Efficient Distributed Threshold-based Offloading for Large-Scale Mobile Cloud Computing

Xudong Qin, *Student Member, IEEE,* Bin Li, *Senior Member, IEEE,* and Lei Ying, *Fellow, IEEE*

***Abstract*—Mobile cloud computing enables compute-limited mobile devices to perform real-time intensive computations such as speech recognition or object detection by leveraging powerful cloud servers. An important problem in large-scale mobile cloud computing is computational offloading, where each mobile device decides when and how much computation should be uploaded to cloud servers by considering the local processing delay and the cost of using cloud servers. In this paper, we develop a distributed threshold-based offloading algorithm where it uploads an incoming computing task to cloud servers if the number of tasks queued at the device reaches the threshold and processes it locally otherwise. The threshold is updated iteratively based on the computational load and the cost of using cloud servers. We formulate the problem as a symmetric game, and characterize the sufficient and necessary conditions for the existence and uniqueness of the Nash Equilibrium (NE) assuming exponential service times. Then, we show the convergence of our proposed distributed algorithm to the NE when the NE exists. Further, we characterize the performance gap between cost under our proposed distributed algorithm and the minimum cost in terms of Price of Anarchy (PoA) when the cost of using cloud servers is high. Finally, we perform extensive simulations to validate our theoretical findings, demonstrate the efficiency of our proposed distributed algorithm under various scenarios such as hyperexponential service times, imperfect server utilization estimation, and asynchronous threshold updates, and reveal the superior performance of threshold-based policies over their probabilistic counterpart.**

***Index Terms*—Mobile Cloud Computing, Distributed Offloading, Nash Equilibrium, Price of Anarchy, Convergence.**

I. Introduction

REAL-TIME mobile cloud applications (see [1], [2]) have grown rapidly over the last few years and have become ubiquitous. For example, in an international trade show such as Consumer Electronics Show, people in the same convention center may need real-time translation services on their mobile devices at the same time, making it challenging to provide low latency language translation with a low service cost. On the one hand, computing limited devices may not have the required computational capability to process the data locally; and on the other hand, offloading the computing tasks to a cloud-computing center incurs both communication and computing costs. Mobile cloud computing, which utilizes both mobile and cloud computing powers, is a vital solution to address this challenge. A central question in mobile cloud computing is: how much to offload and when? This paper addresses this important question and proposes a distributed offloading algorithm where each device aims at minimizing a cost function, including both the local processing delay and offloading cost at the cloud computing center.

Mobile cloud computing has received significant research interest in recent years (see [3], [4] for the most recent thorough surveys). Much of prior works considered the static model in various application scenarios, where all computation demands and their required computing time are known at the beginning of the system operation. For example, [5] considered the cost as a weighted sum of average energy consumption and computation time, and developed an iterative algorithm that minimizes the cost. [6] formulated a multi-objective optimization problem with the goal of minimizing CPU usage, memory overhead, energy consumption, and execution time, and proposed an efficient heuristic algorithm. [7] jointly optimized both energy and latency for mobile edge computing in Internet of Things applications. [8] considered minimizing the weighted sum of energy consumption and task computation time as their objective function and proposed an algorithm that determined both task offloading decisions and CPU frequency. Some other works focused on the distributed offloading design for mobile cloud/edge computing based on game-theoretical approaches (e.g., [9], [10], [11], [12], [13]).

Recent works (e.g., [14], [15]) considered the dynamic model, where computing tasks dynamically arrive at mobile devices and are served by either themselves or edge servers. However, they considered the scenario where all mobile devices share the same wireless networks and compete for wireless transmissions. This is not the case for large-scale mobile cloud computing, where each mobile device may use a different wireless network and need to pay a certain amount of cost for using the cloud servers. Despite some works considering the dynamic models for large-scale mobile cloud computing, they focused on the class of probabilistic offloading policies which offload the computing tasks with a certain probability. For example, [16] considered the offloading design in satellite edge computing and proposed an iterative algorithm that achieves the Nash Equilibrium of the probabilistic offloading strategies. [17] studied mobile edge computing in 5G networks and adopted deep learning techniques to obtain the optimal offloading probability. [18] analyzed the computation latency of probabilistic offloading in

An earlier version of this paper has appeared in the IEEE International Conference on Computer Communications (INFOCOM), 10-13 May 2021, Virtual Conference.

Xudong Qin (xfq5024@psu.edu) and Bin Li (binli@psu.edu) are with the Department of Electrical Engineering, The Pennsylvania State University, State College, PA 16802 USA. Lei Ying (leiying@umich.edu) is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA.

The work of Bin Li has been supported in part by NSF under the grants CNS-2152657 and CNS-2152658. The work of Lei Ying has been supported in part by NSF under the grants CNS-2002608 and CNS-2001687.

mobile edge computing systems under stochastic task arrival and departure processes. [19] focused on jointly minimizing energy consumption, execution delay, and price cost, and designed an algorithm based on the interior point method to find the optimal offloading probability.

In this paper, we focus on the class of threshold-based offloading policies under which each mobile device uploads an incoming computing task to cloud servers if its number of awaiting tasks is greater than a certain threshold and processes it by itself otherwise. This is motivated by the following three facts: (i) threshold-based policies exhibit distributed nature and are suitable for large-scale mobile cloud computing; (ii) the threshold-based policy has been proved to be optimal in some related models (e.g., [20]); (iii) threshold-based policies outperform the existing state-independent probabilistic offloading policies (i.e., uploading an incoming task with a certain probability and processing it locally otherwise) in our considered system, as demonstrated via simulations (cf. Section IV-D). While some prior works (e.g., [21]) considered the threshold-based offloading design, the service cost is independent of the threshold decisions. This is not the case in our considered setup since the service cost depends on the server utilization relying on the users' threshold decisions.

For the class of threshold-based policies, the following five fundamental questions naturally arise:

(i) **Algorithm Design**: How should a device adapt its threshold to minimize its cost?
(ii) **Existence**: Does there exist an equilibrium point such that each device will settle on a threshold and have no incentive to deviate from it?
(iii) **Uniqueness**: Is the equilibrium point unique?
(iv) **Convergence**: Does the system converge to the equilibrium when each device adapts its threshold to minimize its own cost?
(v) **Efficiency (or Price of Anarchy)** How efficient is the distributed threshold-based policy compared with the optimal centralized solution?

There are two main challenges to answer the questions above: (i) Since each user's offloading decision (i.e., threshold) is discrete and unbounded, classical fixed point theorems, which have been used successfully for proving the existence and uniqueness of Nash Equilibrium in many applications (e.g., [22], [23], [24], [25], [26], [27]), do not directly apply in our model; (ii) Since thresholds take integer values, it is challenging to show that the integer sequences will converge to the equilibrium point. In fact, it is *not* clear whether a distributed threshold-based algorithm, where each device chooses the optimal threshold given the current state of the cloud service, converges. However, we are able to show that an incremental distributed threshold-based policy, where each device increases/decreases its threshold to move it closer to the current optimal threshold, converges under some minor conditions.

The main results and contributions are listed below:

• Under the exponential service time assumption, we analytically characterize the sufficient and necessary conditions for the existence and uniqueness of the Nash Equilibrium (see Theorem 1).

• We develop a distributed implementation of the threshold-based offloading algorithm (see Section III-A) so that each user iteratively and incrementally updates its own threshold based on its own cost function. We prove the convergence of our proposed algorithm to the Nash Equilibrium offloading decision if it exists under the exponential service time distribution (see Theorem 2).

• We characterize the efficiency of the Nash Equilibrium offloading decision via the Price of Anarchy (PoA), capturing efficiency loss compared with the optimal centralized offloading (cf. Theorem 3).

• We perform extensive simulations (see Section IV) to validate our theoretical findings. Under various scenarios (e.g., hyperexponential service time distributions, transmission delay when offloads, imperfect server utilization estimation, and asynchronous threshold updates), we also demonstrate the convergence of our proposed distributed algorithm to the Nash Equilibrium offloading decision, which is computed via numerical calculations. We further reveal the superior performance of threshold-based policies over their probabilistic counterpart.

This work extends our previous work [28] in the following aspects: (i) we analytically characterize the PoA performance when the cost of using cloud servers is large and validate it via simulations; (ii) more detailed proofs for Theorem 1 and Theorem 3 are included; (iii) we add additional simulations to demonstrate the superior performance of threshold-based offloading algorithms over the existing probabilistic offloading algorithms in our considered large-scale mobile cloud computing system.

The remainder of this paper is organized as follows: we introduce our system model in Section II. In Section III, we propose a distributed threshold-based offloading algorithm and present our main theoretical results. In Section IV-B, we perform extensive simulations to validate our theoretical findings as well as the efficiency of our proposed distributed algorithm. Section V concludes our paper.

II. System Model

We consider a mobile cloud computing system of N users, as shown in Fig. 1. Tasks arrive at each user according to a Poisson process of rate $\lambda > 0$. Each user can process a task either locally or upload it to cloud servers with a total service rate of Nc, where $c > \lambda$ ensures that all tasks can be processed at cloud servers if necessary[1]. The mean service time of a task at a local device is $1/\mu$, where $\mu > 0$. We assume that each user n $(n = 1, 2, \ldots, N)$ maintains a queue to hold tasks awaiting processing locally, and we use $q_n(t)$ to denote the queue length at time t, i.e., the number of awaiting tasks of user n at time t.

For each incoming task, it experiences both queueing and processing delays when being processed locally. We assume

[1]In this paper, we do not consider the detailed modeling of cloud computing, such as virtual machines. From the cloud service provider's perspective, its goal is to ensure that the processing delay of each computing task is small, and thus we neglect the processing delay in our theoretical model. However, in our simulations, we add the latency, including both communication latency and processing latency, and demonstrate that our proposed algorithm still performs well.

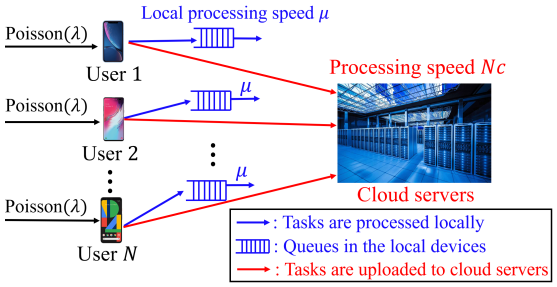


Fig. 1: System Model

the user will be charged with a service cost based on *server utilization* at the time if a task is offloaded to the cloud, where server utilization is the current load of the cloud servers. Here, we assume that cloud servers are high-capacity servers so that the cloud processing time is negligible compared with local processing and queueing delays. Under these assumptions, each individual user makes an *offloading decision* that determines whether its incoming task is processed by itself or is uploaded to cloud servers with the goal of minimizing both its delay and service cost.

Since the offloading problem shares a similar spirit of the optimal admission control of a single queue whose solution has a threshold-based structure (see [20]), we focus on the following Threshold-Based Offloading (TBO) Policy.

Threshold-Based Offloading (TBO) Policy with integer parameters $\mathbf{B} \triangleq (B_n)_{n=1}^N$: For each user n, an incoming computing task will be processed by itself if its current queue length $q_n(t)$ is less than B_n. Otherwise, the task is uploaded to cloud servers for computation.

