# Computation Offloading Strategy in Heterogeneous Fog Computing with Energy and Delay Constraints

Mithun Mukherjee[*], Vikas Kumar[†], Suman Kumar[‡], Rakesh Matam[‖], Constandinos X. Mavromoustakis[¶],
Qi Zhang[§], M. Shojafar[**], and George Mastorakis[††]
[*] Guangdong University of Petrochemical Technology, Maoming, China, m.mukherjee@ieee.org
[‡] Bharat Sanchar Nagar Limited, India, vikaskumar@iitp.ac.in
[‡] Department of Mathematics, IGNTU Amarkantak, MP, India, suman@igntu.ac.in
[§] Indian Institute of Information Technology Guwahati, India, rakesh@iiitg.ac.in
[¶] University of Nicosia, Nicosia, Cyprus, mavromoustakis.c@unic.ac.cy
[‖] DIGIT, Department of Engineering, Aarhus University, Aarhus, Denmark, qz@eng.au.dk
[**] 5GIC/ICS, University of Surrey, Guildford, UK, m.shojafar@surrey.ac.uk
[††] Hellenic Mediterranean University, Greece gmastorakis@staff.teicrete.gr

***Abstract*—In fog computing, end-users can offload the computation-intensive tasks to the fog node in the proximity. Additionally, the fog nodes also offload the tasks to the cloud and neighboring fog node, thereby forming a vertical and horizontal collaboration, respectively. In this paper, we propose a offloading strategy in fog computing to minimize the cost that is a weighted sum of energy consumption and total delay for the task processing per end-user. We take the heterogeneous nature of the fog computing nodes that have different CPU frequency to process the tasks. We aim to find an optimal amount of task data to be either locally processed or offloaded to the preferable fog node.cloud under the energy and delay constraints. We then formulate the optimization problem into a non-convex quadratically constrained quadratic program and provide an efficient solution to this problem by semidefinite relaxation. Finally, our proposed offloading scheme is evaluated by simulation to demonstrate the offloading profile and optimal cost of the offloading with a wide range of parameter settings.**

## I. Introduction

The fog computing framework [1]–[3] addresses the limitations/challenges faced in the cloud environment by bringing in part of computing, storage and processing capabilities closer to the end-devices that require them. By doing so, the fog environment reduces the communication cost and latency that are crucial for delay sensitive Internet of Things (IoT) applications and Cyber Physical Systems (CPS). That is, the fog layer acts as an intermediate facility between the cloud and end-devices. It is comprised of a network of devices that are typically located at the edge of the network like gateways, routers, access points and base-stations, which are connected to power supply and have storage and processing capabilities that can be leveraged upon. Constrained end-devices that usually to rely on cloud for their storage or computing needs will now offload their computation and storage requests to the fog layer. A set of end-devices are usually served by a single fog node, typically referred to as primary fog node. Fog nodes that are interconnected may either collaborate horizontally by offloading part of the tasks to another fog node(s) or vertically by in-turn collaborating with the cloud. To achieve this, several offloading techniques [4]–[10] have been proposed that aim to offload efficiently by optimizing either delay, communication cost or to achieve load balancing of jobs.

### A. Motivation

Generally, the concept of the fog-cloud network plays an important role in the delay sensitive task processing for the end-user. Moreover, to minimize the energy consumption due to the local processing of a large number of tasks, the end-users often prefer to offload their tasks to the fog nodes in the proximity. The collaborative nature of fog computing (i.e., horizontal collaboration with neighboring fog node and the vertical collaboration with the remote cloud) aims to provide additional computational resources with a low energy consumption per bit compared to the end-user. A few downsides exist as follows. First, the computational capacity of each fog node (i.e., the CPU clock speed of the device) is different. The main reason is that the fog computing nodes consist of various network devices, such as a switch, gateway, routers, or mini-data centers. Therefore, the CPU clock speed of fog node is different from each other. Second, we cannot ignore the local energy consumption at the fog nodes, and the delay occurred due to transmission time from fog node to neighboring fog node and cloud. Thus, we require to ensure that *the end-user must obtain reduced task processing time with minimum energy consumption at the computing devices compared to the local processing at the end-user side.*

### B. Our contributions

In this paper, we study the task offloading in the fog-cloud network to minimize the task processing delay and energy consumption. We propose an offloading policy to find the optimal place where to offload the task data and the amount of offloaded task data. We summarize our main contributions, which are as follows:

- We mainly consider the heterogeneous nature of the fog computing nodes. Therefore, the individual computational

