisions at next slot, i.e., $a_n(t+1) = a_n(t)$.

According to the finite improvement property in Theorem 2, the mechanism will converge to a Nash equilibrium of the decentralized computation offloading game within finite number of decision slots. In practice, we can implement that the computation offloading decision update process terminates when no RTU messages are broadcasted for multiple consecutive decision slots (i.e., no decision update can be further carried out by any users). Then each mobile device user n executes the computation task according to the decision a_n obtained at the last decision slot by the mechanism. Due to the property of Nash equilibrium, no user has the incentive to deviate from the achieved decisions. This is very important to the decentralized computation offloading problem, since the mobile devices are owned by different individuals and they may act in their own interests. By following the decentralized computation offloading mechanism, the users adopt the best response to improve their decision makings and eventually self-organize into a mutually satisfactory solution (i.e., Nash equilibrium).

We then analyze the computational complexity of the algorithm. In each iteration, N mobile users will execute the operations in Lines $5-15$. Since the operations in Lines $5-15$ only involve some basic arithmetical calculations, the computational complexity in each iteration is $\mathcal{O}(N)$. Suppose that it takes C iterations for the algorithm to converge. Then the total computational complexity of the algorithm is $\mathcal{O}(CN)$. Numerical results in Section 6 show that the number of iterations C for convergence increases linearly with the number of users N. This demonstrates that the decentralized computation offloading mechanism can converge in a fast manner in practice.

Algorithm 2 Decentralized computation offloading mechanism

1: **initialization:**
2: each mobile device user n **chooses** the computation decision $a_n(0) = 1$.
3: **end initialization**

4: **repeat** for each user n and each decision slot t in parallel:
5: **measure** the interference $\mu_n(t)$.
6: **compute** the best response set $\Delta_n(t)$.
7: **if** $\Delta_n(t) \neq \varnothing$ **then**
8: **contend** for the decision update opportunity.
9: **if win** the decision update contention **then**
10: **choose** the decision $a_n(t+1) \in \Delta_n(t)$ for next slot.
11: **broadcast** the RTU message to other users.
12: **else** choose the original decision $a_n(t+1) = a_n(t)$ for next slot.
13: **end if**
14: **else choose** the original decision $a_n(t+1) = a_n(t)$ for next slot.
15: **end if**
16: **until** no RTU messages are broadcasted for M consecutive slots

5.2 Performance Analysis

We then discuss the efficiency of Nash equilibrium by the decentralized computation offloading mechanism. Note that the decentralized computation offloading game may have multiple Nash equilibria, and the proposed decentralized computation offloading mechanism will randomly select one Nash equilibrium (since a random user is chosen for decision update). Following the definition of price of anarchy (PoA) in game theory [30], we will quantify the efficiency ratio of the worst-case Nash equilibrium over the centralized optimal solution. Let Υ be the set of Nash equilibria of the decentralized computation offloading game. Then the PoA is defined as

$$\text{PoA} = \frac{\max_{\boldsymbol{a} \in \Upsilon} \sum_{n \in \mathcal{N}} V_n(\boldsymbol{a})}{\min_{\boldsymbol{a} \in \prod_{n=1}^{N} \mathcal{A}_n} \sum_{n \in \mathcal{N}} V_n(\boldsymbol{a})},$$

which is lower bounded by 1. A larger PoA implies that the set of Nash equilibrium is less efficient (in the worst-case sense) using the centralized optimum as a benchmark. Let $\overline{Z_n^c} = \frac{\left(\gamma_n^T + \gamma_n^E P_n\right) B_n}{W \log_2\left(1 + \frac{P_n H_{n,s}}{\omega_n}\right)} + \gamma_n^T T_{n,exe}^c$. We can show the following result.

Theorem 3. *The PoA of the decentralized computation offloading game is at most* $\frac{\sum_{n=1}^{N} Z_n^l}{\sum_{n=1}^{N} \min\{Z_n^l, \overline{Z_n^c}\}}$.