In the TBO policy, B_n is the *threshold* of user n. If B_n is set to 0, then user n will upload all its incoming tasks to cloud servers. If $B_n \uparrow \infty$, then all computing tasks are processed by user n. Under the TBO policy, the queue length of each user only depends on its own threshold B_n when the threshold is fixed. Therefore, we use $Q(B_n)$ to denote the average queue length and $\pi(B_n)$ to denote probability that an incoming task is uploaded to cloud servers (also referred to as *offloading probability*). For each user n, an incoming task is processed by itself with probability $1-\pi(B_n)$. In such a case, it experiences the average delay of $\frac{Q(B_n)}{\lambda(1-\pi(B_n))}$ by Little's Law, where we use the fact that the average rate of tasks processed by user n is $\lambda(1-\pi(B_n))$. With offloading probability $\pi(B_n)$, an incoming task is uploaded to cloud servers and experiences a cost depending on the server utilization, i.e., $g(\beta(\mathbf{B}))$, where $\mathbf{B} \triangleq (B_n)_{n=1}^N$, $\beta(\mathbf{B}) \triangleq \lambda\sum_{n=1}^N \pi(B_n)/(Nc)$ is the utilization of cloud servers, $\lambda\sum_{n=1}^N \pi(B_n)$ is the average number of tasks that are uploaded to cloud servers, and $g(\cdot)$ is some convex, non-decreasing, and non-negative function. This is motivated by the fact that a large server utilization results in a high service cost in cloud services (see [29]) and the fact that the cloud service provider would like to ensure a low processing latency and thus charge a much higher cost for high server utilization. In the rest of the paper, we assume that $g(x) = kx^2$, where $k > 0$ is some scaling parameter to take the degree of importance between local processing cost and cloud service cost into consideration. The large the parameter k, the higher the cloud service cost. When $k \to \infty$, the cost of using cloud servers becomes extremely large, and all the users in the system tend to use local devices for computation if the local device can handle the computation traffic. Therefore, the average cost[2] of user n can be expressed as

$$\frac{Q(B_n)}{\lambda} + k\beta^2(\mathbf{B})\pi(B_n). \tag{1}$$

In this paper, we focus on the *large-scale* mobile cloud computing system (i.e., N is large enough). Our goal is to develop a *distributed offloading algorithm* under which each device updates its own threshold, without knowing other users' thresholds, to minimize its cost function. The important questions to answer whether such an algorithm can converge? If it does, where does it converges to, and how efficient is the equilibrium point? We study this problem from a game perspective. In particular, each user n optimizes its own cost function $Q(B_n)/\lambda+k\beta^2\pi(B_n)$ given a fixed server utilization β. $\widetilde{\mathbf{B}} \triangleq (\widetilde{B}_n)_{n=1}^N$ is defined to be the *Nash Equilibrium (NE)* of the system (when it exists) if

$$\widetilde{B}_n \in \arg\min_B Q(B)/\lambda + k\beta^2\pi(B) \tag{2}$$
$$\text{and } \beta = \lambda\sum_{n=1}^N \pi(\widetilde{B}_n)/(Nc).$$

Note the cost function in (2) is different from (1) because the N-player game defined by (1) is difficult to solve, so we approach the problem using a mean-field approximation (or large-system approach) where we assume that each user's choice of the threshold has the minimal impact on the server utilization β, so each user views the server utilization as a fixed constant when optimizing its threshold. The NE has to satisfy two conditions: (i) the threshold is optimal given the server utilization (*optimality condition*) (ii) the server utilization is indeed the one under the chosen thresholds from all users (*consistency condition*).

We define the *Price of Anarchy (PoA)* to be the performance gap between cost under the NE offloading decision and the global minimum cost, i.e.,

$$\text{PoA} \triangleq 1 - \frac{\text{Global minimum cost}}{\text{Average cost under NE offloading decision}}.$$

Note that PoA $\in [0, 1]$. The smaller the PoA, the more efficient the system under the NE offloading decision.

III. Algorithm Design and Main Results

In this section, we first propose a distributed offloading algorithm that incrementally updates the threshold for each user. Then, we present our main theoretical results on the performance of the proposed algorithm.

[2]The user's cost is defined as the weighted sum of local processing latency and cost of using cloud servers, i.e., cost = constant $\times$ latency + service cost, where the unit for the constant can be dollar/second. This is equivalent to minimizing the cost (= latency + constant $\times$ service cost), and our parameter k serves as the constant in this case.

A. Algorithm Description

In this subsection, we introduce an Iterative Threshold Update (ITU) algorithm that constantly updates each user's threshold. Let $B_n^{(m)}$ be the threshold of user n in the m^{th} iteration. Motivated by the fact that the server utilization asymptotically equals to $\beta(\mathbf{B}^{(m)})$ as $N \to \infty$ at the beginning of the $(m+1)^{th}$ iteration, we define the *approximate average cost* of user n given the server utility $\beta(\mathbf{B}^{(m)})$ in the m^{th} iteration as

$$T_n\left(B_n; \mathbf{B}^{(m)}\right) \triangleq \frac{Q(B_n)}{\lambda} + k\beta^2\left(\mathbf{B}^{(m)}\right)\pi(B_n),$$

where $\mathbf{B}^{(m)} \triangleq \left(B_n^{(m)}\right)_{n=1}^{N}$.

Algorithm 1 Iterative Threshold Update (ITU) Algorithm

1: Each user starts from some random threshold $B_n^{(0)}$, where $n = 1, 2, \cdots, N$;
2: **for** $m = 0, 1, 2 \cdots$, **do**
3: **for** $n = 1, 2, \cdots, N$ **do**
4:
$$\widehat{B}_n^{(m+1)} \in \arg\min_{B_n} T_n\left(B_n; \mathbf{B}^{(m)}\right). \quad (3)$$
5: **if** $m = 0$ **then**
6: $B_n^{(m)} \leftarrow \widehat{B}_n^{(m+1)}$
7: **else**
8: **if** $\widehat{B}_n^{(m+1)} < B_n^{(m)}$ **then**
9: $B_n^{(m+1)} \leftarrow B_n^{(m)} - 1$;
10: **else if** $\widehat{B}_n^{(m+1)} > B_n^{(m)}$ **then**
11: $B_n^{(m+1)} \leftarrow B_n^{(m)} + 1$;
12: **else**
13: $B_n^{(m+1)} \leftarrow B_n^{(m)}$.
14: **end if**
15: **end if**
16: **end for**
17: **end for**

We describe our proposed ITU algorithm in Algorithm 1, where each user greedily optimizes its own decision in the first iteration step to speed up the convergence of the ITU algorithm and then gradually adjusts its threshold. Here, the optimal solution to (3) requires the knowledge of the server utilization, which relies on all users' offloading decisions and thus is typically unavailable beforehand. However, the server utilization can be estimated via the ratio of the average offloading rate (i.e., the ratio of the total number of offloaded tasks and the total amount of time) to the total service rate of cloud servers. After we have the server utilization, we use a look-up table to solve the optimization problem (3). Moreover, users in the system may update their offloading decisions asynchronously. In Section IV, we demonstrate via simulations that our proposed ITU algorithm still performs well in the presence of imperfect server utilization estimation and asynchronous threshold updates.

We are interested in whether the proposed ITU algorithm converges and which offloading decisions it converges to if it does converge. We analytically answer these two questions when the service time of each task is independently and identically distributed (i.i.d) and exponentially distributed with mean $1/\mu$. In such a case, when the threshold B_n of user n is fixed, the queue at a device is an $M/M/1/B_n$ queue (see [30]), which has a Poisson arrival process with rate λ, exponentially distributed service time with mean $1/\mu$, and a finite buffer size B_n. We can easily calculate $\pi(B_n)$ and $Q(B_n)$ by using the detailed balance equation of the underlying Markov Chain. Therefore, the average queue length $Q(B_n)$ and probability $\pi(B_n)$ that an incoming task is uploaded to cloud servers (also referred to as *offloading probability*) have the following closed-forms:

$$Q(B_n) = \begin{cases} \frac{B_n+1}{\rho^{B_n+1}-1} + B_n + \frac{1}{1-\rho}, & \rho \neq 1, \\ \frac{B_n}{2}, & \rho = 1, \end{cases} \quad (4)$$

$$\text{and} \quad \pi(B_n) = \begin{cases} \frac{\rho^{B_n} - \rho^{B_n+1}}{1-\rho^{B_n+1}}, & \rho \neq 1, \\ \frac{1}{B_n+1}, & \rho = 1, \end{cases} \quad (5)$$

respectively, where $\rho \triangleq \lambda/\mu > 0$.

B. Main Results

In this subsection, we analyze the performance of our proposed ITU algorithm under the exponential service time distribution assumption. We first characterize the sufficient and necessary conditions for the existence and uniqueness of the NE. Then, we show that the proposed ITU algorithm converges to the unique NE within a finite time when it exists. Finally, we characterize the efficiency of NE offloading decisions via the PoA performance metric in some scenarios.

Theorem 1: If $W(0) < k\lambda^2/c^2$ and $V_1(\lfloor\widetilde{x}\rfloor) < k\lambda^2/c^2 < V_2(\lceil\widetilde{x}\rceil)$, then there is no NE. Otherwise, there exists a unique NE, in particular,

(i) if $W(0) \geq k\lambda^2/c^2$, then the unique NE is $(0)_{N\times 1}$;
(ii) if $W(0) < k\lambda^2/c^2$ and $W(\lfloor\widetilde{x}\rfloor) < k\lambda^2/c^2 \leq V_1(\lfloor\widetilde{x}\rfloor)$, then the unique NE is $(\lfloor\widetilde{x}\rfloor)_{N\times 1}$;
(iii) if $W(0) < k\lambda^2/c^2$ and $V_2(\lceil\widetilde{x}\rceil) \leq k\lambda^2/c^2 < W(\lceil\widetilde{x}\rceil)$, then the unique NE is $(\lceil\widetilde{x}\rceil)_{N\times 1}$.

In the statement above, $\widetilde{x}$ is the unique solution to $W(\widetilde{x}) = k\lambda^2/c^2$ when $W(0) < k\lambda^2/c^2$, and $W(x)$, $V_1(x)$, and $V_2(x)$ are defined as follows:

$$W(x) \triangleq \frac{k\lambda^2}{c^2}\left|\frac{C_L'(x)}{(\partial C_E(x;y)/\partial x)|_{y=x}}\right|, \quad (6)$$

$$V_1(x) \triangleq \frac{k\lambda^2}{c^2}\left|\frac{C_L(x+1) - C_L(x)}{C_E(x+1;x) - C_E(x;x)}\right| \quad (7)$$

$$\text{and } V_2(x) \triangleq \frac{k\lambda^2}{c^2}\left|\frac{C_L(x) - C_L(x-1)}{C_E(x;x) - C_E(x-1;x)}\right|, \quad (8)$$

where $C_L(x) \triangleq Q(x)/\lambda$ denotes the average local computation cost and $C_E(x;y) \triangleq k\left(\lambda\pi(y)/c\right)^2 \cdot \pi(x)$ represents the average service cost.[3]

Proof: If $W(0) \geq k\lambda^2/c^2$, then it is optimal for each user to upload all its incoming tasks to cloud servers and thus $(0)_{N\times 1}$ is the unique NE. If $W(0) < k\lambda^2/c^2$, then

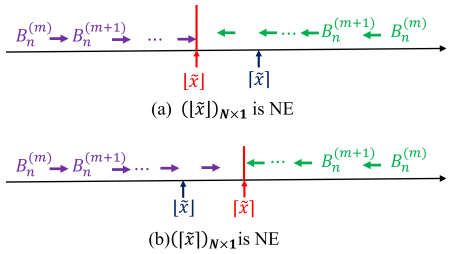
[3]In this paper, $\lfloor y\rfloor$ and $\lceil y\rceil$ denote the maximum integer that is not greater than y and the minimum integer that is not less than y, respectively, $(y)_{N\times 1}$ denotes $N-$ dimensional vector with all y values.

the proof is more involved. It consists of three steps: (i) we show that if $W(0) < k\lambda^2/c^2$, then given all other users' offloading decisions $\widetilde{x}$, the best response decision is also $\widetilde{x}$ for any real numbers $\widetilde{x}$ satisfying $k\lambda^2/c^2 = W(\widetilde{x})$; (ii) we show that NE must be either $(\lfloor \widetilde{x} \rfloor)_{N\times 1}$ or $(\lceil \widetilde{x} \rceil)_{N\times 1}$; (iii) by combining results in (i) and (ii), we only need to find out the best response offloading decision when all other users' offloading decisions are either $\lfloor \widetilde{x} \rfloor$ or $\lceil \widetilde{x} \rceil$. The detailed proof is available in Appendix A. ∎

To apply Theorem 1, we only need to examine the value of $k\lambda^2/c^2$ to check whether the NE exists or not given the system parameters. Note that the term $k\lambda^2/c^2$ denotes the cost of using cloud servers when all the computing tasks are uploaded to cloud servers. Therefore, if $k\lambda^2/c^2 \leq W(0)$, then the cost of using cloud servers is small, and thus, it is better to upload all the tasks to cloud servers (i.e., $\widetilde{x} = 0$). Otherwise, each user partially uploads its incoming computing traffic to the cloud servers with the goal of minimizing its own cost (i.e., $\widetilde{x} > 0$). In addition, when the NE exists, we can further quantify the NE. The next theorem shows that the proposed ITU algorithm converges to the unique NE over a finite number of iterations when the NE exists.

Theorem 2: If the unique NE exists, then the proposed ITU algorithm converges to it over a finite number of iterations.

Proof: The proof relies on the following key property: after each iteration of the ITU algorithm, each user's threshold will get closer and closer to the NE, as it is shown in Fig. 2. Fig. 2 illustrates the convergence of each user's threshold in the case when $\lfloor \widetilde{x} \rfloor$ is the NE and the case when $\lceil \widetilde{x} \rceil$ is the NE, respectively, where we recall that $\widetilde{x}$ is the solution to $k\lambda^2/c = W(\widetilde{x})$ if it exists.



(a) $(\lfloor \widetilde{x} \rfloor)_{N\times 1}$ is NE

(b)$(\lceil \widetilde{x} \rceil)_{N\times 1}$is NE

Fig. 2: Convergence of the n^{th} user's thresholds.

