Decentralized Algorithm for Randomized Task Allocation in Fog Computing Systems

Slađana Jošilo and György Dán, *Senior Member, IEEE*

***Abstract*— Fog computing is identified as a key enabler for using various emerging applications by battery powered and computationally constrained devices. In this paper, we consider devices that aim at improving their performance by choosing to offload their computational tasks to nearby devices or to an edge cloud. We develop a game theoretical model of the problem and use a variational inequality theory to compute an equilibrium task allocation in static mixed strategies. Based on the computed equilibrium strategy, we develop a decentralized algorithm for allocating the computational tasks among nearby devices and the edge cloud. We use the extensive simulations to provide insight into the performance of the proposed algorithm and compare its performance with the performance of a myopic best response algorithm that requires global knowledge of the system state. Despite the fact that the proposed algorithm relies on average system parameters only, our results show that it provides a good system performance close to that of the myopic best response algorithm.**

***Index Terms*— Computation offloading, fog computing, game theory, task placement, decentralized resource management.**

I. Introduction

FOG computing is widely recognized as a key component of 5G networks and an enabler of the Internet of Things (IoT) [1], [2]. The concept of fog computing extends the traditional centralized cloud computing architecture by allowing devices not only to use computing and storage resources of centralized clouds, but also resources distributed across the network including the resources of each other and resources located at the network edge [3].

Traditional centralized cloud computing allows devices to offload the computation to a cloud infrastructure with significant computational power [4]–[6]. Cloud offloading may indeed accelerate the execution of applications, but it may suffer from high communication delays, on the one hand due to the contention of devices for radio spectrum, on the other hand due to the remoteness of the cloud infrastructure. Thus, traditional centralized cloud computing may not be able to meet the delay requirements of emerging IoT applications [7]–[10].

Fog computing addresses this problem by allowing collaborative computation offloading among nearby devices and distributed cloud resources close to the network edge [11]. The benefits of collaborative computation offloading are twofold. First, collaboration among devices can make use of device-to-device (D2D) communication, and thereby it can improve spectral efficiency and free up radio resources for other purposes [12]–[14]. Second, the proximity of devices to each other can enable low communication delays. Thus, fog computing allows to explore the tradeoff between traditional centralized cloud offloading, which ensures low computing time, but may suffer from high communication delay, and collaborative computation offloading, which ensures low communication delay, but may involve higher computing times.

One of the main challenges facing the design of fog computing systems is how to manage fog resources efficiently. Compared to traditional centralized cloud computing, where a device only needs to decide whether to offload the computation of a task, in the case of fog computing the number of offloading choices increases with the number of devices. Furthermore, today's devices are heterogeneous in terms of computational capabilities, in terms of what tasks they have to execute and how often. At the same time, some devices may be autonomous, and hence they would be interested in minimizing their own perceived completion times. Therefore, developing low complexity algorithms for efficient task allocation among nearby devices is an inherently challenging problem.

In this paper we address this problem by considering a fog computing system, where devices can choose either to perform their computation locally, to offload the computation to a nearby device, or to offload the computation to an edge cloud. We provide a game theoretical model of the completion time minimization problem. We show that an equilibrium task allocation in static mixed strategies always exists, i.e., if devices can choose at random whether to offload, and where to offload. Based on the game theoretical model we propose a decentralized algorithm that relies on average system parameters, and allocates the tasks according to a Nash equilibrium in static mixed strategies. We use the algorithm to address the important question whether efficient task allocation is feasible using an algorithm that requires low signaling overhead, and we compare the performance achieved by the proposed algorithm with the performance of a myopic best response algorithm that requires global knowledge of the system state. Our results show that the proposed decentralized algorithm, despite significantly lower signaling overhead, provides good system performance close to that of the myopic best response algorithm.

The rest of the paper is organized as follows. We present the system model in Section II. We present two algorithms in Sections III and IV. In Section V we present numerical results and in Section VI we review related work. Section VII concludes the paper.

Manuscript received October 31, 2017; revised May 11, 2018, September 22, 2018, and November 5, 2018; accepted November 7, 2018; approved by IEEE/ACM Transactions on Networking Editor P. Giaccone. Date of publication December 10, 2018; date of current version February 14, 2019. This work was supported by the Swedish Research Council under Grant 621-2014-6. *(Corresponding author: Slađana Jošilo.)*

The authors are with the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden (e-mail: josilo@kth.se; gyuri@kth.se).

Digital Object Identifier 10.1109/TNET.2018.2880874

II. System Model and Problem Formulation

We consider a fog computing system that consists of a set $\mathcal{N}=\{1,2,...,N\}$ of devices, and an edge cloud. Device $i \in \mathcal{N}$ generates a sequence $(t_{i,1}, t_{i,2}, \ldots)$ of computational tasks. We consider that the size $D_{i,k}$ (e.g., in bytes) of task $t_{i,k}$ of device i can be modeled by a random variable D_i, and the number of CPU cycles $L_{i,k}$ required to perform the task by a random variable L_i. According to results reported in [15]–[17] the number X_i of CPU cycles per data bit can be approximated by a Gamma distribution, and thus we can model the relation between L_i and D_i as $L_i = D_i X_i$. Furthermore, assuming that the first moment $\overline{X}_i$ and the second moment ${}^2\overline{X}_i$ of X_i can be estimated based on the past, the statistics of the number of CPU cycles required to perform the task of device i can be easily obtained. Similar to other works [18]–[20], we assume that the task arrival process of device i can be modeled by a Poisson process with arrival intensity λ_i.

For each task $t_{i,k}$ device i can decide whether to perform the task locally, to offload it to a device $j \in \mathcal{N} \setminus \{i\}$ or to an edge cloud. Thus, device i chooses a member of the set $\mathcal{N} \cup \{0\}$, where 0 corresponds to the edge cloud. We allow for randomized policies, and we denote by $p_{i,j}(k)$ the probability that device i assigns its task $t_{i,k}$ to $j \in \mathcal{N} \cup \{0\}$, and we define the probability vector $p_i(k) = \{p_{i,0}(k), p_{i,1}(k), ..., p_{i,N}(k)\}$, where $\sum_{j \in \mathcal{N} \cup \{0\}} p_{i,j}(k) = 1$. Finally, we denote by $\mathcal{P}$ the set of probability distributions over $\mathcal{N} \cup \{0\}$, i.e., $p_i(k) \in \mathcal{P}$.

The above fog computing system relies on the assumption that all devices faithfully execute the tasks offloaded to them. To ensure this, the devices need to be incentivized to collaborate in executing each others' computational tasks, as discussed in [21]. The collaboration among devices in fog computing systems can be ensured with an adequate incentive scheme similar to those used in peer-to-peer systems [22]–[24]. These schemes ensure the collaboration among the peers through the reputation-based trust supporting mechanism. In the context of fog computing systems, the mechanism would result in an incentive scheme in which only devices that process offloaded tasks themselves are entitled to offload the tasks.

A. Communication Model

We consider that the devices communicate using an orthogonal frequency division multiple access (OFDMA) framework in which there is an assignment of subcarriers to pairs of communicating nodes [25], [26]. Furthermore, we consider that devices use dedicated bandwidth resources, i.e. node-to-node pairs do not share the bandwidth with each other and with the other cellular users [25]. This can be implemented by assigning an orthogonal subcarrier per transmission direction for each pair of communicating nodes, resulting in $N \times N$ subcarriers in total. We denote the transmission rate from device i to device j by $R_{i,j}$, and the transmission rate from device i to the edge cloud through a base station by $R_{i,0}$. Each device maintains N transmission queues, i.e., $N-1$ queues for transmitting to devices $j \in \mathcal{N} \setminus \{i\}$ and one for transmitting to the edge cloud, and the tasks are transmitted in FIFO order.

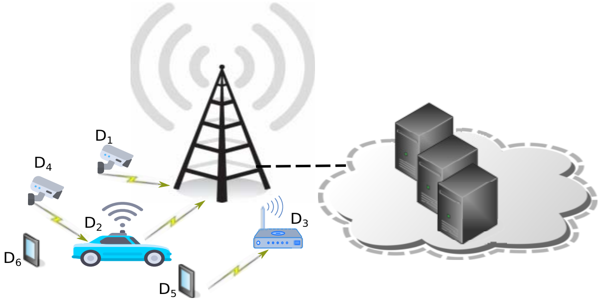

Fig. 1. Fog computing system that consists of 6 devices and an edge cloud.

We consider that the time $T^t_{i,j}(k)$ needed to transmit a task $t_{i,k}$ from device i to $j \in \mathcal{N} \cup \{0\}$ is proportional to its size $D_{i,k}$, and is given by

$$T^t_{i,j}(k) = D_{i,k}/R_{i,j}.$$

Furthermore, the time $T^d_{i,j}(k)$ needed to deliver the input data $D_{i,k}$ from device i to $j \in \mathcal{N} \cup \{0\}$ is the sum of the transmission time $T^t_{i,j}(k)$ and of the waiting time (if any).

Similar to other works [27]–[30], we consider that the time needed to transmit the results of the computation back to the device is negligible. This assumption is justified for many applications including face and object recognition, and anomaly detection, where the size of the result of the computation is much smaller than the size of the input data.

Observe that our system model can accommodate systems in which certain devices $i \in \mathcal{N}$ only serve for performing the computational tasks of others, by setting the arrival intensity $\lambda_i = 0$. These devices can be considered as micro-data centers located at the network edge, whose function in fog computing systems is to perform the computational tasks of the other devices [31], [32]. Furthermore, our system model can accommodate systems in which certain devices $j \in \mathcal{N}$ are not supposed to perform the computational tasks of others, by setting the transmission rates $R_{i,j}$ from the other devices $i \in \mathcal{N} \setminus \{j\}$ to device j to low enough values.

Figure 1 illustrates a fog computing system that consists of six devices and one edge cloud; device 1 and device 2 offload their tasks through a base station to the cloud server, device 4 offloads its tasks to device 2, device 5 offloads its task to device 3 that serves as a micro-data center, and device 6 performs computation locally.

B. Computation Model

To model the time that is needed to compute a task in a device i, we consider that each device i maintains one execution queue with tasks served in FIFO order. We denote by F_i the computational capability of device i. Unlike devices, the cloud server has a large number of processors with computational capability F_0 each, and we assume that computing in the edge cloud begins immediately upon arrival of a task.

