A Nested Two Stage Game-Based Optimization Framework in Mobile Cloud Computing System

Yanzhi Wang, Xue Lin, and Massoud Pedram
Department of Electrical Engineering
University of Southern California
Los Angeles, USA
{yanzhiwa, xuelin, pedram}@usc.edu

Abstract—**The rapidly developing cloud computing and virtualization techniques provide mobile devices with battery energy saving opportunities by allowing them to offload computation and execute applications remotely. A mobile device should judiciously decide whether to offload computation and which portion of application should be offloaded to the cloud. In this paper, we consider a mobile cloud computing (MCC) interaction system consisting of multiple mobile devices and the cloud computing facilities. We provide a nested two stage game formulation for the MCC interaction system. In the first stage, each mobile device determines the portion of its service requests for remote processing in the cloud. In the second stage, the cloud computing facilities allocate a portion of its total resources for service request processing depending on the request arrival rate from all the mobile devices. The objective of each mobile device is to minimize its power consumption as well as the service request response time. The objective of the cloud computing controller is to maximize its own profit. Based on the backward induction principle, we derive the optimal or near-optimal strategy for all the mobile devices as well as the cloud computing controller in the nested two stage game using convex optimization technique. Experimental results demonstrate the effectiveness of the proposed nested two stage game-based optimization framework on the MCC interaction system. The mobile devices can achieve simultaneous reduction in average power consumption and average service request response time, by 21.8% and 31.9%, respectively, compared with baseline methods.**

Keywords-mobile cloud computing; mobile devices; game theory; nested game; resource allocation

I. INTRODUCTION

Cloud computing has been envisioned as the next-generation computing paradigm for its advantages in on-demand service, ubiquitous network access, location dependent resource pooling, and transference of risk [1]. Cloud computing shifts the computation and storage resources from the network edges to a "Cloud" from which businesses and users are able to access applications from anywhere in the world on demand [2][3][4]. In cloud computing, the capabilities of business applications are exposed as sophisticated services that can be accessed over a network. Cloud service providers are incentivized by the profits by charging clients for accessing these services. Clients are attracted by the opportunity for reducing or eliminating costs associated with "in-house" provision of these services.

The cloud service providers own large data centers with massive computation and storage capabilities [5]. Service providers often end up over-provisioning their resources in these data centers in order to meet the clients' response time requirements or service level agreements (SLAs) [6]. Such over-provisioning may increase the cost incurred on the servers in terms of both the electrical energy cost and the carbon emission. Therefore, optimal allocation of the resources in the cloud computing system (or the broader area of distributed computing systems) is imperative in order to reduce the cost incurred on the servers as well as the environmental impact, and has been investigated in [7]-[12].

The emerging paradigm of mobile cloud computing (MCC) moves the processing, memory, and storage requirements all together from the resource limited mobile devices to the resource unlimited cloud [13]-[15]. MCC provides multiple advantages for the mobile devices [16][17], including extension of the storage capacity for mobile users and reducing the risk of data and application lost on mobile devices by backing up users' data. The potentially most important benefit for mobile users is the extension of battery operation time. The MCC paradigm helps the mobile devices to run the computation intensive applications, which may consume a large amount of battery energy when running locally in the mobile devices. This is enabled by the virtualization technique that allows the cloud to run mobile applications for the remote mobile devices [18]. This technique is referred to as *computation offloading* in the reference work [17][19].

The mobile devices should judiciously make decisions about whether to perform computation offloading and which portion of application should be offloaded to the cloud. Reference [17] provides an analysis and guideline on the conditions that computation offloading could help save the energy for mobile devices, i.e., an application or task with high computation but limited data communication

*This research is sponsored in part by the Software and Hardware Foundations program of the NSF's Directorate for Computer & Information Science & Engineering.

requirement could benefit from computation offloading. Reference [19] proposes MAUI to dynamically control the computation (code) offloading for .NET applications at runtime, formulating the computation offloading problem as a linear programming optimization problem. Reference [20] provides a similar approach for Android applications. Moreover, the mobile devices should also be aware of other devices and the potential congestion level in the remote servers if all the mobile devices decide to offload their computations simultaneously. Reference [21] provides a congestion game-based optimization framework, where each mobile device is a player and his strategy is to select one of the available servers in the cloud to offload computation. In the realistic cloud computing facilities, however, a centralized request dispatcher selects the target server for each service request (i.e., request for computation offloading) generated from the mobile devices [4][11]. The mobile devices do not select the target servers themselves.

In this paper, we consider an MCC interaction system consisting of multiple mobile devices and the cloud computing facilities. Each mobile device executes an application and generates service requests, which could either be processed locally in the mobile device or remotely in the cloud through computation offloading. The cloud computing facilities consist of multiple servers dedicated for processing mobile service requests inside a data center. Service requests from the mobile devices are free to be dispatched to any server in the cloud computing system. The total profit in the cloud computing system is the total price gained from serving the service requests, which depends on the average request response time as defined in the *utility function*, subtracted by the energy cost of the active servers.

We provide a two stage (i.e., sequential) game-based formulation [32] for the MCC interaction system. In this game, each mobile device determines the portion of its total service requests for remote processing in the cloud. The objective of each mobile device is to minimize its power consumption as well as the response time of service requests. This is the first stage of the two stage game. The cloud computing controller dispatches the service requests generated from mobile devices to each server and allocates a portion of resources in each server for service request processing. It performs resource allocation depending on the service request arrival rate from all mobile devices. The objective of the cloud computing controller is to maximize its own profit. This is the second stage of the game. Suppose that the portion of allocated resources in the cloud facilities is pre-announced to the mobile devices. Then in the first stage of the game, all the mobile devices compete for the allocated resources, which becomes a normal-form game (i.e., all the players in the game choose strategy simultaneously [32].) We prove that the Nash equilibrium always exists and is unique in this normal-form game. Nash equilibrium is the optimal strategy profile in the sense that no player can find better strategy if he deviates from the current strategy unilaterally.

