IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 25, 2019, accepted August 11, 2019, date of publication September 16, 2019, date of current version October 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2941741

Task Data Offloading and Resource Allocation in Fog Computing With Multi-Task Delay Guarantee

MITHUN MUKHERJEE[1], (Member, IEEE), SUMAN KUMAR[2], QI ZHANG[3], RAKESH MATAM[4], (Member, IEEE), CONSTANDINOS X. MAVROMOUSTAKIS[5], YUNRONG LV[1], AND GEORGE MASTORAKIS[6]

[1]Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology, Maoming 525000, China
[2]Department of Mathematics, IGNTU, Amarkantak 484886, India
[3]DIGIT, Department of Engineering, Aarhus University, 8000 Aarhus, Denmark
[4]Department of Computer Science and Engineering, Indian Institute of Information Technology Guwahati, Guwahati 781015, India
[5]Mobile Systems Laboratory (MoSys Lab), Department of Computer Science, University of Nicosia, 1700 Nicosia, Cyprus
[6]Department of Management Science and Technology, Hellenic Mediterranean University, 72100 Crete, Greece

Corresponding author: Yunrong Lv (lyclyr@yeah.net)

This work was supported by the National Key Research and Development Program under Grant 2018YFC0808600.

ABSTRACT With the emergence of delay-sensitive task completion, computational offloading becomes increasingly desirable due to the end-user's limitations in performing computation-intense applications. Interestingly, fog computing enables computational offloading for the end-users towards delay-sensitive task provisioning. In this paper, we study the computational offloading for the multiple tasks with various delay requirements for the end-users, initiated one task at a time in end-user side. In our scenario, the end-user offloads the task data to its primary fog node. However, due to the limited computing resources in fog nodes compared to the remote cloud server, it becomes a challenging issue to entirely process the task data at the primary fog node within the delay deadline imposed by the applications initialized by the end-users. In fact, the primary fog node is mainly responsible for deciding the amount of task data to be offloaded to the secondary fog node and/or remote cloud. Moreover, the computational resource allocation in term of CPU cycles to process each bit of the task data at fog node and transmission resource allocation between a fog node to the remote cloud are also important factors to be considered. We have formulated the above problem as a Quadratically Constraint Quadratic Programming (QCQP) and provided a solution. Our extensive simulation results demonstrate the effectiveness of the proposed offloading scheme under different delay deadlines and traffic intensity levels.

INDEX TERMS 5G and beyond, computation offloading, mobile edge computing, fog computing, resource allocation, offloading decision.

I. INTRODUCTION

With the emergence of ultra-reliable and low-latency communications (uRLLC) [1]–[4], the latency and reliability-aware mission-critical applications are increasingly growing up. To mention, a few examples are, autonomous driving, virtual and augmented reality, and cloud robotics, remote surgery, and factory automation. However, at the same time, the end-user's computational resources limit the user's experience (e.g., latency and reliability) for the computational-intensive applications. The cloud computing has already proven its significance to process the computational-intensive tasks, however, the physical distance between the end-user and remote cloud data center and burden on fronthaul link are the major barrier for low-latency-aware applications. To address the above challenges, Fog computing [5], [6], often viewed as a middleware between end-user and cloud, extends the computational, communication, and storage resources of the cloud computing close to the network edge.

The associate editor coordinating the review of this manuscript and approving it for publication was Christos Verikoukis.

A. MOTIVATION

For computational-intensive task processing in a fog computing scenario, the end-user offloads the data either partially or entirely to the nearby fog computing node(s). It would be an ideal solution if a single fog computing node (hereinafter referred to as fog nodes) is able to compute, process the task data and deliver the results for the tasks

received from the end-user. However, the computational and storage resources in a single fog node are insufficient to handle all the tasks data under the delay-deadline. As a result, the fog node either finds the assistive fog node under its vicinity or upload the task to a remote cloud for further computational resources. Both cases retain the challenges as a) the assistive fog node also does not always have enough available resources, and b) offloading to the cloud still creates a burden on the upload link, resulting in a delay in the task processing. Thus, it becomes a challenging task to decide where to offload (e.g., assistive fog node and remote cloud) and how much partial task data to be offloaded under delay guarantee imposed by the end-user's application.

B. RELATED WORK

In the last decade, task offloading has been extensively investigated in both academia and industries under different nomenclature/technologies, e.g., mobile cloud computing [7], mobile edge computing [8], [9], cloudlets [10], and computing access points [11]. Recently, Chen *et al.* provided an optimal solution for deciding between fog node (similar to computing access point) and remote cloud server to offload the task data considering single user with a single task [12], a single user with multiple tasks [13], multiple users with more than one task per user [11]. Basically, all these approaches select remote cloud if the fog node does not meet the latency and energy consumption deadline requirement – fog node collaboration was not considered in the network model. Most recently, with an assumption that the end-user has dual connectivity [14], one is with an access point (can be referred as fog node) and another is with a base station (with the higher computational capability), an offloading strategy was suggested. Several work [15], [16] considered the fog node collaboration with transmission delay between fog nodes, however, these work did not provide any insights considering multi-user and multi-delay guarantee. In our recent work [17], a joint optimization of task data offloading and computational resource allocation for fog network is addressed. In this work, we further study the multi-task scenario with different delay deadline for each task, that was not considered in [17].

C. OBJECTIVE AND CONTRIBUTIONS

In this paper, a fog network is considered where the end-user partially uploads its task data to a nearby fog node.[1] Considering multiple tasks with different delay deadline received from the end-users, it becomes a challenging issue for the fog node to allocate computing resources for each task. In addition, a fog node take the tasks from its neighor node, therefore, fog node has to optimize the offloading decision to the neighbour fog node and remote cloud. The main contributions of this paper are summarized as follows.