Similar to common practice [21], [27] we consider that the time $T^c_{i,j}(k)$ needed to compute a task $t_{i,k}$, on $j \in \mathcal{N} \cup \{0\}$ is proportional to its complexity $L_{i,k}$, and is given by

$$T^c_{i,j}(k) = L_{i,k}/F_j.$$

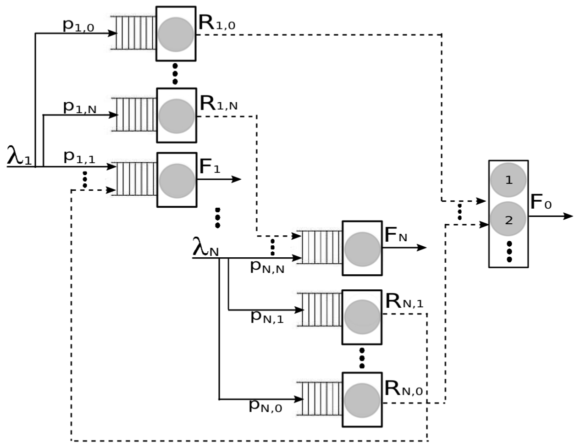

Fig. 2. Fog computing system modeled as a queuing network.

```
$p_i(k) = \textit{MyopicBestResponse}(t_{i,k})$
$p_{i,j}(k) = 0, \quad \forall j \in \mathcal{N} \cup \{0\}$
/* Estimate completion time of $t_{i,k}$ in $\forall j \in \mathcal{N} \cup \{0\}$ */
for $j = 0, \ldots, N$ do
    if $j = i$ then
        $ECompleteT(j) = T^e_{i,j}(k)$
    else
        $ECompleteT(j) = T^d_{i,j}(k) + T^e_{i,j}(k)$
    end if
end for
/* Make a greedy decision */
$i' \leftarrow \underset{\{j \in \mathcal{N} \cup \{0\}}{\arg\min}\, ECompleteT(j)$
$p_{i,i'}(k) = 1$
return $p_i(k)$
```

Fig. 3. Pseudo code of myopic best response.

Furthermore, the execution time $T^e_{i,j}(k)$ of a task $t_{i,k}$ on device j is the sum of the computation time $T^c_{i,j}(k)$ and of the waiting time (if any). Figure 2 illustrates the queuing model of a computation offloading system.

C. Problem Formulation

We define the cost C_i of device i as the mean completion time of its tasks. Given a sequence $(t_{i,1}, t_{i,2}, \ldots)$ of computational tasks, we can thus express the cost C_i as

$$C_i = \lim_{K\to\infty} \frac{1}{K}\left[\sum_{k=1}^{K}\Bigg(p_{i,i}(k)T^e_{i,i}(k) + \sum\nolimits_{j\in\mathcal{N}\setminus\{i\}\cup\{0\}} p_{i,j}(k)\big(T^d_{i,j}(k) + T^e_{i,j}(k)\big)\Bigg)\right]. \quad (1)$$

Since the devices are autonomous, we consider that each device aims at minimizing its cost by solving

$$\min C_i \quad \text{s.t.} \quad (2)$$

$$p_i(k) \in \mathcal{P}. \quad (3)$$

Since devices' decisions affect each other, the devices play a dynamic non-cooperative game, and we refer to the game as the *multi user computation offloading game* (MCOG). The game is closest to an undiscounted stochastic game with countably infinite state space, but the system state evolves according to a semi-Markov chain (instead of a Markov chain, depending on the distribution of D_i and L_i) and payoffs (the completion times) are unbounded. We are not aware of existence results for Markov equilibria for this class of problem, and even for the case when the state evolves according to a Markov chain with countable state space and unbounded payoffs, there are only a few results on the existence of equilibria in Markov strategies [33]–[35].

D. Decentralized Solution Supported by a Centralized Entity

Since fog computing architecture is decentralized in nature, and devices in fog computing systems are expected to be autonomous [11], [36] we are interested in developing decentralized algorithms that will allow devices to make their offloading decisions locally. Motivated by widely considered hierarchical fog computing architectures [37], [38], we consider that there is a single central entity with a high level of hierarchy that collects and stores the information about the fog computing system. The entity need not be a single physical entity, but a single logically centralized entity that can handle high loads and can be resilient to failure.

Furthermore, we consider that the entity periodically sends the needed information to the devices and thus supports them in making their offloading decisions. Intuitively, more information about the system state will allow devices to make better offloading decisions, but at the cost of increased signaling overhead. Therefore, one important objective when developing decentralized algorithms for allocating the computational tasks is to achieve good system performance at the cost of an acceptable signaling overhead. With this in mind, in what follows we propose and discuss two decentralized solutions for the MCOG problem in the form of a Markov strategy and in static mixed strategies, respectively.

III. Myopic Best Response

The first algorithm we consider, called *Myopic Best Response* (MBR), requires global knowledge of the system state, but decisions are made locally at the devices. Similar to the *WaterFilling* algorithm proposed in [39], in the MBR algorithm every device i makes a decision based on a myopic best response strategy, i.e., every device i chooses a node $j \in \mathcal{N} \cup \{0\}$ that minimizes the completion time of its task $t_{i,k}$, given the instantaneous state of the queuing network. The pseudo-code for computing the myopic best response strategy is shown in Figure 3. Note that since the devices make their decisions based on the instantaneous states of the queues, they do not take into account the tasks that may arrive to the other devices' execution queues while transmitting a task. Futhermore, if the devices' execution queues were stable if all devices perform all tasks locally, then under the MBR algorithm the queue lengths do not grow unbounded since each device chooses the destination node based on the instantaneous state of the queues.

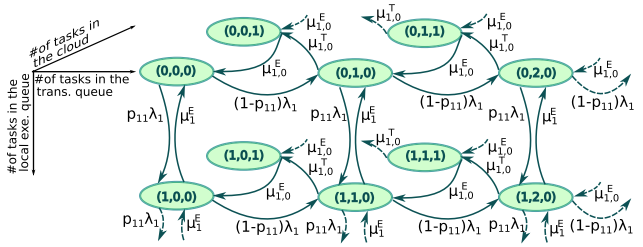

Fig. 4. State transition diagram of the semi-Markov process induced by the offloading decisions for the single device case ($N = 1$).

Note that if we define the system state upon the arrival of task $t_{i,k}$ as the number of jobs in the transmission and execution queues, then the devices' decisions depend on the instantaneous system state only, and hence the myopic best response is a Markov strategy for the *MCOG*. Nonetheless, it is not necessarily a Markov perfect equilibrium.

In a system with N devices we have $N \times N$ transmission queues and $N + 1$ execution queues, and we can thus model the system as an $N \times (N + 1) + 1$ dimensional semi-Markov process.

Example 1: Figure 4 shows the state transition diagram for a single device, i.e., $N = 1$, which is three dimensional. We use the triplet (n_l, n_t, n_0) to denote the system state, where n_l, n_t and n_0 stand for the number of tasks in the local execution queue, number of tasks in the transmission queue and the number of tasks in the cloud server, respectively. Since $N = 1$, a device only needs to decide whether to offload the computation to the edge cloud or to perform the computation locally and hence the transition intensities from state (n_l, n_t, n_0) to state $(n_l, n_t + 1, n_0)$ and from state (n_l, n_t, n_0) to state $(n_l + 1, n_t, n_0)$ are $(1 - p_{1,1})\lambda_1$ and $p_{1,1}\lambda_1$, respectively. In the case of computation offloading, the task with size D_1 and complexity L_1 needs to be transmitted to the edge cloud at rate $R_{1,0}$ and executed with computational capability F_0 and thus the transition intensities from state (n_l, n_t, n_0) to state $(n_l, n_t - 1, n_0 + 1)$ and from state (n_l, n_t, n_0) to state $(n_l, n_t, n_0 - 1)$ are $\mu^T_{1,0} = D_1/R_{1,0}$ and $\mu^E_{1,0} = n_0 L_1/F_0$, respectively. Finally, in the case of local execution the task with complexity L_1 needs to be executed locally with local computational capability F_1 and hence the transition intensity from state (n_l, n_t, n_0) to state $(n_l - 1, n_t, n_0)$ is $\mu^E_1 = L_1/F_1$.

A significant detriment of the MBR algorithm is its signaling overhead, as it requires global information about the system state upon the arrival of each task. To reduce the signaling requirements, in what follows we propose an algorithm that is based on a strategy that relies on average system parameters only.

IV. Equilibrium in Static Mixed Strategies

As a practical alternative to the MBR algorithm, in this section we propose a decentralized algorithm, which we refer to as the *Static Mixed Nash Equilibrium* (SM-NE) algorithm. The algorithm is based on an equilibrium $(p_i)_{i\in\mathcal{N}}$ in static mixed strategies, that is, device i chooses the node where to execute an arriving task at random according to the probability vector p_i, which is the same for all tasks. For computing a static mixed strategy, it is enough for a device to know the average task arrival intensities, transmission rates, and the first and second moments of the task size and the task complexity distribution. Therefore, the SM-NE algorithm requires significantly less signaling than the MBR algorithm.

In order to compute an equilibrium strategy, we start with expressing the (approximate) equilibrium cost of device i as a function of strategy profile $(p_i)_{i\in\mathcal{N}}$, i.e., the mean completion time of its tasks in steady state. Throughout the section we denote by $\overline{D}_i$ and ${}^2\overline{D}_i$ the first and the second moment of D_i, respectively, and by $\overline{L}_i$ and ${}^2\overline{L}_i$ the first and the second moment of L_i, respectively.

A. Transmission Time in Steady State

Since tasks arrive to each device as a Poisson process and we aim for a constant probability vector p_i as a solution, the arrival processes to the transmission queues are Poisson processes. If the transmission queues are sufficiently large, we can approximate them as infinite, similar to [20], and thus we can model each transmission queue as an $M/G/1$ system. Let us denote by $\overline{T^t}_{i,j}$ and ${}^2\overline{T^t}_{i,j}$ the mean and the second moment of the time needed to transmit a task from device i to $j \in \mathcal{N}\setminus\{i\}\cup\{0\}$, respectively. Then the mean time $\overline{T^d}_{i,j}$ needed to deliver the input data from device i to $j \in \mathcal{N}\setminus\{i\}\cup\{0\}$ is the sum of the mean waiting time in the transmission queue and the mean transmission time $\overline{T^t}_{i,j}$, and can be expressed as

$$\overline{T^d}_{i,j} = \frac{p_{i,j}\lambda_i{}^2\overline{T^t}_{i,j}}{2(1 - p_{i,j}\lambda_i\overline{T^t}_{i,j})} + \overline{T^t}_{i,j}, \qquad (4)$$

and the queue is stable as long as the offered load $\rho^t_{i,j} = p_{i,j}\lambda_i\overline{T^t}_{i,j} < 1$.