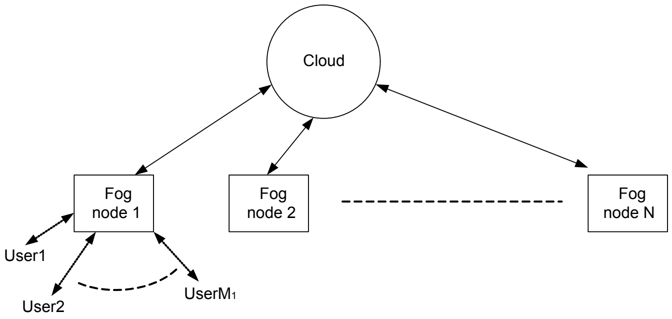

Fig. 1. An illustration of fog computing system model.

capability of the fog node will play an important role in the local task processing time at the fog node, thereby affecting the total task processing delay.

- We focus on the minimization of the total system cost that is calculated as the weighted sum of total energy consumption and task processing delay. We also analyze the tradeoff between the task processing delay and the energy consumption in the task offloading.
- We further apply the semidefinite relaxation (SDR) to the Quadratically Constraint Quadratic Programming (QCQP) problem. We then solve the separable semidefinite programming (SDP) problem. Finally, the simulation results show that our proposed offloading policy can reduce the energy consumption and task processing delay compared with other benchmark offloading schemes.

The rest of the paper is organized as follows. In Section II, we describe the system model with the delay model and the energy consumption model. We formulate the optimization problem in Section III and transform the problem to a QCQP problem and solve it through the SDR method. The simulation results are presented in Section IV. Finally, we conclude our work in Section V.

## II. System Model

As illustrated in Fig. 1, we consider a fog computing network with a set of fog nodes $\mathcal{N} = \{1, 2, \ldots, N\}$, a set of end-users $\mathcal{K} = \{1, 2, \ldots, K\}$, and one remote cloud server. We assume that these fog nodes and end-users are uniformly and randomly distributed over the entire network. We denote $\mathcal{M}_i = \{1, 2, \ldots, M_i\}$, $\sum_{i=1}^{N} M_i = K$ as the set of end-users that chose the $i$th fog node as their primary fog node. We take the scenario when the the end-user can offload the task to only one primary fog node. If the computational resources offered by the primary fog node is not sufficient, then the primary fog node decides to further offload the task to neighboring fog node and cloud. We further assume that the end-user and the fog node can simultaneously offload and execute the tasks.

We write the task arrival rate from the $k$th end-user , $k \in \mathcal{M}_i$, to the $i$th fog node as $\lambda_{k,i}^{\text{OL}} = \alpha_{k,i}\,\lambda_k$ Then, the remaining tasks are locally executed at the end-user side. The task arrival rate at the local processing queue is calculated as $\lambda_k^{\text{CPU}} = (1 - \alpha_{k,i})\,\lambda_k$ . Then, $\lambda_{\text{fog},i} = \lambda_{\text{fog},i}^{\text{CPU}} + \lambda_{\text{fog},i}^{\text{OL}}$ .

TABLE I
Main Notation Definition

| Symbols | Definition |
|---|---|
| $N$ | The number of total fog nodes in the network |
| $K$ | The number of total end-users in the network |
| $M_i$ | The number of end-users under $i$th primary fog node |
| $D$ | The data size of a single sub-task |
| $\lambda_k$ | The task arrival rate at the $k$th end-user |
| $\lambda_k^{\text{CPU}}$ | The task arrival rate at the local task execution queue of the $k$th end-user |
| $\lambda_k^{\text{OL}}$ | The task offloading rate at the offloading queue of the $k$th end-user |
| $\lambda_{\text{fog},i}$ | The subtask arrival rate at the $i$th fog node |
| $\lambda_{\text{fog},i}^{\text{CPU}}$ | The subtask arrival rate at the local task execution queue of the $i$th fog node |
| $\lambda_{\text{fog},i}^{\text{OL}}$ | The subtask offloading rate at the offloading queue of the $i$th fog node |
| $L_{k,a}$ | The processing density for the $a$th task initiated by the $k$th end-user |
| $f_k$ | The CPU clock speed of the $k$th end-user |
| $f_i$ | The CPU clock speed of the $i$th fog node |
| $\mu_k$ | The service rate at the local execution queue of the $k$th end-user |
| $\mu_{\text{fog},i}$ | The service rate at the local execution queue of the $i$th fog node |
| $r_{i,c}$ | The offloading-rate from the $i$th fog node to the cloud |
| $r_{k,i}$ | The offloading-rate from the $k$th end-user to the $i$th fog node |
| $E_{k,i}^{\text{Local}}$ | Energy consumption per second for task processing at end user. |
| $E_k^{\text{OL}}$ | Energy consumption per second for the task to transmit from end user to its parent fog node. |
| $E_i^{\text{Local}}$ | Energy consumption per second for the task processing locally at fog node. |
| $E_{i,j}^{\text{OL}}$ | Energy consumption per second for the task to transmit from fog node to other fog node. |
| $E_c^{\text{Local}}$ | Energy consumption per second for the task to process at cloud. |
| $E_{i,c}^{\text{OL}}$ | Energy consumption per second for the task to transmit from fog node to cloud. |