- We focus on the offloading decision and the amount of task data to be offloaded considering delay deadline. We consider the multiple tasks with different delay imposed by the application initiated by the end-user.
- To address these challenges, we show a comprehensive delay model considering computational and transmission delay and formulate a multi-task offloading optimization problem that is transformed into a Quadratically Constraint Quadratic Programming (QCQP) problem. We further devise a heuristic approach to solve this problem and show that the proposed solution is able to effectively guarantee the latency deadline compared to fixed computing resource allocation.

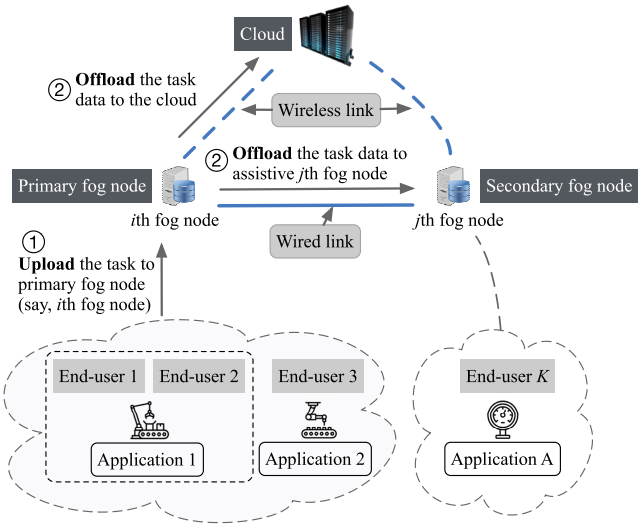

FIGURE 1. Illustration of a fog network for task data offloading with multiple applications.

The rest of the paper is organized as follows. Section II presents the system model. The total delay model including local task execution delay and transmission delay is discussed in Section III. The task offloading and computational resource allocation in primary and secondary fog node are presented in Section IV. The simulation results are presented in Section V. Finally, conclusions are drawn in Section VI.

II. SYSTEM MODEL

Consider a fog network with a set of fog computing nodes $\mathcal{N} = \{1, 2, \ldots, N\}$, a set of end-users $\mathcal{K} = \{1, 2, \ldots, K\}$, and one remote cloud server, as shown in Fig. 1. We consider that the fog nodes and end-users are uniformly distributed over the network. In general, we take a time-slotted system indexed by $t = \{0, 1, , \ldots, t\}$, where the length of each time slot is Δt (in s). Assuming one task arrives at the kth end-user at time slot t, the kth end-user aims to process the task data by itself. However, due to resource constraints (CPU rate and energy consumption,[2]) the end-users are often unable to process the data within the specified delay threshold when the total required computation cycle is high. Therefore, the end-users uploads either a part or an entire task to the nearest

[1] An interesting future work is to further consider fog node selection [18].

[2] Energy consumption issue, although novel, is a part of future work.

fog node that acts as a primary (often termed as *master*) fog node. Assuming two disjoint task queues maintained by the end-user's task scheduler, one queue is for local task data processing at the end-user and another queue is used for the task data offloading, we consider that the end-user simultaneously executes and offloads the task data. The fog computing node has higher computing and storage resources compared to the end-users, however, has less resources in comparison with remote cloud server. Therefore, the primary fog node selects a set of fog nodes within its proximity and/or uploads the task data to the remote cloud for the task processing within the deadline on the delay imposed by the applications. Although the transmission rate between the fog nodes play an significant role in task data offloading to the other fog nodes, however, we assume that the fog nodes are interconnected[3] with Ethernet – the transmission delay is ignored compared to the other delays involved.

Normally, it is assumed that one fog node can be served as primary fog node to several end-users. Let $\mathcal{M}_i = \{1, 2, \ldots, M_i\}$, $\sum_{i=1}^{N} M_i = K$ be the set of end-users that select ith fog node as a primary fog node. Moreover, we consider that these above sets are disjoint in nature, $\mathcal{M}_i \cap \mathcal{M}_{i'} \equiv \emptyset$ for $i \neq i'$, this is due to the reason that one end-user is not allowed to offload the task data to more that one primary fog node directly. In fact, only the primary fog node decides whether to further offload the end-user's task data (to secondary fog node and/or remote cloud) or not. We further assume that the ith primary fog node offloads the task data of only $k \in \mathcal{M}_i$ end-user to the remote cloud. The ith fog node cannot offload the task data that has been received from other end-user $k' \in \mathcal{K} \setminus \mathcal{M}_i$ via J_i neighbor fog nodes, where $\mathcal{J}_i = \{1, 2, \ldots, J_i\}$, $\mathcal{J}_i \in \mathcal{N}$ is the set of fog nodes that can select the ith fog node to offload their task data, to the cloud. The reason is offloading decision to the remote cloud (and other fog node) is co-ordinated by the ith primary fog node of the kth end-user that selects the ith fog node as its primary fog node. Note that a trade-off exists between the computational and transmission latency among the tasks offloaded to other fog nodes and the cloud. As shown in Fig. 1, the ith fog node receives the task data from the end-user $k \in \mathcal{M}_i$ that selects the ith fog node as their primary fog node and other end-user $k' \in \mathcal{K} \setminus \mathcal{M}_i$ via J_i neighbor fog nodes.

A. APPLICATION TYPE

We consider a large-scale industrial application where the data (such as the state-information) collected by the industrial sensors are processed for the assistance of delay-sensitive decision-making applications. Some of the examples are manufacturing process, factory automation, and fault detection. Considering a heterogeneous application scenario, although an end-user (taking industrial sensors in an industrial application) initializes *only* one task at a time from a finite application set $\mathcal{A} = \{1, 2, \ldots, A\}$. Each application requires a different CPU cycles to process each bit of the task data, i.e., processing density is different. Moreover, each application is bounded by different delay requirement. If the kth end-user initializes the ath application, denote the processing density by L_a and the deadline on the delay by τ_a^{task}.

B. TASK AT THE END-USER SIDE

Let $D_k(t)$ (in bits) be the task data size arriving at the kth end-user at the beginning time slot t. This task data can be processed at the starting from next time slot, i.e., $(t + 1)$. As the end-user is assumed to initialize one task at a time, the end-user selects a task say, task a from the application set $\mathcal{A}$. Generally, if a larger-size task that cannot be processed in one time slot can be divided into small sub-tasks which can be computed in a single time slot. For the sake of simplicity, we omit t in the rest of the paper.