B. Computation Time in Steady State

Observe that if the input data size D_i follows an exponential distribution, then departures from the transmission queues can be modeled by a Poisson process, and thus tasks arrive to the devices' execution queues according to a Poisson process. In what follows we use the approximation that the tasks arrive according to a Poisson process even if D_i is not exponentially distributed. Furthermore, following common practice [19], [40], for analytical tractability we approximate the execution queues as being infinite. This approximation is reasonable if the queues are sufficiently large. These two approximations allow us to model the execution queue of each device as an $M/G/1$ system, and the edge cloud as an $M/G/\infty$ system.

Let us denote by $\overline{T^c}_{i,j}$ and ${}^2\overline{T^c}_{i,j}$ the mean and the second moment of the time needed to compute device i's task on $j \in \mathcal{N} \cup \{0\}$, respectively. Then the mean time $\overline{T^e}_{i,j}$ that device $j \in \mathcal{N}$ needs to complete the execution of device i's task is the sum of the mean waiting time in the execution queue and the mean computation time $\overline{T^c}_{i,j}$, and can be expressed as

$$\overline{T^e}_{i,j} = \frac{\sum_{i'\in\mathcal{N}} p_{i',j}\lambda_{i'}{}^2\overline{T^c}_{i',j}}{2(1 - \sum_{i'\in\mathcal{N}} p_{i',j}\lambda_{i'}\overline{T^c}_{i',j})} + \overline{T^c}_{i,j}, \qquad (5)$$

and the queue is stable as long the offered load $\rho^e_j = \sum_{i'\in\mathcal{N}} p_{i',j}\lambda_{i'}\overline{T^c}_{i',j} < 1$.

Since computing in the edge cloud begins immediately upon arrival of a task, the mean time $\overline{T^e}_{i,0}$ that the cloud needs to complete the execution of device i's task is equal to the mean computation time $\overline{T^c}_{i,0}$, i.e.,

$$\overline{T^e}_{i,0} = \overline{L}_i / F_0. \tag{6}$$

C. Existence of Static Mixed Strategy Equilibrium

We can rewrite (1) to express the cost C_i of device i in steady state as a function of $(p_i)_{i\in\mathcal{N}}$,

$$C_i(p_i,p_{-i}) = p_{i,i}\overline{T^e}_{i,i} + \sum\nolimits_{j\in\mathcal{N}\setminus\{i\}\cup\{0\}} p_{i,j}\big(\overline{T^d}_{i,j} + \overline{T^e}_{i,j}\big),$$

where we use p_{-i} to denote the strategies of all devices except device i.

Observe that static mixed strategy profile $(p_i)_{i\in\mathcal{N}}$ of the devices has to ensure that the entire system is stable in steady state, and we assume that the load is such that there is at least one strategy profile that satisfies the stability condition of the entire system. Now, we can define the set of feasible strategies of device i as the set of probability vectors that ensure stability of the transmission and the execution queues

$$\mathcal{K}_i(p_{-i}) = \{p_i \in \mathcal{P} | \rho^t_{i,j} \leq S_t, \rho^e_{i'} \leq S_t, \forall j \in \mathcal{N}\setminus\{i\}\cup\{0\}, \forall i'\},$$

where $0 < S_t < 1$ is the stability threshold associated with the transmission and the execution queues.

Note that due to the stability constraints the set of feasible strategies $\mathcal{K}_i(p_{-i})$ of device i depends on the other devices' strategies, and we are interested in whether there is a strategy profile $(p^*_i)_{i\in\mathcal{N}}$, such that

$$C_i(p^*_i, p^*_{-i}) \leq C_i(p_i, p^*_{-i}), \quad \forall p_i \in \mathcal{K}_i(p^*_{-i}).$$

We are now ready to formulate the first main result of the section.

Theorem 1: The MCOG has at least one equilibrium in static mixed strategies.

In the rest of this subsection we use *variational inequality* (VI) theory to prove the theorem and for computing an equilibrium. For a given set $\mathcal{K} \subseteq \mathbb{R}^n$ and a function $F : \mathcal{K} \to \mathbb{R}^n$, the $VI(\mathcal{K}, F)$ problem is the problem of finding a point $x^* \in \mathcal{K}$ such that $F(x^*)^T(x - x^*) \geq 0$, for $\forall x \in \mathcal{K}$. We define the set $\mathcal{K}$ as

$$\mathcal{K} = \{(p_i)_{i\in N} | p_i \in \mathcal{P}, \rho^t_{i,j} \leq S_t, \rho^e_i \leq S_t, j \in \mathcal{N}\setminus\{i\}\cup\{0\}, \forall i\}.$$

Before we prove the theorem, in the following we formulate an important result concerning the cost function $C_i(p_i, p_{-i})$.

Lemma 1: $C_i(p_i,p_{-i})$ is a convex function of p_i for any fixed p_{-i} and $(p_i,p_{-i}) \in \mathcal{K}$.

Proof: For notational convenience let us start the proof with introducing a few shorthand notations,

$$\gamma_{i,j} = p_{i,j}\lambda_i^2 \overline{T^t}_{i,j}, \quad \delta_i = \sum_{j\in\mathcal{N}} p_{j,i}\lambda_j^2 \overline{T^c}_{j,i},$$
$$\epsilon_{i,j} = 1 - \rho^t_{i,j}, \quad \zeta_i = 1 - \rho^e_i.$$

Using this notation we expand the cost $C_i(p_i, p_{-i})$ as

$$C_i(p_i,p_{-i}) = p_{i,i}\big(\frac{\delta_i}{2\zeta_i} + \overline{T^c}_{i,i}\big) + p_{i,0}\big(\frac{\gamma_{i,0}}{2\epsilon_{i,0}} + \overline{T^t}_{i,0} + \overline{T^c}_{i,0}\big)$$
$$+ \sum_{j\in\mathcal{N}\setminus\{i\}} p_{i,j}\big(\frac{\gamma_{i,j}}{2\epsilon_{i,j}} + \overline{T^t}_{i,j} + \frac{\delta_j}{2\zeta_j} + \overline{T^c}_{i,j}\big).$$

To prove convexity we proceed with expressing the first order derivatives $h_{i,j} = \frac{\partial C_i(p_i,p_{-i})}{\partial p_{i,j}}$,

$$h_{i,0} = \overline{T^t}_{i,0} + \overline{T^c}_{i,0} + \frac{\gamma_{i,0}}{2\epsilon_{i,0}} + p_{i,0}\lambda_i\big(\frac{{}^{2}\overline{T^t}_{i,0}}{2\epsilon_{i,0}} + \frac{\overline{T^t}_{i,0}\gamma_{i,0}}{2\epsilon^2_{i,0}}\big),$$
$$h_{i,i} = \overline{T^c}_{i,i} + \frac{\delta_i}{2\zeta_i} + p_{i,i}\lambda_i\big(\frac{{}^{2}\overline{T^c}_{i,i}}{2\zeta_i} + \frac{\overline{T^c}_{i,i}\delta_i}{2\zeta_i^2}\big),$$
$$h_{i,j}\big|_{j\neq i} = \overline{T^t}_{i,j} + \overline{T^c}_{i,j} + \frac{\gamma_{i,j}}{2\epsilon_{i,j}} + \frac{\delta_j}{2\zeta_j}$$
$$+ p_{i,j}\lambda_i\big(\frac{{}^{2}\overline{T^t}_{i,j}}{2\epsilon_{i,j}} + \frac{{}^{2}\overline{T^c}_{i,j}}{2\zeta_j} + \frac{\overline{T^t}_{i,j}\gamma_{i,j}}{2\epsilon^2_{i,j}} + \frac{\overline{T^c}_{i,j}\delta_j}{2\zeta_j^2}\big).$$

We can now express the Hessian matrix

$$H_i(p_i,p_{-i}) = \begin{pmatrix} h^i_{i,0} & 0 & \dots & 0 \\ 0 & h^i_{i,1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & h^i_{i,N} \end{pmatrix},$$

where $h^i_{i,j} = \frac{\partial^2 C_i(p_i,p_{-i})}{\partial p^2_{i,j}}$, and

$$h^i_{i,0} = \frac{\lambda_i}{\epsilon_{i,0}}\big({}^{2}\overline{T^t}_{i,0} + \frac{\gamma_{i,0}T^t_{i,0}}{\epsilon_{i,0}}\big)\big(1 + p_{i,0}\frac{\lambda_i T^t_{i,0}}{\epsilon_{i,0}}\big),$$
$$h^i_{i,i} = \frac{\lambda_i}{\zeta_i}\big({}^{2}\overline{T^c}_{i,i} + \frac{\delta_i T^c_{i,i}}{\zeta_i}\big)\big(1 + p_{i,i}\frac{\lambda_i T^c_{i,i}}{\zeta_i}\big),$$
$$h^i_{i,j}\big|_{j\neq i} = \frac{\lambda_i}{\epsilon_{i,j}}\big({}^{2}\overline{T^t}_{i,j} + \frac{\gamma_{i,j}T^t_{i,j}}{\epsilon_{i,j}}\big)\big(1 + p_{i,j}\frac{\lambda_i T^t_{i,j}}{\epsilon_{i,j}}\big)$$
$$+ \frac{\lambda_i}{\zeta_j}\big({}^{2}\overline{T^c}_{i,j} + \frac{\delta_j T^c_{i,j}}{\zeta_j}\big)\big(1 + p_{i,j}\frac{\lambda_i T^c_{i,j}}{\zeta_j}\big).$$

Observe that all diagonal elements of $H_i(p_i,p_{-i})$ are non-negative, and thus the Hessian matrix $H_i(p_i,p_{-i})$ is positive semidefinite on $\mathcal{K}$, which implies convexity. □

We are now ready to prove Theorem 1.