According to the above discussion, the MCC interaction system is essentially a *nested two stage game* [33] since its first stage itself is a normal-form game. Based on the backward induction principle [32], we derive the optimal or near-optimal strategy for all the mobile devices as well as the cloud computing controller in the nested two stage game using convex optimization approach [31]. Experimental results demonstrate the effectiveness of the proposed nested two stage game-based optimization framework on the MCC interaction system. The mobile devices can achieve simultaneous reduction in average power consumption and average service request response time, by 21.8% and 31.9%, respectively, compared with baseline methods.

The rest of this paper is organized as follows. The MCC system model, including models for mobile devices and resource allocation in the cloud computing facilities, is presented in Section II. The nested two stage game-based formulation and optimization of the MCC system are provided in Section III and Section IV, respectively. Experimental results are presented in Section V and the conclusion is in the last section.

II. Mobile Cloud Computing System Model

We consider an MCC system (i.e., an interaction system of mobile devices and cloud computing) consisting of N mobile devices. These mobile devices such as smart phones, tablet computers are connected to the cloud through WiFi or 3G networks. Each mobile device in the MCC system is identified by a unique ID, represented by index i. Figure 1 illustrates the i-th ($1 \leq i \leq N$) mobile device. Each i-th mobile device executes an application and generates service requests, which could either be processed locally or remotely in the cloud through computation offloading.

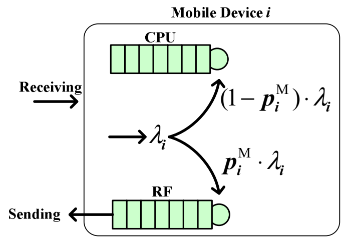
Mobile Device i
CPU
Receiving
$(1-p_i^{\mathrm{M}}) \cdot \lambda_i$
λ_i
$p_i^{\mathrm{M}} \cdot \lambda_i$
RF
Sending

Figure 1. Conceptual structure for the mobile device: local or remote processing?

In order to find an analytical form of the average response time, service requests generated from the i-th ($1 \leq i \leq N$) mobile device are assumed to follow a Poisson process with an average generating rate of λ_i, which is predicted based on the behavior of the application. The mobile device chooses to offload each service request for remote processing in the cloud with probability $p_i^{\mathbf{M}}$, where the superscript $\mathbf{M}$ stands for 'mobile'. We call $p_i^{\mathbf{M}}$ the *offloading probability* of the *i*-th mobile device. These probability values for mobile devices are the optimization variables in the MCC optimization framework. According to the properties of the Poisson distribution [29], service requests that are generated from the *i*-th mobile device and processed remotely in the cloud follow a Poisson process with an average rate of $p_i^{\mathbf{M}} \cdot \lambda_i$, called the *offloading rate*. The service requests that are generated from the *i*-th mobile device and processed locally in the device follow a Poisson process with an average rate of $\left(1 - p_i^{\mathbf{M}}\right) \cdot \lambda_i$. When $p_i^{\mathbf{M}}$ becomes larger, the average response time for the locally processed service requests decreases while the average response time for remotely processed requests increases (due to the average delay increasing in sending/receiving a service request and request processing in the cloud.) In the perspective of power consumption of the *i*-th mobile device, the power consumption in the mobile CPU (for locally processed service requests) decreases while the power consumption in the radio frequency (RF) components for sending the service requests increases. Therefore, it is crucial for each mobile device to judiciously choose the optimal $p_i^{\mathbf{M}}$ considering the characteristics of service requests (i.e., computation and data communication requirements), the anticipated offloading rate $p_{i'}^{\mathbf{M}} \cdot \lambda_{i'}$ of other mobile devices, as well as the anticipated congestion level in the servers.

Let $\mu_i^{\mathbf{M}}$ denote the average service request processing rate in the *i*-th mobile device. Then the average response time of the locally processed service requests in the *i*-th device is calculated as:

$$R_i^{\mathbf{M}}\left(p_i^{\mathbf{M}}\right) = \frac{1}{\mu_i^{\mathbf{M}} - \left(1 - p_i^{\mathbf{M}}\right) \cdot \lambda_i} \tag{1}$$

Let $\mu_i^{\mathbf{S}}$ denote the average speed in service request sending in the *i*-th mobile device, where the superscript **S** stands for 'sending'. We calculate as follows the average time for a service request to wait in the mobile device before it is completely sent out:

$$R_i^{\mathbf{S}}\left(p_i^{\mathbf{M}}\right) = \frac{1}{\mu_i^{\mathbf{S}} - p_i^{\mathbf{M}} \cdot \lambda_i} \tag{2}$$

$\mu_i^{\mathbf{S}}$ is proportional to the wireless channel capacity from the mobile device to the access point [24].

The power consumption in the *i*-th mobile device consists of two parts: (i) power consumption in the mobile CPU for local service request processing, and (ii) power consumption in the RF components (e.g., WiFi, 3G) for sending the service requests to the cloud [26][27]. Both the CPU power consumption and the RF components power consumption can be further separated into a *dynamic power consumption* part when the CPU or RF components are active (i.e., when they are processing or sending service requests) and a *static power consumption* part. The average dynamic power consumption in the CPU of the *i*-th mobile device, denoted by $P_{CPU,i}^{dyn}\left(p_i^{\mathbf{M}}\right)$, is proportional to the portion of time that the CPU is active, given by $\left(1 - p_i^{\mathbf{M}}\right) \cdot \lambda_i / \mu_i^{\mathbf{M}}$. We calculate $P_{CPU,i}^{dyn}\left(p_i^{\mathbf{M}}\right)$ as:

$$P_{CPU,i}^{dyn}\left(p_i^{\mathbf{M}}\right) = \frac{\left(1 - p_i^{\mathbf{M}}\right) \cdot \lambda_i}{\mu_i^{\mathbf{M}}} \cdot P_{CPU,i}^{dyn,max} \tag{3}$$

where $P_{CPU,i}^{dyn,max}$ is the dynamic power consumption when the mobile CPU is active. Similarly, the average dynamic power consumption in the RF components of the *i*-th mobile device is given by:

$$P_{RF,i}^{dyn}\left(p_i^{\mathbf{M}}\right) = \frac{p_i^{\mathbf{M}} \cdot \lambda_i}{\mu_i^{\mathbf{S}}} \cdot P_{RF,i}^{dyn,max} \tag{4}$$