Let D_k^{CPU} be the amount of task (in bits) locally computed at the kth end-user side. Based on the task data size, processing density, and available computing resources, if the end-user estimates that the task data cannot be processed within the tolerable delay τ_a^{task}, then the task scheduler in end-user starts to offload[4] the task data to the primary fog node in parallel with the local task processing. Therefore, we have

$$D_k = D_k^{\text{CPU}} + \mu_{k,i} D_{k,i}^{\text{OL}}, \tag{1}$$

where $D_{k,i}^{\text{OL}}$ is the task data (in bits) offloaded from the kth end-user to the ith fog node, $\mu_{k,i}$ is the offloading decision variable at the end-user side and is expressed as $\mu_{k,i} = 1$ if the kth end-user selects the ith fog node as primary fog node to offload the task data, and 0 otherwise.

C. TASK AT THE FOG NODE SIDE

The primary fog node receives the task data from the end-users under its coverage. However, due to the resource constraints, the primary fog node is not able to process all the task data offloaded by the end-users within the imposed deadline by the different applications. Thus, the fog node has to offload a part of the task to the neighbor fog node (we call it *secondary* fog node) that has sufficient amount of resources.

We define $\beta_{k,i,j}$ as the *inter-fog offloading decision* variable and express as $\beta_{k,i,j} = 1$ if the ith primary fog node offloads the kth end-user's task data to the jth secondary fog node, and 0 otherwise. Moreover, we introduce another variable, *fog-cloud offloading decision variable*, $\lambda_{k,i}$ equals to 1 if the ith primary fog node offloads the kth end-user's task data to the remote cloud, and 0 otherwise.

Let $D_{k,i}^{\text{CPU, fog}}$ be the locally processed task data of the kth end-user at the ith primary fog node. Therefore,

$$D_{k,i}^{\text{OL}} = D_{k,i}^{\text{CPU, fog}} + \beta_{k,i,j} D_{k,i\to j}^{\text{OL}} + \lambda_{k,i} D_{k,i\to c}^{\text{OL}}, \tag{2}$$

where $D_{k,i\to j}^{\text{OL}}$ and $D_{k,i\to c}^{\text{OL}}$ are the offloaded task data of the kth end-user from the ith primary fog node to the jth secondary fog node and the remote cloud, respectively.

[3] In several cases, the fog nodes are connected via WiFi Direct (using IEEE 802.11n) with a data rate more than 300 Mbps.

[4] Several tasks, e.g., OS-level processing cannot be offloaded. We consider to offload the task that can only be offloaded to avail higher computational resources for processing.

III. DELAY MODEL: LOCAL TASK EXECUTION DELAY AND TRANSMISSION DELAY

For the delay model, we only consider a) local task execution delay and b) transmission delay. An interesting future work is to further consider task queue model, task prefetching, and resource allocation delay.

A. LOCAL TASK EXECUTION DELAY

The task execution delay mainly depends on processing density, i.e., required cycles to process the task data and CPU clock speed. Consider that the kth end-user initializes the task a. Then, the local task execution delay at the kth end-user is

$$\tau_k^{\text{CPU}} = \frac{L_a\, D_k^{\text{CPU}}}{f_k}\ [\text{s}], \tag{3}$$

where L_a is the processing density (in cycles/bit) for the ath task served by the kth user and f_k denotes the CPU clock speed (in cycles/s) of the kth end-user.

In the similar way, the local task execution delay (in [s]) for the kth end-user's task at the ith fog node becomes

$$\tau_{k,i}^{\text{CPU, fog}} = \frac{L_a\, D_{k,i}^{\text{CPU, fog}}}{f_{k,i}}\ [s], \tag{4}$$

where $f_{k,i}$ refers to the CPU clock speed (in cycles/second) of the ith fog node assigned for the kth user task data processing. As the offloaded task data for the kth user from the ith primary fog node to the jth secondary fog node must be processed at the secondary fog node by itself, the local task execution delay for the kth end-user's task at the jth secondary fog node is given by

$$\tau_{k,j}^{\text{CPU, fog}} = \frac{L_a\, D_{k,i\to j}^{\text{OL}}}{f_{k,j}}\ [s]. \tag{5}$$

In our present work, we will not consider the local task processing time at the remote cloud since the cloud is generally equipped with a sufficient amount of computational and storage resources [19]. Therefore, compared to the resource constraint fog node and end-users, the task execution delay is significantly lower in cloud server.

B. TRANSMISSION DELAY

The transmission delay mainly depends on the transmission rate (sometimes, called as *offloading rate*). In general, the total transmission delay consists of both uploading and downloading time. Similar to [20], in our system model, the downloading time is ignored due to the small data size of the results compared to the uploaded task data size from end-user to fog node, from fog node to the cloud, and from primary fog node to secondary fog node.

1) END-USER TO PRIMARY FOG NODE

The transmission delay between the kth end-user and the ith primary fog node is

$$\tau_{k,i}^{\text{Tx}} = \frac{\mu_{k,i}\, D_{k,i}^{\text{OL}}}{r_{k,i}}\ [s], \tag{6}$$

where $r_{k,i}$ denotes the transmission rate between the kth end-user and the ith fog node.

2) INTER-FOG TRANSMISSION DELAY

It is assumed that the fog nodes can be interconnected via IEEE 802.3 ah/av 1/10 Gbps Ethernet. Thus, compared to the transmission rate between end-user to primary fog node and primary fog node to the remote cloud, the inter-fog transmission delay can be ignored.

3) FOG NODE TO CLOUD TRANSMISSION DELAY

We consider that the fog nodes use orthogonal bands to upload data to the cloud as in 4G cellular networks [21], [22]. The transmission delay between ith fog node to the cloud for the kth end-user is

$$\tau_{k,i\to c}^{\text{Tx}} = \frac{\lambda_{k,i}\, D_{k,i\to c}^{\text{OL}}}{r_{k,i,c}}\ [s], \tag{7}$$

where $r_{k,i,c}$ is the offloading-rate for the kth user from the ith fog node to the cloud.

