This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2018.2888481, IEEE Transactions on Network and Service Management

QoS-Aware Fog Resource Provisioning and Mobile Device Power Control in IoT Networks

Jingjing Yao, *Student Member, IEEE*, and Nirwan Ansari, *Fellow, IEEE*

***Abstract*—Fog-aided Internet of Things (IoT) addresses the resource limitations of IoT devices in terms of computing and energy capacities, and enables computational intensive and delay-sensitive tasks to be offloaded to the fog nodes attached to the IoT gateways. A fog node, utilizing the cloud technologies, can lease and release virtual machines (VMs) in an on-demand fashion. For the power-limited mobile IoT devices (e.g., wearable devices and smart phones), their quality of service (QoS) may be degraded owing to the varying wireless channel conditions. Power control helps maintain the wireless transmission rate and hence the QoS. The QoS (i.e., task completion time) is affected by both the fog processing and wireless transmission; it is thus important to jointly optimize fog resource provisioning (i.e., decisions on the number of VMs to rent) and power control. Our work addresses this joint optimization problem to minimize the system cost (VM rentals) while guaranteeing QoS requirements, formulated as a mixed integer nonlinear programming (MINLP) problem. An approximation algorithm is then proposed to solve the problem. Simulation results demonstrate the performance of our proposed algorithm.**

***Index Terms*—Internet of Things, Smart Devices, Fog Resource Provisioning, Power Control, Mobility Management.**

I. Introduction

Internet of Things (IoT) connects billions of physical objects, including sensors, smart meters, smart cars and actuators, to collect and exchange data to facilitate various applications, such as smart city, smart grid, e-healthcare and home automation [1], [2]. The fundamental application of IoT is the sensing service which deploys several sensors to detect the ambient environment and send the sensed data (e.g., temperature and humidity) to the users for monitoring purposes [3]. The sensed data are usually collected by the IoT gateway and then transmitted to the users. According to Cisco, the IoT devices connected to the Internet are predicted to grow to 50 billions by 2020 and will reach 500 billions by 2025 [4].

Many IoT devices have severely limited resources (e.g., computating and energy capacity), and hence many computing tasks are offloaded to the cloud data center for intensive processing [5], [6]. However, many IoT applications require real-time processing and event response (e.g., disaster response and virtual reality applications). Hence, data processing by the remote cloud may not satisfy the strict latency requirement. Fog computing is a promising paradigm to improve the performance of IoT services by performing data analytics and time-sensitive tasks close to IoT devices [7]. In fog computing, fog nodes, equipped with computing, storage and networking resources, are attached to an IoT gateway (GW) to assume a substantial amount of computing tasks instead of performing all tasks in the remote cloud, thus enabling immediate service response [8].

The fundamental issue of fog computing in IoT is resource provisioning, which addresses how to allocate computing resources and how to process service requests over a set of computing resources [9], [10]. These computing resources usually include virtual machines (VMs), which are scheduled to process different tasks [11]. Fog computing adopts the virtualization and cloud technologies, and hence can lease and release computing resources in an on-demand utility-like fashion [12]. As renting VMs incurs rentals (i.e., system cost), the service provider should determine how many VMs to rent to minimize its system cost (i.e., fog resource provisioning problem) [13], and at the same time ensure that the service completion time does not surpass tasks' expected deadlines.

An ever-increasing number of services are provisioned for mobile IoT devices (e.g., wearable devices and smart phones), including cognitive assistance, face recognition, object recognition, etc [14], [15]. Owing to the resource-limited IoT devices, the computing tasks are usually offloaded to fog nodes for processing [16]. Hence, the communication between a device and its corresponding fog node should be efficient. On the other hand, if the device moves far away from the IoT GW, the quality of service (QoS) could be degraded owing to the long transmission latency [17]. Therefore, how to maintain QoS is critical, and this involves the issue of mobility management, which tries to prevent mobile terminals from experiencing service quality degradation when they move to another place [18]. Mobility management of mobile terminals in fog-aided networks involves several options. The first one is to properly set the transmission power of the device (i.e., power control problem). The second one is to perform handover between different IoT GWs [19] and to migrate virtual machines (VMs) to comply with the requirements of QoS [20]. We focus on the power control of a device in our work and assume the device always moves within the coverage of this GW.

For the deadline-driven real-time mobile IoT applications (e.g., cognitive assistance and object recognition), computing tasks are usually offloaded to the fog node for processing, and the processed results are sent back to the user. Hence, the QoS can be affected by both wireless transmission and fog processing. For example, provisioning more VMs in the fog node increases the computing capacity and reduces the processing time, and thus leads to the better QoS [21]; provisioning more transmission power increases the wireless transmission rate and reduces data transmission time, and hence improves the QoS. Although fog resource provisioning and power control have been studied separately to meet QoS requirements, the joint consideration of both has not been readily reported. In our work, we jointly optimize the fog

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2018.2888481, IEEE Transactions on Network and Service Management

resource provisioning and power control problem to minimize the system cost under the constraint of QoS requirements of mobile IoT applications. We formulate our problem as a mixed integer nonlinear programming (MINLP) problem, and then design an approximation algorithm to solve it. We first transform the MINLP problem into a convex optimization problem by relaxing its integer variables, and then design an integer recovery scheme to obtain the feasible solution.

The rest of this paper is organized as follows. Section II provides a summary of related works. The system model is described in Section III. Section IV discusses the problem formulation. In Section V, we propose an approximation algorithm to solve this problem. Simulation results are analyzed in Section VI. Section VII concludes the paper.

II. Related Work

Several mobile IoT applications have been proposed. Satyanarayanan *et al.* [22] proposed a mobile eye wearable device to provide deep cognitive assistance which offers hints for social interaction by real-time scene analysis. Ha *et al.* [15] proposed a wearable cognitive assistance system to help people with reduced mental acuity. In order to provide real-time responses, computing tasks are offloaded to the edge devices in their system.

Owing to the limited computing capacity of IoT devices, the computationally demanding real-time applications are usually offloaded to fog nodes, which are attached to the IoT GW and have large computing resources. In order to improve the fog node performance, resources should be well provisioned. Chiang and Zhang [23] presented the opportunities and challenges of fog-aided IoT, where fog nodes enable computation closer to the sensors and data analytics at the network edge to support time-sensitive applications. Ansari and Sun [24] proposed a mobile edge internet of things (MEIoT) architecture, which was demonstrated to speed up data sharing and analytics. Skarlat *et al.* [25] proposed a fog resource provisioning approach to distribute task requests and data among fog nodes. Gu *et al.* [26] investigated the joint radio and computational resource allocation problem in fog computing to assign suitable fog nodes and radio spectrum to satisfy QoS requirements. However, the above works consider neither the resource provisioning within the fog nodes nor the system cost of the fog nodes.