Proof of Theorem 1: Let us define the generalized Nash equilibrium problem $\Gamma^s =< \mathcal{N}, (\mathcal{P})_{i\in\mathcal{N}}, (C_i)_{i\in\mathcal{N}} >$, subject to $(p_i)_{i\in\mathcal{N}} \in \mathcal{K}$. Γ^s is a strategic game, in which each device $i \in \mathcal{N}$ plays a *mixed strategy* $p_i \in \mathcal{K}_i(p_{-i})$, and aims at minimizing its cost C_i by solving

$$\min_{p_i} C_i(p_i,p_{-i}) \quad \text{s.t.} \tag{7}$$
$$p_i \in \mathcal{K}_i(p_{-i}). \tag{8}$$

Clearly, a pure strategy Nash equilibrium $(p^*_i)_{i\in\mathcal{N}}$ of Γ^s is an equilibrium of the *MCOG* in static mixed strategies, as

$$C_i(p^*_i, p^*_{-i}) \leq C_i(p_i, p^*_{-i}), \quad \forall p_i \in \mathcal{K}_i(p^*_{-i}).$$

We thus have to prove that Γ^s has a pure strategy Nash equilibrium.

To do so, let us first define the function

$$F = \begin{pmatrix} \nabla_{p_1} C_1(p_1, p_{-1}) \\ \vdots \\ \nabla_{p_N} C_N(p_N, p_{-N}) \end{pmatrix},$$

where $\nabla_{p_i} C_i(p_i, p_{-i})$ is the gradient vector given by

$$\nabla_{p_i} C_i(p_i, p_{-i}) = \begin{pmatrix} h_{i,0} \\ h_{i,1} \\ \vdots \\ h_{i,N} \end{pmatrix}.$$

We prove the theorem in two steps based on the VI($\mathcal{K}, F$) problem, which corresponds to Γ^s.

First, we prove that the solution set of the VI($\mathcal{K}, F$) problem is nonempty and compact. Since the first order derivatives $h_{i,j}$ are rational functions, the function F is infinitely differentiable at every point in $\mathcal{K}$, and hence it is continuous on $\mathcal{K}$. Furthermore, the set $\mathcal{K}$ is compact and convex. Hence, the solution set of the VI($\mathcal{K}, F$) problem is nonempty and compact ([41, Corollary 2.2.5]).

Second, we prove that any solution of the VI($\mathcal{K}, F$) problem is an equilibrium of the MCOG. Since the function F is continuous on $\mathcal{K}$, it follows that $C_i(p_i, p_{-i})$ is continuously differentiable on $\mathcal{K}$. Furthermore, by Lemma 1 we know that $C_i(p_i, p_{-i})$ is a convex function. Therefore, any solution of the VI($\mathcal{K}, F$) problem is a pure strategy Nash equilibrium of Γ^s [42], and is thus an equilibrium in static mixed strategies of *MCOG*. This proves the theorem. □

Theorem 1 guarantees that the MCOG possesses at least one equilibrium in static mixed strategies, according to which the SM-NE algorithm allocates the tasks among the devices and the edge cloud. The next important question is whether there is an efficient algorithm for solving the VI problem, and hence for computing an equilibrium $(p_i^*)_{i\in\mathcal{N}}$ of the *MCOG* in static mixed strategies.

In what follows we show that an equilibrium can be computed efficiently under certain conditions. To do so, we show that the function F is monotone if the execution queue of each device can be modeled by an $M/M/1$ system and all task arrival intensities are equal. Monotonicity of F is a sufficient condition for various algorithms proposed for solving VIs [43], e.g., for the *Solodov-Tseng Projection-Contraction* (ST-PC) method.

Theorem 2: If the task sizes and complexities are exponentially distributed, arrival intensities $\lambda_i = \lambda$ and

$$\lambda \max_{j\in\mathcal{N}} \overline{T^c}_{j,i} \leq \frac{1 - S_t}{N}, \quad \forall i \in \mathcal{N},$$

then the function F is monotone.
The proof is given in Appendix A.

Note that the sufficient condition provided by Theorem 2 ensures stability of all execution queues in the worst case scenario, i.e., when $\overline{T^c}_{j,i} = \max_{j\in\mathcal{N}} \overline{T^c}_{j,i}$ for all devices. This condition is, however, not necessary for function F to be monotone in realistic scenarios. In fact, our simulations showed that the ST-PC method converges to an equilibrium for various considered scenarios.

V. Numerical Results

In what follows we show simulation results obtained using an event driven simulator, in which we implemented the MBR and SM-NE algorithms. For the *ST-PC* method we set $p_{i,i} = 1$, $\forall i \in \mathcal{N}$ as starting point, which corresponds to the strategy profile in which each device performs all tasks locally. The *ST-PC* method stops when the norm of the difference of two successive iterations is less than 10^{-4}.

Similar to [44] and [45], we placed the devices at random on a *regular grid* with 10^4 points defined over a square area of $1\ km \times 1\ km$, and we placed the edge cloud at the center of the grid as in [44]. Unless otherwise noted, we consider that the wired link latency τ_c incurred during communication with the cloud server can be neglected since the cloud is located in close proximity of devices [46]. For simplicity, we consider a static bandwidth assignment for the simulations; we assign a bandwidth of $B_{i,j} = 5$ *MHz* for communication between device i and device j [47], [48], and for the device to cloud bandwidth assignment we consider two scenarios. In the *elastic* scenario the bandwidth $B_{i,0}$ assigned for communication between device i and the edge cloud is independent of the number of devices. In the *fixed* scenario the devices share a fixed amount of bandwidth B_0 when they want to offload a task to the edge cloud, and the bandwidth $B_{i,0}$ scales directly proportional with the number of devices, i.e., $B_{i,0} = \frac{1}{N} B_0$. We consider that the channel gain of device i to a node $j \in \mathcal{N} \setminus \{i\} \cup \{0\}$ is proportional to $d_{i,j}^{-\alpha}$, where $d_{i,j}$ is the distance between device i and node j, and α is the path loss exponent, which we set to 4 according to the path loss model in urban and suburban areas [49]. We set the data transmit power P_i^t of every device i to 0.4 *W* according to [50] and given the bandwidth $B_{i,j}$ available for the communication between nodes i and j we calculate the noise power P_n as $P_n = B_{i,j} N_0$, where $N_0 = 1.38065 \times 10^{-23} T$ is the spectral density for the termal noise at the temperature $T = 290K$. Finally, we calculate the transmission rate $R_{i,j}$ from device i to node $j \in \mathcal{N} \setminus \{i\} \cup \{0\}$ as $R_{i,j} = B_{i,j} log_2(1 + P_i^t d_{i,j}^{-\alpha} / P_n)$.

The input data size D_i follows a uniform distribution on $[a_i^d, b_i^d]$, where a_i^d and b_i^d are uniformly distributed on $[0.1, 1.4]$ *Mb* and on $[2.2, 3.4]$ *Mb*, respectively. The arrival intensity λ_i of the tasks of device i is uniformly distributed on $[0.01, 0.03]$ *tasks/s*, and the stability threshold is $S_t = 0.6$. Note that for the above set of parameters the maximum arrival intensity does not satisfy the sufficient condition of Theorem 2 already for $N = 20$ devices. Yet, our evaluation shows that the *ST-PC* method converges even for larger instances of the problem.

The computational capability F_i of device i is drawn from a continuous uniform distribution on $[1, 4]$ *GHz*, while the computation capability of the edge cloud is $F_0 = 64$ *GHz* [51]. The task complexity L_i follows a uniform distribution on $[a_i^l, b_i^l]$, where a_i^l and b_i^l are uniformly distributed on $[0.2, 0.5]$ *Gcycles* and $[0.7, 1]$ *Gcycles*, respectively.

We use three algorithms as a basis for comparison. The first algorithm computes the socially optimal static mixed strategy profile $(\bar{p}_i)_{i\in\mathcal{N}}$ that minimizes the system cost $C = \frac{1}{N}\sum_{i\in\mathcal{N}} C_i$, i.e., $(\bar{p}_i)_{i\in\mathcal{N}} = \arg\min_{(p_i)_{i\in\mathcal{N}}} C$. We refer to this algorithm as the *Static Mixed Optimal* (SM-OPT) algorithm. The second algorithm considers that the devices are allowed to offload the tasks to the edge cloud only (i.e., $p_{i,i} + p_{i,0} = 1$), and we refer to this algorithm as the *Static Mixed Cloud Nash Equilibrium* (SMC-NE) algorithm. The third

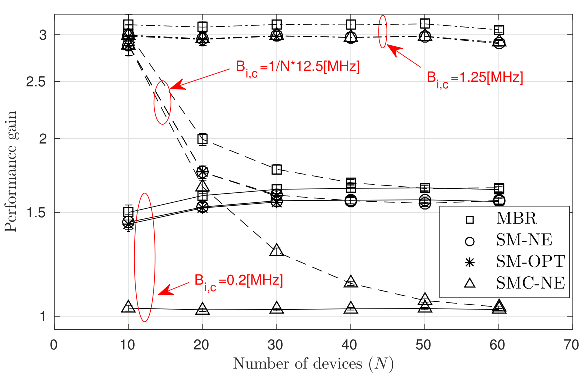

Fig. 5. *Performance gain* vs. number of devices for $B_{i,0} = 0.2$ *MHz*, $B_{i,0} = 1.25$ *MHz* and $B_{i,0} = \frac{1}{N}12.5$ *MHz*.

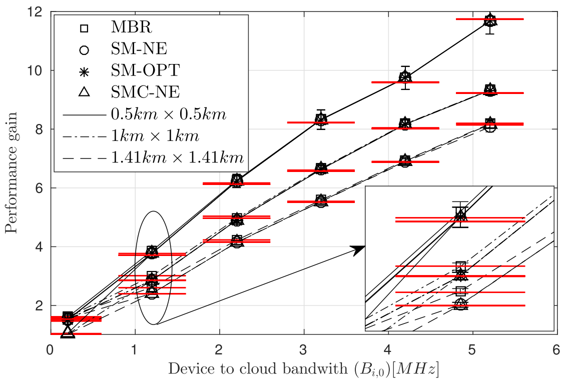

Fig. 6. *Performance gain* vs. device to cloud bandwidth $B_{i,0}$ for $N = 8$ devices placed over $0.5\ km \times 0.5\ km$ square area, for $N = 30$ devices placed over $1\ km \times 1\ km$ square area, and for $N = 60$ devices placed over $1.41\ km \times 1.41\ km$ square area.

algorithm considers that all devices perform local execution (i.e., $p_{i,i} = 1$). Furthermore, we define the performance gain of an algorithm as the ratio between the system cost reached when all devices perform local execution and the system cost reached by the algorithm. For the SM-OPT algorithm the results are shown only up to 30 or 35 devices, because the computation of the socially optimal strategy profile was computationally infeasible for larger problem instances. The results shown in all figures are the averages of 50 simulations, together with 95% confidence intervals.