On the other hand, the (average) static power consumptions in the CPU and the RF components of the *i*-th mobile device are constant values denoted by $P_{CPU,i}^{sta}$ and $P_{RF,i}^{sta}$, respectively. The overall power consumption in the *i*-th mobile device is given by

$$\begin{aligned} P_{Mobile,i}\left(p_i^{\mathbf{M}}\right) = P_{CPU,i}^{dyn}\left(p_i^{\mathbf{M}}\right) + P_{RF,i}^{dyn}\left(p_i^{\mathbf{M}}\right) \\ + P_{CPU,i}^{sta} + P_{RF,i}^{sta} \end{aligned} \tag{5}$$

Figure 2 shows the structure of the target resource allocation system in cloud computing with a service request pool, a data center as the service provider as well as a central resource management node. We consider homogeneous data center in this paper. The data center consists of M homogeneous servers that are dedicated for service request processing from the mobile devices (clients). We use *j* as the index of the servers in the data center.

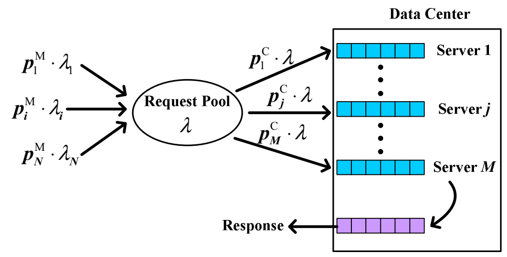

Figure 2. Conceptual structure of the resource allocation system in cloud computing.

The service request pool contains the remote service requests generated from all the mobile devices. According to the properties of the Poisson distribution [29], the total service request generating rate of the request pool, denoted by λ, is given by

$$\lambda = \sum_{i=1}^{N} p_i^{\mathbf{M}} \cdot \lambda_i \tag{6}$$

according to the properties of the Poisson distribution [29]. A service request can be dispatched to any server in the data center. The request dispatcher assigns a request to the *j*-th server with probability $p_j^{\mathbf{C}}$, where the superscript **C** stands for 'cloud'. These probability values are the optimization variables in the resource allocation optimization framework of the cloud. According to the properties of the Poisson distribution [29], mobile service requests that are dispatched to the *j*-th server follow a Poisson process with an average rate of $p_j^{\mathbf{C}} \cdot \lambda$, which is the average service request arrival rate of that server. As long as a service request is dispatched to a server, the server creates a dedicated virtual machine (VM) for that service request, loads the application executable and starts execution.

Each *j*-th server in the MCC system allocates a portion of its total resources, denoted by $\phi_j^{\mathbf{C}}$ ($0 \le \phi_j^{\mathbf{C}} \le 1$), for servicing the mobile service requests. By using the well-known formula in M/M/1 queues [30], the average processing time of the service requests dispatched to that server is calculated as

$$R_j^{\mathbf{C}}\left(p_j^{\mathbf{C}}; \phi_j^{\mathbf{C}}; \boldsymbol{p}^{\mathbf{M}}\right) = \frac{1}{\phi_j^{\mathbf{C}} \cdot \mu_j^{\mathbf{C}} - p_j^{\mathbf{C}} \cdot \lambda} \tag{7}$$

where $\mu_j^{\mathbf{C}}$ denotes the average service request processing speed when all the resources in the server are allocated for request processing. The $\mu_j^{\mathbf{C}}$ values are equal to each other since the servers are homogeneous. λ is a function of $\boldsymbol{p}^{\mathbf{M}} = \{p_1^{\mathbf{M}}, p_2^{\mathbf{M}}, \dots, p_N^{\mathbf{M}}\}$ as given in (6).

The data center sends back the response to a service request after finishing processing it. We calculate as follows the average time for the response to wait in the data center before it is completely sent out:

$$R^{\mathbf{R}}(\boldsymbol{p}^{\mathbf{M}}) = \frac{1}{\mu^{\mathbf{R}} - \lambda} \tag{8}$$

where the superscript **R** stands for 'receiving' (i.e., the mobile device receives the response from the data center.)

Therefore, the average response time of a service request generated from the *i*-th mobile device (either processed locally or remotely) is given by:

$$\begin{aligned} R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{p}^{\mathbf{C}}; \boldsymbol{\phi}^{\mathbf{C}}) &= \left(1 - p_i^{\mathbf{M}}\right) \cdot R_i^{\mathbf{M}}\left(p_i^{\mathbf{M}}\right) + p_i^{\mathbf{M}} \cdot \\ &\left(R_i^{\mathbf{S}}\left(p_i^{\mathbf{M}}\right) + \sum_{j=1}^{M} p_j^{\mathbf{C}} \cdot R_j^{\mathbf{C}}\left(p_j^{\mathbf{C}}; \phi_j^{\mathbf{C}}; \boldsymbol{p}^{\mathbf{M}}\right) + R^{\mathbf{R}}(\boldsymbol{p}^{\mathbf{M}}) \right) \end{aligned} \tag{9}$$

where $\boldsymbol{p}^{\mathbf{C}} = \{p_1^{\mathbf{C}}, p_2^{\mathbf{C}}, \dots, p_M^{\mathbf{C}}\}$ and $\boldsymbol{\phi}^{\mathbf{C}} = \{\phi_1^{\mathbf{C}}, \phi_2^{\mathbf{C}}, \dots, \phi_M^{\mathbf{C}}\}$.