Adjusting the transmission power helps the mobile IoT devices maintain the wireless transmission rate and hence maintain the QoS when they move. Kamoun *et al.* [27] studied the power control and computation offloading decision problem, which determines which tasks should be offloaded to the cloud attached to the BS, with the objective to minimize the energy consumption of the mobile terminals while satisfying predefined execution delay. Mach and Becvar [28] proposed a cloud-aware power control algorithm to adjust the transmission power and offload the computationally demanding real-time applications to the mobile edge computing enabled BS, so that the ratio of delivered computing tasks is maximized. However, none of the above works consider the fog provisioning problem within the edge cloud (i.e., fog node).

Li *et al.* [29] proposed the edge computing IoT architecture (ECIoT) which jointly considered the admission control and resource allocation to offload tasks to different fog nodes and to allocate power to IoT devices. They utilized a simple idea to allocate more power to IoT devices with bad channel conditions and offload tasks to fog nodes with higher computing resources. However, their work did not consider the fog provisioning problem which determines how many VMs should be rented to minimize the system cost. They also did not provide any insights on how much power should be adopted to guarantee the QoS requirement. In our work, we address the joint optimization of fog provisioning and mobile device power control while meeting QoS requirements.

III. System Model

A. System Description

We consider a fog-aided IoT architecture, as shown in Fig. 1. A fog node with computing resources is attached to the IoT GW which is located at Location 0 (denoted as L_0). We investigate a representative user with IoT devices who moves within the coverage of the GW. Within a certain time period (e.g., an hour or a day), the user initiates a deadline-specific applications at N different locations, denoted as $L_i,\ i \in \mathcal{N} = \{1, ..., N\}$. At each location i, the application involves several tasks to be sent and further processed in the fog node. There are x_i VMs initiated in the fog node to process the tasks of the application at location i. The system cost is incurred by renting the VMs at each location. We denote the cost of each VM as C (we only consider homogenous VMs in this work), and we assume the VMs are released after all tasks are processed at each location. Therefore, the total system cost can be calculated as $C\sum_{i=1}^{N} x_i$. Note that this is the popular commercial cloud service model, e.g., Amazon Elastic Compute Cloud (EC2) [30]. To take the face recognition application as an example, each task can be a picture or recorded video of other people. At a location, the tasks containing pictures and videos are transmitted to be analyzed in the fog node, which then sends the results via downlink to the user. In our work, we assume the application completion time is relatively short as compared with the time during which the user transits from one location to another, and hence the wireless channel condition does not change in processing all tasks of the application.

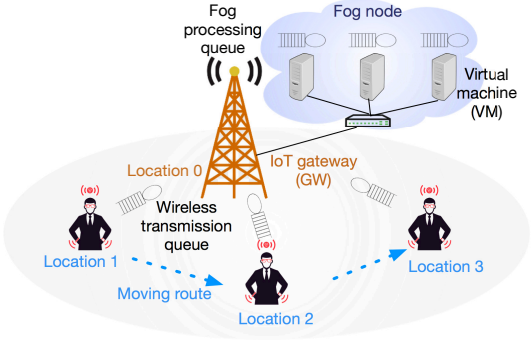


Fig. 1. Fog-aided IoT architecture.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2018.2888481, IEEE Transactions on Network and Service Management

B. QoS Model

To characterize each computing task, we adopt the widely used three-parameter model [31]. A task at location i is defined as a tuple $< l_i, v_i, D_i >,\ i \in \mathcal{N}$, where l_i denotes the input data size in bits, v_i the computation intensity in CPU cycles per bit, and D_i the completion deadline in seconds. Since all tasks from the same location come from the same application, we assume they have the same completion deadline and computation intensity. We also assume they are independent and identically distributed. Note that the received result from the fog node after processing is relatively small in size [26] (e.g., names in the face recognition application) and the downlink capacity is usually large, and so we neglect the downlink transmission delay in our work. Hence, the total delay of each task can be modeled as the summation of the uplink wireless transmission delay and fog processing delay. We model the wireless transmission delay and fog processing delay as a two-layer queueing system, which behaves in a first in first out (FIFO) manner.

1) Wireless Transmission Queue: The task arrivals at each location i are assumed to follow a Poisson process with the arrival rate λ_i and the input data size of each task follows the exponential distribution with average value l_i. The wireless transmission rate r_i at location i can be calculated as

$$r_i = Wlog_2(1 + \frac{p_i H_i}{N_0 W}),\ \forall i \in \mathcal{N}, \tag{1}$$

where W denotes the wireless bandwidth, H_i indicates the wireless channel gain between the device and the GW, N_0 is the noise power spectrum density, and p_i is the wireless transmission power. Since l_i is exponentially distributed, the service time (i.e., $\frac{l_i}{r_i}$) follows an exponential distribution at each location i. Therefore, the wireless transmission queue realizes an M/M/1 queue [32]. Note that the M/M/1 queuing model with the wireless channel capacity as the service rate has been widely used to characterize wireless communications systems [33]. The traffic load in the wireless transmission queue at location i can be calculated as

$$\rho_i = \frac{\lambda_i l_i}{r_i},\ \forall i \in \mathcal{N}. \tag{2}$$

Then, the wireless transmission delay is

$$t_i^w = \frac{l_i}{r_i(1-\rho_i)} = \frac{l_i}{r_i - \lambda_i l_i},\ \forall i \in \mathcal{N}. \tag{3}$$

2) Fog Processing Queue: The fog node switch can schedule tasks to different VMs. The probability that a task is assigned to VM j at location i is denoted as P_{ij}. In order to balance the workload, we assume the tasks are distributed evenly to all VMs (i.e., $P_{ij} = \frac{1}{x_i}, \forall i \in \mathcal{N}, j \in \{1, ..., x_i\}$), where x_i denotes the number of rented VMs at location i. Although different task assignment strategies can be applied to assign tasks to different VMs, we only consider evenly distribution strategy (i.e., tasks are evenly assigned to VMs) in our work for simplicity. According to Burke's Theorem, the traffic departure process of an M/M/1 queue is still a Poisson process with the same departure rate as the arrival rate [32]. Hence, the task arrivals in the fog processing queue still follow a Poisson process with the traffic arrival rate $\lambda_i P_{ij} = \frac{\lambda_i}{x_i}$. The service time can be calculated as $\frac{l_i v_i}{u}$ where u is the computing capacity of the VM in CPU cycles per second, and hence follows an exponential distribution (l_i follows an exponential distribution; v_i and u are constants). Therefore, the fog processing queue realizes an M/M/1 queue [32]. Note that the M/M/1 model has been widely adopted in resource provisioning problem of cloud/fog computing for performance analysis [34]. The traffic load in each fog processing queue is

$$\tilde{\rho}_i = \frac{\lambda_i l_i v_i}{x_i u},\ \forall i \in \mathcal{N}. \tag{4}$$