A. *Performance Gain*

We start with evaluating the performance gain as a function of the number of devices. Figure 5 shows the performance gain for the MBR, SM-NE, SM-OPT and SMC-NE algorithms as a function of the number of devices for the two scenarios of device to cloud bandwidth assignment. For the *elastic* scenario $B_{i,0} = 0.2$ *MHz* and $B_{i,0} = 1.25$ *MHz*, and for the *fixed* scenario $B_0 = 12.5$ *MHz*.

The results show that the SM-NE and the SM-OPT algorithms perform close to the MBR algorithm, despite the fact that they are based on average system parameters only. We can also observe that when the device to cloud bandwidth is low (about 0.2 *MHz*), SMC-NE does not provide significant gain compared to local execution (the performance gain is close to one for all values of N). On the contrary, the MBR, SM-NE and SM-OPT algorithms, which allow collaborative offloading, provide a performance gain of about 50%, and the gain slightly increases with the number of devices. The reason for the slight increase of the gain is that when there are more devices, devices are closer to each other on average, which allows higher transmission rates between devices.

Compared to the case when $B_{i,0} = 0.2$ *MHz*, the results for $B_{i,0} = 1.25$ *MHz* show that all algorithms achieve very high performance gains (up to 300%). Furthermore, the performance gain of the SMC-NE algorithm is similar to that of the SM-NE and the SM-OPT algorithms, while the MBR algorithm performs slightly better. The reason is that for high device to cloud bandwidth in the static mixed equilibrium most devices offload to the edge cloud, as on average it is best to do so, even if given the instantaneous system state it may be better to offload to a device, as done by the MBR algorithm. Furthermore, unlike for $B_{i,0} = 0.2$ *MHz*, for $B_{i,0} = 1.25$ *MHz* the performance gain becomes fairly insensitive to the number of devices, which is again due to the increased reliance on the cloud resources for computation offloading.

The results are fairly different for the *fixed* device to cloud bandwidth assignment scenario, as in this scenario the number of devices affects the device to cloud bandwidth. In this scenario collaboration among the devices improves the system performance (SMC-NE vs. SM-NE algorithms). We can also observe that as N increases, the curves for *fixed* scenario approach the curves for the *elastic* scenario for $B_{i,0} = 0.2$ *MHz*. This is due to that for large values of N the device to cloud bandwidth $B_{i,0}$ becomes low and the devices offload more to each other than to the edge cloud.

Finally, the results show that the gap between the *SM-NE* and the *SM-OPT* algorithms is almost negligible for all scenarios, and hence we can conclude that the price of stability of the MCOG game in static mixed strategies is close to one.

B. *Impact of Cloud Availability*

In order to analyse the impact of the possibility to offload to the edge cloud, in the following we vary the bandwidth $B_{i,0}$ between 0.2 *MHz* and 5.2 *MHz*.

Figure 6 shows the average and the median performance gain for the MBR, SM-NE, SM-OPT and SMC-NE algorithms as a function of the device to cloud bandwidth for 8 devices placed over a square area of $0.5\ km \times 0.5\ km$, for 30 devices placed over a square area of $1\ km \times 1\ km$, and for 60 devices placed over a square area of $1.41\ km \times 1.41\ km$. Note that the three scenarios have approximately the same density of devices. We first observe that the median performance gain is almost equal to the average performance gain for all algorithms and for all considered scenarios, which suggests that distribution of the completion times of the tasks is approximately symmetrical. The figure shows that the performance gain achieved by the algorithms increases with the bandwidth $B_{i,0}$. Furthermore, we observe that the gap between the algorithms decreases as the device to cloud bandwidth increases, and for reasonably high bandwidths the SM-NE algorithm performs almost equally well as the MBR algorithm. The results also show that collaboration among the devices has highest impact

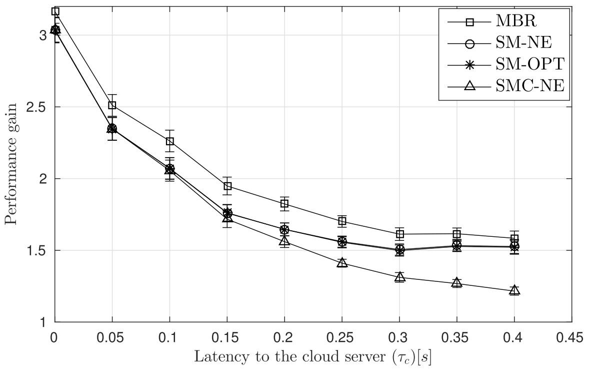

Fig. 7. *Performance gain* vs. latency τ_c to the cloud server, for $N = 30$ devices placed over 1 $km \times 1$ km square area, and $B_{i,0} = 1.25$ *MHz*.

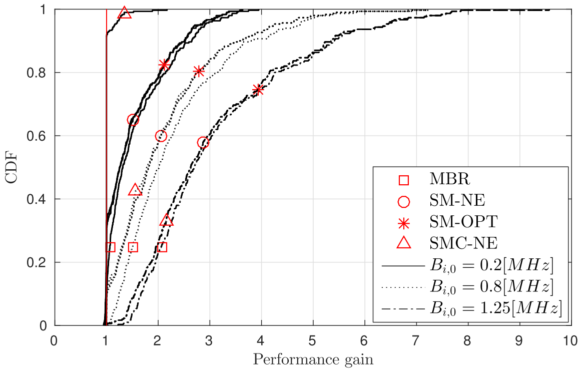

Fig. 8. Distribution of the performance gain for $N = 30$ devices, $B_{i,0} = 0.2$ *MHz*, $B_{i,0} = 0.8$ *MHz* and $B_{i,0} = 1.25$ *MHz*.

on the system performance when the bandwidth $B_{i,0}$ is low, and for $B_{i,0} = 1.2$ *MHz* offloading to the edge cloud only (SMC-NE) is as good as the SM-NE and SM-OPT algorithms.

Comparing the performance for different sized areas we observe that the performance gain decreases as the size of the area increases, which is due to that the devices are closer to the cloud server on average in a smaller area.

C. Impact of Cloud Remoteness

In order to evaluate the impact of the cloud access latency, in the following we vary the latency τ_c between 0 s and 0.4 s. A low latency (0 $ms \leq \tau_c < 20$ ms) would correspond to the case of an edge cloud or a home gateway, a moderate latency (20 $ms \leq \tau_c < 100$ ms) would correspond to an edge cloud located deeper in the network (e.g., metro network), and high latency (100 $ms \leq \tau_c$) would correspond to remote cloud servers.

In Figure 7 we show the average performance gain as a function of the latency τ_c for the MBR, SM-NE, SM-OPT and SMC-NE algorithms in a fog computing system that serves $N = 30$ devices, each of them assigned a bandwidth of $B_{i,0} = 1.25$ *MHz* for communication with the cloud. The figure shows that the performance gain of all algorithms decreases as the latency to the cloud server increases. Furthermore, we observe that the performance gain of the SMC-NE algorithm approaches one, as in the case of a high latency it is better for most of devices to perform the computation locally. On the contrary, the performance gain of the MBR, SM-NE and SM-OPT algorithms remains slightly above 1.5 even for high values of the latency ($\tau_c \geq 300$ ms), which additionally confirms that devices can decrease the average completion times of their tasks through collaboration even in systems where they cannot entirely rely on the cloud resources.

D. Performance Gain Perceived Per Device

In order to evaluate the performance gain perceived per device, we use the notion of ex-ante and ex-post individual rationality. These are important in situations when the devices are allowed to decide whether or not to participate in the collaboration before and after learning their types (i.e., the exact size and complexity of their tasks), respectively. The results in Figure 5 show that on average the devices benefit from collaboration, as the performance gain is greater than one, and hence collaboration among the devices is ex-ante individually rational. In order to investigate whether collaboration among the devices is ex-post individually rational, in Figure 8 we plot the CDF of the performance gain for the *elastic* device to cloud bandwidth assignment scenario with 30 devices and for $B_{i,0} = 0.2$ *MHz*, $B_{i,0} = 0.8$ *MHz*, and $B_{i,0} = 1.25$ *MHz*.

The results for $B_{i,0} = 0.2$ *MHz* show that the SMC-NE algorithm is ex-post individually rational, as devices always gain compared to local computation. At the same time, the SM-NE and MBR algorithms achieve a performance gain below one for a small fraction of the devices, and hence collaboration among devices is not ex-post individually rational. On the contrary, the results for $B_{i,0} = 0.8$ *MHz* show that the MBR algorithm is ex-post individually rational, since the performance gain of every device is larger than one, but the SM-NE is not. Finally, the results for $B_{i,0} = 1.25$ *MHz* show that all algorithms ensure that every device achieves a performance gain at least one, and hence for $B_{i,0} = 1.25$ *MHz* collaboration among devices is ex-post individually rational using all algorithms.

The above results show that collaboration among the devices is ex-post individually rational only if sufficient bandwidth is provided for communication to the edge cloud. Thus, if ex-post individual rationality is important then the device to cloud bandwidth has to be managed appropriately.

E. Utilization Ratio of Collaboration Among Devices

In order to evaluate the impact of collaboration on the system performance, we consider the ratio of the tasks executed at different nodes in the system. To obtain this ratio, we simulated stochastic task arrivals over a period of $10^4 s$. We recorded the N_t tasks generated in the system during this period, and for an algorithm $A \in \{MBR, SM-NE, SM-OPT\}$ we recorded N_l^A and N_c^A, the number of tasks executed locally and the number of tasks executed in the edge cloud, respectively. Figure 9 shows the ratio $\frac{N_l^A}{N_t}$ of the tasks executed locally, and the ratio $\frac{N_t - N_c^A}{N_t}$ of the tasks executed either locally or at one of the other devices for the MBR, SM-NE and SM-OPT algorithms as a function of the number of devices for $B_{i,0} = \frac{1}{N} 12.5$ *MHz*.