Power consumption in each server consists of a *dynamic power consumption* part when the server is active (i.e., when it is processing service requests) and a *static power consumption part.* The average dynamic power consumption in each *j*-th server is proportional to the portion of time that the server is active, given by $\left(p_j^{\mathbf{C}} \cdot \lambda\right) / \left(\phi_j^{\mathbf{C}} \cdot \mu_j^{\mathbf{C}}\right)$, as well as the portion $\phi_j^{\mathbf{C}}$ of the resources that have been allocated for request processing:

$$\begin{aligned} P_{Serv,j}^{dyn}\left(p_j^{\mathbf{C}}; \boldsymbol{p}^{\mathbf{M}}\right) &= \frac{p_j^{\mathbf{C}} \cdot \lambda}{\phi_j^{\mathbf{C}} \cdot \mu_j^{\mathbf{C}}} \cdot \phi_j^{\mathbf{C}} \cdot P_{Serv,j}^{dyn,max} \\ &= \frac{p_j^{\mathbf{C}} \cdot \lambda}{\mu_j^{\mathbf{C}}} \cdot P_{Serv,j}^{dyn,max} \end{aligned} \tag{10}$$

where $P_{Serv,j}^{dyn,max}$ is the dynamic power consumption when the server is active and all resources have been allocated for service request processing. On the other hand, the (average) static power consumption in each *j*-th server in the MCC system is the sum of a constant term $\varepsilon_{Serv,j}$ and another term proportional to the portion $\phi_j^{\mathbf{C}}$ of allocated resources for request processing:

$$P_{Serv,j}^{sta}\left(\phi_j^{\mathbf{C}}\right) = \varepsilon_{Serv,j} + \phi_j^{\mathbf{C}} \cdot \left(P_{Serv,j}^{sta,max} - \varepsilon_{Serv,j}\right) \tag{11}$$

We neglect the power consumption in the data center for response sending since it is much smaller than the power consumption for request processing. Therefore, the overall power consumption of the data center is the sum of the total power consumptions of all the servers residing in it, i.e.,

$$P_{DC} = \sum_{j=1}^{M} \left(P_{Serv,j}^{dyn}\left(p_j^{\mathbf{C}}; \boldsymbol{p}^{\mathbf{M}}\right) + P_{Serv,j}^{sta}\left(\phi_j^{\mathbf{C}}\right) \right) \tag{12}$$

Let $U(R) = \beta - \gamma \cdot R$ denote the utility function of the cloud computing system with the average service request response time equal to R. Then the total profit of the cloud computing system is calculated by:

$$\begin{aligned} &\lambda \cdot \left(\beta - \gamma \cdot \sum_{j=1}^{M} p_j^{\mathbf{C}} \cdot \left(R_j^{\mathbf{C}}\left(p_j^{\mathbf{C}}; \phi_j^{\mathbf{C}}; \boldsymbol{p}^{\mathbf{M}}\right) + R^{\mathbf{R}}(\boldsymbol{p}^{\mathbf{M}}) \right) \right) \\ &\quad - price \cdot \sum_{j=1}^{M} \left(P_{Serv,j}^{dyn}\left(p_j^{\mathbf{C}}; \boldsymbol{p}^{\mathbf{M}}\right) + P_{Serv,j}^{sta}\left(\phi_j^{\mathbf{C}}\right) \right) \end{aligned} \tag{13}$$

where $price$ denotes the unit energy price.

III. Game Theoretic Problem Formulation

We consider the MCC interaction system comprised of the mobile devices and cloud computing, and provide a two stage game-based formulation [32] of the interaction system as follows. In the first stage of the game, each mobile device i determines the portion $p_i^{\mathbf{M}}$ of its total service requests for remote processing in the cloud. The objective of the mobile device is to minimize the following objective function:

$$w_1 \cdot P_{Mobile,i}\left(p_i^{\mathbf{M}}\right) + w_2 \cdot R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{p}^{\mathbf{C}}; \boldsymbol{\phi}^{\mathbf{C}}) \tag{14}$$

which is a linear combination of the mobile device's power consumption $P_{Mobile,i}\left(p_i^{\mathbf{M}}\right)$ and average request response time $R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{p}^{\mathbf{C}}; \boldsymbol{\phi}^{\mathbf{C}})$. Please note that $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$ are NOT given to the mobile device since they are determined by the cloud controller. Let $p_{-i}^{\mathbf{M}} = \{p_1^{\mathbf{M}}, p_2^{\mathbf{M}}, \dots, p_{i-1}^{\mathbf{M}}, p_{i+1}^{\mathbf{M}}, \dots, p_N^{\mathbf{M}}\}$ denote the offloading probabilities of all the mobile devices other than the i-th one. $p_{-i}^{\mathbf{M}}$ is also not given in prior to the i-th mobile device. Thus we can write $R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{p}^{\mathbf{C}}; \boldsymbol{\phi}^{\mathbf{C}}) = R_i^{\mathbf{Avg}}(p_i^{\mathbf{M}}, p_{-i}^{\mathbf{M}}; \boldsymbol{p}^{\mathbf{C}}; \boldsymbol{\phi}^{\mathbf{C}})$. The weight coefficients w_1 and w_2 do not have to be the same for a mobile device at all times. For example, when a mobile device's battery is full, it could reduce the value of w_1 because the battery energy is not a bottleneck at this time; when its battery drops under a critical level, it could increase the weight on $P_{Mobile,i}(p_i^{\mathbf{M}})$ and perhaps offload more computation.

In the second stage, the cloud computing controller dispatches the service requests to various servers and allocates a portion of its total resources for service request processing, i.e., finding the optimal $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$, depending on the request offloading rate of all mobile devices. The objective of the cloud computing controller is to maximize its own profit given by (13), which is equivalent to minimizing the following objective function:

$$\begin{aligned}\lambda \cdot \gamma \cdot \sum_{j=1}^{M} p_j^{\mathbf{C}} \cdot R_j^{\mathbf{C}}(p_j^{\mathbf{C}}; \phi_j^{\mathbf{C}} | \boldsymbol{p}^{\mathbf{M}}) + \\ price \cdot \sum_{j=1}^{M} \left(P_{Serv,j}^{dyn}(p_j^{\mathbf{C}} | \boldsymbol{p}^{\mathbf{M}}) + P_{Serv,j}^{sta}(\phi_j^{\mathbf{C}}) \right)\end{aligned} \tag{15}$$

where $R_j^{\mathbf{C}}(p_j^{\mathbf{C}}; \phi_j^{\mathbf{C}} | \boldsymbol{p}^{\mathbf{M}})$ and $P_{Serv,j}^{dyn}(p_j^{\mathbf{C}} | \boldsymbol{p}^{\mathbf{M}})$ are the functions $R_j^{\mathbf{C}}(p_j^{\mathbf{C}}; \phi_j^{\mathbf{C}}; \boldsymbol{p}^{\mathbf{M}})$ and $P_{Serv,j}^{dyn}(p_j^{\mathbf{C}}; \boldsymbol{p}^{\mathbf{M}})$ when $\boldsymbol{p}^{\mathbf{M}}$ is given, respectively.