C. TOTAL DELAY

Since the task scheduler at the end-user simultaneously executes and uploads the task data, the total delay will be the maximum value of local task execution delay and the summation of transmission delay and task execution delay of the offloaded task data. Moreover, the primary fog node simultaneously a) locally executes the task data, b) offloads the task data to the secondary fog node, and c) offloads the task data to the remote cloud. Therefore, the maximum value of $\tau_{k,i}^{\text{fog, CPU}}$, $\tau_{k,j}^{\text{fog, CPU}}$, and $\tau_{k,i\to c}^{\text{Tx}}$ will mainly contribute to the total delay. Therefore, the total delay is expressed as

$$\tau_k = \max\Bigg(\underbrace{\tau_k^{\text{CPU}}}_{\substack{\text{executed}\\ \text{at end-user}}},\ \underbrace{\left(\tau_{k,i}^{\text{Tx}} + \max\left(\tau_{k,i}^{\text{CPU, fog}}, \tau_{k,j}^{\text{CPU, fog}}, \tau_{k,i\to c}^{\text{Tx}}\right)\right)}_{\text{for the offloaded task data}}\Bigg). \tag{8}$$

IV. PROBLEM FORMULATION FOR TASK OFFLOADING AND RESOURCE ALLOCATION

The main objective is to complete the task execution within the delay deadline (i.e., τ_a^{task}) imposed by the certain application initiated by the end-user. As we do not consider the energy consumption issue, we let the end-user locally executes the task data until the delay deadline τ_a^{task}. Therefore, we drop τ_k^{CPU} in (9) with an assumption that $\tau_k^{\text{CPU}} \approxeq \tau_a^{\text{task}}$. As a result, we aim to obtain the following: $\min \tau_k'$, where

$$\tau_k' = \tau_{k,i}^{\text{Tx}} + \max\left(\tau_{k,i}^{\text{CPU, fog}}, \tau_{k,j}^{\text{CPU, fog}}, \tau_{k,i\to c}^{\text{Tx}}\right). \tag{9}$$

Therefore,

$$D_{k,i}^{\text{OL}} = D_k - D_k^{\text{CPU}} \equiv D_k - \frac{\tau_a^{\text{task}} f_k}{L_a}. \tag{10}$$

As an end-user executes one task at a time, we allocate the maximum CPU clock speed $f_k^{\max}$ to the task data processing, i.e., $f_k = f_k^{\max}$. As in [23], we take the assumption that a fog node adjusts its CPU rate to meet different amount of CPU resources for processing the certain task data. Let $f_i^{\max}$ be the maximum CPU rate for the ith fog node, then

$$\underbrace{\sum_{k=1}^{M_i} \mu_{k,i} f_{k,i}}_{\text{for } \forall k \in \mathcal{M}_i} + \underbrace{\sum_{j=1}^{J_i} \sum_{k'=1}^{|\mathcal{M}_j|} \beta_{k',j,i} f_{k',i}}_{\text{for } \forall k' \in \mathcal{K} \setminus \mathcal{M}_i} \leq f_i^{\max}. \tag{11}$$

A. PROBLEM FORMULATION

Our main objective is to find the optimal way (where to offload, i.e., secondary fog node or remote cloud server, and amount of task data to offload) to offload task data that cannot be entirely processed at the end-user side within the latency deadline. As discussed, a primary fog node receives the task data from the end-users directly under its coverage and from its neighbor fog nodes. Thus, we need to jointly optimize the computing resources (CPU rate allocated for end-user's task data execution) and transmission resources for offloading task data from primary fog node to the remote cloud. At the same time, the interference from the end-users under same primary fog node also plays an important role in transmission rate between the end-user to the primary fog node since these end-users share the same channel to offload their task data. Therefore, to find the number of end-users that select the ith fog node as their primary node, M_i is also an important factor to be considered.

We aim to jointly optimize the computing and transmission resource allocation in fog nodes (both primary and secondary fog node) to guarantee the minimum delay for each end-user's task completion considering tasks which are arrived from multi-users. The task offloading vector for the kth user is defined as $\Gamma_k = \left[\mu_{k,i}, \beta_{k,i,j}, \lambda_{k,c}, D_{k,i}^{\text{CPU}}, D_{k,i\to j}^{\text{OL}}, D_{k,i\to c}^{\text{OL}}\right]$. Next, we formulate the above optimization problem as:

$$\underset{M_i, \Gamma_k, x_i, f_{k,i}, f_{k,j}}{\text{minimize}} \ \tau_k' \ \forall k \tag{12a}$$

$$\text{subject to } \mu_{k,i}, \beta_{k,i,j}, \lambda_{k,i} \in \{0, 1\} \tag{12b}$$

$$D_{k,i}^{\text{CPU, fog}} + D_{k,i\to j}^{\text{OL}} + D_{k,i\to c}^{\text{OL}} \geq D_{k,i}^{\text{OL}} \tag{12c}$$

$$\sum_{k=1}^{M_i} r_{k,i} \leq r_i^{\max} \tag{12d}$$

$$\sum_{i=1}^{N} \lambda_{k,i,c}\, r_{k,i,c} \leq r_{k,c}^{\max} \tag{12e}$$

$$\text{and (11)}, \tag{12f}$$

where the constraint (12b) implies the offloading decision variables for the kth end-user. Moreover, the constraint (12c) corresponds to the condition that the total offloaded task of the kth end-user to the primary fog node must be processed in primary and secondary fog node and cloud server. Moreover, the constraint (12d) denotes that the total transmission rate between the ith fog node and all the users is under the maximum value $r_i^{\max}$. The constraint (12e) corresponds to the total transmission rate between the ith fog node and the cloud for all the users is limited by the maximum value $r_{k,c}^{\max}$.