Then, in both cases, the updated threshold of each user exhibits bisection property, i.e., if $B_n^{(m)} < \widetilde{x}$, then the threshold will increase by one in each iteration until it reaches the NE. If $B_n^{(m)} > \widetilde{x}$, then the threshold will decrease by one in each iteration until it converges to the NE. Therefore, the threshold increases or decreases until it is below or above the equilibrium point to ensure the convergence of the ITU algorithm. Please see Appendix B for the detailed proof. ∎

Finally, we characterize the PoA of the NE when it exists, which captures the efficiency of our proposed ITU algorithm when it converges. In particular, we provide conditions under which the proposed ITU algorithm is optimal, i.e., the PoA is equal to zero.

Theorem 3: Whenever the NE exists:

(i) If $W(0) \geq k\lambda^2/c^2$, then PoA $= 0$;

(ii) If $W(0) < k\lambda^2/c^2$ and $\rho = 1$, then PoA ≤ 0.12 as $k \to \infty$;

(iii) If $W(0) < k\lambda^2/c^2$ and $\rho \neq 1$, then PoA $\to 0$ as $k \to \infty$;

Proof: The proof consists of the following three cases:

(i) If $W(0) \geq k\lambda^2/c^2$, according to Theorem 1, the unique NE is $(0)_{N\times 1}$ and thus it is optimal to offload all the tasks to the cloud server. We can also show that in such a case, the thresholds of the global optimal solution for all users are 0 using a similar argument and is omitted here due to space limitations. Therefore, PoA $= 0$ in this case.

(ii) If $W(0) < k\lambda^2/c^2$ and $\rho = 1$, then we can obtain an upper bound of PoA as

$$\text{PoA} \leq 1 - \frac{4x^* + 1}{6\widetilde{x} + 9},$$

where $x^* = \left(6k\lambda^3/c^2\right)^{\frac{1}{4}} - 1$ and $\widetilde{x} = \left(2k\lambda^3/c^2\right)^{\frac{1}{4}} - 1$ are the solutions of equations $k\lambda^2/c^2 = W(x^*)/3$ and $k\lambda^2/c^2 = W(\widetilde{x})$, respectively. By letting $k \to \infty$ in the upper bound, we have the desired results.

(iii) If $W(0) < k\lambda^2/c^2$, $0 < \rho < 1$ and $\rho > 1$, we first obtain the following upper bound on PoA using its definition:

$$\text{PoA} \leq 1 - \frac{(1-\rho)\log(\rho)}{3(\rho\log(\rho) + \rho^{\lceil \widetilde{x} \rceil + 1}(1-\rho))} \cdot \left(\frac{2\lfloor x^* \rfloor + 2}{\rho^{\lfloor x^* \rfloor + 1} - 1} + 2\lfloor x^* \rfloor + \frac{2+\rho}{1-\rho} + \frac{\rho^{\lfloor x^* \rfloor + 1}}{\log(\rho)} \right),$$

where $\widetilde{x}$ and x^* are the unique solutions to the equations $k\lambda^2/c^2 = W(\widetilde{x})$ and $k\lambda^2/c^2 = W(x^*)/3$, respectively. In Appendix C, we further show that this upper bound converges to 0 as $k \to \infty$ when $0 < \rho < 1$. This, together with the fact that PoA ≥ 0, implies that PoA $\to 0$ as $k \to \infty$ when $0 < \rho < 1$.

If $W(0) < k\lambda^2/c^2$ and $\rho > 1$, we first show that $\lim_{k\to\infty} 3\rho^{\widetilde{x}}/\rho^{x^*} = 1$, where $\widetilde{x}$ and x^* are the unique solutions to the equations $k\lambda^2/c^2 = W(\widetilde{x})$ and $k\lambda^2/c^2 = W(x^*)/3$, respectively. Then, we further show that $\limsup_{k\to\infty} \{\text{PoA upper bound}\} \leq 0$. Therefore, we can conclude that PoA $\to 0$ as $k \to \infty$ in this case. The detailed proof is available in Appendix C. ∎

In case (i), the inequality $W(0) \geq k\lambda^2/c^2$ indicates that the cost of uploading all the tasks to cloud servers is relatively small. Therefore, in this case, the threshold under both the ITU algorithm and global optimal solution is zero, which means that users will upload all the tasks to cloud servers under both solutions and thus, the ITU algorithm is optimal. In case (ii), we have $\rho = 1$ and the cost of using cloud servers is high when $k \to \infty$. In this case, users under the ITU algorithm tend to use local devices more, and users under the global optimal solution will still upload some computing traffic to the cloud since local devices can not process all the arrival tasks. Therefore, there is a performance gap. However, in the last case (i.e., $\rho \neq 1$), the cost difference between the ITU algorithm and the global optimal solution diminishes (and hence the PoA converges to zero) as the cost of using cloud servers increases to infinity.

Note that these theoretical results are obtained under the assumption that the service time follows an exponential distribution. In the next section, we perform simulations to

validate our theoretical results and to demonstrate the efficiency of our proposed ITU algorithm under various practical scenarios, such as hyperexponential service time distribution, transmission delay when offloads, imperfect server utilization estimations, and asynchronous threshold updates.

IV. Simulations

In this section, we perform simulations to validate our theoretical findings, especially conditions for the existence and uniqueness of the NE (cf. Theorem 1) and the convergence of our proposed ITU algorithm (cf. Theorem 2) under the exponential service time distribution with $\mu = 4$. Then, we demonstrate the convergence property of the proposed ITU algorithm in the presence of the task transmission delay when the service time follows a hyperexponential distribution, i.e., it follows an exponential distribution with a rate of $8p$ with probability p and another exponential distribution with a rate of $8(1-p)$ otherwise. Note that the mean of the hyperexponential distribution is $1/4$, and the variance is $1/(8p(1-p))-1/16$. We consider hyperexponential distribution in the simulations mainly for two reasons: (i) the variance of the service time can be easily configured; (ii) it simplifies the simulation (using uniformization) and allows for a large-scale simulation. Moreover, we evaluate the efficiency of the NE via the PoA performance that characterizes the gap between the cost under the NE and the global minimum cost. Finally, we demonstrate the superior performance of threshold-based policies over their probabilistic counterpart in our considered mobile cloud computing system. In our simulations, we consider $N = 1000$ users, each of which has the Poisson arrival process with the rate of $\lambda = 6$, and $c = 10$ unless we explicitly mention it.

A. Existence of the NE

In this subsection, we perform numerical simulations to validate the conditions such that the NE exists under the exponential service time distribution. We consider three different values of k, i.e., $k = 22$, $k = 30$, and $k = 40$. These corresponds to the case with $W(\lfloor\widetilde{x}\rfloor) < k\lambda^2/c^2 \leq V_1(\lfloor\widetilde{x}\rfloor)$, $V_1(\lfloor\widetilde{x}\rfloor) < k\lambda^2/c^2 \leq V_2(\lfloor\widetilde{x}\rfloor)$, and $V_2(\lceil\widetilde{x}\rceil) < k\lambda^2/c^2 \leq V_1(\lceil\widetilde{x}\rceil)$, respectively, under which the unique NE is $(2)_{N\times 1}$, NE does not exist, and the unique NE is $(3)_{N\times 1}$, according to Theorem 1. Table I summarize the NE under the above three different cases[4]. From Table I, we can see that there exists a unique NE $(2)_{N\times 1}$ when $k = 22$, and NE $(3)_{N\times 1}$ when $k = 40$, which means that the optimal threshold of one user is the same as all other users' threshold. However, we can observe from Table I that there does not exist a NE when $k = 30$. This validates the conditions for the existence and uniqueness of the NE, as shown in Theorem 1.

B. Convergence under the ITU Algorithm

In this subsection, we perform simulations to validate the convergence of the ITU algorithm. We randomly select 5 users to study their convergences. Fig. 3 shows the convergence property of the ITU algorithm when the calculation of the server utilization uses the exact offloading probability (cf. (5)). We can see from Fig. 3a and Fig. 3c that our proposed ITU algorithm can quickly converge to the corresponding NE. The NE does not exist in the setup for Fig. 3b, in which case the updated threshold under the ITU algorithm oscillates between 2 and 3. This indicates the bisection property of the updated threshold of ITU, and validates the convergence property of the ITU algorithm, as revealed in Theorem 2.

[4]Here, the NE is obtained by plotting the best response threshold of one user with respect to all other users' threshold and finding its intersection with linear function, where the NE must be an integer vector.

System Setup	NE
$k = 22$, i.e., $W(\lfloor\widetilde{x}\rfloor) < \frac{k\lambda^2}{c^2} \leq V_1(\lfloor\widetilde{x}\rfloor)$	$(2)_{N\times 1}$
$k = 30$, i.e., $V_1(\lfloor\widetilde{x}\rfloor) < \frac{k\lambda^2}{c^2} < V_2(\lceil\widetilde{x}\rceil)$	NE does not exist.
$k = 40$, i.e., $V_2(\lceil\widetilde{x}\rceil) \leq \frac{k\lambda^2}{c^2} < W(\lceil\widetilde{x}\rceil)$	$(3)_{N\times 1}$

TABLE I: NE under exponential distribution service time distribution.

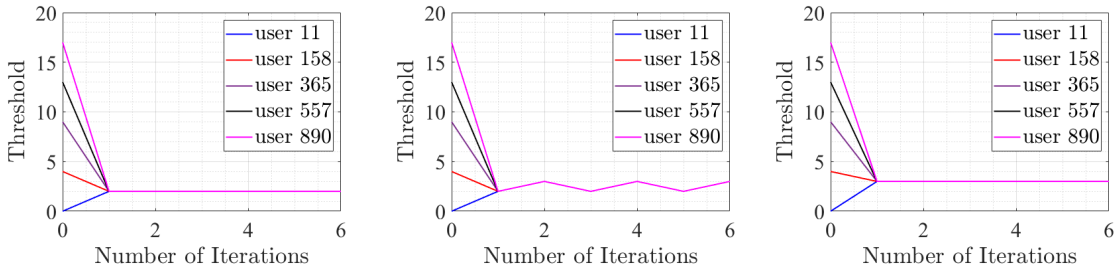
In practice, the service time of local devices may not follow the exponential distribution, and there exists transmission delay when uploading tasks to cloud servers. In addition, the knowledge of the server utilization is not available beforehand and requires estimating over time. As such, we use the ratio between the average offloading rate (i.e., the ratio of the total number of offloaded tasks and the total amount of time) and the total service rate of cloud servers. Moreover, each user may not synchronously update its threshold.

System Setup	NE
$k = 30$	NE does not exist.
$k = 40$	$(3)_{N\times 1}$

TABLE II: NE under hyperexponential distribution service time distribution.

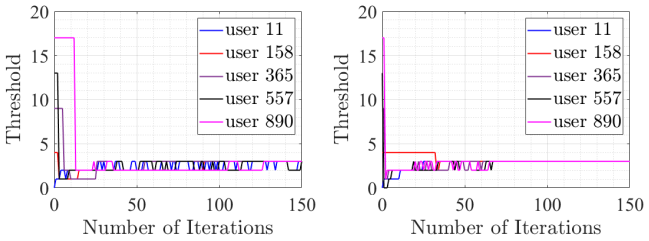
To this end, we consider a hyperexponential distribution service time distribution with $p = 1/8$. Each user updates its threshold asynchronously (i.e., updates with a fixed probability) to optimize its cost function in the presence of imperfect server utilization estimation. We further consider the case that each task offloading to the cloud server experiences both the transmission delay and processing delay at the cloud. In particular, the cost function of each user n is defined as $Q(B_n)/\lambda + (k\beta(\mathbf{B})^2 + \tau_n)\pi(B_n)$, and τ_n is the mean delay and is sampled from an uniform distribution $U[0, 1]$. Note that we do not know the theoretical NE, and thus we first perform numerical simulations to find the NE. The numerical results are summarized in Table II. From Table II, we can observe that the NE is $(3)_{N\times 1}$ when $k = 40$, while the NE does not exist when $k = 30$. Fig. 4 shows the convergence of the ITU algorithm under the hyperexponential distribution service time distribution together with asynchronous threshold update and imperfect server utilization estimation. From Fig. 4, we can observe that the updated threshold converges to the corresponding NE when the NE exists (see Fig. 4b), and oscillates between 2 and 3 otherwise (see Fig. 4a).