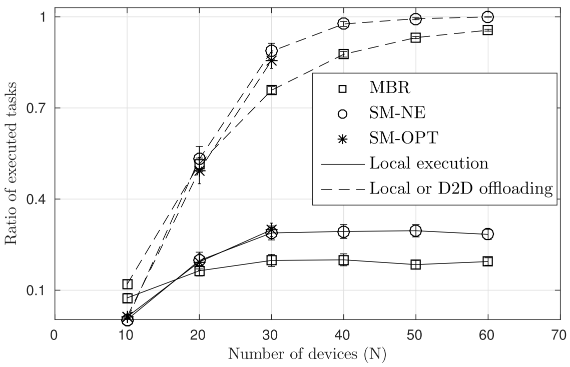

Fig. 9. Ratio of the tasks executed locally and the tasks executed at any of the devices for $B_{i,0} = \frac{1}{N}12.5$ *MHz*.

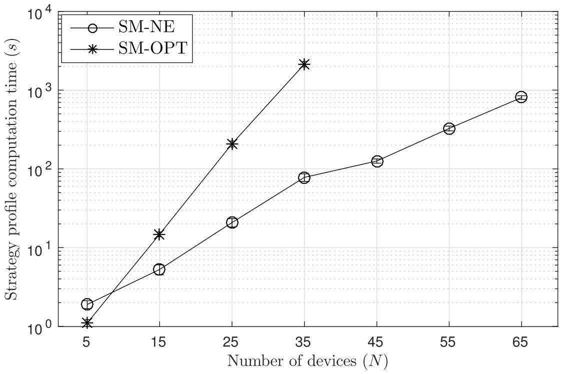

Fig. 10. Time needed to compute a static mixed strategy equilibrium and a socially optimal static mixed strategy profile for $B_{i,0} = 1.25$ *MHz*.

The results in Figure 9 show that for $N = 10$, i.e., when the bandwidth assigned to each device for communication with the edge cloud is 1.25 *MHz*, the devices offload more tasks to the edge cloud in the case of the SM-NE and SM-OPT algorithms than in the case of the MBR algorithm, which coincides with the observation made in Figure 5 for $B_{i,0} = 1.25$ *MHz*. On the contrary, when $N \geq 20$ the devices offload more tasks to the edge cloud in the case of the MBR algorithm than in the case of the SM-NE and SM-OPT algorithms that achieve approximately the same performance. Furthermore, we observe that while the ratio of the tasks executed locally increases up to 30 users and remains constant for more devices, the ratio of the tasks executed either locally or at one of the other devices continues to increase with the number of devices for all algorithms. These results confirm the observation made for $B_{i,0} = \frac{1}{N}12.5$ *MHz* in Figure 5 that the collaboration among the devices improves the system performance.

F. Computational Efficiency of the SM-NE Algorithm

Recall that the SM-NE algorithm is based on the static mixed strategy equilibrium, and that the SM-OPT algorithm is based on the socially optimal static mixed strategy profile. In order to assess the computational efficiency of the SM-NE algorithm we measured the time needed to compute a static mixed strategy equilibrium by the *ST-PC* method and the time needed to compute a socially optimal static mixed strategy profile by the quasi-Newton method. Figure 10 shows the measured times as a function of the number of devices. We observe that the time needed to compute the socially optimal static mixed strategy profile increases exponentially with the number of devices at a fairly high rate, and already for 30 devices it is more than an order of magnitude faster to compute a static mixed strategy equilibrium than to compute the socially optimal static mixed strategy profile. Therefore, we conclude that the SM-NE algorithm, which is based on an equilibrium in static mixed strategies, is a computationally efficient solution for medium to large scale collaborative computation offloading systems.

VI. Related Work

There is a large body of work on augmenting the execution of computationally intensive applications using cloud resources [27], [52]–[56]. Yang *et al.* [52] studied the problem of maximizing the throughput of mobile data stream applications through partitioning, and proposed a genetic algorithm as a solution. Rahimi *et al.* [53] considered multiple QoS factors in a 2-tiered cloud infrastructure, and proposed a heuristic for minimizing the users' cost. Sardellitti *et al.* [54] proposed an iterative algorithm that minimizes the users' overall energy consumption, while meeting latency constraints. Lyu *et al.* [55] considered the joint optimization of the offloading decisions, and the allocation of communication and computation resources, proved the NP-hardness of the problem and proposed a heuristic offloading decision algorithm for minimizing the completion time and the energy consumption of devices. Chen [27] considered a single wireless link and an elastic cloud, provided a game theoretic treatment of the problem of minimizing completion time and showed that the game is a potential game. Cardellini *et al.* [56] considered multiple wireless links, elastic and non-elastic cloud, provided a game theoretic analysis of the problem and proposed a polynomial complexity algorithm for computing an equilibrium allocation. In [19] the authors considered a three-tier cloud architecture with stochastic task arrivals, provided a game theoretical formulation of the problem, and used a variational inequality to prove the existence of a solution and to provide a distributed algorithm for computing an equilibrium. Unlike these works, we allow devices to offload computations to each other as well.

A few recent works considered augmenting the execution of computationally intensive applications using the computational power of nearby devices in a collaborative way [18], [39], [57]–[59]. Xiang *et al.* [57] modeled the collaboration among mobile devices as a coalition game, and proposed a heuristic method for solving a $0-1$ integer quadratic programing problem that minimizes the overall energy consumption. Ghasemi-Falavarjani *et al.* [58] formulated the resource allocation problem among neighboring mobile devices as a multi-objective optimization that aims to minimize the completion times of the tasks as well as the overall energy consumption, and as a solution proposed a two-stage approach based on enumerating Pareto optimal solutions. Wang *et al.* [59] formulated the problem of maximizing the probability of computing tasks before their deadlines through mobility-assisted opportunistic computation offloading as a convex optimization problem, and used the barrier method to solve the problem. Bohez *et al.* [18] considered a collaborative cloudlet that consists of devices that can perform shared offloading, and proposed two heuristic allocation algorithms that minimize the average relative usage of all the nodes in the cloudet. Shi *et al.* [39] proposed an architecture that enables a mobile device to remotely access computational resources on other mobile devices, and proposed two greedy algorithms that require complete information about devices' states, for minimizing the job completion time and the energy consumption, respectively. Our work differs from these works, as we consider computation offloading to an edge cloud and nearby devices, and provide a non-cooperative game theoretic treatment of the problem.

Only a few recent works considered the computation offloading problem in fog computing systems [60]–[63]. Du *et al.* [60] considered a fog computing system in which the tasks can be performed locally at the devices, at a fog node or at a remote cloud server, and proposed a suboptimal algorithm for computing the offloading decisions and allocating resources with the objective to minimize the delay and the energy consumption of devices. In [61] the authors considered a fog computing system, where devices may offload their computation to small cell access points that provide computation and storage capacities, and designed a heuristic for a joint optimization of radio and computational resources with the objective of minimizing the energy consumption. Unlike this work, we consider stochastic task arrivals, and we provide a game theoretical treatment of the completion time minimization problem. Deng *et al.* [62] authors formulated the power consumption-delay tradeoff problem in fog computing system that consists of a set of fog devices and a set of cloud servers, and proposed a heuristic for allocating the workload among fog devices and cloud servers. Zeng *et al.* [63] the authors considered the joint optimization problem of task allocation and task image placement in a fog computing system that consists of a set of storage srevers, a set of computation servers and a set of users, and proposed a low-complexity three-stage algorithm for the task completion time minimization problem. Our work differs from these works, as we consider heterogeneous computational tasks, and our queueing system model captures the contention for both communication and computational resources.

To the best of our knowledge ours is the first work based on a game theoretical analysis that proposes a decentralized algorithm with low signaling overhead for solving the completion time minimization problem in fog computing systems.

VII. Conclusion

We have provided a game theoretical analysis of a fog computing system. We proposed an efficient decentralized algorithm based on an equilibrium task allocation in static mixed strategies. We compared the performance achieved by the proposed algorithm that relies on average system parameters with the performance of a myopic best response algorithm that requires global knowledge of the system state. Our numerical results show that the proposed algorithm achieves good system performance, close to that of the myopic best response algorithm, and could be a possible solution for coordinating collaborative computation offloading with low signaling overhead. There is a number of interesting extensions of our model. First, one could consider a communication model in which devices share the bandwidth with each other. Another direction is to consider the energy cost of offloading, e.g., use it as a constraint for offloading optimization.

Appendix
Proof of Theorem 2

Observe that if $\lambda_i = \lambda$ then the cost C_i can equivalently be defined as $N_i = \lambda C_i$, i.e., the number of tasks in the system. Furthermore, since task complexities are assumed to be exponentially distributed, the execution queues are $M/M/1$ systems. We can thus rewrite $\overline{T^e}_{i,j}$ as

$$\overline{T^e}_{i,j} = \frac{\overline{T^c}_{i,j}}{1-\rho^e_j}, \tag{9}$$

and the cost $N_i(p_i, p_{-i})$ of device i as

$$N_i(p_i,p_{-i}) = p_{i,i}\lambda\frac{\overline{T^c}_{i,i}}{\zeta_i} + p_{i,0}\lambda\big(\frac{\gamma_{i,0}}{2\epsilon_{i,0}} + \overline{T^t}_{i,0} + \overline{T^c}_{i,0}\big) + \sum_{j\in\mathcal{N}\setminus\{i\}} p_{i,j}\lambda\big(\frac{\gamma_{i,j}}{2\epsilon_{i,j}} + \overline{T^t}_{i,j} + \frac{\overline{T^c}_{i,j}}{\zeta_j}\big).$$

Next, we express the first order derivatives $h_{i,j}$ of $N_i(p_i, p_{-i})$ as

$$h_{i,0}=\lambda\big(\overline{T^t}_{i,0}+\overline{T^c}_{i,0}+\frac{\gamma_{i,0}}{2\epsilon_{i,0}}\big)+p_{i,0}\lambda^2\big(\frac{{}^2\overline{T^t}_{i,0}}{2\epsilon_{i,0}}+\frac{\overline{T^t}_{i,0}\gamma_{i,0}}{2\epsilon^2_{i,0}}\big),$$

$$h_{i,i} = \lambda\frac{\overline{T^c}_{i,i}}{\zeta_i} + p_{i,i}\lambda^2\frac{\overline{T^c}^2_{i,i}}{\zeta_i^2},$$

$$h_{i,j}\big|_{j\neq i} = \lambda\big(\overline{T^t}_{i,j} + \frac{\gamma_{i,j}}{2\epsilon_{i,j}} + \frac{\overline{T^c}_{i,j}}{\zeta_j}\big) + p_{i,j}\lambda^2\big(\frac{{}^2\overline{T^t}_{i,j}}{2\epsilon_{i,j}} + \frac{\overline{T^t}_{i,j}\gamma_{i,j}}{2\epsilon^2_{i,j}} + \frac{\overline{T^c}^2_{i,j}}{\zeta_j^2}\big).$$