### A. Delay Model

In our delay model, we consider the a) average response time in end-user, fog node (primary and neighboring fog node), and cloud server, and b) uploading time from end-user to the primary fog node, offloading time from primary fog node to neighboring fog node, and uploading time from the primary fog node to the remote cloud.

*1) Average response time:* As the service rate depends on the task processing density and CPU clock speed, the service time of the $k$th end-user is

$$\mathfrak{T}_k^{\text{CPU}} = \frac{L\,D}{f_k}\,. \tag{1}$$

Thus, the the service rate at the $k$th end-user can be written as $\mu_k = 1/\mathfrak{T}_k^{\text{CPU}}$. Moreover, assuming M/M/1 queue model with a mean task arrival rate $\lambda_k^{\text{CPU}}$ at the $k$th end-user's local processing queue, the average response time including waiting time at the local queue and local task execution time at local processing queue for the $k$th end-user is expressed as

$$\mathfrak{T}_k^{\mathsf{Local}} = \frac{1}{\mu_k - \lambda_k^{\mathsf{CPU}}} \, . \tag{2}$$

Assuming a mean task arrival rate $\lambda_i^{\mathsf{CPU}}$ at the $i$th fog node's local processing queue, the average response time including waiting time at the local queue and local task execution time at local processing queue for the $k$th end-user is expressed as

$$\mathfrak{T}_i^{\mathsf{Local}} = \frac{1}{\mu_i - \lambda_i^{\mathsf{CPU}}} \, , \tag{3}$$

where the service rate $\mu_i = 1/\mathfrak{T}_i^{\mathsf{CPU}}$ with $\mathfrak{T}_i^{\mathsf{CPU}} = L\,D/f_i$.

*2) Uploading time:* When the $k$th end-user offloads the task data to the $i$th fog node, the uploading time from the $k$th end-user to the $i$th fog node becomes

$$\mathfrak{T}_{k,i}^{\mathsf{OL}} = \lambda_k^{\mathsf{OL}}\, D/r_{k,i} \, . \tag{4}$$

Now, the offloading time from the $i$th fog node to the $j$th neighboring fog node is expressed as

$$\mathfrak{T}_{\text{fog},i,j}^{\mathsf{OL}} = \beta_{i,j}\, \lambda_{\text{fog},i}^{\mathsf{OL}}\, D/r_{i,j} \, . \tag{5}$$

Further, the uploading time from the $i$th fog node to the cloud is written as

$$\mathfrak{T}_{\text{fog},i,c}^{\mathsf{OL}} = \gamma_{i,c}\, \lambda_{\text{fog},i}^{\mathsf{OL}}\, D/r_{i,c} \, . \tag{6}$$

As a result, the total time to process the $k$th end-user's task is expressed as

$$\mathfrak{T}_k = \mathfrak{T}_{k,i}^{\mathsf{OL}} + \text{min}\Big(\mathfrak{T}_k^{\mathsf{CPU}}, \mathfrak{T}_i^{\mathsf{CPU}}, \big(\mathfrak{T}_{\text{fog},i,j}^{\mathsf{OL}} + \mathfrak{T}_j^{\mathsf{CPU}}\big), \mathfrak{T}_{\text{fog},i,c}^{\mathsf{OL}}\Big) \tag{7}$$

### *B. Energy Consumption Model*

The energy consumption due to the uploading the task from the $k$th end-user to the $i$th primary fog node is expressed as $E_k^{\mathsf{OL}} = (P_0 + k_t\, P_t)\mathfrak{T}_{k,i}^{\mathsf{OL}}$,

The energy consumption due to the local task processing at the $i$th fog node is expressed as $E_i^{\mathsf{Local}} =$