In practice, our proposed algorithm still works even when the number of users is small (e.g., $N = 10$). As shown in Fig. 5, our proposed ITU algorithm converges to the same



(a) $k = 22$ (i.e., $W(\lfloor \widetilde{x} \rfloor) < \frac{k\lambda^2}{c^2} \leq V_1(\lfloor \widetilde{x} \rfloor)$)
(b) $k = 30$ (i.e., $V_1(\lfloor \widetilde{x} \rfloor) < \frac{k\lambda^2}{c^2} < V_2(\lceil \widetilde{x} \rceil)$)
(c) $k = 40$ (i.e., $V_2(\lceil \widetilde{x} \rceil) \leq \frac{k\lambda^2}{c^2} < W(\lceil \widetilde{x} \rceil)$)

Fig. 3: Convergence of the ITU algorithm under exponential distribution service time.



(a) $k = 30$ (b) $k = 40$

Fig. 4: The convergence of the ITU algorithm under the hyperexponential distribution service time distribution.

threshold when the number of users N varies from 10 to 990.

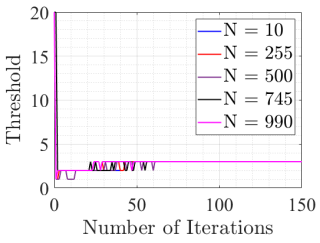


Fig. 5: Thresholds at 100 iterations w.r.t. number of users

C. Price of Anarchy

In this subsection, we perform simulations to evaluate the efficiency of the NE via the PoA performance under three different cases: $\rho = 0.75$ (i.e., $\lambda = 3$), $\rho = 1$ (i.e., $\lambda = 4$), and $\rho = 1.25$ (i.e., $\lambda = 5$). We consider both exponential service time distribution and hyperexponential service time distribution with $p \in \{1/4, 1/8, 1/16\}$. We first verify our theoretical results in Theorem 3 under exponential service time distribution. In Fig. 6, we plot the value of PoA with respect to k varying from 20 to 10^8. We can see from Fig. 6a, Fig. 6b, and Fig. 6c that PoA converges to $0, 0.12$ and 0 as $k \to \infty$, respectively, which validates Theorem 3. In Fig. 7, we plot PoA performance with respect to k varying from 0 to 100. Here, we ignore the trivial case with $k\lambda^2/c^2 \leq W(0)$, where PoA is always equal to zero. We can see from Fig. 7 that both the PoA under different service time distributions share the similar properties. In addition, PoA only exists in certain range of k since system parameters have a significant impact on the existence of the NE (see Theorem 1). From Fig. 7, we can also observe that PoA < 0.3 in all of our simulation scenarios, which implies that our proposed ITU algorithm is at least 70% efficient compared to the global optimal solution.

D. Comparing with Probabilistic Offloading Policies

In this subsection, we demonstrate via simulations that the threshold-based policies outperform the probabilistic offloading policies that offload computing tasks to cloud servers with a certain probability and are widely considered in existing literature (e.g., see [16], [17], [18], and [19]). In particular, we compared our ITU algorithm with the following three policies:

(i) Distributed probabilistic offloading policy: Each mobile device minimizes its own cost function by choosing an offloading probability until the system reaches the NE, where the NE is the offloading probability that users have no incentive to deviate from.

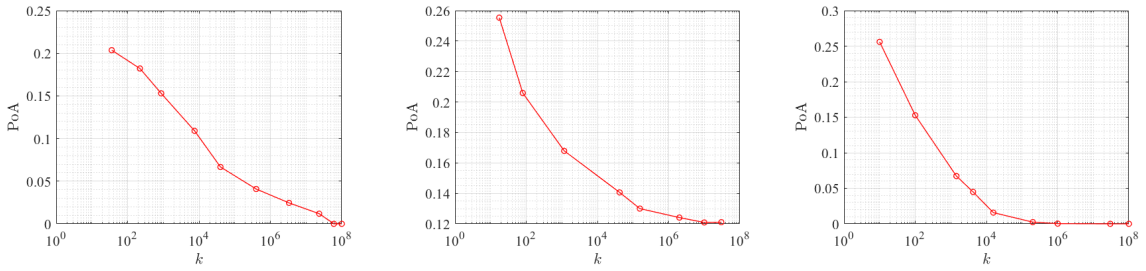
(ii) Global optimal probabilistic offloading policy: A centralized controller minimizes the cost function by choosing an offloading probability for each mobile user in the mobile cloud computing system.

(iii) Global optimal threshold-based offloading policy: A centralized controller minimizes the cost function by choosing a threshold for each user in the system.

We perform numerical simulations in two different setups. From Fig. 8, we can observe that the distributed threshold-based offloading policy outperforms the probabilistic counterpart and even performs better than the global optimal probabilistic offloading policy in most of our considered scenarios.

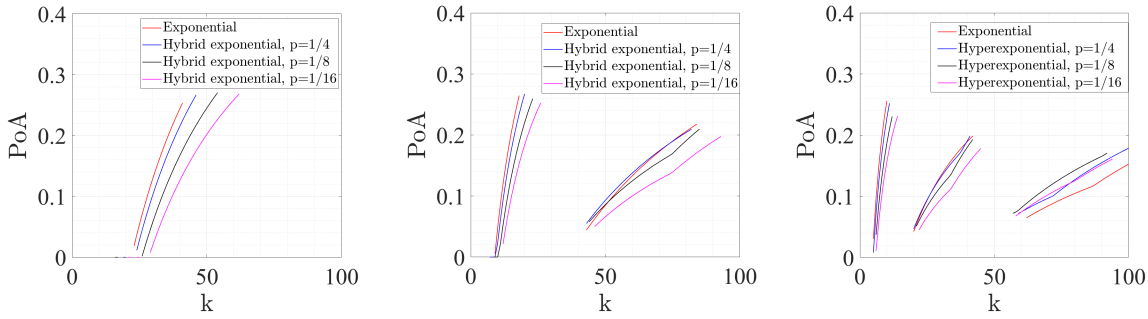
V. Conclusion

In this paper, we proposed a distributed threshold-based offloading algorithm so that each user gradually updates its own threshold with the goal of minimizing its own cost function consisting of average processing delay and the cost of using the cloud services depending on the server utilization in large-scale mobile cloud computing. We then characterized the sufficient and necessary conditions for the existence and uniqueness of the Nash Equilibrium offloading decision under the exponential service time distribution. Furthermore, we showed the convergence of our proposed distributed algorithm to Nash Equilibrium when it exists. Then, we characterized



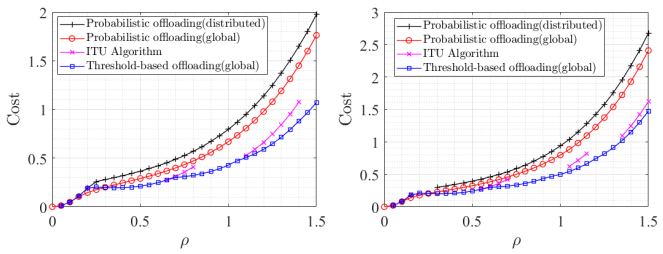
(a) $\rho = 0.75$ (b) $\rho = 1$ (c) $\rho = 1.25$

Fig. 6: PoA performance as $k \to \infty$.



(a) $\rho = 0.75$ (b) $\rho = 1$ (c) $\rho = 1.25$

Fig. 7: PoA performance.



(a) $k = 30$ (b) $k = 50$

Fig. 8: Policies comparison.

the performance of PoA when the cost of using cloud servers is high. Finally, we performed extensive simulations to confirm our theoretical findings, exhibited the efficiency of our proposed algorithm under various practice scenarios such as hyperexponential service time distributions, imperfect server utilization estimation, and asynchronous threshold updates, and demonstrated the superior performance of threshold-based policies over their probabilistic counterpart.

Appendix A
Proof of Theorem 1

We first consider the cases when the NE exists and then the case when NE does not exist. Hence, we first consider the following three cases:

Case (i) $W(0) \geq k\lambda^2/c^2$: In such a case, given all other users' threshold y, the best response is 0. This can be verified by monotonic increasing property of cost function $T(x;y)$ when $W(0) \geq k\lambda^2/c^2$. The proof follows from the basic calculus and thus is omitted due to the lack of space.

Next, we need two lemmas to prove the second case, whose proofs are available at the end of this section.

Lemma 1: If $W(0) < k\lambda^2/c^2$ and given all other user's offloading decisions $\widetilde{x}$, then the best response is also $\widetilde{x}$, where $\widetilde{x}$ is the unique solution to

$$W(\widetilde{x}) = k\lambda^2/c^2. \tag{9}$$

The proof of Lemma 1 follows from the following step: (i) First, we let $T(x;\widetilde{x})$ be the cost function when current threshold is x given thresholds of all other users being $\widetilde{x}$; (ii) We take derivative of $T(x;\widetilde{x})$ with respect to x and let $\frac{T(x;\widetilde{x})}{dx}|_{x=\widetilde{x}} = 0$. Then, we can obtain equation $k\lambda^2/c^2 = W(\widetilde{x})$; (iii) Finally, we show that $\widetilde{x}$ is the unique solution to $k\lambda^2/c^2 = W(\widetilde{x})$. The detailed proof is available at the end of this section.

Lemma 2: (i) $W(x)$ is strictly increasing on $[0,\infty)$;

(ii) $W(x)$, $V_1(x)$ and $V_2(x)$ satisfy the following inequality:

$$W(x) < V_1(x) < V_2(x+1) < W(x+1), \quad \forall x \geq 0.$$

Lemma 3: Functions $C_L(x) = Q(x)/\lambda$ and $C_E(x;y) = k(\lambda\pi(y)/c)^2\pi(x)$ (their definitions are defined in Theorem 1) are strictly increasing and strictly decreasing on the interval $[0,\infty)$ independently of all other users' offloading decisions y, respectively.

Lemma 4: Given all other users' offloading decisions $\widetilde{B}$, if $\widetilde{B} \notin \{\lfloor\widetilde{x}\rfloor, \lceil\widetilde{x}\rceil\}$, then NE does not exist.

From Lemma 1, we can see that if $W(0) < k\lambda^2/c^2$, there exists the unique solution $\widetilde{x}$ to equation (9), i.e., $k\lambda^2/c^2 = W(\widetilde{x})$. Therefore, according to the monotonic increasing property of $W(x)$ (cf. Lemma 2), we have $W(\lfloor\widetilde{x}\rfloor) < k\lambda^2/c^2 < W(\lceil\widetilde{x}\rceil)$ when $\widetilde{x}$ is not an integer. Next, we characterize

the conditions for the existence and uniqueness of the NE by considering a partition of the interval $(W(\lfloor\widetilde{x}\rfloor), W(\lceil\widetilde{x}\rceil))$. Therefore, we consider the following cases under the condition $W(0) < k\lambda^2/c^2$.

Case (ii) $W(\lfloor x\rfloor) < k\lambda^2/c^2 \leq V_1(\lfloor\widetilde{x}\rfloor)$: In such a case, we would like to show that $(\lfloor\widetilde{x}\rfloor)_{N\times 1}$ is the unique NE. From Lemma 4, we know that the NE must be either $(\lfloor\widetilde{x}\rfloor)_{N\times 1}$ or $(\lceil\widetilde{x}\rceil)_{N\times 1}$. Therefore, it is sufficient to show that $T(\lfloor\widetilde{x}\rfloor; \lfloor\widetilde{x}\rfloor) \leq T(\lceil\widetilde{x}\rceil; \lfloor\widetilde{x}\rfloor)$, i.e., the best response of an individual user is $\lfloor\widetilde{x}\rfloor$ given all other users' offloading decisions $\lfloor\widetilde{x}\rfloor$. Indeed, according to the condition $k\lambda^2/c^2 \leq V_1(\lfloor\widetilde{x}\rfloor)$ and the definition of $V_1(x)$ (cf. (7)), we have

$$\frac{k\lambda^2}{c^2} \leq \frac{k\lambda^2}{c^2}\left|\frac{C_L(\lceil\widetilde{x}\rceil) - C_L(\lfloor\widetilde{x}\rfloor)}{C_E(\lceil\widetilde{x}\rceil; \lfloor\widetilde{x}\rfloor) - C_E(\lfloor\widetilde{x}\rfloor; \lfloor\widetilde{x}\rfloor)}\right|.$$

By using Lemma 3, this immediately implies that

$$C_E(\lfloor\widetilde{x}\rfloor; \lfloor\widetilde{x}\rfloor) - C_E(\lceil\widetilde{x}\rceil; \lfloor\widetilde{x}\rfloor) \leq C_L(\lceil\widetilde{x}\rceil) - C_L(\lfloor\widetilde{x}\rfloor).$$

By rearranging items of the above inequality, we have

$$C_L(\lfloor\widetilde{x}\rfloor) + C_E(\lfloor\widetilde{x}\rfloor; \lfloor\widetilde{x}\rfloor) \leq C_L(\lceil\widetilde{x}\rceil) + C_E(\lceil\widetilde{x}\rceil; \lfloor\widetilde{x}\rfloor),$$

i.e., $T(\lfloor\widetilde{x}\rfloor; \lfloor\widetilde{x}\rfloor) \leq T(\lceil\widetilde{x}\rceil; \lfloor\widetilde{x}\rfloor)$.