First, the constraint (12b) is transformed into a quadratic equation as $x(x-1) = 0$, where $x \in \{\mu_{k,i}, \beta_{k,i,j}, \lambda_{k,i}\}$. Then, we introduce the auxiliary variables to convert the above optimization problem into a convex QCQP. The CVX toolbox [24] is used to obtain the optimum points, feasibility analysis is left for future work. Afterward, we introduce the auxiliary variables $\zeta_{k,i}^{\text{L}}$, $\zeta_{k,j}^{\text{O}}$, and $\zeta_{k,c}^{\text{O}}$, such as

$$\tau_{k,i}^{\text{CPU, fog}} \leq \zeta_{k,i}^{\text{L}}, \tag{13a}$$

$$\tau_{k,j}^{\text{CPU, fog}} \leq \zeta_{k,j}^{\text{O}}, \tag{13b}$$

$$\tau_{k,i\to c}^{\text{Tx}} \leq \zeta_{k,c}^{\text{O}}. \tag{13c}$$

Let $\alpha_k = \max\left[\zeta_{k,i}^{\text{L}}, \zeta_{k,j}^{\text{O}}, \zeta_{k,c}^{\text{O}}\right]$, such that $\{\zeta_{k,i}^{\text{L}}, \zeta_{k,j}^{\text{O}}, \zeta_{k,c}^{\text{O}}\} \leq \alpha_k \forall k \in \mathcal{M}_i$. Let $\mathbf{w}_k = [\mu_{k,i}, \beta_{k,i,j}, \mu_{k,j}, \lambda_{k,i}, \zeta_{k,i}^{\text{L}}, f_{k,i}, \zeta_{k,j}^{\text{O}}, f_{k,j}, r_{k,i,c}, \zeta_{k,c}^{\text{O}}, r_{k,i}]_{10\times 1}^{\intercal}$ denotes the variable matrix, where $(\cdot)^{\intercal}$ denotes the transpose of a matrix. Let $\mathbf{b}_{k,i}^{\text{L}} = [L_a D_{k,i}^{\text{CPU, fog}}, \mathbf{0}_{1\times 9}]^{\intercal}$, $\mathbf{b}_{k,j}^{\text{O}} = [L_a D_{k,i\to j}^{\text{OL}}, \mathbf{0}_{1\times 9}]^{\intercal}$, $\mathbf{b}_{k,c}^{\text{O}} = [0, 0, D_{k,i\to c}^{\text{OL}}, \mathbf{0}_{1\times 7}]^{\intercal}$, and $\mathbf{e}_q = [\mathbf{0}_{1\times(q-1)}\ 1\ \mathbf{0}_{1\times(10-q)}]^{\intercal}$ for $1 \leq q \leq 10$.

Then, we rewrite (13a)–(13c) as

$$\mathbf{e}_1^{\intercal}\, \mathbf{b}_{k,i}^{\text{L}} + \mathbf{w}_k^{\intercal}\, \mathbf{A}_{k,i}^{\text{L}} \mathbf{w}_k \leq 0, \tag{14a}$$

$$\mathbf{e}_1^{\intercal}\, \mathbf{b}_{k,j}^{\text{O}} + \mathbf{w}_k^{\intercal}\, \mathbf{A}_{k,c}^{\text{O}} \mathbf{w}_k \leq 0, \tag{14b}$$

$$\mathbf{w}_k^{\intercal}\, \mathbf{b}_{k,c}^{\text{O}} + \mathbf{w}_k^{\intercal}\, \mathbf{A}_{k,c}^{\text{O}} \mathbf{w}_k \leq 0, \tag{14c}$$

where

$$\mathbf{A}_{k,i}^{\text{L}} = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 2} & \mathbf{0}_{3\times 5} \\ \mathbf{0}_{2\times 3} & \mathbf{A}_{k,i}^{\text{L}'} & \mathbf{0}_{2\times 5} \\ \mathbf{0}_{5\times 3} & \mathbf{0}_{5\times 2} & \mathbf{0}_{5\times 5} \end{bmatrix}, \quad \mathbf{A}_{k,i}^{\text{L}'} = \begin{bmatrix} 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix},$$

$$\mathbf{A}_{k,j}^{\text{O}} = \begin{bmatrix} \mathbf{0}_{5\times 5} & \mathbf{0}_{5\times 2} & \mathbf{0}_{5\times 3} \\ \mathbf{0}_{2\times 5} & \mathbf{A}_{k,j}^{\text{O}'} & \mathbf{0}_{2\times 3} \\ \mathbf{0}_{3\times 5} & \mathbf{0}_{3\times 2} & \mathbf{0}_{3\times 3} \end{bmatrix}, \quad \mathbf{A}_{k,j}^{\text{O}'} = \begin{bmatrix} 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix},$$

and

$$\mathbf{A}_{k,c}^{\text{O}} = \begin{bmatrix} \mathbf{0}_{7\times 7} & \mathbf{0}_{7\times 2} & \mathbf{0}_{7\times 1} \\ \mathbf{0}_{2\times 7} & \mathbf{A}_{k,c}^{\text{O}'} & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 7} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 1} \end{bmatrix}, \quad \mathbf{A}_{k,c}^{\text{O}'} = \begin{bmatrix} 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix}.$$