In order to prove the monotonicity of the function F in what follows we show that the Jacobian J of F is positive semidefinite. The Jacobian J has the structure as shown in the first equation at the top of the next page,
where the second order derivatives can be expressed as

$$h^i_{i,0} = \frac{\lambda^2}{\epsilon_{i,0}}\big({}^2\overline{T^t}_{i,0} + \frac{\gamma_{i,0}\overline{T^t}_{i,0}}{\epsilon_{i,0}}\big)\big(1 + p_{i,0}\frac{\lambda\overline{T^t}_{i,0}}{\epsilon_{i,0}}\big)$$

$$h^i_{i,i} = \left(\frac{\lambda\overline{T^c}_{i,i}}{\zeta_i}\right)^2 \big(2 + 2\frac{\lambda}{\zeta_i}p_{i,i}\overline{T^c}_{i,i}\big),$$

$$h^i_{i,j}\big|_{j\neq i} = \left(\frac{\lambda\overline{T^c}_{i,j}}{\zeta_j}\right)^2 \big(2 + 2\frac{\lambda}{\zeta_j}p_{i,j}\overline{T^c}_{i,j}\big) + h^t_{i,j},$$

where

$$h^t_{i,j} = \frac{\lambda^2}{\epsilon_{i,j}}\big({}^2\overline{T^t}_{i,j} + \frac{\gamma_{i,j}\overline{T^t}_{i,j}}{\epsilon_{i,j}}\big)\big(1 + p_{i,j}\frac{\lambda\overline{T^t}_{i,j}}{\epsilon_{i,j}}\big),$$

and

$$h^i_{i',j}\big|_{i'\neq i} = \frac{\lambda\overline{T^c}_{i,j}\lambda\overline{T^c}_{i',j}}{\zeta_j^2}\big(1 + 2\frac{\lambda}{\zeta_j}p_{i,j}\overline{T^c}_{i,j}\big).$$

$$\begin{pmatrix}
h^1_{1,0} & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0\\
0 & h^1_{1,1} & \dots & 0 & 0 & h^1_{2,1} & \dots & 0 & \dots & 0 & h^1_{N,1} & \dots & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots & \\
0 & 0 & \dots & h^1_{1,N} & 0 & 0 & \dots & h^1_{2,N} & \dots & 0 & 0 & \dots & h^1_{N,N}\\
\vdots & & & & & & & & & & & & \vdots\\
0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & h^N_{N,0} & 0 & \dots & 0\\
0 & h^N_{1,1} & \dots & 0 & 0 & h^N_{2,1} & \dots & 0 & \dots & 0 & h^N_{N,1} & \dots & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots & \\
0 & 0 & \dots & h^N_{1,N} & 0 & 0 & \dots & h^N_{2,N} & \dots & 0 & 0 & \dots & h^N_{N,N}
\end{pmatrix},$$

$$\begin{pmatrix}
p_{1,i}\overline{T^c}_{1,i} & \dfrac{p_{1,i}\overline{T^c}_{1,i}+p_{2,i}\overline{T^c}_{2,i}}{2} & \dots & \dfrac{p_{1,i}\overline{T^c}_{1,i}+p_{N,i}\overline{T^c}_{N,i}}{2}\\
\dfrac{p_{2,i}\overline{T^c}_{2,i}+p_{1,i}\overline{T^c}_{1,i}}{2} & p_{2,i}\overline{T^c}_{2,i} & \dots & \dfrac{p_{2,i}\overline{T^c}_{2,i}+p_{N,i}\overline{T^c}_{N,i}}{2}\\
\vdots & \vdots & \ddots & \vdots\\
\dfrac{p_{N,i}\overline{T^c}_{N,i}+p_{1,i}\overline{T^c}_{1,i}}{2} & \dfrac{p_{N,i}\overline{T^c}_{N,i}+p_{2,i}\overline{T^c}_{2,i}}{2} & \dots & p_{N,i}\overline{T^c}_{N,i}
\end{pmatrix}.$$

Reordering the rows and columns, the Jacobian J can be rewritten as

$$J=\begin{pmatrix} C & 0 & \dots & 0\\ 0 & M_1 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & M_N \end{pmatrix},$$

where

$$C=\begin{pmatrix} h^1_{1,0} & 0 & \dots & 0\\ 0 & h^2_{2,0} & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & h^N_{N,0} \end{pmatrix},$$

$$M_i=\begin{pmatrix} h^1_{1,i} & h^1_{2,i} & \dots & h^1_{N,i}\\ h^2_{1,i} & h^2_{2,i} & \dots & h^2_{N,i}\\ \vdots & \vdots & \ddots & \vdots\\ h^N_{1,i} & h^N_{2,i} & \dots & h^N_{N,i} \end{pmatrix}.$$

Observe that all diagonal elements of C are nonnegative, and thus the matrix C is positive definite. In order to show that J is positive semidefinite we have to show that the symmetric matrix $M^s_i=\frac{1}{2}(M^T_i+M_i)$ is positive semidefinite.

The diagonal elements ${}^dh^s_{j,i}$ of M^s_i are given by

$$\begin{aligned}
{}^dh^s_{j,i}\big|_{j=i} &= \left(\frac{\lambda\overline{T^c}_{i,i}}{\zeta_i}\right)^2\big(2+2\frac{\lambda}{\zeta_i}p_{i,i}\overline{T^c}_{i,i}\big),\\
{}^dh^s_{j,i}\big|_{j\neq i} &= \left(\frac{\lambda\overline{T^c}_{j,i}}{\zeta_i}\right)^2\big(2+2\frac{\lambda}{\zeta_i}p_{j,i}\overline{T^c}_{j,i}\big)+h^t_{j,i},
\end{aligned}$$

where

$$h^t_{j,i}=\frac{\lambda^2}{\epsilon_{j,i}}\big({}^2\overline{T^t}_{j,i}+\frac{\gamma_{j,i}\overline{T^t}_{j,i}}{\epsilon_{j,i}}\big)\big(1+p_{j,i}\frac{\lambda\overline{T^t}_{j,i}}{\epsilon_{j,i}}\big),$$

and the off-diagonal elements ${}^oh^s_{j,i}=\frac{1}{2}(h^i_{j,i}+h^j_{i,i})\Big|_{j\neq i}$ are given by

$${}^oh^s_{j,i}=\frac{\lambda\overline{T^c}_{i,i}\lambda\overline{T^c}_{j,i}}{\zeta^2_i}\big(1+\frac{\lambda}{\zeta_i}(p_{i,i}\overline{T^c}_{i,i}+p_{j,i}\overline{T^c}_{j,i})\big)$$

Let us define the vector $\overline{T^c}_i=(\overline{T^c}_{1,i}\ \overline{T^c}_{2,i}\dots\overline{T^c}_{N,i})^T$ and matrix $\overline{T^t}_i$

$$\overline{T^t}_i=\begin{pmatrix} \operatorname{diag}(h^t_{j,i})\big|_{j\in\mathcal{N}\setminus\{i\}} & 0\\ 0 & 0 \end{pmatrix}.$$

Furthermore, let us define matrix T^p_i as shown in the second equation at the top of the page.
Now, matrix M_i can be rewritten as

$$M_i=\frac{\lambda^2}{\zeta^2_i}\left(\overline{T^c}_i\,\overline{T^c}^T_i\circ\left(I+E+\frac{2\lambda}{\zeta_i}T^p_i\right)\right)+\overline{T^t}_i,$$

where $\circ$ denotes the Hadamard product, i.e., the component-wise product of two matrices.

It is well known that the identity I and unit E matrices are positive definite, while positive definiteness of matrix $\overline{T^c}_i\,\overline{T^c}^T_i$ follows from the definition. Observe that matrix $\overline{T^t}_i$ is positive semidefinite as well, since it is a diagonal matrix with nonnegative elements. Since the sum of two positive semidefinite matrices is positive semidefinite and the Hadamard product of two positive semidefinite matrices is also positive semidefinite [64], the proof reduces to showing that matrix $I+E+\frac{2\lambda}{\zeta_i}T^p_i$ is positive semidefinite. To do so, we will show that the minimum eigenvalue of the matrix $\frac{2\lambda}{\zeta_i}T^p_i$ is greater than or equal to -1. To do so, let us denote by e the all-ones vector and define the vector $t^p_i=(p_{1,i}\overline{T^c}_{1,i}\ p_{2,i}\overline{T^c}_{2,i}\dots p_{N,i}\overline{T^c}_{N,i})$. Now, we can express matrix T^p_i as

$$T^p_i=\frac{1}{2}\big(t^p_ie^T+e(t^p_i)^T\big).$$

The characteristic polynomial of the symmetric matrix T_i^p is given by [65]

$$\frac{k^{N-2}}{2}\left(k^2 - 2(e^T t_i^p)k + (e^T t_i^p)^2 - N\|t_i^p\|^2\right).$$

We observe that T_i^p has $N-2$ zero eigenvalues, and one non-negative and one non-positive eigenvalue given by $k_+ = \left(e^T t_i^p + \sqrt{N}\|t_i^p\|\right)/2$ and $k_- = \left(e^T t_i^p - \sqrt{N}\|t_i^p\|\right)/2$, respectively. Therefore, the minimum eigenvalue of the matrix $\frac{2\lambda}{\zeta_i}T_i^p$ is greater than -1 if

$$\frac{\lambda}{\zeta_i}\left(\sqrt{N}\|t_i^p\| - e^T t_i^p\right) \le 1. \tag{10}$$

Since t_i^p is a vector with non-negative elements, we have that $e^T t_i^p \ge \|t_i^p\|$ and it also holds that $\|t_i^p\| \le \sqrt{N}\max_{j\in\mathcal{N}} t_{j,i}$. Therefore, the following inequalities hold

$$\begin{aligned}\frac{\lambda}{\zeta_i}\left(\sqrt{N}\|t_i^p\| - e^T t_i^p\right) &\le \frac{\lambda}{\zeta_i}\left(\sqrt{N}\max_{j\in\mathcal{N}} t_{j,i}(\sqrt{N}-1)\right)\\ &\le \frac{N\lambda}{\zeta_i}\max_{j\in\mathcal{N}} t_{j,i} \le \frac{N\lambda}{\zeta_i}\max_{j\in\mathcal{N}} \overline{T^c}_{j,i}.\end{aligned}$$

Since $\rho_i^e \le S_t$, we have that $\zeta_i \ge 1 - S_t$, and therefore

$$\frac{N\lambda}{\zeta_i}\max_{j\in\mathcal{N}} \overline{T^c}_{j,i} \le \frac{N\lambda}{1-S_t}\max_{j\in\mathcal{N}} \overline{T^c}_{j,i}. \tag{11}$$

Based on (11) a sufficient condition for (10) is that $\lambda\max_{j\in\mathcal{N}} \overline{T^c}_{j,i} \le \frac{1-S_t}{N}$. This proves the theorem.