Suppose that the resource allocation results in the cloud computing facilities, i.e., the vectors $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$, are pre-announced to the mobile devices. Then in the first stage of the game, all the mobile devices compete for the allocated resources, which becomes a normal-form game (i.e., all the players in the game choose strategy simultaneously [32].) Each i-th mobile device chooses the optimal strategy $p_i^{\mathbf{M}}$ so as to minimize the following cost function:

$$\begin{aligned} &w_1 \cdot P_{Mobile,i}(p_i^{\mathbf{M}}) + w_2 \cdot R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}} | \boldsymbol{p}^{\mathbf{C}}; \boldsymbol{\phi}^{\mathbf{C}}) \\ &= w_1 \cdot P_{Mobile,i}(p_i^{\mathbf{M}}) + w_2 \cdot R_i^{\mathbf{Avg}}(p_i^{\mathbf{M}}, p_{-i}^{\mathbf{M}} | \boldsymbol{p}^{\mathbf{C}}; \boldsymbol{\phi}^{\mathbf{C}})\end{aligned} \tag{16}$$

where $R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}} | \boldsymbol{p}^{\mathbf{C}}; \boldsymbol{\phi}^{\mathbf{C}})$ (and $R_i^{\mathbf{Avg}}(p_i^{\mathbf{M}}, p_{-i}^{\mathbf{M}} | \boldsymbol{p}^{\mathbf{C}}; \boldsymbol{\phi}^{\mathbf{C}})$) is the function $R_i^{\mathbf{Avg}}(\boldsymbol{p}^{\mathbf{M}}; \boldsymbol{p}^{\mathbf{C}}; \boldsymbol{\phi}^{\mathbf{C}})$ when $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$ are given.

Therefore, the MCC interaction system is essentially a *nested two stage game* [33] since its first stage itself is a normal-form game. We provide the optimization procedure of the nested two stage game in Section IV.

IV. GAME THEORETIC OPTIMIZATION

In this section, we provide the optimization method for the nested two stage game of the MCC system. Based on the backward induction principle, we start with the optimization procedure in the second stage, i.e., the cloud computing controller. After that, we derive the optimal strategy for the mobile devices.

A. Optimization for the Cloud Computing Controller

The cloud computing controller finds the optimal control variables $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$ in order to minimize the objective function stated in (15), when $\boldsymbol{p}^{\mathbf{M}}$ is given. We name this profit optimization problem the *Resource Allocation and Request Dispatching* (RARD) problem. The constraints of the RARD problem are:

$$0 \le p_j^{\mathbf{C}} \le 1, \qquad \text{for } \forall j \tag{17}$$

$$0 \le \phi_j^{\mathbf{C}} \le 1, \qquad \text{for } \forall j \tag{18}$$

$$\sum_{j=1}^{M} p_j^{\mathbf{C}} = 1 \tag{19}$$

$$p_j^{\mathbf{C}} \cdot \sum_{i=1}^{N} p_i^{\mathbf{M}} \cdot \lambda_i < \phi_j^{\mathbf{C}} \cdot \mu_j^{\mathbf{C}}, \qquad \text{for } \forall j \tag{20}$$

We propose the following Theorem I, which provides the optimal solution for the request dispatching phase, i.e., finding the optimal $\boldsymbol{p}^{\mathbf{C}}$, in the RARD problem. The detailed proof of Theorem I is omitted in this paper due to space limitation.

Theorem I (*Optimal Request Dispatching*): In some optimal solution of the RARD problem, the optimal request dispatching probabilities are all equal, i.e., $p_j^{\mathbf{C}} = 1/M$ for $1 \le j \le M$.

Based on Theorem I, the original RARD problem becomes the optimal resource allocation problem, i.e., finding the optimal $\boldsymbol{\phi}^{\mathbf{C}}$ to minimize the objective function (15) with given $\boldsymbol{p}^{\mathbf{C}}$ (all components in $\boldsymbol{p}^{\mathbf{C}}$ are equal to $1/M$.) The constraints of the optimal resource allocation problem are (18) and (20).

This problem is a convex optimization problem since the objective function (15) is a convex function of $\boldsymbol{\phi}^{\mathbf{C}}$ when $\boldsymbol{p}^{\mathbf{C}}$ is given, and constraints (18) and (20) are linear inequality constraints. It can be solved optimally with polynomial time complexity using standard convex optimization techniques [31]. We have the following corollary about the optimal solution of the optimal resource allocation problem.

Corollary I (*Optimal Resource Allocation*): In the optimal solution of the resource allocation problem, the components in the optimal optimization variable vector $\boldsymbol{\phi}^{\mathbf{C}}$

are all equal to each other. This conclusion also applies to the original RARD problem.