Then, the fog processing delay is

$$t_i^c = \frac{l_i v_i}{u(1-\tilde{\rho}_i)} = \frac{l_i v_i}{u - \frac{\lambda_i l_i v_i}{x_i}},\ \forall i \in \mathcal{N}. \tag{5}$$

The QoS requirement stipulates that the total delay should not surpass the computation deadline of each task, i.e.,

$$t_i^c + t_i^w \leq D_i,\ \forall i \in \mathcal{N}. \tag{6}$$

Although we only consider one presentative user in our system model, our system model can be extended to multiple users by summing the task arrival rates λ. We will address this multi-user model in our future work. Note that the network status is dynamic, and hence we re-operate our proposed model and algorithm periodically (e.g., one hour).

IV. Problem Formulation

In our work, we jointly consider the QoS-aware fog resource provisioning and power control problem to minimize the system cost incurred by renting VMs. Specifically, we provide insights on how to determine the number of VMs to rent and transmission power of the IoT device at each location. Our problem is formulated as a MINLP problem:

$$\textbf{P0:}\quad \min_{\boldsymbol{x},\boldsymbol{p}}\quad C\sum_{i=1}^{N} x_i \tag{7}$$

$$s.t.\quad \frac{l_i}{r_i - \lambda_i l_i} + \frac{l_i v_i}{u - \frac{\lambda_i l_i v_i}{x_i}} \leq D_i,\ \forall i \in \mathcal{N}, \tag{8}$$

$$r_i = Wlog_2(1 + \frac{p_i H_i}{N_0 W}),\ \forall i \in \mathcal{N}, \tag{9}$$

$$\sum_{i=1}^{N} p_i \leq P^T, \tag{10}$$

$$p_i \leq P^m,\ \forall i \in \mathcal{N}, \tag{11}$$

$$r_i > \lambda_i l_i,\ \forall i \in \mathcal{N}, \tag{12}$$

$$u > \frac{\lambda_i l_i v_i}{x_i},\ \forall i \in \mathcal{N}. \tag{13}$$

$$x_i \in \mathbb{Z}^+,\ \forall i \in \mathcal{N}. \tag{14}$$

Eq. (7) states that the objective is to minimize the system cost. Eq. (8) indicates the QoS requirement. Owing to the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2018.2888481, IEEE Transactions on Network and Service Management

device battery limitation, the total power consumption should not exceed the power budget P^T of the IoT device, which is shown in Eq. (10). The maximum transmission power is defined as P^m in Eq. (11). Eqs. (12) and (13) are the constraints of stability of the wireless transmission queue and fog processing queue, respectively.

Lemma 1. *We can find an optimal solution $(\boldsymbol{x}^*, \boldsymbol{p}^*)$ so that Eq. (8) is active (equality holds), i.e.,*

$$\frac{l_i}{r_i^* - \lambda_i l_i} + \frac{l_i v_i}{u - \frac{\lambda_i l_i v_i}{x_i^*}} = D_i,\ \forall i \in \mathcal{N}, \tag{15}$$

$$r_i^* = W log_2(1 + \frac{p_i^* H_i}{N_0 W}),\ \forall i \in \mathcal{N}. \tag{16}$$

Proof: If there exists $\frac{l_i}{r_i^* - \lambda_i l_i} + \frac{l_i v_i}{u - \frac{\lambda_i l_i v_i}{x_i^*}} < D_i$ for any i, we can always decrease p_i^* to $\tilde{p}_i^*$ so that the equality $\frac{l_i}{\tilde{r}_i^* - \lambda_i l_i} + \frac{l_i v_i}{u - \frac{\lambda_i l_i v_i}{x_i^*}} = D_i$ holds. The updated solution is still feasible and can be the optimal solution, which completes the proof. ∎

Lemma 1 implies the interaction between the wireless transmission and fog processing. When the transmission power increases, the wireless transmission delay decreases, and hence the fog processing delay increases, thus resulting in the reduction of the number of VMs and the system cost. On the contrary, if the transmission power decreases, more VMs are required to satisfy the QoS requirement and hence the system cost increases. Therefore, we can transform Eq. (15) into

$$\begin{aligned} r_i &= \lambda_i l_i + \frac{l_i(u x_i - \lambda_i l_i v_i)}{(D_i u - l_i v_i) x_i - D_i \lambda_i l_i v_i} \\ &= \lambda_i l_i + \frac{l_i u}{m_i} + (\frac{u D_i}{m_i} - 1) \frac{\lambda_i l_i^2 v_i}{m_i x_i - D_i \lambda_i l_i v_i} \\ &= \lambda_i l_i + \frac{l_i u}{m_i} + \frac{\lambda_i l_i^3 v_i^2}{m_i (m_i x_i - D_i \lambda_i l_i v_i)} \\ &\triangleq f_i(x_i),\ \forall i \in \mathcal{N}, \end{aligned} \tag{17}$$

where we denote $m_i = D_i u - l_i v_i$ for simplicity.

Lemma 2. *If $x_i \in \mathbb{R}^+$, $f_i(x_i)$ is a monotonically decreasing and convex function.*

Proof: We obtain the first and second order derivative of $f_i(x_i)$ in terms of x_i as

$$f_i^{'}(x_i) = -\frac{\lambda_i l_i^3 v_i^2}{(m_i x_i - D_i \lambda_i l_i v_i)^2}, \tag{18}$$

$$f_i^{''}(x_i) = \frac{2 \lambda_i l_i^3 v_i^2 m_i}{(m_i x_i - D_i \lambda_i l_i v_i)^3}. \tag{19}$$

From Lemma 1, we know that $\frac{l_i v_i}{u - \frac{\lambda_i l_i v_i}{x_i}} < D_i$, which proves that $l_i v_i < D_i(u - \frac{\lambda_i l_i v_i}{x_i}) < D_i u$. Hence,

$$m_i = D_i u - l_i v_i > 0. \tag{20}$$

Note that in Eq. (17), $\frac{l_i(u x_i - \lambda_i l_i v_i)}{m_i x_i - D_i \lambda_i l_i v_i} > 0$ because $r_i > \lambda_i l_i$. Since $u x_i - \lambda_i l_i v_i > 0$ (i.e., Eq. (13)), we then have

$$m_i x_i - D_i \lambda_i l_i v_i > 0. \tag{21}$$

From Eqs. (18)-(21), we can deduce that $f_i^{'}(x_i) < 0, f_i^{''}(x_i) > 0$, which completes the proof. ∎