Case (iii) $V_2(\lceil\widetilde{x}\rceil) \leq k\lambda^2/c^2 < W(\lceil\widetilde{x}\rceil)$: In such a case, we would like to show that $(\lceil\widetilde{x}\rceil)_{N\times 1}$ is the unique NE. Again following Lemma 4, the NE is either $(\lfloor\widetilde{x}\rfloor)_{N\times 1}$ or $(\lceil\widetilde{x}\rceil)_{N\times 1}$. Therefore, it is sufficient to show that $T(\lceil\widetilde{x}\rceil; \lceil\widetilde{x}\rceil) \leq T(\lfloor\widetilde{x}\rfloor; \lceil\widetilde{x}\rceil)$, i.e., the best response of an individual user is $\lceil\widetilde{x}\rceil$ given all other users' offloading decisions $\lceil\widetilde{x}\rceil$. Indeed, according to the condition $V_2(\lceil\widetilde{x}\rceil) \leq k\lambda^2/c^2$ and the definition of $V_2(x)$ (cf. (8)), we have

$$\frac{k\lambda^2}{c^2}\left|\frac{C_L(\lceil\widetilde{x}\rceil) - C_L(\lfloor\widetilde{x}\rfloor)}{C_E(\lceil\widetilde{x}\rceil; \lceil\widetilde{x}\rceil) - C_E(\lfloor\widetilde{x}\rfloor; \lceil\widetilde{x}\rceil)}\right| \leq \frac{k\lambda^2}{c^2}.$$

By using Lemma 3 again, we have

$$C_L(\lceil\widetilde{x}\rceil) - C_L(\lfloor\widetilde{x}\rfloor) \leq C_E(\lfloor\widetilde{x}\rfloor; \lceil\widetilde{x}\rceil) - C_E(\lceil\widetilde{x}\rceil; \lceil\widetilde{x}\rceil),$$

which immediately implies the desired result.

Finally, we will show the case when NE does not exist. In this case, we have $V_1(\lfloor\widetilde{x}\rfloor) < k\lambda^2/c^2 < V_2(\lceil\widetilde{x}\rceil)$. Indeed, by following the same arguments in the previous two cases, we are able to show that the best response of an individual user is $\lceil\widetilde{x}\rceil$ and $\lfloor\widetilde{x}\rfloor$ given all other users' offloading decisions $\lfloor\widetilde{x}\rfloor$ and $\lceil\widetilde{x}\rceil$, respectively. ■

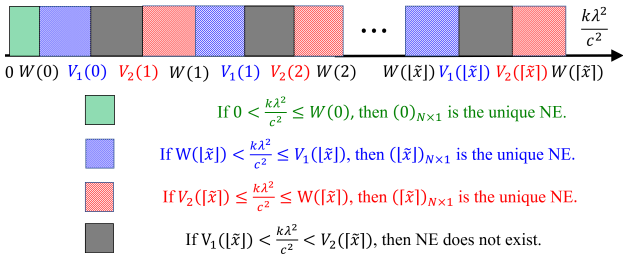


Fig. 9: Conditions for the existence and uniqueness of NE.

Fig. 9 summarizes the sufficient and necessary conditions of the existence and uniqueness of the NE.

The proofs of Lemma 2 and Lemma 3 follow from basic calculus and thus are omitted due to space limit. Next, we prove Lemma 1 and Lemma 4 to complete the proof.

Proof of Lemma 1: Here, we consider three different cases, i.e., $\rho = 1, 0 < \rho < 1$ and $\rho > 1$:

Case (i) $\rho = 1$: In this case, $T(x; \widetilde{x})$ can be calculated as follows.

$$T(x; \widetilde{x}) = \frac{1}{\lambda}\cdot\frac{x}{2} + \frac{k\lambda^2}{c^2}\left(\frac{1}{\widetilde{x}+1}\right)^2\cdot\frac{1}{x+1}.$$

Taking the derivative of $T(x; \widetilde{x})$ with respect to x and set $x = \widetilde{x}$ we have the following:

$$\frac{1}{2\lambda} - \frac{k\lambda^2}{c^2}\left(\frac{1}{\widetilde{x}+1}\right)^4 = 0. \tag{10}$$

If $k\lambda^2/c^2 > W(0) = 1/2\lambda$, then equation (10) has one unique solution

$$\widetilde{x} = \left(\frac{2k\lambda^3}{c^2}\right)^{\frac{1}{4}} - 1.$$

Indeed, when $x \in (0, \widetilde{x})$, we have

$$\begin{aligned}\frac{dT(x;\widetilde{x})}{dx} &= \frac{1}{2\lambda} - \frac{k\lambda^2}{c^2}\left(\frac{1}{\widetilde{x}+1}\right)^2\cdot\left(\frac{1}{x+1}\right)^2 \\ &< \frac{1}{2\lambda} - \frac{k\lambda^2}{c^2}\left(\frac{1}{\widetilde{x}+1}\right)^4 = 0,\end{aligned}$$

implying that $T(x; \widetilde{x})$ is decreasing in $(0, \widetilde{x})$. Similarly, we can show that $T(x; \widetilde{x})$ is increasing in $x \in (\widetilde{x}, \infty)$. Therefore, $\widetilde{x}$ is the unique solution to $k\lambda^2/c^2 = W(\widetilde{x})$ in this case.

Next, we consider the cases when $\rho > 1$ and $0 < \rho < 1$.

By the definition of $W(x)$ (cf. (2), (4) and (5)), we have:

$$\begin{aligned}T(x; \widetilde{x}) = \frac{1}{\lambda}&\left(\frac{x+1}{\rho^{x+1}-1} + x + \frac{1}{1-\rho}\right) \\ &+ \frac{k\lambda^2}{c^2}\left(\frac{\rho^{\widetilde{x}} - \rho^{\widetilde{x}+1}}{1-\rho^{\widetilde{x}+1}}\right)^2\cdot\frac{\rho^x - \rho^{x+1}}{1-\rho^{x+1}}.\end{aligned} \tag{11}$$

Taking derivative of $T(x; \widetilde{x})$ with respect to x, then set $x = \widetilde{x}$ and let $dT(x; \widetilde{x})/dx = 0$, we have

$$\frac{k\lambda^3(1-\rho)^3}{c^2\rho^3}\left(\frac{\rho^{\widetilde{x}+1}}{\rho^{\widetilde{x}+1}-1}\right)^2 + \frac{\rho^{\widetilde{x}+1}-1}{\log(\rho)} = \widetilde{x}+1, \tag{12}$$

where $\log(\cdot)$ is the logarithm with the natural base e.

Next, we will find the condition such that the equation (12) has one unique solution. To simply the notations, we let $a = k\lambda^3(1-\rho)^3/(c^2\rho^3)$, $b = 1/\log(\rho)$ and $u = \rho^{x+1} - 1$. Then we rewrite (12) as follows.

$$h_1(u) = h_2(u),$$

where

$$h_1(u) \triangleq a\left(1 + \frac{1}{u}\right)^2 + bu,\ u > -1,$$
$$\text{and } h_2(u) \triangleq \log_\rho(u+1),\ u > -1.$$

Then, we consider the following cases when $0 < \rho < 1$ and $\rho > 1$, respectively.

Case (ii) $0 < \rho < 1$: In this case, we have $-1 < u \leq \rho - 1 < 0, a > 0$ and $b < 0$. In addition, we have both

$d^2h_1(u)/du^2 > 0$ and $d^2h_2(u)/du^2 > 0$, which implies that $h_1(u)$ and $h_2(u)$ are strictly convex. We also have $dh_2(u)/du < 0$ implying that $h_2(u)$ is strictly decreasing. Moreover, we have $h_2(\rho-1)=1$. As shown in Fig. 10, the tangent line $l_T(u)$ to the function $h_2(u)$ at the point $(\rho-1,1)$ can be expressed as follows.

$$l_T(u) = \frac{u-\rho+1}{\rho\log(\rho)} + 1.$$

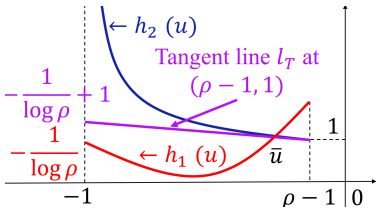


Fig. 10: Relations between $h_1(u)$ and $h_2(u)$: $0<\rho<1$ and $k\lambda^2/c^2 > W(0)$

Since $l_T(-1) = -1/\log(\rho)+1$, we have

$$l_T(-1) > h_1(-1) = -\frac{1}{\log(\rho)}. \tag{13}$$

If $k\lambda^2/c^2 > W(0)$, then we have $h_1(\rho-1) > h_2(\rho-1)$. Since $h_2(u)$ is strictly decreasing in $u\in(-1,\rho-1]$ and $l_T(u)$ is the tangent line $l_T(u)$ to the function $h_2(u)$ at the point $(\rho-1,1)$, we have

$$h_2(u) \geq l_T(u), \quad \forall u\in(-1,\rho-1],$$

which implies that $dT(x;\widetilde{x})/dx > 0$. This together with (13) implies that $\lim_{u\to-1} h_2(u) > h_1(-1)$. Therefore, we have that $h_1(u)$ and $h_2(u)$ have only one intersection point in the interval $(-1,\rho-1]$, which implies that equation $h_1(u)=h_2(u)$ has one unique solution $\widetilde{u}$ and thus the unique solution $\widetilde{x}$.

Next, we will show that such $\widetilde{x}$ is indeed unique best response of $T(x;\widetilde{x})$ given that all other users' offloading decisions $\widetilde{x}$. If $x\in[0,\widetilde{x})$, then we have $u\in(\widetilde{u},\rho-1]$. Therefore, we have

$$\frac{dT(x;\widetilde{x})}{dx} = \frac{u+1}{b\lambda u^2}\left(a\left(1+\frac{1}{\widetilde{u}}\right)^2 + bu - h_2(u)\right)$$
$$\overset{(a)}{<} \frac{u+1}{b\lambda u^2}(h_1(\widetilde{u}) - h_2(\widetilde{u})) \overset{(b)}{=} 0,$$

where step (a) follows from the fact that $b=1/\log(\rho)<0$ and $u+1>0$ imply $(u+1)/(b\lambda u^2)<0$ and the fact that $bu-h_2(u) > b\widetilde{u}-h_2(\widetilde{u})$ (will be shown shortly); step (b) follows from the fact that $h_1(\widetilde{u})-h_2(\widetilde{u})=0$. Next, we will show that $bu-h_2(u) > b\widetilde{u}-h_2(\widetilde{u})$. To that end, let's consider function

$$h_w(u) \triangleq bu - h_2(u), \quad u\in[\widetilde{u},\rho-1].$$

The derivative of $h_w(u)$ can be expressed as

$$\frac{dh_w(u)}{du} = \frac{bu}{u+1} > 0 \quad \text{(due to } b<0,\ u<0 \text{ and } u+1>0),$$

which implies that $h_w(u)$ is increasing on $[\widetilde{u},\rho-1]$. Thus, we have $h_w(u) > h_w(\widetilde{u})$ when $u>\widetilde{u}$.

Similarly, we can show $dT(x;\widetilde{x})/dx > 0, \forall x\in[\widetilde{x},\infty)$. Thus, $T(x;\widetilde{x})$ is decreasing in $[0,\widetilde{x})$, and is increasing in $[\widetilde{x},\infty)$.

Case (iii) $\rho>1$: To facilitate our proof, we let $h_d(u)$ denote the difference between $h_1(u)$ and $h_2(u)$, i.e.,

$$h_d(u) \triangleq h_1(u) - h_2(u).$$

By taking derivative of $h_d(u)$ we can show that: If $\rho>1$, then $h_d(u)$ is strictly increasing when $u\in[\rho-1,\infty)$.