The proof is given in Section 8.5 of the separate supplementary file. Intuitively, Theorem 3 indicates that when users have lower cost of local computing (i.e., Z_n^l is smaller), the Nash equilibrium is closer to the centralized optimum and hence the PoA is lower. Moreover, when the communication efficiency is higher (i.e., $P_n H_{n,s}$ is larger and hence $\overline{Z_n^c}$ is larger), the performance of Nash equilibrium can be improved. Numerical results in Section 6 demonstrate that the Nash equilibrium by the decentralized computation offloading mechanism is efficient, with at most 10% performance loss, compared with the centralized optimal solution.

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/TPDS.2014.2316834, IEEE Transactions on Parallel and Distributed Systems

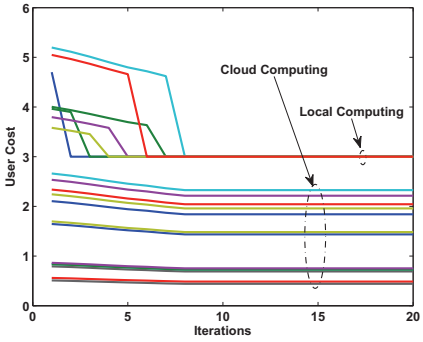


Fig. 2. Dynamics of user cost by the decentralized computation offloading mechanism

6 Numerical Results

In this section, we evaluate the proposed decentralized computation offloading mechanism by numerical studies. We first consider the mobile cloud computing scenario that $N = 20$ mobile device users are randomly scattered over a $50\text{m}\times 50\text{m}$ region and the wireless access base-station is located in the center of the region. For the wireless access, we set the channel bandwidth $W = 5$ MHz, the transmission power $P_n = 100$ mWatts, and the background noise $\omega_n = -100$ dBm. According to the physical interference model [24], we set the channel gain $H_{n,s} = d_{n,s}^{-\alpha}$, where $d_{n,s}$ is the distance between mobile device user n and the cloudlet and $\alpha = 4$ is the path loss factor. We set the decision weights $\gamma_n^T = \gamma_n^E = 0.5$. For the computation task, we use the face recognition application in [1], where the data size for the computation offloading $B_n = 420$ KB and the total number of CPU cycles $D_n = 1000$ Megacycles. The CPU computational capability F_n^l of a mobile device user n is randomly assigned from the set $\{0.5, 0.8, 1.0\}$ GHz and the computational capability on the cloud $F_n^c = 100$ GHz [1].

We first show the dynamics of mobile device users' computation cost $V_n(\boldsymbol{a})$ by the proposed decentralized computation offloading mechanism in Figure 2. We see that the mechanism can keep mobile users' cost decreasing and converge to an equilibrium. To verify that the convergent equilibrium is a Nash equilibrium, we further show the dynamics of the potential function value $\Phi(\boldsymbol{a})$ of the decentralized computation offloading game in Figure 3. It demonstrates that the proposed decentralized computation offloading mechanism can lead the potential function of the game to the minimum point, which is a Nash equilibrium according to the property of potential game.

To investigate the impact of computation size on decentralized computation offloading, we then implement the simulations with different number of CPU processing cycles D_n required for completing the computing task. Upon comparison, we also implement the local mobile computing solution such that all the mobile device users compute their tasks locally on the mobile devices. The results are shown in Figure 4. We see that the system-wide computing cost $\sum_{n\in\mathcal{N}} V_n(\boldsymbol{a})$ by decentralized computation offloading and local mobile computing solutions increases as the number of CPU processing cycles D_n increases. However, the system-wide computing cost $\sum_{n\in\mathcal{N}} V_n(\boldsymbol{a})$ by decentralized computation offloading increases much slower than that of local mobile computing. This is because that as the number of CPU processing cycles D_n increases, more mobile device users choose to utilize the cloud computing via computation offloading to mitigate the heavy cost of local computing.