Lemma 2 indicates that the minimum number of VMs should be guaranteed to ensure the transmission power does not surpass the power threshold P^m. Combining Eqs. (17), (12) and (13) yields

$$\begin{aligned} x_i &= \frac{D_i \lambda_i l_i v_i}{m_i} + \frac{\lambda_i l_i^3 v_i^2}{(r_i - \lambda_i l_i) m_i^2 - u l_i m_i} \\ &> \frac{D_i \lambda_i l_i v_i}{m_i} + \frac{\lambda_i l_i^3 v_i^2}{(W log_2(1 + \frac{P^m H_i}{N_0 W}) - \lambda_i l_i) m_i^2 - u l_i m_i} \\ &\triangleq \tilde{n}_i,\ \forall i \in \mathcal{N}. \end{aligned} \tag{22}$$

Eq. (21) indicates that

$$x_i > \frac{D_i \lambda_i l_i v_i}{m_i},\ \forall i \in \mathcal{N}. \tag{23}$$

Combining Eqs. (13), (22) and (23) yields

$$x_i > \max\{\tilde{n}_i, \frac{D_i \lambda_i l_i v_i}{m_i}, \frac{\lambda_i l_i v_i}{u}\} \triangleq n_i,\ \forall i \in \mathcal{N}, \tag{24}$$

which is the lower bound of x_i. Similarly, by combining Eqs. (17), (20) and (21), we have $r_i > \lambda_i l_i + \frac{l_i u}{m_i}$, which implies the lower bound of p_i, i.e.,

$$p_i > \frac{N_0 W}{H_i}(2^{\frac{\lambda_i l_i + \frac{l_i u}{m_i}}{W}} - 1) \triangleq \tilde{p}_i,\ \forall i \in \mathcal{N}. \tag{25}$$

Therefore, Problem **P0** can then be transformed into

$$\textbf{P1:}\quad \min_{\boldsymbol{x}, \boldsymbol{p}} \quad C \sum_{i=1}^{N} x_i$$

$$s.t. \quad (10), (14)$$

$$f_i(x_i) - W log_2(1 + \frac{p_i H_i}{N_0 W}) = 0,\ \forall i \in \mathcal{N}, \tag{26}$$

$$x_i \geq \lceil n_i \rceil,\ \forall i \in \mathcal{N}, \tag{27}$$

$$\tilde{p}_i < p_i \leq P^m,\ \forall i \in \mathcal{N}, \tag{28}$$

where $f_i(x_i)$ is defined in Eq. (17); Eq. (27) shows the lower bound of x_i and $\lceil n_i \rceil$ denotes the ceiling of n_i (defined in Eq. (24)); Eq. (28) indicates the lower and upper bound of p_i. Note that problem (**P1**) is still intractable because of the non-linear constraints (i.e., Eq. (26)) and the integer variables $\boldsymbol{x}$. To solve this problem, we propose an approximation algorithm based on relaxation in the next section.

V. Approximation Algorithm

In this section, we propose a QoS-aware Fog Resource provisioning and Power Allocation (FRPA) algorithm to solve problem **P1**. Specifically, FRPA first relaxes the integer variables $\boldsymbol{x}$ to continuous rational numbers so that problem **P1** can be transformed into a convex optimization problem, which can then be solved by the gradient projection algorithm (GPA) [35]. Then, we design an integer recovery scheme to obtain the feasible solution.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2018.2888481, IEEE Transactions on Network and Service Management

A. Gradient Projection Algorithm (GPA)

GPA is used to solve bound constrained optimization problem by taking steps in the direction of the gradient of the objective function and projecting the solutions onto the feasible set of constraints [35]. Our proposed algorithm first relaxes the integer variables x_i in problem **P1** and then transforms **P1** into a convex problem **P2**. The convex problem guarantees that the original problem is optimal when its Lagrangian dual problem is optimal [36].

We first relax $\boldsymbol{x}$ and substitute Eq. (26) into Eq. (10); problem **P1** can then be transformed into

$$\textbf{P2:} \quad \min_{\boldsymbol{x}} \quad C\sum_{i=1}^{N} x_i$$

$$s.t. \quad (27),$$

$$\sum_{i=1}^{N} \frac{N_0 W}{H_i}(2^{\frac{f_i(x_i)}{W}} - 1) \leq P^T, \tag{29}$$

$$x_i \in \mathbb{R}^+, \ \forall i \in \mathcal{N}, \tag{30}$$

Lemma 3. *Problem **P2** is a convex optimization problem.*

Proof: Define $h_i(x_i) = 2^{\frac{f_i(x_i)}{W}} - 1$, and the first derivative of $h_i(x_i)$ is $h_i^{'}(x_i) = 2^{\frac{f_i(x_i)}{W}}(\ln 2)\frac{f_i^{'}(x_i)}{W}$. The second derivative is $h_i^{''}(x_i) = \frac{\ln 2}{W}2^{\frac{f_i(x_i)}{W}} f_i^{''}(x_i) + (\frac{\ln 2}{W} f_i^{'}(x_i))^2 2^{\frac{f_i(x_i)}{W}} > 0$, which demonstrate that $h_i(x_i)$ is convex function. $\sum_{i=1}^{N} \frac{N_0 W}{H_i}(2^{\frac{f_i(x_i)}{W}} - 1)$ is a combination of positive weighted convex functions and hence is a convex function. Therefore, problem **P2** is a convex optimization problem. ■

Note that it is easier to solve problem **P2** than problem **P1** because it only contains the variable $\boldsymbol{x}$. The Lagrangian function of problem **P2** can be expressed as

$$\begin{aligned} L(\boldsymbol{x}, b) &= C\sum_{i=1}^{N} x_i + b[\sum_{i=1}^{N} \frac{N_0 W}{H_i}(2^{\frac{f_i(x_i)}{W}} - 1) - P^T] \\ &= \sum_{i=1}^{N}(Cx_i + \frac{bN_0 W}{H_i}2^{\frac{f_i(x_i)}{W}}) - b(\sum_{i=1}^{N} \frac{N_0 W}{H_i} + P^T) \end{aligned} \tag{31}$$

where $b \geq 0$ is the Lagrangian multiplier associated with Eq. (29). Then, the Lagrangian dual function is

$$\begin{aligned} g(b) &= \inf_{x_i \geq \lceil n_i \rceil} L(\boldsymbol{x}, b) \\ &= \sum_{i=1}^{N} \inf_{x_i \geq \lceil n_i \rceil}(Cx_i + \frac{bN_0 W}{H_i}2^{\frac{f_i(x_i)}{W}}) - b(\sum_{i=1}^{N} \frac{N_0 W}{H_i} + P^T) \end{aligned} \tag{32}$$