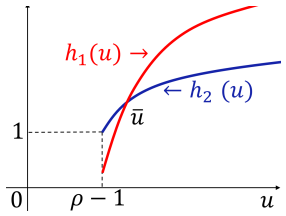


Fig. 11: Relations between $h_1(u)$ and $h_2(u)$: $\rho>1$ and $k\lambda^2/c^2 > W(0)$

In this case, if $k\lambda^2/c^2 > W(0)$, then we have $h_1(\rho-1) < h_2(\rho-1)$, as shown in Fig. 11. Thus, we have $h_d(\rho-1)<0$. Since $h_d(u)$ is strictly increasing in $u\in[\rho-1,\infty)$, then there must exist a unique $\widetilde{u}>\rho-1$ satisfying $h_d(\widetilde{u})=0$. Therefore, there exists some $\widetilde{x}=\log(\widetilde{u}+1)-1$, which is the solution to

$$\frac{dT(x;\widetilde{x})}{dx} = \frac{u+1}{b\lambda u^2}h_d(\widetilde{u}) = 0.$$

Next, we will show that such $\widetilde{x}$ is the unique best response of $T(x;\widetilde{x})$ given all other users' offloading decisions $\widetilde{x}$. Indeed, if $x\in[0,\widetilde{x})$, then we have $u\in[\rho-1,\widetilde{u})$. Hence, we have

$$\frac{dT(x;\widetilde{x})}{dx} = \frac{u+1}{b\lambda u^2}\left(a\left(1+\frac{1}{\widetilde{u}}\right)^2 + bu - h_2(u)\right)$$
$$\overset{(a)}{<} \frac{u+1}{b\lambda u^2}h_d(u) \overset{(b)}{<} 0,$$

where step (a) follows from the fact that $b=1/\log(\rho)>0$ implies $(u+1)/(b\lambda u^2)>0$ and $(1+1/\widetilde{u})^2 < (1+1/u)^2$; step (b) follows from the fact that $h_d(u)<0$ for all $u\in[\rho-1,\widetilde{u})$, which follows from the fact that $h_d(u)$ is increasing in $[\rho-1,\infty)$ and the fact that $h_d(\widetilde{u})=0$.

Similarly, we can show that $dT(x;\widetilde{x})/dx > 0, \forall x\in[\widetilde{x},\infty)$. Thus, $T(x;\widetilde{x})$ is decreasing in $x\in[0,\widetilde{x})$ and is increasing in $x\in[\widetilde{x},\infty)$. Therefore, $\widetilde{x}$ is the unique solution to $k\lambda^2/c^2 = W(\widetilde{x})$. ■

Proof of Lemma 4: We want to show that $(\widetilde{B})_{N\times1}$ is not the NE when $\widetilde{B}\notin\{\lfloor\widetilde{x}\rfloor,\lceil\widetilde{x}\rceil\}$. To that end, we consider the best response of an individual user given all other users' integer offloading decisions $\widetilde{B}$, denoted by $x_{\widetilde{B}}$, where $x_{\widetilde{B}}$ is real number and satisfies the following equations.

$$\frac{1}{2\lambda} - \frac{k\lambda^2}{c^2}\left(\frac{1}{\widetilde{B}+1}\right)^2\cdot\left(\frac{1}{x_{\widetilde{B}}+1}\right)^2 = 0 \tag{14}$$

when $\rho = 1$, and

$$\rho\left(\rho^{x_{\widetilde{B}}+1}-1\right)-(x_{\widetilde{B}}+1)\rho\log(\rho)= \frac{k\lambda^3(\rho-1)^3\log(\rho)}{c^2}\cdot\left(\frac{\rho^{\widetilde{B}}}{\rho^{\widetilde{B}+1}-1}\right)^2, \quad (15)$$

when $\rho \neq 1$, which is obtained by setting $dT(x_{\widetilde{B}};\widetilde{B})/dx_{\widetilde{B}} = 0$. It can be shown shortly that the best response $x_{\widetilde{B}}$ is decreasing with respect to $\widetilde{B}$. Since $\widetilde{B} \notin \{\lfloor\widetilde{x}\rfloor, \lceil\widetilde{x}\rceil\}$, we have $\widetilde{B} < \lfloor\widetilde{x}\rfloor$ or $\widetilde{B} > \lceil\widetilde{x}\rceil$. Then, we have the following two different cases:

• If $\widetilde{B} < \lfloor\widetilde{x}\rfloor < \widetilde{x}$, then according to the monotonic decreasing property of the best response $x_{\widetilde{B}}$, we have $x_{\widetilde{B}} > \widetilde{x}$, where we use the fact that the best response of an individual user is $\widetilde{x}$ given all other users' offloading decisions $\widetilde{x}$. This implies that $\lfloor x_{\widetilde{B}}\rfloor \geq \lfloor\widetilde{x}\rfloor > \widetilde{B}$ and hence $(\widetilde{B})_{N\times 1}$ is not a NE.

• If $\widetilde{B} > \lceil\widetilde{x}\rceil > \widetilde{x}$, then following the same arguments as in the case of $\widetilde{B} < \lfloor\widetilde{x}\rfloor < \widetilde{x}$, we again can show that $(\widetilde{B})_{N\times 1}$ is not a NE.

Next, we show the monotonic decreasing property of $x_{\widetilde{B}}$ with respect to $\widetilde{B}$ in both $\rho = 1$ and $\rho \neq 1$ cases to complete the proof.

• $\rho = 1$ case: From (14), we have

$$f_1(x_{\widetilde{B}}) = g_1(\widetilde{B}), \quad (16)$$

where

$$f_1(x) \triangleq \frac{(x+1)^2}{2} \text{ and } g_1(x) \triangleq \frac{k\lambda^3}{c^2}\left(\frac{1}{x+1}\right)^2, x \geq 0.$$

From (16), we have

$$f_1\left(x_{\widetilde{B}+1}\right) - f_1\left(x_{\widetilde{B}}\right) = g_1\left(\widetilde{B}+1\right) - g_1\left(\widetilde{B}\right) < 0,$$

where the last step follows from the monotonic decreasing property of $g_1(x)$. Therefore, we have $f_1\left(x_{\widetilde{B}+1}\right) < f_1\left(x_{\widetilde{B}}\right)$. Since $f_1(x)$ is strictly increasing with respect to x, we have that $x_{\widetilde{B}+1} < x_{\widetilde{B}}$ holds for any non-negative integer $\widetilde{B}$.

• $\rho \neq 1$ case: We first rearrange terms in (15),

$$f_2(x_{\widetilde{B}}) = g_2(\widetilde{B}), \quad (17)$$

where

$$f_2(x) \triangleq \rho^{x+1} - 1 - (x+1)\log(\rho),$$
$$\text{and } g_2(x) \triangleq \frac{k\lambda^3(\rho-1)^3\log(\rho)}{c^2\rho}\left(\frac{\rho^x}{\rho^{x+1}-1}\right)^2,$$

for all $x \geq 0$. It can be easily shown by calculus that $f_2(x)$ and $g_2(x)$ are strictly decreasing and increasing, respectively. The proofs are omitted due to the lack of space. Then, from (17), we have

$$f_2\left(x_{\widetilde{B}+1}\right) - f_2\left(x_{\widetilde{B}}\right) = g_2\left(\widetilde{B}+1\right) - g_2\left(\widetilde{B}\right) < 0,$$

where the last step follows from the monotonic decreasing and increasing property of $f_2(x)$ and $g_2(x)$, respectively. Hence, we have $x_{\widetilde{B}+1} < x_{\widetilde{B}}$ holds for any non-negative integer $\widetilde{B}$. ■

APPENDIX B
PROOF OF THEOREM 2

From Theorem 1, we know that if the NE exists, it is either $(0)_{N\times 1}$, $(\lfloor\widetilde{x}\rfloor)_{N\times 1}$ or $(\lceil\widetilde{x}\rceil)_{N\times 1}$. Hence, We will consider these three cases, respectively.

(i) $(0)_{N\times 1}$ is the NE: In this case, we have $k\lambda^2/c^2 \leq W(0)$. (cf. Theorem 1) and $T(x;\widetilde{x})$ is increasing with respect to x, which can be easily verified by taking derivative with respect to x. Therefore, we have $\widehat{B}_n^{(m+1)} = 0$. Then, for any $B_n^{(1)} > 0$, the threshold will decrease by one in each iteration and goes to zero within $B_n^{(1)} + 1$ steps .

In order to prove the convergence in the other two cases, we need the following lemma that shows the bisection property of the updated threshold under the ITU algorithm.

Lemma 5: If $k\lambda^2/c^2 > W(0)$, then for any $\widetilde{x} > 0$, where $\widetilde{x}$ satisfies $k\lambda^2/c^2 = W(\widetilde{x})$, and $m \geq 1$, we have:
(i) If $B_n^{(m)} < \lfloor\widetilde{x}\rfloor$ or $B_n^{(m)} = \lfloor\widetilde{x}\rfloor$ but $(\lfloor\widetilde{x}\rfloor)_{N\times 1}$ is not NE, then $\widehat{B}_n^{(m+1)} \geq B_n^{(m)}$;
(ii) If $B_n^{(m)} > \lceil\widetilde{x}\rceil$ or $B_n^{(m)} = \lceil\widetilde{x}\rceil$ but $(\lceil\widetilde{x}\rceil)_{N\times 1}$ is not NE, then $\widehat{B}_n^{(m+1)} \leq B_n^{(m)}$.

From Lemma 5 we have that for any $B_n^{(m)} < \widetilde{x}$, then in the next iteration, $B_n^{(m+1)}$ will move closer to $\widetilde{x}$. Similarly, for any $B_n^{(m)} > \widetilde{x}$, $B_n^{(m+1)}$ will also move closer to $\widetilde{x}$. In either cases, in each iteration, the threshold will get closer and closer to $\widetilde{x}$, as shown in Fig. 2. Note that in the first iteration (i.e., $m = 0$), all users in the system solve the same optimization problem and obtain the same threshold, since the server utilization is the same for all users. Then, after the first iteration (i.e., $m > 1$), all users will adjust their threshold in the same way and thus we just need to focus on a particular user n. Now, we are ready to prove the convergence of the ITU algorithm when the NE is not $(0)_{N\times 1}$.

(ii) $(\lfloor\widetilde{x}\rfloor)_{N\times 1}$ is the NE: In this case , for any $B_n^{(1)} < \lfloor\widetilde{x}\rfloor < \widetilde{x}$, the threshold will increase by one in each iteration until reaching to $\lfloor\widetilde{x}\rfloor$. For any $B_n^{(1)} > \widetilde{x} > \lfloor\widetilde{x}\rfloor$, the threshold will decrease by one in each iteration until reaching to $\lfloor\widetilde{x}\rfloor$. Thus, it will take at most $\left|\lfloor\widetilde{x}\rfloor - B_n^{(1)}\right| + 1$ iterations for user n's threshold to converge to $\lfloor\widetilde{x}\rfloor$.

(iii) $(\lceil\widetilde{x}\rceil)_{N\times 1}$ is the NE: In this case , for any $B_n^{(1)} < \widetilde{x} < \lceil\widetilde{x}\rceil$, the threshold will increase in each iteration until reaching to $\lceil\widetilde{x}\rceil$. For any $B_n^{(1)} > \lceil\widetilde{x}\rceil > \widetilde{x}$, the threshold will decrease by one in each iteration until reaching to $\lceil\widetilde{x}\rceil$. Thus, it will take at most $\left|\lceil\widetilde{x}\rceil - B_n^{(1)}\right| + 1$ iterations for user n's threshold to converge to $\lceil\widetilde{x}\rceil$. ■

Next, we prove Lemma 5 to complete the proof.

Proof of Lemma 5: We first define the following two functions:

$$U_1(x) \triangleq \begin{cases} x, & \rho = 1, \\ \rho^{x+2} - (x+1)\rho\log\rho - \rho, & \rho \neq 1. \end{cases}$$

$$\text{and} \quad U_2(x) \triangleq \begin{cases} \frac{\sqrt{2k\lambda}\lambda}{c(x+1)} - 1, & \rho = 1, \\ \frac{k\lambda^3(\rho-1)^3\log\rho}{c^2}\cdot\left(\frac{\rho^x}{1-\rho^{x+1}}\right)^2, & \rho \neq 1. \end{cases}$$

It can be easily showed that function $U_1(x)$ is strictly increasing on $[0,\infty)$ and $U_2(x)$ is strictly decreasing on $[0,\infty)$. Therefore, $U_2(x) - U_1(x)$ is strictly decreasing. The

detailed proofs are omitted due to space limit. Now we are ready to prove Lemma 5.

In the ITU algorithm, all users will solve the same optimization problem in the first iteration. Thus, all users will have the same threshold when $m = 1$. Then, for any $m \geq 1$, we can simplify the cost function in (3) as

$$T\left(x; B_n^{(m)}\right) = \frac{Q(x)}{\lambda} + k\left(\frac{\pi\left(B_n^{(m)}\right)}{c}\right)^2 \pi(x), \qquad (18)$$

where $x \geq 0$ is some real number and $B_n^{(m)}$ is the threshold of user n in the m^{th} iteration.