## III. Problem Formulation and Proposed Offloading Strategy

In this section, we define our problem to jointly optimize the energy consumed and time delay for each individual end user present in the network. Cost function for each user is defined as $\mathbf{C}_k = \mathbf{a}_k^e E_k + \mathbf{a}_k^t \mathfrak{T}_k$. Where $\mathbf{a}_k^e$, $\mathbf{a}_k^t \in [0,1]$, denotes the weights of energy consumed and delay for $k$th user. Here, we formulate the joint optimization of offloading decision, $\mathbf{od} = [\boldsymbol{\alpha}_{k,i}, \boldsymbol{\beta}_{k,i,j}, \boldsymbol{\gamma}_{k,i,c}]$, with assumption that fog node are arranged in descending order with total number of task arrived to each fog node ie. $i = 1 \dots N$, as follows

$$
\begin{aligned}
\min_{\mathbf{od}} \quad & \max\ \mathbf{C}_k \ \forall\ k & \text{(8a)}\\
\text{s.t.} \quad & \text{(C1)} : \alpha_{k,i}, \beta_{k,i,j}, \gamma_{k,i,c} \in [0,1], & \text{(8b)}\\
& \text{(C2)} : \alpha_{k,i} + \sum_{j=i}^{N} \beta_{k,i,j} + \gamma_{k,i,c} = 1, & \text{(8c)}\\
& \text{(C3)} : \lambda_{\text{fog},i}^{\mathsf{CPU}} < \mu_{\text{fog},i}, & \text{(8d)}\\
& \text{(C4)} : r_{k,i} > 0, & \text{(8e)}\\
& \text{(C5)} : \sum_{k=1}^{M_i} r_{k,i} \leq r_i \qquad\qquad , & \text{(8f)}\\
& \text{(C6)} : r_{i,c} > 0, & \text{(8g)}\\
& \text{(C7)} : \sum_{i=1}^{N} r_{i,c} \leq r_c, & \text{(8h)}\\
& \text{(C8)} : \sum_{i=1}^{N} M_i = K\,, & \text{(8i)}
\end{aligned}
$$

Where in constraint (C2) if $i = j$, then $\beta_{k,i,j}$ shows local processing of $k$th user task at its parent node. Constraint (C3) indicates that each $i$th fog node can not exceed its service rate. Constraint (C4) and (C6) shows non-negative transfer rate of $k$th user task to its fog node and $i$th fog node to cloud respectively. Similarly, constraint (C5) and (C7) shows the transmission rate of $k$th user task to its fog node and $i$th fog node to cloud respectively. Constraint (C8) indicates total number of end users present in the network.

Optimization problem defined in (8_change) is not convex due to the variables $\mathbf{od}$. It is a mixed-integer non-linear programming problem, which can be generally NP-hard. We transform (8_change) into a QCQP problem and then semidefinite relaxation (SDR) approach is applied, which can be solved using convex optimization toolbox in CVX. Let us introduce a slack variable $\zeta$, such that max $\mathbf{C}_k = \zeta$. So overall cost with $\mathbf{od}$ can be expressed as

$$
\begin{aligned}
&\sum_{i=1}^{N}\sum_{k=1}^{M_i} \big((\mathbf{a}_k^e E_k^{\text{Local}} + \mathbf{a}_k^t)\mathfrak{T}_k^{\text{CPU}} \alpha_{k,i}\big) + \sum_{i=1}^{N}\sum_{k=1}^{M_i} (\mathbf{a}_k^e E_k^{\text{OL}} + \mathbf{a}_k^t) \\
&\quad \mathfrak{T}_{k,i}^{\text{OL}}(1 - \alpha_{k,i})) + \sum_{i=1}^{N}\sum_{k=1}^{M_i} \big((\mathbf{a}_k^e E_i^{\text{Local}} + \mathbf{a}_k^t)\mathfrak{T}_i^{\text{CPU}} \beta_i\big) \\
&\quad + \sum_{i=1}^{N}\sum_{j=i}^{N}\sum_{k=1}^{M_i} \big((\mathbf{a}_k^e E_{i,j}^{\text{OL}} + \mathbf{a}_k^t)\mathfrak{T}_{fog,i,j}^{\text{OL}} \beta_{i,j}\big) \\
&\quad + \sum_{i=1}^{N}\sum_{k=1}^{M_i} \big((\mathbf{a}_k^e E_{i,c}^{\text{OL}} + \mathbf{a}_k^t)\mathfrak{T}_{i,c}^{\text{OL}}\big) \\
&\quad + \sum_{i=1}^{N}\sum_{k=1}^{M_i} \big((\mathbf{a}_k^e E_c^{\text{Local}} + \mathbf{a}_k^t)\big)\, \mathfrak{T}_c^{\text{CPU}} \gamma_{i,c} \leq \zeta
\end{aligned} \tag{9}
$$