To obtain $g(b)$, we can decompose $g(b)$ into N independent subproblems as follows,

$$\textbf{P3:} \quad \min_{x_i} \quad \xi(x_i) = Cx_i + \frac{bN_0 W}{H_i}2^{\frac{f_i(x_i)}{W}}$$

$$s.t. \quad (27).$$

Note that if $b = 0$, $\xi(x_i)$ is an increasing function because $f_i(x_i)$ is a decreasing function based on Lemma 2. In this case, $x_i = \lceil n_i \rceil$. On the contrary, if $b > 0$, the solution of problem **P3** is $\max\{\lceil n_i \rceil, a\}$, where a satisfies $\xi^{'}(a) = 0$. The first derivative of $\xi(x_i)$ can be expressed as

$$\xi^{'}(x_i) = C + \frac{(\ln 2) bN_0 W}{H_i W}2^{\frac{f_i(x_i)}{W}} f_i^{'}(x_i) = 0 \mid x_i = a, \tag{33}$$

which leads to

$$\begin{aligned} \varphi(x_i) \triangleq & f_i(x_i) - 2W\log_2(m_i x_i - D_i\lambda_i l_i v_i) \\ & - W\log_2(\frac{CH_i}{(\ln 2) bN_0\lambda_i l_i^3 v_i^2}) = 0 \mid x_i = a. \end{aligned} \tag{34}$$

Note that $\varphi(x_i)$ is a decreasing function because $f_i(x_i)$ is a decreasing function. In order to obtain the value of a which makes $\varphi(x_i) = 0$, we utilize the binary search algorithm [37], which searches for a within the interval $[L, R]$. It compares the value $\varphi(M)$, where $M = (L+R)/2$ is the middle element of the interval, with 0. If $\varphi(M)$ equals 0, we have $a = M$. If $\varphi(M) < 0$, the search continues in the upper half of the interval, i.e., $[L, M]$. Otherwise, the lower half of the interval $[M, R]$ should be searched. The binary search algorithm is described in Alg. 1. Lines 1-8 initialize the search interval. Note that we can only search within the interval $[\lceil n_i \rceil, +\infty)$ and hence if a satisfying $\varphi(a) = 0$ falls out of the interval, we assign $a = \lceil n_i \rceil$, as shown in line 3. Lines 9-18 iteratively halves the search interval to find the solution.

Algorithm 1: Binary Search Algorithm

Input : $C, N, W, H_i, N_0, D_i, P^T, P^m, \lambda_i, u_i, v_i$
Output: Value a satisfying $\varphi(a) = 0$

```
1  Initialize L = ⌈n_i⌉ ;
2  if φ(L) <= 0 then
3  |  return a = ⌈n_i⌉ ;
4  end
5  Initialize R = L + 1 ;
6  while φ(R) > 0 do
7  |  R = R + 1 ;
8  end
9  do
10 |  Calculate M = (L+R)/2 ;
11 |  if φ(M) = 0 then
12 |  |  break ;
13 |  else if φ(M) < 0 then
14 |  |  M = R ;
15 |  else
16 |  |  M = L ;
17 |  end
18 while L < R;
19 return a = M;
```

Then, the solution of problem **P3** can be expressed as

$$x_i^* = \begin{cases} \lceil x_i \rceil, \ \text{if } b = 0, \\ \max\{\lceil n_i \rceil, a\}, \ \text{otherwise}, \end{cases} \tag{35}$$

where a is obtained by Alg. 1 and satisfies $\xi^{'}(a) = 0$ (i.e., $\varphi(a) = 0$).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2018.2888481, IEEE Transactions on Network and Service Management

Problem **P2** can then be solved by GPA which updates b based on the gradient search algorithm, i.e.,

$$b(t) = \max\{b(t-1) + \alpha[\sum_{i=1}^{N} \frac{N_0 W}{H_i}(2^{\frac{f_i(x_i^*(t))}{W}} - 1) - P^T], 0\}, \tag{36}$$

where α is the step size; t denotes the iteration number; $x_i^*(t)$ can be obtained from Eq. (35) by substituting the Lagrangian multiplier from the previous iteration, i.e., $b(t-1)$; $\sum_{i=1}^{N} \frac{N_0 W}{H_i}(2^{\frac{f_i(x_i^*(t))}{W}} - 1) - P^T$ is the gradient of $g(b)$ based on Eqs. (31) and (32). The GPA iteratively updates $\boldsymbol{x^*}$ and b until $g(b)$ does not change or the maximum number T of iteration is reached.

B. Integer Recovery Algorithm (IRA)

Since we have relaxed problem **P1** into a convex optimization problem **P2**, the solution obtained by the gradient projection algorithm may not be feasible if any x_i^* is not an integer. In order to address this issue, we design an integer recovery algorithm to get the feasible solution. The basic idea of IRA is to choose between the floor and ceiling of x_i^* (i.e., $\lfloor x_i^* \rfloor$ and $\lceil x_i^* \rceil$) and obtain the feasible solution of x_i and p_i.

Lemma 4. *If x_i^* is the solution of problem **P2**, both $\lfloor x_i^* \rfloor$ and $\lceil x_i^* \rceil$ are still greater than $\lceil n_i \rceil$.*

Proof: Since x_i^* is the solution of problem **P2**, we have $x_i \geq \lceil n_i \rceil$. If x_i^* is an integer, $\lfloor x_i^* \rfloor = x_i^* = \lceil x_i^* \rceil$ and hence $\lfloor x_i^* \rfloor = \lceil x_i^* \rceil \geq \lceil n_i \rceil$. Contrarily, if x_i^* is not an integer, $x_i^* > \lceil n_i \rceil$ because $\lceil n_i \rceil$ is an integer. Then, we can deduce that $\lceil x_i^* \rceil > x_i^* > \lfloor x_i^* \rfloor \geq \lceil n_i \rceil$. Hence, the lemma is proved. ■