References

[1] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[2] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the internet of things realize its potential," *Comput.*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[3] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for Internet of Things: A primer," *Digit. Commun. Netw.*, vol. 4, no. 2, pp. 77–86, 2018.

[4] E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. ACM MobiSys*, 2010, pp. 49–62.

[5] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129–140, Feb. 2013.

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

[7] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[8] G. P. Fettweis, "The tactile Internet: Applications and challenges," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar. 2014.

[9] M. S. Elbamby, M. Bennis, and W. Saad, "Proactive edge computing in latency-constrained fog networks," in *Proc. IEEE Netw. Commun. (EuCNC)*, Jun. 2017, pp. 1–6.

[10] S. Li, L. Xu, and S. Zhao, "The Internet of Things: A survey," *Inf. Syst. Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

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

[12] G. Fodor *et al.*, "Design aspects of network assisted device-to-device communications," *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 170–177, Mar. 2012.

[13] K. Doppler, C.-H. Yu, C. B. Ribeiro, and P. Janis, "Mode selection for device-to-device communication underlaying an LTE-advanced network," in *Proc. IEEE WCNC*, Apr. 2010, pp. 1–6.

[14] M. Zulhasnine, C. Huang, and A. Srinivasan, "Efficient resource allocation for device-to-device communication underlaying LTE network," in *Proc. IEEE WiMob*, Oct. 2010, pp. 368–375.

[15] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. USENIX Conf. Hot Topics Cloud Comput.*, 2010, p. 4.

[16] J. R. Lorch and A. J. Smith, "Improving dynamic voltage scaling algorithms with pace," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 29, no. 1, pp. 50–61, 2001.

[17] W. Yuan and K. Nahrstedt, "Energy-efficient cpu scheduling for multimedia applications," *ACM Trans. Comput. Syst.*, vol. 24, no. 3, pp. 292–331, 2006.

[18] S. Bohez, T. Verbelen, P. Simoens, and B. Dhoedt, "Discrete-event simulation for efficient and stable resource allocation in collaborative mobile cloudlets," *Simul. Model. Pract. Theory*, vol. 50, pp. 109–129, Jan. 2015.

[19] V. Cardellini *et al.*, "A game-theoretic approach to computation offloading in mobile cloud computing," *Math. Program.*, vol. 157, no. 2, pp. 421–449, 2015.

[20] Y. Wang, X. Lin, and M. Pedram, "A nested two stage game-based optimization framework in mobile cloud computing system," in *Proc. IEEE Int. Symp. Service Oriented Syst. Eng.*, Mar. 2013, pp. 494–502.

[21] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.

[22] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2001, pp. 310–317.

[23] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *Proc. ACM Int. Conf. World Wide Web*, 2003, pp. 640–651.

[24] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, Jul. 2004.

[25] P. Mach, Z. Becvar, and T. Vanek, "In-band device-to-device communication in OFDMA cellular networks: A survey and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1885–1922, 4th Quart., 2015.

[26] S. Sharma, N. Gupta, and V. A. Bohara, "OFDMA-based device-to-device communication frameworks: Testbed deployment and measurement results," *IEEE Access*, vol. 6, pp. 12019–12030, 2018.

[27] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

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

[29] S. Jošilo and G. Dan, "Selfish decentralized computation offloading for mobile cloud computing in dense wireless networks," *IEEE Trans. Mobile Comput.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8345749

[30] S. Jošilo and G. Dán, "Decentralized scheduling for offloading of periodic tasks in mobile edge computing," in *Proc. IFIP Netw.*, 2018, pp. 496–504.

[31] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of Everything*. Singapore: Springer, 2018, pp. 103–130.

[32] A. Brogi, S. Forti, A. Ibrahim, and L. Rinaldi, "Bonsai in the fog: An active learning lab with fog computing," in *Proc. IEEE Int. Conf. Fog Mobile Edge Comput.*, Apr. 2018, pp. 79–86.

[33] L. I. Sennott, "Nonzero-sum stochastic games with unbounded costs: Discounted and average cost cases," *Math. Methods Oper. Res.*, vol. 40, no. 2, pp. 145–162, 1994.

[34] E. Altman, A. Hordijk, and F. Spieksma, "Contraction conditions for average and α-discount optimality in countable state Markov games with unbounded rewards," *Math. Oper. Res.*, vol. 22, no. 3, pp. 588–618, 1997.

[35] A. S. Nowak, "Sensitive equilibria for ergodic stochastic games with countable state spaces," *Math. Methods Oper. Res.*, vol. 50, no. 1, pp. 65–76, 1999.

[36] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G.-J. Ren, "Foggy clouds and cloudy fogs: A real need for coordinated management of fog-to-cloud computing systems," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 120–128, Oct. 2016.

[37] B. Tang *et al.*, "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in *Proc. ASE BigData Social Informat.*, 2015, p. 28.

[38] O. Consortium *et al.*, "OpenFog reference architecture for fog computing," OpenFog Consortium, Fremont, CA, USA, Tech. Rep. OPFRA001.020817, Feb. 2017.

[39] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: Enabling remote computing among intermittently connected mobile devices," in *Proc. ACM MobiHoc*, 2012, pp. 145–154.

[40] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Feb. 2018.

[41] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*. New York, NY, USA: Springer, 2007.

[42] F. Facchinei, A. Fischer, and V. Piccialli, "On generalized Nash games and variational inequalities," *Oper. Res. Lett.*, vol. 35, no. 2, pp. 159–164, 2007.

[43] F. Tinti, "Numerical solution for pseudomonotone variational inequality problems by extragradient methods," in *Variational Analysis and Applications*. Boston, MA, USA: Springer, 2005, pp. 1101–1128.

[44] E. Balevi and R. D. Gitlin, "Optimizing the number of fog nodes for cloud-fog-thing networks," *IEEE Access*, vol. 6, pp. 11173–11183, 2018.

[45] S. Sigg, P. Jakimovski, and M. Beigl, "Calculation of functions on the RF-channel for IoT," in *Proc. IEEE IOT*, Oct. 2012, pp. 107–113.

[46] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 26–35, Mar. 2017.

[47] Y.-L. Chung, "Rate-and-power control based energy-saving transmissions in OFDMA-based multicarrier base stations," *IEEE Syst. J.*, vol. 9, no. 2, pp. 578–584, Jun. 2015.

[48] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-device communication in 5G cellular networks: Challenges, solutions, and future directions," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 86–92, May 2014.

[49] A. Aragon-Zavala, *Antennas and Propagation for Wireless Communication Systems*. Hoboken, NJ, USA: Wiley, 2008.

[50] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: A measurement study and implications for network applications," in *Proc. ACM Internet Meas. Conf. (IMC)*, 2009, pp. 280–293.

[51] M. Satyanarayanan, "A brief history of cloud offload: A personal journey from odyssey through cyber foraging to cloudlets," *GetMobile, Mobile Comput. Commun.*, vol. 18, no. 4, pp. 19–23, 2015.

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

[53] M. R. Rahimi, N. Venkatasubramanian, S. Mehrotra, and A. V. Vasilakos, "On optimal and fair service allocation in mobile cloud computing," *IEEE Trans. Cloud Comput.*, vol. 6, no. 3, pp. 815–828, Jul./Sep. 2015.

[54] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[55] X. Lyu, H. Tian, P. Zhang, and C. Sengul, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2016.

[56] S. Jošilo and G. Dán, "A game theoretic analysis of selfish mobile computation offloading," in *Proc. IEEE INFOCOM*, May 2017, pp. 1–9.

[57] L. Xiang, B. Li, and B. Li, "Coalition formation towards energy-efficient collaborative mobile computing," in *Proc. IEEE ICCCN*, Aug. 2015, pp. 1–8.

[58] S. Ghasemi-Falavarjani, M. Nematbakhsh, and B. S. Ghahfarokhi, "Context-aware multi-objective resource allocation in mobile cloud," *Comput. Elect. Eng.*, vol. 44, pp. 218–240, May 2015.

[59] C. Wang, Y. Li, and D. Jin, "Mobility-assisted opportunistic computation offloading," *IEEE Commun. Lett.*, vol. 18, no. 10, pp. 1779–1782, Oct. 2014.

[60] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2017.

[61] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *Proc. IEEE Veh. Technol. Conf.*, May 2015, pp. 1–6.

[62] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.

[63] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Trans. Comput.*, vol. 65, no. 12, pp. 3702–3712, Dec. 2016.

[64] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[65] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas With Application to Linear Systems Theory*, vol. 41. Princeton, NJ, USA: Princeton Univ. Press, 2005.

Slađana Jošilo received the M.Sc. degree in electrical engineering from the University of Novi Sad, Serbia, in 2012. She is currently pursuing the Ph.D. degree with the Department of Network and Systems Engineering, KTH Royal Institute of Technology. From 2013 to 2014, she was a Research Engineer with the Department of Power, Electronics and Communication Engineering, University of Novi Sad. Her research interests are the design and analysis of distributed algorithms for exploiting resources available at the network edge using game theoretical tools.

György Dán (M'07–SM'17) is a professor at KTH Royal Institute of Technology, Stockholm, Sweden. He received the M.Sc. degree in computer engineering from the Budapest University of Technology and Economics, Hungary, in 1999, the M.Sc. degree in business administration from the Corvinus University of Budapest, Hungary, in 2003, and the Ph.D. degree in telecommunications from KTH in 2006. From 1999 to 2001, he was a consultant in the fields of access networks, streaming media, and videoconferencing. He was a Visiting Researcher with the Swedish Institute of Computer Science in 2008, a Fulbright Research Scholar with the University of Illinois at Urbana–Champaign from 2012 to 2013, and an invited Professor with EPFL from 2014 to 2015. He is currently an Associate Professor with the KTH Royal Institute of Technology, Stockholm, Sweden. His research interests include the design and analysis of content management and computing systems, game theoretical models of networked systems, and cyber-physical system security and resilience.