Proof: When $\boldsymbol{p}^{\mathbf{C}}$ is given, the optimal resource allocation problem can be separated into a set of M local optimization problems, one for each server. Each local optimization problem finds the optimal $\phi_j^{\mathbf{C}}$ value such that the following objective function is minimized:

$$\lambda \cdot \gamma \cdot p_j^{\mathbf{C}} \cdot R_j^{\mathbf{C}}\left(p_j^{\mathbf{C}}; \phi_j^{\mathbf{C}} \middle| \boldsymbol{p}^{\mathbf{M}}\right) + price \cdot \left(P_{Serv,j}^{dyn}\left(p_j^{\mathbf{C}} \middle| \boldsymbol{p}^{\mathbf{M}}\right) + P_{Serv,j}^{sta}\left(\phi_j^{\mathbf{C}}\right)\right) \tag{21}$$

The constraints are $0 \leq \phi_j^{\mathbf{C}} \leq 1$ and $p_j^{\mathbf{C}} \cdot \sum_{i=1}^{N} p_i^{\mathbf{M}} \cdot \lambda_i < \phi_j^{\mathbf{C}} \cdot \mu_j^{\mathbf{C}}$. The local optimization problems for different servers are exactly the same since (i) the $p_j^{\mathbf{C}}$ values are equal to each other and (ii) the servers are homogeneous. Hence the corollary is proved. ■

Based on Corollary I, we only need to find the optimal solution of the above-mentioned local optimization problem for only one server. We effectively reduce the computation complexity in solving the RARD problem through this observation.

B. Optimization for the Mobile Devices

Suppose that the portion of allocated resources in the cloud facilities is pre-announced to the mobile devices, i.e., $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$ are given in prior. Then in the first stage of the nested two stage game, each mobile device i determines the portion $p_i^{\mathbf{M}}$ ($0 \leq p_i^{\mathbf{M}} \leq 1$) of service requests for remote processing, in order to minimize the objective function (16). The constraints on $p_i^{\mathbf{M}}$ are given as follows:

$$0 \leq p_i^{\mathbf{M}} \leq 1, \qquad \text{for } \forall i \tag{22}$$

$$\left(1 - p_i^{\mathbf{M}}\right) \cdot \lambda_i \leq \mu_i^{\mathbf{M}} - \varepsilon, \qquad \text{for } \forall i \tag{23}$$

$$p_i^{\mathbf{M}} \cdot \lambda_i \leq \mu_i^{\mathbf{S}} - \varepsilon, \qquad \text{for } \forall i \tag{24}$$

$$p_j^{\mathbf{C}} \cdot \sum_{i=1}^{N} p_i^{\mathbf{M}} \cdot \lambda_i \leq \phi_j^{\mathbf{C}} \cdot \mu_j^{\mathbf{C}} - \varepsilon, \qquad \text{for } \forall j \tag{25}$$

$$\sum_{i=1}^{N} p_i^{\mathbf{M}} \cdot \lambda_i \leq \mu^{\mathbf{R}} - \varepsilon \tag{26}$$

where $\varepsilon \ll 1$ is a small positive number, which is incorporated to make the domain of $\boldsymbol{p}^{\mathbf{M}}$ a closed set. Constraints (23), (24), (25), and (26) are derived from equations (1), (2), (7), and (8), respectively. This distributed optimization problem is essentially a normal-form game in which every player (i.e., mobile device) chooses its strategy $p_i^{\mathbf{M}}$ simultaneously in order to minimize (16). We name the game the *Offloading Probability Decision* (OPD) game for each mobile device.

As the mobile devices are considered to be non-cooperative among each other, we are interested in the existence and uniqueness of the Nash equilibrium [32]. Nash equilibrium is the optimal strategy profile for all the players in the sense that no player can find better strategy if he deviates from the current strategy unilaterally. In other words, no player (mobile device) will have incentive to leave this strategy. Therefore, the Nash equilibrium is of particular interest to a non-cooperative normal-form game. We prove the existence and uniqueness of the Nash equilibrium in the OPD game.

Theorem II (*Nash Equilibrium in the OPD Game*): The Nash equilibrium in the OPD game exists and is unique.

Proof: The OPD game is a strictly concave *n*-player game [22][23] since (i) we minimize a convex objective function (16) for each player i, which is equivalent to maximizing a concave payoff function for each player, and (ii) the domain of the strategy profile $\boldsymbol{p}^{\mathbf{M}}$, which is constrained by (22) – (26), is a closed convex set. In this case, the existence and uniqueness of the Nash equilibrium are directly resulted from the first and third theorem in [23].

Each mobile device finds the Nash equilibrium of the corresponding OPD game using standard convex optimization technique [31], with detailed procedure shown in Algorithm 1.

Algorithm 1: Finding the Nash Equilibrium in the OPD Game for Each Mobile Device *i*.

```
Initialize $p_i^{\mathbf{M}}$ (the offloading probability of the $i$-th mobile device itself) as well as $p_{-i}^{\mathbf{M}}$ (the anticipation of the offloading probabilities of other mobile devices.)
Do the following procedure iteratively:
    For each $1 \leq i' \leq N$:
        Find the optimal $p_{i'}^{\mathbf{M}}$ (i.e., the best response of the $i'$-th mobile device) with respect to $p_{-i'}^{\mathbf{M}}$, by solving the convex optimization problem for the $i'$-th mobile device with objective function (16) and constraints (22) – (26).
        Update $p_{i'}^{\mathbf{M}}$ to be the new value.
    End
Until the solution converges.
```

In the MCC system, however, $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$ are not given in prior to the mobile devices since these values are determined by the cloud computing controller. Based on the backward induction principle in sequential games [32], each mobile device optimizes its offloading probability $p_i^{\mathbf{M}}$ based on an anticipation of $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$. We provide an iterative algorithm, stated in Algorithm 2, in order to find the optimal strategy for each mobile device in the nested two stage game. In each iteration, Algorithm 2 has a *Nash equilibrium finding* phase and a *resource allocation anticipation updating* phase. In the Nash equilibrium finding phase, the mobile device runs Algorithm 1 to find the Nash equilibrium of the corresponding OPD game with anticipated resource allocation results in the cloud computing facilities. In the resource allocation anticipation updating phase, the mobile device updates the anticipation of $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$ based on the updated Nash equilibrium. An optimal or near-optimal strategy for each mobile device in the nested two stage game can be derived by executing Algorithm 2.