As we have assumed that the fog nodes are arranged in descending order based on task arrived from their respective end-users. Thus, for $i = 1$, *i.e.*, the first fog node does not receive any offloaded data from any other fog node. Considering this assumption and (9) let us define a $8 \times 1$ vector $\mathbf{w}_{k,i,j} = [\alpha_{k,i}, \beta_{i,j}, \gamma_{i,c}, r_{k,i}, r_{i,c}, \zeta, \lambda_{fog,i}, M_i]^\intercal$ be the variable matrix. Then, the matrix form of (8_change) becomes

Let $\mathbf{e}_q = [\mathbf{0}_{1\times(q-1)}\ 1\ \mathbf{0}_{1\times(8-q)}]^\intercal$ for $1 \le q \le 8$, therefore, the objective function is written as

$$\min_{\mathbf{w}_{k,i,j}} \quad \mathbf{b}_k^\intercal \mathbf{w}_{k,i,j} \tag{10a}$$

$$\text{s.t.} \quad 0 \le \mathbf{e}_u^\intercal \mathbf{w}_{k,i,j} \le 1 \quad \forall u \in \{1,2,3\}, \tag{10b}$$

$$\mathbf{e}_1^\intercal \mathbf{w}_{k,i,j} + \sum_{j=i}^{N} \mathbf{e}_2^\intercal \mathbf{w}_{k,i,j} + \mathbf{e}_3^\intercal \mathbf{w}_{k,i,j} = 1, \tag{10c}$$

$$\mathbf{e}_7^\intercal \mathbf{w}_{k,i,j} \le \mu_{\text{fog},i}\,, \tag{10d}$$

$$\mathbf{e}_4^\intercal \mathbf{w}_{k,i,j} > 0, \tag{10e}$$

$$\sum_{k=1}^{M_i} \mathbf{e}_4^\intercal \mathbf{w}_{k,i,j} \le r_i, \tag{10f}$$

$$\mathbf{e}_5^\intercal \mathbf{w}_{k,i,j} > 0, \tag{10g}$$

$$\sum_{i=1}^{N} \mathbf{e}_5^\intercal \mathbf{w}_{k,i,j} \le r_c, \tag{10h}$$

$$\sum_{i=1}^{N} \mathbf{e}_8^\intercal \mathbf{w}_{k,i,j} = K \tag{10i}$$

Now, we transform the optimization problem into a homogeneous separable QCQP form. Define $\mathbf{Z}_{k,i,j} = \begin{bmatrix}\mathbf{w}_{k,i,j}^\intercal & 1\end{bmatrix}^\intercal$. Thus, the above optimization problem becomes

$$\min_{\mathbf{Z}_{k,i,j}} \quad \mathbf{Z}_{k,i,j}^\intercal\, \mathbf{Q}^k\, \mathbf{Z}_{k,i,j} \tag{11a}$$

$$\text{s.t.} \quad 0 \le \mathbf{Z}_{k,i,j}^\intercal\, \mathbf{Q}^u\, \mathbf{Z}_{k,i,j} \le 1 \quad \forall u \in \{1,2,3\}, \tag{11b}$$

$$\sum_{j=i+1}^{N} \mathbf{Z}_{k,i,j}^\intercal\, \mathbf{Q}_1\, \mathbf{Z}_{k,i,j} + \mathbf{Z}_{k,i,j}^\intercal\, \mathbf{Q}_2\, \mathbf{Z}_{k,i,j} = 1, \tag{11c}$$

$$\mathbf{Z}_{k,i,j}^\intercal\, \mathbf{Q}^\lambda\, \mathbf{Z}_{k,i,j} \le \mu_{\text{fog},i}, \tag{11d}$$

$$\mathbf{Z}_{k,i,j}^\intercal\, \mathbf{Q}^r\, \mathbf{Z}_{k,i,j} > 0, \tag{11e}$$

$$\sum_{k=1}^{M_i} \mathbf{Z}_{k,i,j}^\intercal\, \mathbf{Q}^r\, \mathbf{Z}_{k,i,j} \le r_i, \tag{11f}$$

$$\mathbf{Z}_{k,i,j}^\intercal\, \mathbf{Q}^c\, \mathbf{Z}_{k,i,j} > 0, \tag{11g}$$