To evaluate the impact of communication data size on the decentralized computation offloading, we next implement the simulations with different data size for computation offloading B_n in Figure 5. We observe that the system-wide computing cost $\sum_{n\in\mathcal{N}} V_n(\boldsymbol{a})$ by decentralized computation offloading as the data size for computation offloading B_n increases, due to the fact that a larger data size requires higher overhead for computation offloading via wireless communication. Moreover, we see that the system-wide computing cost $\sum_{n\in\mathcal{N}} V_n(\boldsymbol{a})$ by decentralized computation offloading increases slowly when the data size for computation offloading B_n is large. This is because that the data size for computation offloading B_n is large, more mobile device users choose to compute the tasks locally on the mobile devices, in order to avoid the heavy cost of computation offloading via wireless access.

To benchmark the performance of the decentralized computation offloading mechanism, we further implement the system-wide computing cost minimization solution by centralized optimization, i.e., $\max_{\boldsymbol{a}} \sum_{n\in\mathcal{N}} V_n(\boldsymbol{a})$. Notice that the centralized optimization solution requires the complete information of all mobile device users, such as the details of computing tasks, the transmission power, the channel gain, and the CPU frequency of all mobile devices. While the decentralized computation offloading mechanism only requires each mobile device user to measure its received interference and make the decision locally. We run experiments with the number of $N = 10, 15, ..., 50$ mobile device users being randomly scattered over the square area, respectively. We repeat each experiment 100 times and show the average system-wide computing cost in Figure 6. We see that the system-wide computing cost by all the computation offloading solutions increases as the number of mobile device users N increases. The proposed incentive compatible computation offloading solution can reduce up-to 33% and 38% computing cost over the solutions of all the users choosing the local computing and choosing the cloud computing, respectively. Compared with the centralized optimization solution, the performance loss of the decentralized computation offloading mechanism is less than 10% in all cases. This demonstrates the efficiency of the proposed decentralized computation offloading mechanism. We next evaluate the convergence time of the decentralized

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/TPDS.2014.2316834, IEEE Transactions on Parallel and Distributed Systems

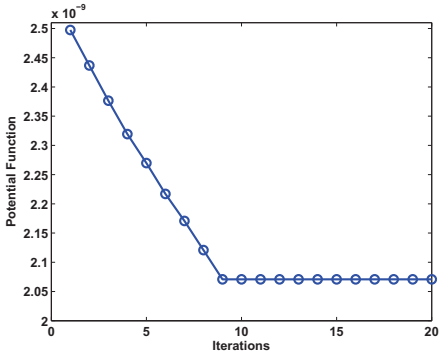


Fig. 3. Dynamics of potential function by the decentralized computation offloading mechanism

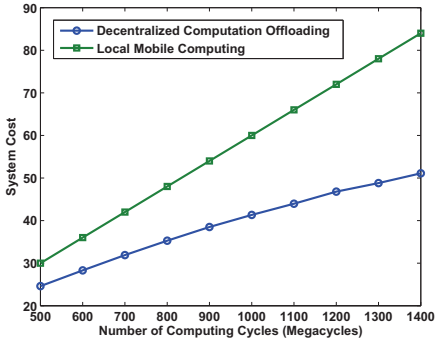


Fig. 4. System-wide computing cost with different number of CPU processing cycles

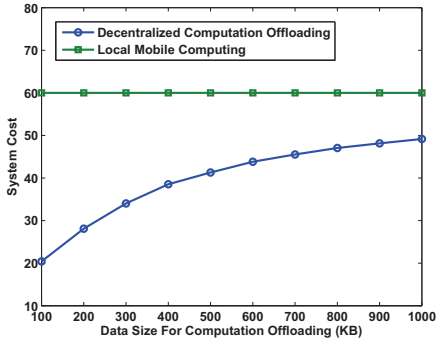


Fig. 5. System-wide computing cost with different data size for the computation offloading