Let $\mathbf{D}_k^{\text{I}} = [D_{k,i}^{\text{OL}}, -D_{k,i}^{\text{CPU}}, -D_{k,i,j}^{\text{OL}}, -D_{k,i,c}^{\text{OL}}, \mathbf{0}_{1\times 6}]^{\intercal}$. Then, (12c) becomes $(\mathbf{e}_1^{\intercal} + \mathbf{e}_2^{\intercal} + \mathbf{e}_3^{\intercal} + \mathbf{e}_4^{\intercal})\, \mathbf{D}_k^{\text{I}} \leq 0$. and (12d) becomes $\mathbf{e}_{10}^{\intercal}\, \mathbf{w}_k \leq r_i^{\max}\, \forall i \in \mathcal{N}$. We further rewrite (12e) as

$$\sum_{i=1}^{N} \mathbf{w}_k^{\intercal}\, \mathbf{A}_{k,i,c}\, \mathbf{w}_k \leq r_{k,c}^{\max}, \tag{15}$$

where

$$\mathbf{A}_{k,i,c} == \begin{bmatrix} \mathbf{0}_{2\times 10} \\ \mathbf{0}_{1\times 7} \ \frac{1}{2} \ \mathbf{0}_{1\times 2} \\ \mathbf{0}_{4\times 10} \\ \mathbf{0}_{1\times 2} \ \frac{1}{2} \ \mathbf{0}_{1\times 7} \\ \mathbf{0}_{2\times 10} \end{bmatrix}.$$

Moreover, we rewrite the constraint (12f) as

$$\sum_{k=1}^{M_i} \mathbf{w}_k^{\mathsf{T}} \mathbf{Q}_{k,i}\, \mathbf{w}_k + \sum_{j=1}^{J_i} \sum_{k'=1}^{M_j} \mathbf{w}_k^{\mathsf{T}} \mathbf{Q}_{k',i,j} \mathbf{w}_k \leq f_i^{\max}, \quad (16)$$

where

$$\mathbf{Q}_{k,i} = \begin{bmatrix} \mathbf{0}_{3\times 3} & \begin{matrix}\frac{1}{2}\\ \mathbf{0}_{2\times 1}\end{matrix} & \mathbf{0}_{3\times 6} \\ \begin{matrix}\frac{1}{2} & \mathbf{0}_{1\times 9}\end{matrix} & & \\ \multicolumn{3}{c}{\mathbf{0}_{6\times 10}} \end{bmatrix}$$

and

$$\mathbf{Q}_{k',i,j} = \begin{bmatrix} \mathbf{0}_{10\times 1} & \begin{matrix}\mathbf{0}_{3\times 1}\\ \frac{1}{2}\\ \mathbf{0}_{6\times 1}\end{matrix} & \mathbf{0}_{10\times 1} & \begin{matrix}0\\ \frac{1}{2}\\ \mathbf{0}_{8\times 1}\end{matrix} & \mathbf{0}_{10\times 6} \end{bmatrix}.$$

V. SIMULATION RESULTS

This section evaluates the performance of the proposed solution for task offloading in multiple task delay sensitive fog networks with Monte Carlo simulations. We consider that total $N = 5$ fog nodes and total $K = 15$ end-users are uniformly distributed over the network. We set $r_{k,c}^{\max}$ =1 Mbps, $r_i^{\max}$ =10 Mbps, $f_k^{\max} = 600 \times 10^6$ [cycles/s], $f_i^{\max} = 5 \times 10^9$ [cycles/s], and $f_c = 10 \times 10^9$ [cycles/s]. For multiple tasks, we consider two tasks with different processing density as $L_a = 1900$ [cycles/byte] (e.g., x264 constant bit rate encoding [25]) for task 1 and $L_a = 2500$ [cycles/byte] for task 2.

We show that the performance of average total delay versus input task data size in Fig. 2. The total delay for the task increases with the increase of input data size. We further compare the performance of proposed scheme with a baseline approach, called fixed resource allocation where the transmission resources are equally distributed over all the fog nodes and the fog node allocates an equal amount of CPU resources for each tasks. It is interesting to observe that the proposed approach outperforms the fixed resource allocation.

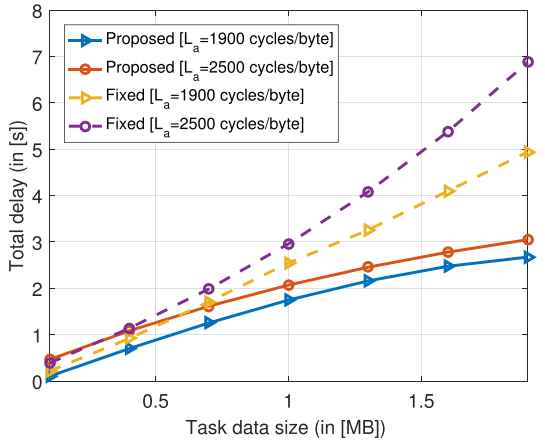

FIGURE 2. Total delay with task data size.

Moreover, Fig. 3 demonstrates the impact of delay deadline of the two tasks with different task processing density on the

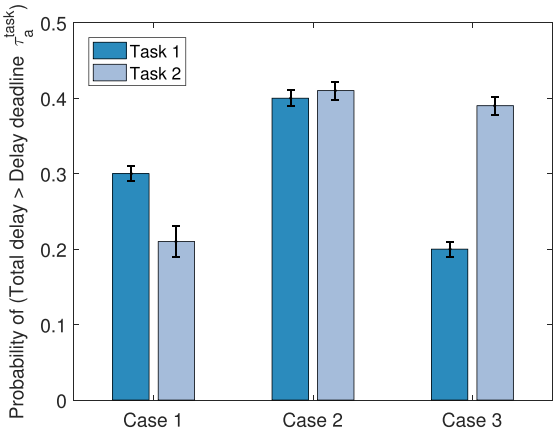

FIGURE 3. Delay violation with different deadline for the tasks. We consider the following cases Case 1: $[\tau_a^{task\ 1}=2s, \tau_a^{task\ 2}=4s]$, Case 2: $[\tau_a^{task\ 1}=2s, \tau_a^{task\ 2}=2s]$, and Case 3: $[\tau_a^{task\ 1}=4s, \tau_a^{task\ 2}=2s]$.

delay violation (i.e., probability of occurrence when the total delay does not meet the delay deadline). From Fig. 3, we see that the delay violation is higher for task 1 compared to the task 2 due to reason that the task 1 has more strict delay deadline compared to the task 2. In case 2, we reduce the delay deadline for the task 2 from 4s to 2s. Although we keep the same delay deadline for the task 1 in case 2 as that in case 1, because of the more strict deadline on task 2, the delay violation increases for both tasks. It is worthwhile to note that in case 2, although both tasks have same delay deadline, due to higher processing density of the task 2 compared to the task 1, the delay violation is slightly higher in task 2 than task 1. Furthermore, we have relaxed the delay deadline for the task 1 in case 3 compared to the deadline in case 2. As evident, the delay violation decreases with the increase of delay deadline. However, the reduction in delay violation is less in task 2 (note that, the delay deadline is the same as the previous case) as compared to the reduction in task 1. Thus, we can say that the if we relax the delay deadline for the tasks with lower processing density, it has a negligible impact on the reduction of the delay violation reduction for the tasks with high processing density.