$$\sum_{i=1}^{N} \mathbf{Z}_{k,i,j}^\intercal\, \mathbf{Q}^c\, \mathbf{Z}_{k,i,j} \le r_c, \tag{11h}$$

$$\sum_{i=1}^{N} \mathbf{Z}_{k,i,j}^\intercal\, \mathbf{Q}^m\, \mathbf{Z}_{k,i,j} = K \tag{11i}$$

where

$$\mathbf{Q}^u = \begin{bmatrix}\mathbf{0}_{8\times8} & \frac{1}{2}\mathbf{e}_u \\ \frac{1}{2}\mathbf{e}_u^\intercal & 0\end{bmatrix} \ \forall u \in \{1,2,3\}\,,\ \mathbf{Q}^k = \begin{bmatrix}\mathbf{0}_{8\times8} & \frac{1}{2}\mathbf{b}_k \\ \frac{1}{2}\mathbf{b}_k^\intercal & 0\end{bmatrix}$$

$$\mathbf{Q}^r = \begin{bmatrix}\mathbf{0}_{8\times8} & \frac{1}{2}\mathbf{e}_4 \\ \frac{1}{2}\mathbf{e}_4^\intercal & 0\end{bmatrix}, \mathbf{Q}^c = \begin{bmatrix}\mathbf{0}_{8\times8} & \frac{1}{2}\mathbf{e}_5 \\ \frac{1}{2}\mathbf{e}_5^\intercal & 0\end{bmatrix},$$

TABLE II
SIMULATION PARAMETERS

| Parameters | Values |
|---|---|
| CPU frequency of the end-user ($f_k$) | $0.5 \times 10^9$ [cycles/s] |
| Maximum CPU frequency of fog node ($f_i$) | $2.5 \times 10^9$ [cycles/s] |
| CPU frequency of the cloud ($f_c$) | $4 \times 10^9$ [cycles/s] |
| Uploading rate from end-user to fog ($r_{k,i}$) | 72.2 Mbps |
| Uploading rate from fog to cloud ($r_{i,c}$) | 30 Mbps |
| Task density ($L$) | 2300 [cycles/byte] |
| Task size ($D$) | 50 Kbit |

$$\mathbf{Q}^m = \begin{bmatrix}\mathbf{0}_{8\times8} & \frac{1}{2}\mathbf{e}_8 \\ \frac{1}{2}\mathbf{e}_8^\intercal & 0\end{bmatrix}, \mathbf{Q}^\lambda = \begin{bmatrix}\mathbf{0}_{8\times8} & \frac{1}{2}\mathbf{e}_7 \\ \frac{1}{2}\mathbf{e}_7^\intercal & 0\end{bmatrix},$$

$$\mathbf{Q}_1 = \begin{bmatrix}\mathbf{0}_{8\times8} & \frac{1}{2}\mathbf{e}_2 \\ \frac{1}{2}\mathbf{e}_2 & 0\end{bmatrix}, \mathbf{Q}_2 = \begin{bmatrix}\mathbf{0}_{8\times8} & \frac{1}{2}\mathbf{e}_3 \\ \frac{1}{2}\mathbf{e}_3 & 0\end{bmatrix}.$$

where $\mathbf{b}_k = [\lambda_k\, D, \mathbf{0}_{1\times7}]_{1\times8}^\intercal$. We apply SDR to solve above QCQP problem. Define $\mathbf{Y}_{k,i,j} \equiv \mathbf{Z}_{k,i,j}\, \mathbf{Z}_{k,i,j}^\intercal$. It is clearly observed that by dropping the $\texttt{rank}(\mathbf{Y}_{k,i,j}) = 1$, we obtain the following SDP problem

$$\min_{\mathbf{Y}_{k,i,j}} \quad \text{Tr}(\mathbf{Q}^k\, \mathbf{Y}_{k,i,j}) \tag{12a}$$

$$\text{s.t.} \quad 0 \le \text{Tr}(\mathbf{Q}^u\, \mathbf{Y}_{k,i,j}) \le 1 \quad \forall u \in \{1,2,3\}, \tag{12b}$$

$$\sum_{j=i+1}^{N} \text{Tr}(\mathbf{Q}_1\, \mathbf{Y}_{k,i,j}) + \text{Tr}(\mathbf{Q}_2\, \mathbf{Y}_{k,i,j}) = 1, \tag{12c}$$