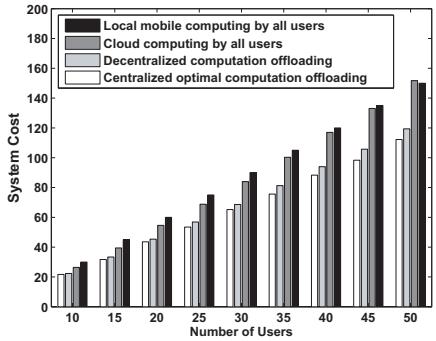


Fig. 6. Average system-wide computing cost

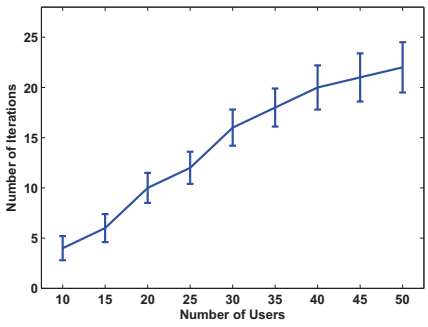


Fig. 7. Number of iterations by decentralized computation offloading mechanism

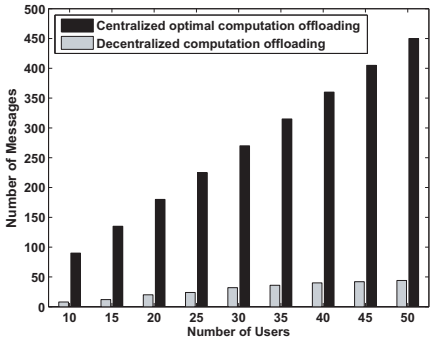


Fig. 8. Number of controlling and signaling messages by the centralized optimal and decentralized computation offloading mechanisms

computation offloading mechanism. Figure 7 shows that the average convergence time increases linearly with the number of mobile device users N. This shows that the decentralized computation offloading mechanism scales well with the size of mobile device users. This is critical since computing the centralized optimal computation offloading solution involves solving the integer programming problem (i.e., the decision variables $a_n \in \{0, 1\}$) and the computational complexity grows exponentially as the number of mobile device users N increases.

To evaluate the controlling and signaling overhead reduction by the decentralized computation offloading mechanism, we further show the number of controlling and signaling messages exchanged among the mobile users and between the users and the cloud in Figure 8. It demonstrates that the decentralized computation offloading mechanism can reduce the number of controlling and signaling messages by at least 89% over the centralized optimal computation offloading scheme in all cases. This is because that for the decentralized computation offloading mechanism, a mobile user would exchange messages (for interference measurement and decision update announcement) only when it updates its computation decision. While for the centralized optimal computation offloading scheme, each mobile user needs to report all its local parameters to the cloud, including the transmission power, the channel gain, the background interference power, the local computation capability, and many other parameters. Moreover, in some application scenarios, due to privacy concerns some mobile users may be sensitive to the revealing of their local parameters and hence do not have the incentive to participate in the centralized optimal computation offloading scheme. While the decentralized computation offloading mechanism does not have this issue since each mobile user can make the computation offloading decision locally without exposing its local parameters.

7 Conclusion

In this paper, we consider the computation offloading decision making problem among mobile device users for mobile cloud computing and propose as a decentralized computation offloading game formulation. We show that the game always admits a Nash equilibrium for both cases of homogenous and heterogenous wireless access. We also design a decentralized computation offloading

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/TPDS.2014.2316834, IEEE Transactions on Parallel and Distributed Systems

mechanism that can achieve a Nash equilibrium of the game and further quantify its price of anarchy. Numerical results demonstrate that the proposed mechanism is efficient and scales well as the system size increases.

For the future work, we are going to consider the more general case that mobile users may depart and leave dynamically within a computation offloading period. In this case, the user mobility patterns might play an important role in the problem formulation.

REFERENCES

[1] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2012, pp. 000 059–000 066.

[2] J. Cohen, "Embedded speech recognition applications in mobile phones: Status, trends, and challenges," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 5352–5355.

[3] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: making smartphones last longer with code offload," in *the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 49–62.

[4] P. Bahl, R. Y. Han, L. E. Li, and M. Satyanarayanan, "Advancing the state of mobile cloud computing," in *the third ACM workshop on Mobile cloud computing and services*, 2012.

[5] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *the sixth conference on Computer systems*. ACM, 2011, pp. 301–314.