Algorithm 2: Deriving an Optimal or Near-Optimal Strategy for Each Mobile Device in the Nested Two Stage Game.

```
Initialize the anticipation of $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$.
Do the following procedure iteratively:
    Finding the Nash equilibrium: Find the Nash equilibrium of the OPD game based on the anticipation of $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$, by executing Algorithm 1.
    Updating the anticipation of resource allocation results: Find and update the anticipation of $\boldsymbol{p}^{\mathbf{C}}$ and $\boldsymbol{\phi}^{\mathbf{C}}$, by solving the RARD problem based on $\boldsymbol{p}^{\mathbf{M}}$ obtained from the Nash equilibrium.
Until the solution converges.
```

V. Experimental Results

In this section, we implement the interaction system of multiple mobile devices and the cloud computing facilities and demonstrate the effectiveness of the proposed game theory-based optimization framework.

We consider an MCC interaction system comprised of $N = 20$ (we will change this parameter later) mobile devices, as well as a cloud computing infrastructure. The data center in the cloud computing system consists of 10 servers. We use normalized amounts of most of the parameters in the MCC interaction system instead of their real values. The service request generating rate λ_i of each mobile device is a uniformly distributed random variable between 1 and 1.5. The average service request processing rate $\mu_i^{\mathbf{M}}$ in the mobile CPU is 1.6. The average service request sending rate $\mu_i^{\mathbf{S}}$ in every mobile device is 2. The maximum dynamic power consumption values in each mobile CPU and RF components, $P_{CPU,i}^{dyn,max}$ and $P_{RF,i}^{dyn,max}$, are uniformly distributed between 4 and 6, and between 1 and 1.5, respectively. The static power consumption values in each mobile CPU and RF components, $P_{CPU,i}^{sta}$ and $P_{RF,i}^{sta}$, are uniformly distributed between 2 and 3, and between 1 and 1.5, respectively. In the cloud computing system, the maximum average service request processing rate $\mu_j^{\mathbf{C}}$ in each homogenous server (i.e., when all its resources are allocated for request processing) is 3. The maximum dynamic power consumption $P_{Serv,j}^{dyn,max}$ and maximum static power consumption $P_{Serv,j}^{sta,max}$ of each server are 10 and 5, respectively. The average response sending rate $\mu^{\mathbf{R}}$ in the cloud computing system is 50. For the utility function in the cloud computing system, parameter β is set to 5 and parameter γ is 1. The unit energy price $price$ is 0.2.

In the first experiment, we test on the MCC interaction system and compare the average power consumption and the average request response time of all the mobile devices using the nested two stage game-based optimization framework and using the two baseline methods. In the first baseline system, service requests generated from all the mobile devices are processed locally in the mobile device. In the second baseline system, the mobile devices send all of their generated service requests to the cloud computing system for remote processing, and then the cloud computing controller performs optimal request dispatching and resource allocation based on the total service request arrival rate as described in Section IV.A. Figure 3 illustrates the tradeoff between the normalized average power consumption and the normalized average service request response time, obtained by changing the weight coefficients w_1 and w_2 simultaneously for each mobile device. Figure 3 also illustrates the normalized average power consumption and average service request response time of the two baseline systems.

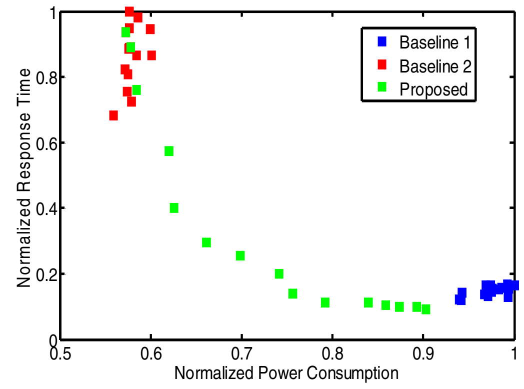

Figure 3. The normalized average power consumption versus normalized average service request response time of all the mobile devices, using the proposed game theory-based optimization framework and the baseline methods.

We observe from Figure 3 that the mobile devices can achieve simultaneous reduction in average power consumption and average service request response time, by 21.8% and 31.9%, respectively, when using the proposed game theory-based optimization framework compared with Baseline 1. This is because the mobile devices have high power consumption in the mobile CPU and large service request response time due to congestion in request processing, if all the service requests are processed locally. On the other hand, the mobile devices can achieve significant average service request response time reduction, by up to 89%, compared with Baseline 2. However, the mobile devices cannot achieve reduction in average power consumption compared with Baseline 2. This is a natural result since offloading all the service requests for remote processing turns out to be the most energy-efficient policy for the mobile devices, although it may often incur unbearable delay.

For more comparisons, we compare the average values of the objective function (14) in each mobile device. This objective function shows a desirable tradeoff of power consumption and service request response time in the mobile devices. The data are collected in the above experiment. We show in Table I the minimum and maximum reduction in the average values of the objective function (14) when employing the game theoretic optimization framework compared with the baseline methods. We can observe that the maximum reduction in average objective function value is up to 89.5%, demonstrating the effectiveness of the nested two stage game-based optimization framework.

TABLE I. REDUCTION OF THE AVERAGE OBJECTIVE FUNCTION VALUE USING THE PROPOSED GAME THEORY-BASED OPTIMIZATION FRAMEWORK

Minimum reduction over Baseline 1	Maximum reduction over Baseline 1
15.4%	41.2%

Minimum reduction over Baseline 2	Maximum reduction over Baseline 2
0%	89.5%

In the second experiment, we fix the weight coefficients w_1 and w_2 to be 5 and 1, respectively, whereas we change the number N of mobile devices in the MCC interaction system. We test on the MCC interaction system under the nested two stage game-based optimization framework, and compare the normalized average power consumption and the normalized average service request response time of each mobile device with respect to the number N. Figure 4 illustrates the results of this experiment. We can observe from Figure 4 that both the average power consumption and the average service request response time increase with the increase of N. This is because of the increasing in the congestion level and therefore the increasing in the average service request response time in the data center. The mobile devices are aware of this congestion and assign more service requests for local processing, which in turn increases their power consumption levels.