The IRA aims to solve problem **P1** by substituting x_i with either $\lfloor x_i^* \rfloor$ or $\lceil x_i^* \rceil$, where x_i^* is obtained from the relaxed problem **P2**. From Eq. (26), we have $p_i = \frac{N_0 W}{H_i}(2^{\frac{f_i(x_i)}{W}} - 1)$. Hence, the value of p_i is from $\{p_i^l, p_i^r\} \triangleq \{\frac{N_0 W}{H_i}(2^{\frac{f_i(\lfloor x_i^* \rfloor)}{W}} - 1), \frac{N_0 W}{H_i}(2^{\frac{f_i(\lceil x_i^* \rceil)}{W}} - 1)\}$ ($p_i^l > p_i^r$). From Eq. (22), we have $x_i = \frac{D_i \lambda_i l_i v_i}{m_i} + \frac{\lambda_i l_i^3 v_i^2}{(W log_2(1+\frac{P_i H_i}{N_0 W}) - \lambda_i l_i) m_i^2 - u l_i m_i} \triangleq \phi_i(p_i)$ with regard to p_i. Then, problem **P1** can be transformed into

$$\textbf{P4:} \quad \min_{\boldsymbol{p}} \quad C \sum_{i=1}^{N} \phi_i(p_i)$$

$$s.t. \quad (10),$$

$$p_i \in \{p_i^l, p_i^r\}, \ \forall i \in \mathcal{N}.$$

Note that Eq. (27) is eliminated because of Lemma 4. We denote a Boolean variable y_i to indicate whether $\{\lfloor x_i^* \rfloor, p_i^l\}$ is chosen (i.e., $y_i = 1$), or $\{\lceil x_i^* \rceil, p_i^r\}$ (i.e., $y_i = 0$). Eq. (10) is then transformed into $\sum_{i=1}^{N} [y_i p_i^l + (1 - y_i) p_i^r] \leq P^T$. Similarly, the objective function of problem **P4** becomes $C \sum_{i=1}^{N} [y_i \phi_i(p_i^l) + (1 - y_i)\phi_i(p_i^r)]$. Therefore, problem **P4** is transformed into

$$\begin{aligned} \textbf{P5:} \quad \min_{\boldsymbol{y}} \quad & C \sum_{i=1}^{N} (\phi_i(p_i^l) - \phi_i(p_i^r)) y_i + C \sum_{i=1}^{N} \phi_i(p_i^r) \\ = \quad & - C \sum_{i=1}^{N} (\lceil x_i^* \rceil - \lfloor x_i^* \rfloor) y_i + C \sum_{i=1}^{N} \lceil x_i^* \rceil \\ s.t. \quad & \sum_{i=1}^{N} (p_i^l - p_i^r) y_i \leq P^T - \sum_{i=1}^{N} p_i^r, \\ & y_i \in \{0, 1\}, \ \forall i \in \mathcal{N}. \end{aligned}$$

Algorithm 2: Fog Resource Provisioning and Power Allocation Algorithm (FRPA)

Input : $C, N, W, H_i, N_0, D_i, P^T, P^m, \lambda_i, u_i, v_i, T$
Output: Number of rented VMs $\boldsymbol{x}$; Transmission power $\boldsymbol{p}$

1 Initialize the Lagrangian multiplier $b(0)$;
2 Initialize $t = 0, x_i^*(0) = +\infty$;
3 **do**
4 t = t + 1 ;
5 **for** *each* i **do**
6 Calculate $x_i^*(t)$ according to Eq. (35) by substituting $b(t-1)$;
7 **end**
8 Calculate $b(t)$ according to Eq. (36) ;
9 **while** $g(b(t)) \not\approx g(b(t-1))$ *and* $t < T$;
10 Calculate $y_i, \ i \in \mathcal{N}$ by solving the 0-1 knapsack problem **P5** with $\boldsymbol{x^* = x^*(t)}$;
11 Calculate $p_i, x_i, \forall i \in \mathcal{N}$ according to Eqs. (37) and (38), respectively ;
12 **return** $\boldsymbol{x}, \boldsymbol{p}$;

Problem **P5** is equivalent to the 0-1 knapsack problem [38], where N locations are considered as N items, $p_i^l - p_i^r$ is the weight of item i, $P^T - \sum_{i=1}^{N} p_i^r$ is the capacity of the knapsack, and $C(\lceil x_i^* \rceil - \lfloor x_i^* \rfloor)$ is the value of each item. The objective of the 0-1 knapsack problem is to maximize the total value of all items in the knapsack constrained by the knapsack weight capacity. Many algorithms have been proposed to solve the 0-1 knapsack problem (e.g., approximation algorithms and dynamic programming approaches) [38]. Then, the optimal solution of problem **P5** is

$$p_i = \begin{cases} p_i^l, \ \text{if } y_i = 1, \\ p_i^r, \ \text{if } y_i = 0, \end{cases} \quad \forall i \in \mathcal{N}, \tag{37}$$

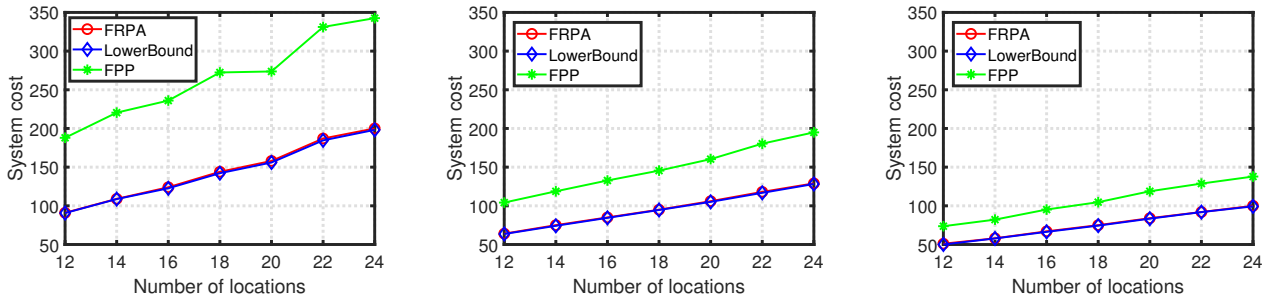
and the optimal solution of x_i is

$$x_i = \begin{cases} \lfloor x_i^* \rfloor, \ \text{if } y_i = 1, \\ \lceil x_i^* \rceil, \ \text{if } y_i = 0, \end{cases} \quad \forall i \in \mathcal{N}, \tag{38}$$

where y_i is the optimal solution of problem **P5**.