[6] Y. Wen, W. Zhang, and H. Luo, "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones," in *IEEE INFOCOM*. IEEE, 2012, pp. 2716–2720.

[7] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, "To offload or not to offload? the bandwidth and energy costs of mobile cloud computing," in *IEEE INFOCOM*, vol. 2013, 2013.

[8] K. Kumar, J. Liu, Y. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 129–140, 2013.

[9] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: A survey, state of art and future directions," *ACM/Springer Mobile Networks and Applications (MONET)*, pp. 1–11, 2013.

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

[11] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.

[12] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, "Saving portable computer battery power through remote process execution," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 2, no. 1, pp. 19–26, 1998.

[13] G. Huertacanepa and D. Lee, "An adaptable application offloading scheme based on application behavior," in *22nd International Conference on Advanced Information Networking and Applications-Workshops*, 2008.

[14] C. Xian, Y. Lu, and Z. Li, "Adaptive computation offloading for energy conservation on battery-powered systems," in *International Conference on Parallel and Distributed Systems*, vol. 2. IEEE, 2007, pp. 1–8.

[15] R. Wolski, S. Gurun, C. Krintz, and D. Nurmi, "Using bandwidth data to make computation offloading decisions," in *IEEE International Symposium on Parallel and Distributed Processing*. IEEE, 2008, pp. 1–8.

[16] M. R. Rahimi, N. Venkatasubramanian, S. Mehrotra, and A. V. Vasilakos, "Mapcloud: mobile applications on an elastic and scalable 2-tier cloud architecture," in *the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing*, 2012.

[17] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 4, pp. 23–32, 2013.

[18] M. R. Rahimi, N. Venkatasubramanian, and A. V. Vasilakos, "Music: Mobility-aware optimal service allocation in mobile cloud computing," in *IEEE International Conference on Cloud Computing (Cloud)*, July 2013.

[19] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Joint allocation of computation and communication resources in multiuser mobile cloud computing," in *IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2013, pp. 26–30.

[20] S. Wu, Y. Tseng, C. Lin, and J. Sheu, "A multi-channel mac protocol with power control for multi-hop mobile ad hoc networks," *The Computer Journal*, vol. 45, no. 1, pp. 101–110, 2002.

[21] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "An iterative double auction mechanism for mobile data offloading," in *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2013.

[22] Y. Wu, P. A. Chou, and S. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1906–1918, 2005.

[23] V. Chandrasekhar, J. Andrews, and A. Gatherer, "Femtocell networks: a survey," *IEEE Communications Magazine*, vol. 46, no. 9, pp. 59–67, 2008.

[24] T. S. Rappaport, *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 1996, vol. 2.

[25] M. Xiao, N. B. Shroff, and E. K. Chong, "A utility-based power-control scheme in wireless cellular systems," *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, pp. 210–221, 2003.

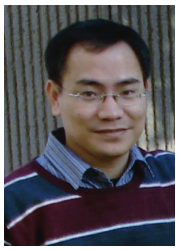
[26] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 291–303, 2002.

[27] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *the 2nd USENIX conference on Hot topics in cloud computing*. USENIX Association, 2010, pp. 4–4.

[28] M. J. Osborne, *A course in game theory*. Cambridge, Mass.: MIT Press, 1994.

[29] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[30] T. Roughgarden, *Selfish routing and the price of anarchy*. MIT press, 2005.



Xu Chen (S'10-M'12) received the B.S. degree in electronic engineering from the South China University of Technology (Guangzhou, Guangdong, China) in 2008, and the Ph.D. degree in information engineering from the Chinese University of Hong Kong (Hong Kong, China) in 2012. Dr. Chen is currently a postdoctoral research fellow in the School of Electrical, Computer and Energy Engineering, Arizona State University (Tempe, Arizona, USA). His general research interests include cognitive radio networks, wireless resource allocation, network economics, mobile social networks, and game theory. He is the recipient of the Honorable Mention Award (the first runner-up of the best paper award) in IEEE international conference on Intelligence and Security Informatics (ISI), 2010.