Since we have shown that cost function $T(x;\widetilde{x})$ is decreasing and increasing in $[0,\widetilde{x})$ and $[\widetilde{x},\infty)$ when $k\lambda^2/c^2 > W(0)$, respectively (cf. Proof of Lemma 1). We notice that $T(x;\widetilde{x})$ (cf. (11)) and $T(x;B_n^{(m)})$ share the similar form. Therefore, we take derivative of $T(x;B_n^{(m)})$ with respect to x and set to zero. Then, we have $U_1(\widehat{x}) = U_2\left(B_n^{(m)}\right)$, where $\widehat{x}$ is a real number such that $\left(\frac{dT\left(x;B_n^{(m)}\right)}{dx}\right)\Big|_{x=\widehat{x}} = 0$. Therefore, we have

$$\widehat{B}_n^{(m+1)} \in \left\{\lfloor \widehat{x} \rfloor, \lceil \widehat{x} \rceil\right\}. \qquad (19)$$

Note that $\widetilde{x}$ satisfies equation $W(\widetilde{x}) = k\lambda^2/c^2$ and through simple algebraic operations, we have $U_1(\widetilde{x}) = U_2(\widetilde{x})$. Next, we consider the following two different cases:

(i) If $B_n^{(m)} < \lfloor \widetilde{x} \rfloor$ or $B_n^{(m)} = \lfloor \widetilde{x} \rfloor$ but is not NE, then we have $B_n^{(m)} < \widetilde{x}$. Then we have

$$\begin{aligned} U_1(\widehat{x}) - U_1\left(B_n^{(m)}\right) &= U_2\left(B_n^{(m)}\right) - U_1\left(B_n^{(m)}\right) \\ &\overset{(a)}{>} U_2(\widetilde{x}) - U_1(\widetilde{x}) = 0, \end{aligned}$$

where step (a) follows from the fact $U_2(x) - U_1(x)$ is strictly decreasing. Therefore, we have $U_1(\widehat{x}) > U_1\left(B_n^{(m)}\right)$. Since $U_1(x)$ is strictly increasing, we have $\widehat{x} > B_n^{(m)}$. Thus, by (19), we have $\widehat{B}_n^{(m+1)} \geq B_n^{(m)}$.

(ii) If $B_n^{(m)} > \lceil \widetilde{x} \rceil$ or $B_n^{(m)} = \lceil \widetilde{x} \rceil$ but is not NE, then we have $B_n^{(m)} > \widetilde{x}$. Then we have

$$\begin{aligned} U_1(\widehat{x}) - U_1\left(B_n^{(m)}\right) &= U_2\left(B_n^{(m)}\right) - U_1\left(B_n^{(m)}\right) \\ &\overset{(a)}{<} U_2(\widetilde{x}) - U_1(\widetilde{x}) = 0, \end{aligned}$$

where step (a) follows from the fact that $U_2(x) - U_1(x)$ is strictly decreasing. Therefore, we have $U_1(\widehat{x}) < U_1\left(B_n^{(m)}\right)$. Since $U_1(x)$ is strictly increasing, we have $\widehat{x} < B_n^{(m)}$. Therefore, by (19), we have $\widehat{B}_n^{(m+1)} \leq B_n^{(m)}$. ∎

Appendix C
Proof of Theorem 3

Here, we show that the PoA converges to zero as $k \to \infty$ in both $0 < \rho < 1$ and $\rho > 1$ cases. We first present the following two lemmas to facilitate our proof.

Lemma 6: Let B^* denote the optimal threshold determined by some central controller. If $W(0) < k\lambda^2/c^2$, then $B^* = \lfloor x^* \rfloor$, where x^* satisfies function $3k\lambda^2/c^2 = W(x^*)$.

Lemma 6 characterizes the optimal threshold determined by a central controller in the system when $W(0) < k\lambda^2/c^2$.

Lemma 7: If $W(0) < k\lambda^2/c^2$, then we have

$$\text{PoA} \leq 1 - \frac{4x^* + 1}{6\widetilde{x} + 9}$$

when $\rho = 1$ and

$$\begin{aligned} \text{PoA} \leq 1 - &\frac{(1-\rho)\log(\rho)}{3(\rho\log(\rho) + \rho^{\lceil \widetilde{x} \rceil + 1}(1-\rho))} \\ &\cdot \left(\frac{2\lfloor x^* \rfloor + 2}{\rho^{\lfloor x^* \rfloor + 1} - 1} + 2\lfloor x^* \rfloor + \frac{2+\rho}{1-\rho} + \frac{\rho^{\lfloor x^* \rfloor + 1}}{\log(\rho)}\right) \end{aligned}$$

when $\rho \neq 1$, where x^* satisfies function $3k\lambda^2/c^2 = W(x^*)$.

Lemma 7 characterizes a upper bound of PoA when $W(0) < k\lambda^2/c^2$ and $\rho \neq 1$. Having characterized the optimal threshold under some central controller and PoA upper bound when $W(0) < k\lambda^2/c^2$, we are ready to prove Theorem 3.

Case (i) $0 < \rho < 1$: First, we will show that as $k \to \infty$, both $\widetilde{x} \to \infty$ and $x^* \to \infty$. By Theorem 1 and Lemma 6, we have $k\lambda^2/c^2 = W(\widetilde{x})$ and $3k\lambda^2/c^2 = W(x^*)$. As $k \to \infty$, we have both $W(\widetilde{x}) \to \infty$ and $W(x^*) \to \infty$. By Lemma 2 we know that $W(x)$ is strictly increasing on $[0,\infty)$. Therefore, both $\widetilde{x} \to \infty$ and $x^* \to \infty$ as $k \to \infty$. Hence, we have $\rho^{\lceil \widetilde{x} \rceil} \to 0$ and $\rho^{\lfloor x^* \rfloor} \to 0$. This combines with the upper bound on PoA (cf. Lemma 7), yielding

$$\begin{aligned} \text{PoA} &\leq 1 - \frac{(1-\rho)\log(\rho)}{3(\rho\log(\rho))}\left(-2\lfloor x^* \rfloor - 2 + 2\lfloor x^* \rfloor + \frac{2+\rho}{1-\rho}\right) \\ &= 0. \end{aligned}$$

Case (ii) $\rho > 1$: From the previous case we know that if $\widetilde{x} \to \infty$ and $x^* \to \infty$, then we have both $\rho^{\widetilde{x}} \to \infty$ and $\rho^{x^*} \to \infty$. By Taylor series we have $\rho^{\lceil \widetilde{x} \rceil} \geq \rho^{\widetilde{x}} = 1 + \widetilde{x}\log\rho + O\left(\widetilde{x}^2\right)$ and $\rho^{\lfloor x^* \rfloor} = 1 + \lfloor x^* \rfloor \log\rho + O\left(\lfloor x^* \rfloor^2\right)$. Therefore, we have both $\widetilde{x}/\rho^{\lceil \widetilde{x} \rceil} \to 0$ as $\widetilde{x} \to \infty$ and $\lfloor x^* \rfloor / \rho^{\lfloor x^* \rfloor} \to 0$ as $x^* \to \infty$. Therefore, by dividing $\rho^{\lceil \widetilde{x} \rceil + 1}$ for the numerator and denominator terms of PoA upper bound (cf. Lemma 7), we have

$$\begin{aligned} 1 - &\frac{(1-\rho)\log(\rho)}{3(\rho\log\rho/\rho^{\lceil \widetilde{x} \rceil + 1}) + (1-\rho))}\left(\frac{2\lfloor x^* \rfloor + 2}{(\rho^{\lfloor x^* \rfloor + 1} - 1)\rho^{\lceil \widetilde{x} \rceil + 1}}\right. \\ &\left. + \frac{2\lfloor x^* \rfloor}{\rho^{\lceil \widetilde{x} \rceil + 1}} + \frac{2+\rho}{(1-\rho)\rho^{\lceil \widetilde{x} \rceil + 1}} + \frac{\rho^{\lfloor x^* \rfloor + 1}}{\log(\rho)\rho^{\lceil \widetilde{x} \rceil + 1}}\right) \\ \overset{(a)}{=} 1 - &\frac{\log\rho}{3} \cdot \left(\frac{2\lfloor x^* \rfloor \log\rho + \rho^{\lfloor x^* \rfloor + 1}}{\log(\rho)\rho^{\lceil \widetilde{x} \rceil + 1}}\right), \qquad (20) \end{aligned}$$

where step (a) follows from the fact that both $\widetilde{x}/\rho^{\lceil \widetilde{x} \rceil} \to 0$ as $\widetilde{x} \to \infty$ and $\lfloor x^* \rfloor / \rho^{\lfloor x^* \rfloor} \to 0$ as $x^* \to \infty$.

Next, we will analyze the relation between x^* and $\widetilde{x}$. By Theorem 1 and Lemma 6 we have $k\lambda^2/c^2 = W(\widetilde{x})$ and $3k\lambda^2/c^2 = W(x^*)$. Therefore, we have $3W(\widetilde{x}) = W(x^*)$. By (6) we have

$$\begin{aligned} &\frac{3(1-\rho^{\widetilde{x}+1})^2}{\lambda(1-\rho)^3\rho^{2\widetilde{x}-1}}\left(\widetilde{x} + 1 - \frac{\rho^{\widetilde{x}+1} - 1}{\log\rho}\right) \\ =&\frac{(1-\rho^{x^*+1})^2}{\lambda(1-\rho)^3\rho^{2x^*-1}}\left(x^* + 1 - \frac{\rho^{x^*+1} - 1}{\log\rho}\right). \qquad (21) \end{aligned}$$

Note that as $x^*, \widetilde{x} \to \infty$, we have $\left(1-\rho^{\widetilde{x}+1}\right)^2/\rho^{2\widetilde{x}} \to \rho^2$ and $\left(1-\rho^{x^*+1}\right)^2/\rho^{2x^*} \to \rho^2$. Therefore, as $x^*, \widetilde{x} \to \infty$, we have

$$\lim_{k\to\infty} \frac{3\left(\widetilde{x}+1-\frac{\rho^{\widetilde{x}+1}-1}{\log\rho}\right)}{x^*+1-\frac{\rho^{x^*+1}-1}{\log\rho}} = \lim_{k\to\infty}\frac{3\rho^{\widetilde{x}}}{\rho^*} = 1. \quad (22)$$

Next, we want to show that $\rho^{\widetilde{x}} > O\left(\lfloor x^*\rfloor^2\right)$. We show this result by contradiction. Assume that $\rho^{\widetilde{x}} \le O\left(\lfloor x^*\rfloor^2\right)$. Then from (22) we have that $\rho^{x^*} \le 3O\left(\lfloor x^*\rfloor^2\right)$. However, by Taylor series expansion we have

$$\rho^{x^*} = 1 + x^*\log\rho + \frac{(x^*\log\rho)^2}{2!} + O\left(x^{*3}\right) > O\left(\lfloor x^*\rfloor^2\right).$$

Therefore, we have a contradiction, which means that $\rho^{\widetilde{x}} > 3O(\lfloor x^*\rfloor^2)$. Therefore, by letting $\widetilde{x}, x^* \to \infty$ we have

$$\text{PoA} \le 1 - \frac{\log\rho}{3}\left(\frac{2\lfloor x^*\rfloor\log\rho + \rho^{\lfloor x^*\rfloor+1}}{\log(\rho)\rho^{\lceil\widetilde{x}\rceil+1}}\right) = 1 - \frac{\rho^{\lfloor x^*\rfloor}}{3\rho^{\lceil\widetilde{x}\rceil}},$$

Thus, we have

$$\limsup_{k\to\infty}\text{PoA} \le 1 - \limsup_{k\to\infty}\frac{\rho^{\lfloor x^*\rfloor}}{3\rho^{\lceil\widetilde{x}\rceil}} = 1 - \lim_{k\to\infty}\frac{\rho^{x^*}}{3\rho^{\widetilde{x}}} = 0,$$

where the last step follows directly from (22). ∎

Next, we prove Lemma 6 and 7 to complete the proof.