$$\text{Tr}(\mathbf{Q}^\lambda\, \mathbf{Y}_{k,i,j}) \le C,, \tag{12d}$$

$$\text{Tr}(\mathbf{Q}^r\, \mathbf{Y}_{k,i,j}) > 0, \tag{12e}$$

$$\sum_{k=1}^{M_i} \text{Tr}(\mathbf{Q}^r\, \mathbf{Y}_{k,i,j}) \le r_i, \tag{12f}$$

$$\text{Tr}(\mathbf{Q}^c\, \mathbf{Y}_{k,i,j}) > 0, \tag{12g}$$

$$\sum_{i=1}^{N} \text{Tr}(\mathbf{Q}^c\, \mathbf{Y}_{k,i,j}) \le r_c, \tag{12h}$$

$$\sum_{i=1}^{N} \text{Tr}(\mathbf{Q}^m\, \mathbf{Y}_{k,i,j}) = K \tag{12i}$$

We solve the above SDP problem in a polynomial time using a standard SDP software SeDuMi [11].

## IV. SIMULATION RESULTS

This section evaluates the performance of proposed optimal solution for task offloading with Monte Carlo simulations. The results are averaged over 10,000 different runs. Extensive simulations are conducted in MATLAB. The simulation parameters are summarized in Table II.

In Fig. 2, we study the cost with various number of end-users per fog node. In particular, the cost increases with the number of end-users per fog node. From the figure, we observe that fog CPU rate has a positive impact to reduce the cost per end-user. However, the cost performance of the high CPU fog node converges with the other low CPU fog node when the number of end-users per fog node increases. The reason is that

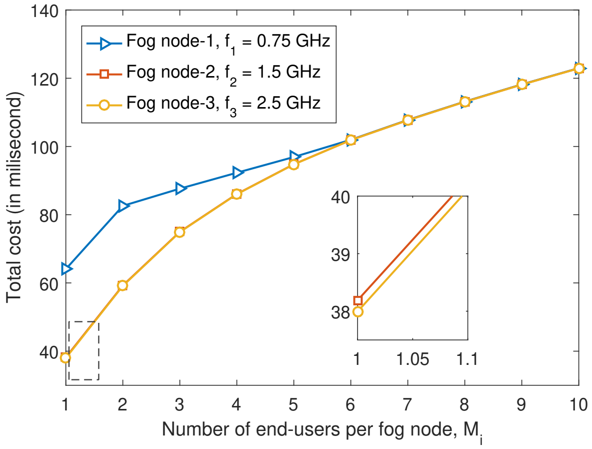

Fig. 2. Cost performance, $\rho = 0.2$ (millisecond/Jule), task arrival rate per end-user $\lambda_k$ = 35 tasks/second, number of fog node $N = 3$.

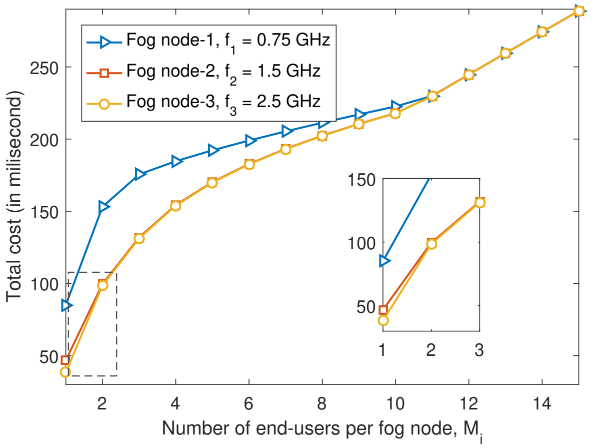

Fig. 3. Cost performance, $\rho = 1.4$ (millisecond/Jule), task arrival rate per end-user $\lambda_k$ = 35 tasks/second, number of fog node $N = 3$.

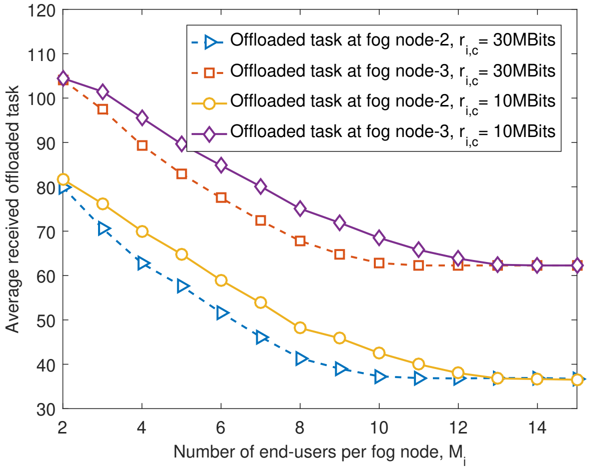

Fig. 4. Offloading profile, i.e., received offloaded tasks from neighboring fog nodes, $\rho = 1.4$ (millisecond/Jule).