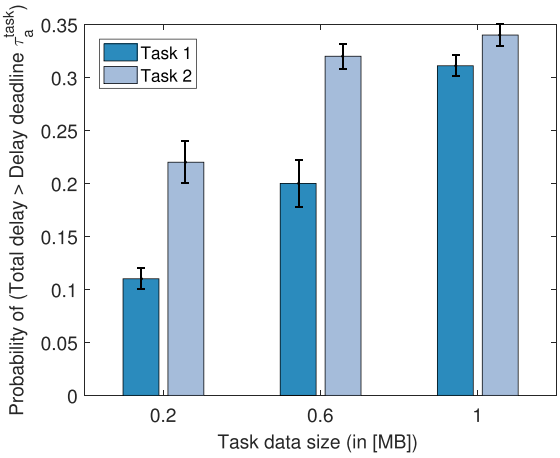

FIGURE 4. Delay violation with the task data size.

Finally, Fig. 4 shows the results for delay violation with task data size. It is clearly observed that the task data size has an adverse impact on the delay violation for both the tasks.

VI. CONCLUSION

In this paper, we address the issues of task data offloading in fog computing considering different delay deadline for the tasks initiated by the end-users. Our approach takes into account the delay deadline for different tasks, the transmission delay between primary fog node to the cloud, and secondary fog node's available computing resources while offloading the task data. Simulation results have shown that the proposed solution outperforms fixed computational and transmission resource allocation to satisfy the delay deadline. Moreover, a trade-off between the deadline on the latency and the delay violation is observed with numerous parameter settings. Further extensions of this work may include the investigation of task offloading for carrier-grade reliability and latency constraints with joint and competitive caching designs based on network utility maximization.

REFERENCES

[1] K. S. Kim, D. K. Kim, C.-B. Chae, S. Choi, Y.-C. Ko, J. Kim, Y.-G. Lim, M. Yang, S. Kim, B. Lim, K. Lee, and K. L. Ryu, "Ultrareliable and low-latency communication techniques for tactile Internet services," *Proc. IEEE*, vol. 107, no. 2, pp. 376–393, Feb. 2019.

[2] J. Liu and Q. Zhang, "Code-partitioning offloading schemes in mobile edge computing for augmented reality," *IEEE Access*, vol. 7, pp. 11222–11236, 2019.

[3] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12825–12837, 2018.

[4] P.-V. Mekikis, K. Ramantas, L. Sanabria-Russo, J. Serra, A. Antonopoulos, D. Pubill, E. Kartsakli, and C. Verikoukis, "NFV-enabled experimental platform for 5G Tactile Internet support in industrial environments," *IEEE Trans. Ind. Informat.*, to be published.

[5] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1826–1857, 3rd Quart., 2018.

[6] R. Vilalta, V. Lopez, A. Giorgetti, S. Peng, V. Orsini, L. Velasco, R. Serral-Gracia, D. Morris, S. De Fina, F. Cugini, P. Castoldi, A. Mayoral, R. Casellas, R. Martinez, C. Verikoukis, and R. Munoz, "TelcoFog: A unified flexible fog and cloud computing architecture for 5G networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 36–43, Aug. 2017.

[7] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 945–953.

[8] L. Jiao, H. Yin, H. Huang, D. Guo, and Y. Lyu, "Computation offloading for multi-user mobile edge computing," in *Proc. IEEE HPCC/SmartCity/DSS*, Jun. 2018, pp. 422–429.

[9] I. Sarrigiannis, E. Kartsakli, K. Ramantas, A. Antonopoulos, and C. Verikoukis, "Application and network VNF migration in a MEC-enabled 5G architecture," in *Proc. 23rd IEEE Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD)*, Sep. 2018, pp. 1–6.

[10] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[11] M.-H. Chen, B. Liang, and M. Dong, "Multi-user multi-task offloading and resource allocation in mobile cloud systems," *IEEE Wireless Commun.*, vol. 17, no. 10, pp. 6790–6805, Oct. 2018.

[12] M.-H. Chen, M. Dong, and B. Liang, "Joint offloading decision and resource allocation for mobile cloud with computing access point," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 3516–3520.

[13] M.-H. Chen, B. Liang, and M. Dong, "A semidefinite relaxation approach to mobile cloud offloading with computing access point," in *Proc. IEEE 16th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun./Jul. 2015, pp. 1–5.

[14] Y. Wu, Y. He, L. Qian, J. Huang, and X. S. Shen, "Optimal resource allocations for mobile data offloading via dual-connectivity," *IEEE Trans. Mobile Comput.*, vol. 17, no. 10, pp. 2349–2365, Oct. 2018.

[15] Y.-Y. Shih, W.-H. Chung, A.-C. Pang, T.-C. Chiu, and H.-Y. Wei, "Enabling low-latency applications in fog-radio access networks," *IEEE Netw.*, vol. 31, no. 1, pp. 52–58, Jan. 2017.

[16] M. Mukherjee, Y. Liu, J. Lloret, L. Guo, R. Matam, and M. Aazam, "Transmission and latency-aware load balancing for fog radio access networks," in *Proc. IEEE GLOBECOM*, Dec. 2018, pp. 1–6.

[17] M. Mukherjee, S. Kumar, M. Shojafar, Q. Zhang, and C. X. Mavromoustakis, "Joint task offloading and resource allocation for delay-sensitive fog networks," in *Proc. IEEE ICC*, May 2019, pp. 1–7.