Proof of Lemma 6: The proof of Lemma 6 consists of three parts: (i) We first take derivative of cost function $T(x;x)$ (cf. (18)) with respect to x and set the derivative equal to 0. By rearranging terms we have the equation $3k\lambda^2/c^2 = W(x)$; (ii) We can show that the equation $3k\lambda^2/c^2 = W(x)$ has one unique solution x^* when $W(0) < k\lambda^2/c^2$ using the similar argument in the proof of Lemma 1(cf. Appendix A). Furthermore, we can show that $T(x;x)$ is decreasing on $[0,x^*)$ and increasing on $[x^*,\infty)$, which implies that optimal threshold can be either $\lfloor x^*\rfloor$ or $\lceil x^*\rceil$; (iii) Finally, we can show that $T(\lfloor x^*\rfloor;\lfloor x^*\rfloor) < T(\lceil x^*\rceil;\lceil x^*\rceil)$ when $W(0) < k\lambda^2/c^2$. The detailed proof is omitted due to space limitation. ∎

Proof of Lemma 7: First, we consider the cost function of optimal threshold decisions determined by some central controller, we have

$$T\left(\lfloor x^*\rfloor;\lfloor x^*\rfloor\right) \ge \frac{Q(x^*)}{\lambda} + \frac{W(x^*)}{3}\pi^3(x^*) = \frac{4x^*+1}{6\lambda} \quad (23)$$

when $\rho = 1$ and

$$\begin{aligned} T\left(\lfloor x^*\rfloor;\lfloor x^*\rfloor\right) &= \frac{Q(\lfloor x^*\rfloor)}{\lambda} + \frac{W(\lfloor x^*\rfloor)}{3}\pi^3(\lfloor x^*\rfloor) \\ &= \frac{1}{3\lambda}\left(\frac{2\lfloor x^*\rfloor+2}{\rho^{\lfloor x^*\rfloor+1}-1} + 2\lfloor x^*\rfloor + \frac{2+\rho}{1-\rho} + \frac{\rho^{\lfloor x^*\rfloor+1}}{\log(\rho)}\right) \quad (24) \end{aligned}$$

when $\rho \ne 1$, where the inequality in (23) follows from the fact that $k\lambda^2/c^2 = W(x^*)/3 \ge W(\lfloor x^*\rfloor)/3$ (cf. Lemma 2 and Lemma 6).

By Theorem 1, we have that the NE must be either $(\lfloor\widetilde{x}\rfloor)_{N\times1}$ or $(\lceil\widetilde{x}\rceil)_{N\times1}$ if it exists. Therefore, we have the following two cases:

- $(\lfloor\widetilde{x}\rfloor)_{N\times1}$ is NE (under the condition $W(\lfloor\widetilde{x}\rfloor) < k\lambda^2/c^2 \le V_1(\lfloor\widetilde{x}\rfloor)$). In such a case, according to the definition of the individual cost function $T(\cdot;\cdot)$, the fact that $k\lambda^2/c^2 \le V_1(\lfloor\widetilde{x}\rfloor)$ and $\lfloor\widetilde{x}\rfloor \le \widetilde{x}$, we have

$$T(\lfloor\widetilde{x}\rfloor;\lfloor\widetilde{x}\rfloor) \le \frac{\widetilde{x}+1}{\lambda},$$

when $\rho = 1$ and

$$T(\lfloor\widetilde{x}\rfloor;\lfloor\widetilde{x}\rfloor) \le \frac{1}{\lambda}\frac{\rho-\rho^{\lfloor\widetilde{x}\rfloor+2}}{1-\rho},$$

when $\rho \ne 1$.

- $(\lceil\widetilde{x}\rceil)_{N\times1}$ is NE (under the condition $V_2(\lceil\widetilde{x}\rceil) \le k\lambda^2/c^2 < W(\lceil\widetilde{x}\rceil)$). In such a case, again according to the definition of the individual cost function $T(\cdot;\cdot)$, the fact that $k\lambda^2/c^2 < W(\lceil\widetilde{x}\rceil)$ and $\lceil\widetilde{x}\rceil \le \widetilde{x}+1$, we have

$$T(\lceil\widetilde{x}\rceil;\lceil\widetilde{x}\rceil) \le \frac{2\widetilde{x}+3}{2\lambda},$$

when $\rho = 1$ and

$$T(\lceil\widetilde{x}\rceil;\lceil\widetilde{x}\rceil) \le \frac{1}{\lambda}\left(\frac{\rho}{1-\rho} + \frac{\rho^{\lceil\widetilde{x}\rceil+1}}{\log(\rho)}\right),$$

when $\rho \ne 1$.

Therefore, we have

$$T(\widetilde{B};\widetilde{B}) \le \max\left\{\frac{\widetilde{x}+1}{\lambda}, \frac{2\widetilde{x}+3}{2\lambda}\right\} = \frac{2\widetilde{x}+3}{2\lambda}, \quad (25)$$

when $\rho = 1$ and

$$T(\widetilde{B};\widetilde{B}) \le \max\left\{\frac{1}{\lambda}\frac{\rho-\rho^{\lfloor\widetilde{x}\rfloor+2}}{1-\rho}, \frac{1}{\lambda}\left(\frac{\rho}{1-\rho} + \frac{\rho^{\lceil\widetilde{x}\rceil+1}}{\log(\rho)}\right)\right\},$$

when $\rho \ne 1$. Next, we show

$$\frac{1}{\lambda}\frac{\rho-\rho^{x+2}}{1-\rho} < \frac{1}{\lambda}\left(\frac{\rho}{1-\rho} + \frac{\rho^{x+2}}{\log(\rho)}\right)$$

to complete the proof. As such, we consider

$$\begin{aligned} &\frac{1}{\lambda}\frac{\rho-\rho^{x+2}}{1-\rho} - \frac{1}{\lambda}\left(\frac{\rho}{1-\rho} + \frac{\rho^{x+2}}{\log(\rho)}\right) \\ =&\frac{-\rho^{x+2}}{\lambda}\left(\frac{\log(\rho)+1-\rho}{(1-\rho)\log(\rho)}\right). \end{aligned}$$

Note that we have $\log(\rho)+1-\rho < 0$, which can be easily shown by basic calculus. Therefore, we have

$$\frac{-\rho^{x+1}}{\lambda}\left(\frac{\log(\rho)+1-\rho}{(1-\rho)\log(\rho)}\right) < 0$$

in both $0 < \rho < 1$ and $\rho > 1$ cases. Hence, we have

$$T\left(\widetilde{B};\widetilde{B}\right) \le \frac{1}{\lambda}\left(\frac{\rho}{1-\rho} + \frac{\rho^{\lceil\widetilde{x}\rceil+1}}{\log(\rho)}\right) \quad (26)$$

when $\rho \ne 1$.

Therefore, substituting (23) and (25) when $\rho = 1$ into the definition of PoA (cf. Section II) or (24) and (26) when $\rho \ne 1$ into the definition of PoA (cf. Section II). Then, we have the desired results. ∎

REFERENCES

[1] A. Sunyaev, "Cloud computing," in *Internet computing*. Springer, 2020, pp. 195–236.

[2] A. Ravulavaru, *Google Cloud AI Services Quick Start Guide: Build Intelligent Applications with Google Cloud AI Services*. Packt Publishing Ltd, 2018.

[3] A. Boukerche, S. Guan, and R. E. D. Grande, "Sustainable offloading in mobile cloud computing: algorithmic design and implementation," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–37, 2019.

[4] A. J. Ferrer, J. M. Marquès, and J. Jorba, "Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–36, 2019.

[5] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 319–333, 2018.

[6] H. Tout, A. Mourad, N. Kara, and C. Talhi, "Multi-persona mobility: Joint cost-effective and resource-aware mobile-edge computation offloading," *IEEE/ACM Transactions on Networking*, 2021.

[7] L. Cui, C. Xu, S. Yang, J. Z. Huang, J. Li, X. Wang, Z. Ming, and N. Lu, "Joint optimization of energy consumption and latency in mobile edge computing for internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4791–4803, 2018.

[8] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571–3584, 2017.

[9] C. Yi, J. Cai, and Z. Su, "A multi-user mobile computation offloading and transmission scheduling mechanism for delay-sensitive applications," *IEEE Transactions on Mobile Computing*, vol. 19, no. 1, pp. 29–43, 2019.

[10] L. Yang, H. Zhang, X. Li, H. Ji, and V. C. Leung, "A distributed computation offloading strategy in small-cell networks integrated with mobile edge computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2762–2773, 2018.

[11] S. Jošilo and G. Dán, "A game theoretic analysis of selfish mobile computation offloading," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[12] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[13] Y. Ge, Y. Zhang, Q. Qiu, and Y.-H. Lu, "A game theoretic resource allocation for overall energy minimization in mobile cloud computing system," in *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*, 2012, pp. 279–284.

[14] X. Qin, W. Xu, and B. Li, "Optimal joint offloading and wireless scheduling for parallel computing with deadlines," in *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2019.

[15] B. Li, "Optimal offloading for dynamic compute-intensive applications in wireless networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[16] Y. Wang, J. Yang, X. Guo, and Z. Qu, "A game-theoretic approach to computation offloading in satellite edge computing," *IEEE Access*, vol. 8, pp. 12 510–12 520, 2019.

[17] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5g services in mobile edge computing systems: Learn from a digital twin," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4692–4707, 2019.

[18] S.-W. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modeling and latency analysis," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5225–5240, 2018.

[19] L. Liu, Z. Chang, X. Guo, and T. Ristaniemi, "Multi-objective optimization for computation offloading in mobile-edge computing," in *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2017, pp. 832–837.

[20] W. Lin and P. Kumar, "Optimal control of a queueing system with two heterogeneous servers," *IEEE Transactions on Automatic control*, vol. 29, no. 8, pp. 696–703, 1984.

[21] M. Shifrin, R. Atar, and I. Cidon, "Optimal scheduling in the hybrid-cloud," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. IEEE, 2013, pp. 51–59.

[22] B. Xia, S. Shakkottai, and V. Subramanian, "Small-scale markets for bilateral resource trading in the sharing economy," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2447–2455.

[23] F. Alotaibi, S. Hosny, H. El Gamal, and A. Eryilmaz, "A game theoretic approach to content trading in proactive wireless networks," in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 2216–2220.

[24] Z. Chen, Y. Liu, B. Zhou, and M. Tao, "Caching incentive design in wireless d2d networks: A stackelberg game approach," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.

[25] J. Li, R. Bhattacharyya, S. Paul, S. Shakkottai, and V. Subramanian, "Incentivizing sharing in realtime d2d streaming networks: A mean field game perspective," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 3–17, 2016.

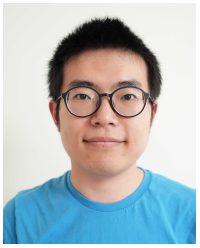
[26] D. Narasimha, S. Shakkottai, and L. Ying, "A mean field game analysis of distributed mac in ultra-dense multichannel wireless networks," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2019, pp. 1–10.

[27] M. A. Abd, S. F. M. Al-Rubeaai, B. K. Singh, K. E. Tepe, and R. Benlamri, "Extending wireless sensor network lifetime with global energy balance," *IEEE Sensors Journal*, vol. 15, no. 9, pp. 5053–5063, 2015.

[28] X. Qin, B. Li, and Y. Lei, "Distributed threshold-based offloading for large-scale mobile cloud computing," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2021.

[29] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir, "Deconstructing amazon ec2 spot instance pricing," *ACM Transactions on Economics and Computation*, vol. 1, no. 3, p. 16, 2013.

[30] M. Harchol-Balter, *Performance modeling and design of computer systems: queueing theory in action*. Cambridge University Press, 2013.



Xudong Qin (Student Member, IEEE) received the B.S. degree in automation from Shandong University, Ji'nan, China, in 2015, and the M.S. degree in control theory and engineering from Northeastern University, Shenyang, China, in 2018. Currently, he is working towards the Ph.D. degree in electrical engineering at The Pennsylvania State University. His current research interests include mobile edge/cloud computing and wireless scheduling design.

Bin Li (S'11-M'16-SM'20) received the B.S. degree in Electronic and Information Engineering, M.S. degree in Communication and Information Engineering, both from Xiamen University, China, and Ph.D. degree in Electrical and Computer Engineering from The Ohio State University. He is currently an associate professor in the Department of Electrical Engineering at the Pennsylvania State University, University Park, PA, USA. His research focuses on the intersection of networking, machine learning, and system developments, and their applications in networking for virtual/augmented reality, mobile edge computing, mobile crowd-learning, and Internet-of-Things. He is a senior member of the IEEE and a member of the ACM. He received both the National Science Foundation (NSF) CAREER Award and Google Faculty Research Award in 2020, and ACM MobiHoc 2018 Best Poster Award.

Lei Ying (F'22) received the B.E. degree from Tsinghua University, Beijing, China, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign. He is currently a Professor with the Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor. His research is broadly in the interplay of complex stochastic systems and big-data, including large-scale communication/computing systems for big-data processing, private data marketplaces, and large-scale graph mining. He coauthored books Communication Networks: An Optimization, Control and Stochastic Networks Perspective (Cambridge University Press, 2014) and Diffusion Source Localization in Large Networks, Synthesis Lectures on Communication Networks (Morgan & Claypool Publishers, 2018). He won the Young Investigator Award from the Defense Threat Reduction Agency (DTRA) in 2009 and NSF CAREER Award in 2010. His research contributions have been recognized as best papers in conferences across different disciplines, including communication networks (INFOCOM and WiOpt), computer systems (SIGMETRICS), and data mining (KDD).