at higher number of end-users at fog node, the CPU utilization saturates. Although, at higher number of fog node, although the high CPU rate can process higher number of tasks, the neighboring fog nodes also offload to high CPU frequency fog node. This results highlight that in our proposed scheduling scheme, the task data can be offloaded from the low CPU fog node to the high CPU fog node to exploit the computational resource utilization provided by the fog nodes. Moreover, when we increase the weight of energy consumption relative to delay in Fig. 3, we have the following observation: a) the total cost increases with the increase of the weight factor $\rho$ and b) advantage of higher CPU frequency in fog node is more visible compared to the low CPU frequency.

Fig. 4 depicts the offloading profile of the fog node with the number of end-user per fog node. As in previous, we take equal number of end-users per fog node. We assume the fog node-1 has lowest and fog node-3 has the highest CPU frequency, whereas the fog node-2 has the intermediate CPU frequency. As obvious, the fog node does not receive any task data from other two fog nodes. It is evident from the Fig. 4 that a more amount of tasks are offloaded to the higher CPU fog node to increase the utilization of the computational resources provided by the fog nodes. However, as the number of end-users per fog node increases, the high CPU fog node also receives a higher number of tasks from its own end-users compared to the scenario with lower number of end-users per fog node. As a result, the amount of tasks offloaded from the low CPU fog node decreases with the increase of end-users per fog node. Therefore, rather than offloading the data to the neighboring fog node, the fog node prefer to offload the task data to the remote cloud.

Moreover, amount of offloaded task to the neighboring node decreases when we increase the the uploading rate from the fog node to the cloud, $r_{i,c} = 30$ Mbits/s from 10 Mbits/s. The main reason is that due to low transmission delay to the cloud, more number of task data is offloaded to the cloud than offloading to the neighboring fog node. Besides, we also have some intriguing findings: when the number of end-users per for node increases, the horizontal collaboration is less due to the saturation of computational resources of higher CPU fog nodes. Therefore, although the transmission delay between the fog node and the cloud is higher with lower transmission rate, the fog node cannot able to process the most of the tasks, thus, fog nodes offload their tasks to the cloud server.

## V. Conclusion

In this paper, we study the task offloading in fog computing network that consists of heterogeneous fog computing nodes with different computational capability. We have proposed an offloading policy to find the optimal amount of offloaded task data under delay and energy constraints. It has been observed that the individual computational capability of the fog node will play an important role in local task processing time thereby thereby affecting the total cost for task processing. To solve the non-convex optimization problem, we apply SDR to the optimization problem. From the extensive simulation, we observed that with an equal number of end-users per fog node, the fog node with higher CPU rate receives more data from the neighboring fog node to reduce the computational load of the fog node with low CPU rate. In particular, the offloading profile of the fog node shows an insight of the task offloading under energy and computational time for individual fog node.

## REFERENCES

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, Jan. 2012, pp. 13–16.

[2] M. Mukherjee, L. Shu, and D. Wang, “Survey of fog computing: Fundamental, network applications, and research challenges,” *IEEE Commun. Surv. Tut.*, vol. 20, no. 3, pp. 1826–1857, 3rd quarter 2018.

[3] M. Aazam and E. N. Huh, “Fog Computing: The Cloud-IoT/IoE middleware paradigm,” *IEEE Potentials*, vol. 35, no. 3, pp. 40–44, May 2016.

[4] 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. 2018.

[5] J. Liu and Q. Zhang, “Offloading schemes in mobile edge computing for ultra-reliable low latency communications,” *IEEE Access*, vol. 6, pp. 12 825–12 837, Feb. 2018.

[6] J. Liu and Q. Zhang, “Code-partitioning offloading schemes in mobile edge computing for augmented reality,” *IEEE Access*, vol. 7, pp. 11 222–11 236, 2019.

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

[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*, June 2018, pp. 422–429.

[9] S. Yu, R. Langar, X. Fu, L. Wang, and Z. Han, “Computation offloading with data caching enhancement for mobile edge computing,” *IEEE Trans. on Vehi. Technol.*, vol. 67, no. 11, pp. 11 098–11 112, Nov 2018.

[10] Z. Liu, X. Yang, Y. Yang, K. Wang, and G. Mao, “DATS: Dispersive stable task scheduling in heterogeneous fog networks,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3423–3436, Apr. 2019.

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