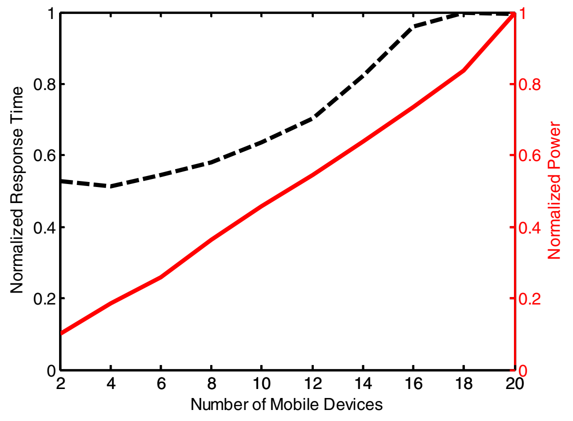

Figure 4. The normalized average power consumption and normalized average response time versus the number of mobile devices using the proposed game theory-based optimization framework.

VI. CONCLUSION

Cloud computing and virtualization techniques provide mobile devices with battery energy saving opportunities by allowing them to offload computation and execute applications remotely. In this paper, we consider an MCC interaction system consisting of multiple mobile devices and the cloud computing facilities. We provide a nested two stage game formulation for the MCC interaction system. In the first stage, each mobile device determines the portion of its service requests for remote processing in the cloud. In the second stage, the cloud computing facilities allocate a portion of its total resources for service request processing depending on the request arrival rate from all the mobile devices. The objective of each mobile device is to minimize its power consumption as well as the service request response time. The objective of the cloud computing controller is to maximize its own profit. Based on the backward induction principle, we derive the optimal or near-optimal strategies for all the mobile devices as well as the cloud computing controller in the nested two stage game using convex optimization approach. Experimental results demonstrate the effectiveness of the proposed nested two stage game-based optimization framework on the MCC interaction system. The mobile devices can achieve simultaneous reduction in average power consumption and average service request response time, by 21.8% and 31.9%, respectively, compared with baseline methods.

REFERENCES

[1] B. Hayes, "Cloud Computing," *Communications of the ACM*, 2008.

[2] R. Buyya, "Market-oriented cloud computing: vision, hype, and reality of delivering computing as the 5th utility," *Proc. of CCGrid*, 2009.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Pabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, 2010.

[4] M. Pedram, "Energy-Efficient Datacenters," *IEEE Trans. on CAD*, 2012.

[5] R. H. Katz, "Tech Titans Building Boom," *IEEE Spectrum*, 2009.

[6] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," *IEEE Computer*, 2007.

[7] K. Krauter, R. Buyya, and M. Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing," *Software Practice and Experience*, 2002.

[8] L. Zhang and D. Ardagna, "SLA based profit optimization in autonomic computing systems," in *2nd Int. Conf. on Service Oriented Computing*, 2004.

[9] D. Ardagna, M. Trubian, and L. Zhang, "SLA based resource allocation policies in autonomic environments," *Journal of Parallel and Distributed Computing*, 2007.

[10] A. Chandra, W. Gongt, and P. Shenoy, "Dynamic resource allocation for shared clusters using online measurements," *International Conference on Measurement and Modeling of Computer Systems* (SIGMETRICS), 2003.

[11] H. Goudarzi and M. Pedram, "Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems," *IEEE Cloud*, 2011.

[12] Y. Wang, S. Chen, H. Goudarzi, and M. Pedram, "Resource allocation and consolidation in a multi-core server cluster using a Markov decision process model," *Proc. of ISQED*, 2013.

[13] H. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," in *Wireless Commun. and Mobile Comput.*, 2011.

[14] A. Khan and K. Ahirwar, "Mobile cloud computing as a future of mobile multimedia database," in *International Journal of Computer Science and Communication*, 2011.

[15] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. of the 2nd USENIX Conference on Hot Topics in Cloud Computing*, 2010.

[16] http://aws.amazon.com/s3/.

[17] K. Kumar and Y. Lu, "Cloud computing for mobile users: can offloading computation save energy?" in *IEEE Computer*, 2010.

[18] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in Proceedings of the 19th ACM Symposium on Operating System Principles (SOSP), 2003.

[19] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making smartphones last longer with code offload," in *Proc. of International Conference on Mobile Systems, Applications, and Services*, 2010.

[20] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: elastic execution between mobile device and cloud," in *Proc. of EuroSys*, 2011.

[21] Y. Ge, Y. Zhang, Q. Qiu, and Y. Lu, "A game theoretic resource allocation for overall energy minimization in mobile cloud computing system," in *Proc. of ISLPED*, 2012.

[22] H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, and R. Schober, "Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid", in *Proc. of IEEE PES ISGT*, 2010.

[23] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave *n*-person games," *Econometrica*, vol. 33, pp. 347-351, 1965.

[24] D. Tse and P. Viswanath, *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.

[25] I. Hwang, T. Kam, and M. Pedram, "A study of the effectiveness of CPU consolidation in a virtualized multi-core server system," in *Proc. of ISLPED*, 2012.

[26] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. Dick, Z. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proc. of Int. Conf. on HW/SW Codesign and System Synthesis* (CODES+ISSS), 2010.

[27] D. Shin, W. Lee, K. Kim, Y. Wang, Q. Xie, M. Pedram, and N. Chang, "Online estimation of the remaining energy capacity in mobile systems considering system-wide power consumption and battery characteristics," to appear in *Proc. of Asia South Pacific Design Automation Conference* (ASP-DAC), 2013.

[28] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta, "VL2: a scalable and flexible data center network," in *Proc. of the ACM SIGCOMM*, 2009.

[29] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 3rd edition, 1991.

[30] L. Kleinrock, *Queueing Systems, Volume I: Theory*, Wiley, 1975.

[31] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[32] K. Leyton-Brown and Y. Shoham, *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*, Morgan & Claypool Publishers, 2008.

[33] G. Tsebelis, *Nested Games: Rational Choice in Comparative Politics*, University of California Press, 1990.