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

[19] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proc. IEEE GLOBECOM*, Dec. 2016, pp. 1–6.

[20] S.-W. Ko, K. Huang, S.-L. Kim, and H. Chae, "Live prefetching for mobile computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3057–3071, May 2017.

[21] J. Huang, V. G. Subramanian, R. Agrawal, and R. Berry, "Joint scheduling and resource allocation in uplink OFDM systems for broadband wireless access networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 2, pp. 226–234, Feb. 2009.

[22] G. Yu, Y. Jiang, L. Xu, and G. Y. Li, "Multi-objective energy-efficient resource allocation for multi-RAT heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2118–2127, Oct. 2015.

[23] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[25] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput. (HotCloud)*, Jun. 2010, p. 4.

MITHUN MUKHERJEE (S'10–M'16) received the B.E. degree in electronics and communication engineering from the University Institute of Technology, Burdwan University, Bardhaman, India, in 2007, the M.E. degree in information and communication engineering from the Indian Institute of Science and Technology, Shibpur, India, in 2009, and the Ph.D. degree in electrical engineering from the Indian Institute of Technology Patna, Patna, India, in 2015. He is currently an Assistant Professor with the Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology, Maoming, China. He has (co)authored more than 80 publications in peer-reviewed international Transactions/journals and conferences. Dr. Mukherjee was a recipient of the 2016 EAI International Wireless Internet Conference, the 2017 International Conference on Recent Advances on Signal Processing, Telecommunications and Computing, the 2018 IEEE Systems Journal, and the 2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) Best Paper Award. He has been an Associate Editor of IEEE Access and a Guest Editor of the IEEE Internet of Things Journal, the IEEE Transactions on Industrial Informatics, ACM/Springer *Mobile Networks and Applications*, and *Sensors*. His current research interests include wireless communications, fog computing, and ultra-reliable low-latency communications.

SUMAN KUMAR received the M.Sc. degree in mathematics from the University of Hyderabad and the Ph.D. degree in mathematics from IIT Patna, India. He has done research in mathematical control theory. He is currently an Assistant Professor of mathematics with IGNTU Amarkantak, India. He has also served as a member with organizing committee in numerous international conferences. His current research interests include control theory, delay differential systems, abstract linear and nonlinear systems, and modeling and mathematical analysis of wireless communication systems.

QI ZHANG received the M.Sc. and Ph.D. degrees in telecommunications from the Technical University of Denmark (DTU), Denmark, in 2005 and 2008, respectively. She is currently an Associate Professor with the DIGIT, Department of Engineering, Aarhus University, Denmark. Besides her academic experiences, she has various industrial experiences. Her current research interests include the tactile Internet, the IoT, URLLC, mobile edge computing, massive machine type communication, non-orthogonal multiple access (NOMA), and compressed sensing. She was a Co-Chair of the Co-operative and Cognitive Mobile Networks (CoCoNet) Workshop in the ICC conference 2010–2015 and a TPC Co-Chair of BodyNets 2015. She is serving as an Editor of the *Journal on Wireless Communications and Networking* (EURASIP).

RAKESH MATAM (M'14) received the bachelor's degree in computer science from Jawaharlal Nehru Technological University at Hyderabad, the master's degree from Kakatiya University Warangal, India, and the Ph.D. degree in computer science from IIT Patna, in 2014. In 2014, he joined the Department of Computer Science, Indian Institute of Information Technology Guwahati (IIIT Guwahati), as an Assistant Professor, where he is currently a member of the Design and Innovation Center, and a Principal Investigator of a funded research project sponsored by the Government of India. His current research interests include wireless networks, network security, and cloud computing. He has also served as a member in organizing committee in numerous international conferences.

CONSTANDINOS X. MAVROMOUSTAKIS received a five-year Dipl.Eng. (B.Sc., B.Eng., and M.Eng.) in electronic and computer engineering from the Technical University of Crete, Greece, the M.Sc. degree in telecommunications from the University College of London, U.K., and the Ph.D. degree from the Department of Informatics, Aristotle University of Thessaloniki, Greece. He is currently a Professor with the Department of Computer Science, University of Nicosia, Cyprus. Prof. Mavromoustakis is leading the Mobile Systems Laboratory (MOSys Lab., http://www.mosys.unic.ac.cy/), Department of Computer Science, University of Nicosia. He has been an Active Member (Vice-Chair) of IEEE/ R8 regional Cyprus section, since 2016, and since 2009, he has been serving as the Chair of C16 Computer Society Chapter of the Cyprus IEEE section. He has a dense research work outcome in Mobile and Wearable computing systems and the Internet-of-Things (IoT), consisting of numerous refereed publications including several books (IDEA/IGI, Springer, and Elsevier). He has served as a Consultant to many industrial bodies including Intel Corporation LLC (www.intel.com), and he is a Management Member of the IEEE Communications Society (ComSoc) Radio Communications Committee (RCC) and a Board Member the IEEE-SA Standards IEEE SCC42 WG2040. He has participated in several FP7/H2020/Eureka and National projects.

YUNRONG LV received the Ph.D. degree from Zhejiang University, China. He is currently a Distinguished Full Professor with the Guanagdong University of Petrochemical University and an Executive Director of the Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, China. He was a Chief Scientist in several national and international companies. He has been involved with several key projects in Guangdong. He leads many industrial Internet-related projects in several national and international companies.

GEORGE MASTORAKIS received the B.Eng. degree in electronic engineering from UMIST, in 2000, the M.Sc. degree in telecommunications from UCL, in 2001, and the Ph.D. degree in telecommunications from the University of the Aegean, in 2008. He is currently an Associate Professor with the Department of Management Science and Technology, Hellenic Mediterranean University, Greece. He has more than 250 publications in various international conferences proceedings, workshops, scientific journals, and book chapters. His current research interests include cognitive radio networks, the Internet of Things, energy-efficient networks, big data analytics, and mobile computing.