We summarize the overall FRPA in Alg. 2. Lines 1-2 initialize all parameters. Lines 3-9 illustrate the process of GPA where we obtain the solution of the relaxed convex optimization problem **P2**. Lines 10-11 delineate the process of IRA

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2018.2888481, IEEE Transactions on Network and Service Management



(a) $D = 130$ *ms* (b) $D = 140$ *ms* (c) $D = 150$ *ms*

Fig. 2. System cost versus number of locations.

where we get the feasible solution for our original problem **P1**. The for-loop in Lines 5-7 is run N times and the iteration in Lines 3-9 may be executed T times in the worst case. The computational complexity in solving the 0-1 knapsack problem in Line 10 is $\mathcal{O}(N(P^T - \sum_{i=1}^{N} p_i^r))$ [39]. Hence, the computational complexity of FRPA is $\mathcal{O}(NT + N^2(P^T - \sum_{i=1}^{N} p_i^r))$.

VI. Simulation Results

We set up simulations to evaluate the performance of our proposed approximation algorithm FRPA which solves the QoS-aware joint problem of fog resource provisioning and power control. We compare FRPA with the lower bound of the problem which is obtained by solving the convex problem after relaxing the integer variable $\boldsymbol{x}$. We also utilize the the existing scheme which only considers Fog Provisioning Problem (FPP) [25] for comparison. Specifically, FPP chooses a fixed transmission power within the interval $(\max_i\{\tilde{p}_i\}, \min\{P^m, \frac{P^T}{N}\}]$ based on Eqs. (10) and (28). We investigate the impacts of different numbers of locations, different arrival rates, and different QoS requirements on the performance of FRPA.

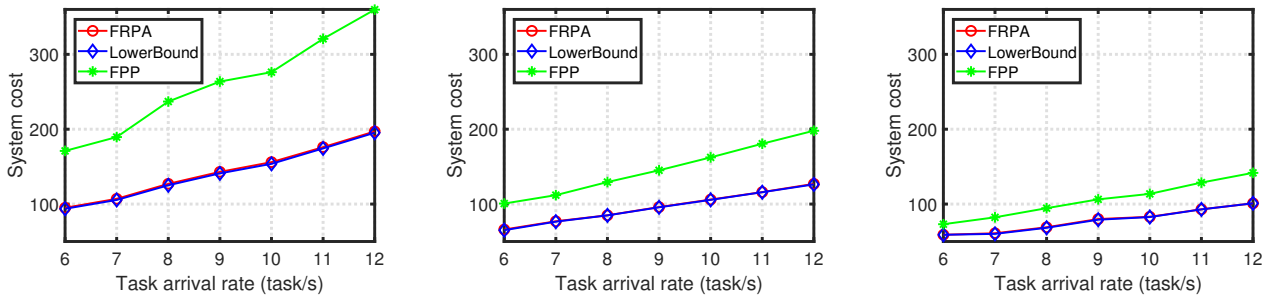
We simulate a 1000m $\times$ 1000m area, where the GW is in the center of this area. The locations where the user requests applications are uniformly distributed in the area. We adopt the path loss model $128.1 + 37.6 \log_{10} d$, as suggested in the 3GPP specification [40], where d refers to the distance in kilometers. The system bandwidth $W = 10$ *MHz* and the noise power density is -174 *dBm/Hz*. The power budget for the total transmission power $P^T = 30$ *W*. We only consider heterogenous applications in our simulation. The average task arrival rate is 10 tasks per second, and the average task size is 1 *Mb* in each location. The computation intensity is 50 cycles per *Mb* and the computing capacity of each VM $u = 5 \times 10^8$ cycles per second, which are consistent with [41]. We normalize the cost of renting a VM as 1, i.e., $C = 1$ per VM. The maximum transmission power budget of the device $P^m = 3$ *W*. Note that the above values are default values if not specified.

We first evaluate the performance of FRPA with different numbers of locations where an application is requested, ranging from 12 to 24 in Fig. 2. We conduct the simulations under three different QoS requirements (i.e., 130 *ms*, 140 *ms* and 150 *ms*), as shown in Fig. 2(a), Fig. 2(b) and Fig. 2(c), respectively. The general trend of the system cost goes up with the increasing number of locations because more locations imply that more applications are requested and hence more VMs are required to process the tasks of the applications (i.e., the increase of the system cost). In Fig. 2, for all QoS requirements, FRPA performs very close to the lower bound. FRPA incurs less system cost as compared with FPP because FPP does not consider the power control problem. For example, if the person moves far from the GW, the wireless transmission rate will degrade because the path loss becomes larger while the transmission power remains the same. In order to satisfy the QoS requirement, more VMs are needed and hence the system cost increases.

Among Fig. 2(a), Fig. 2(b) and Fig. 2(c), all of the system costs for FRPA, LowerBound and FPP decrease if the QoS requirement becomes less strict (i.e., task deadline becomes larger). This is because when the task deadline increases, more processing time can be tolerated, i.e., fewer VMs are required and hence the system cost decreases. We can also observe that the differences between FPP and FRPA become smaller, i.e., the decrease of FPP is much larger than FRPA as the QoS requirement becomes less strict. This is because FPP only considers the fog provisioning problem. For instance, the sum of the wireless transmission delay t_1 and the fog processing delay t_2 equals the task deadline D, i.e., $t_1 + t_2 = D$ and hence their differences $\Delta t_1 + \Delta t_2 = \Delta D$ holds. Consider $\Delta D = 20$ *ms* from Fig. 2(a) to Fig. 2(c). Denote Δt_1^{FRPA} and Δt_2^{FRPA} as the differences of t_1 and t_2 for FRPA, respectively. Similarly, we denote Δt_1^{FPP} and Δt_2^{FPP} as the differences of t_1 and t_2 for FPP, respectively. Since FPP does not consider the power control problem, the wireless transmission rate may degrade because of the varying wireless channel condition. Hence, the wireless transmission delay t_1^{FPP} increases and we assume $\Delta t_1^{FPP} = 10$ *ms*. On the contrary, FRPA utilizes power control to maintain the wireless transmission rate and delay, and so we assume $\Delta t_1^{FRPA} = 0$ *ms*. Therefore, $\Delta t_2^{FPP} = \Delta D - \Delta t_1^{FPP} = 10$ *ms* and $\Delta t_2^{FRPA} = \Delta D - \Delta t_1^{FRPA} = 20$ *ms*, which indicates $\Delta t_2^{FPP} < \Delta t_2^{FRPA}$. Based on Eq. (5), a smaller Δt_2 leads to a larger difference of numbers of VMs Δx. As a result, $\Delta x^{FPP} > \Delta x^{FRPA}$ which explains why FPP and FRPA get closer when the task deadline increases.

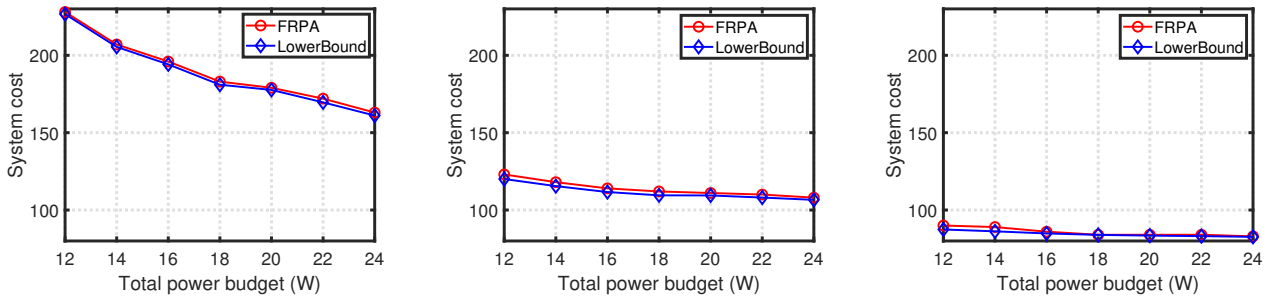
We then compare the performances of FRPA, LowerBound, and FPP with different task arrival rates ranging from 6 to 12 *task/s*. Fig. 3 illustrates the system cost under different task ar-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2018.2888481, IEEE Transactions on Network and Service Management



(a) $D = 130$ *ms* (b) $D = 140$ *ms* (c) $D = 150$ *ms*

Fig. 3. System cost versus task arrival rate.



(a) $D = 130$ *ms* (b) $D = 140$ *ms* (c) $D = 150$ *ms*

Fig. 4. System cost versus total power budget.

rival rates and the comparisons of different QoS requirements are shown in Fig. 3(a), Fig. 3(b) and Fig. 3(c). When the task arrival rate increases, the system cost rises , as shown in Fig. 3(a), Fig. 3(b) and Fig. 3(c), because more VMs are needed to serve the increasing number of tasks. FRPA performs close to the LowerBound and better than the FPP. In comparing Fig. 3(a), Fig. 3(b) and Fig. 3(c), a less strict QoS requirement introduces a smaller system cost and FPP becomes closer to FRPA for the similar reason in Fig. 2.

We also investigate the impacts of different total power budget on the system cost. Fig. 4 depicts the system cost under different total power budget P^T from 12 to 24 *W*. The performances with different QoS requirements are illustrated in Fig. 4(a), Fig. 4(b) and Fig. 4(c). Note that we do not show the results of FPP because when p^T is small, FPP may not be able to find a feasible solution which satisfies $p \in (\max_i\{\tilde{p}_i\}, \min\{P^m, \frac{P^T}{N}\}]$. When P^T is large, FRPA does not change much, in which case we cannot observe the performances of FRPA and hence we ignore that occasion. In Fig. 4, a larger total power budget implies more transmission power can be adopted, and this leads to a higher wireless transmission rate and smaller wireless transmission delay t_1. With certain task deadline D, the fog processing delay $t_2 = D - t_1$ increases and less VMs are required for processing tasks, and that is why the system cost declines when the total power budget gets larger. In addition, FRPA always performs close to LowerBound. Note that FRPA with the stricter QoS requirement (i.e., Fig. 4(a)) exhibits a steeper slope than those in Fig. 4(b) and Fig. 4(c) because when the task deadline is large (i.e., Fig. 4(c)), a relatively low transmission power can satisfy the QoS requirement, and hence increasing the total power budget does not have much impact on the system cost. On the contrary, when the QoS requirement is strict (i.e., Fig. 4(a)), increasing the total power budget helps improve the transmission power and reduce the transmission delay. Therefore, the requirement of fog processing delay is alleviated and hence less VMs are required, and this incurs a smaller system cost.

VII. Conclusion

In this paper, we have investigated joint fog provisioning and power control problem to minimize the system cost incurred by renting VMs while satisfying the QoS requirement. We have modeled our QoS requirement as the sum of delays of two tandem queues, including the wireless transmission queue and the fog processing queue. An MINLP model has been formulated to address this joint optimization problem, which provides insights on how many VMs should be rented and how much transmission power should be budgeted in each location where an application is requested. In order to solve the MINLP, we have proposed an approximation algorithm, FRPA, which first relaxes the integer variable $\boldsymbol{x}$ and transforms the MINLP into a convex problem. Then, the gradient projection algorithm has been designed to obtain the solution of the convex problem. We have further designed the integer recovery algorithm to obtain the feasible solution of the MINLP. Simulation results have demonstrated that our proposed algorithm FRPA performs very close to the lower bound of the relaxed MINLP and much better than the existing work, FPP, which only considers the fog provisioning problem.

REFERENCES

[1] J. Santos, T. Wauters, B. Volckaert, and F. D. Turck, "Resource provisioning for IoT application services in smart cities," in *2017 13th International Conference on Network and Service Management (CNSM)*, Nov. 2017, pp. 1–9.

[2] Y. Hsieh, H. Hong, P. Tsai, Y. Wang, Q. Zhu, M. Y. S. Uddin, N. Venkatasubramanian, and C. Hsu, "Managed edge computing on internet-of-things devices for smart city applications," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, Apr. 2018, pp. 1–2.

[3] J. Yao and N. Ansari, "Joint content placement and storage allocation in C-RANs for IoT sensing service," *IEEE Internet of Things Journal*, DOI: 10.1109/JIOT.2018.2866947, early access.

[4] J. Camhi, "Former Cisco CEO john chambers predicts 500 billion connected devices by 2025," *Business Insider*, 2015.

[5] K. Wu, P. Lu, and Z. Zhu, "Distributed online scheduling and routing of multicast-oriented tasks for profit-driven cloud computing," *IEEE Communications Letters*, vol. 20, no. 4, pp. 684–687, Apr. 2016.

[6] J. Yao, P. Lu, L. Gong, and Z. Zhu, "On fast and coordinated data backup in geo-distributed optical inter-datacenter networks," *Journal of Lightwave Technology*, vol. 33, no. 14, pp. 3005–3015, July 2015.

[7] Z. Zhu, P. Lu, J. J. P. C. Rodrigues, and Y. Wen, "Energy-efficient wideband cable access networks in future smart cities," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 94–100, Jun. 2013.

[8] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*, N. Bessis and C. Dobre, Eds. Cham: Springer International Publishing, 2014, pp. 169–186.

[9] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the internet of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, Dec. 2016.

[10] K. Han, S. Li, S. Tang, H. Huang, S. Zhao, G. Fu, and Z. Zhu, "Application-driven end-to-end slicing: When wireless network virtualization orchestrates with NFV-based mobile edge computing," *IEEE Access*, vol. 6, pp. 26 567–26 577, 2018.

[11] J. Yao and N. Ansari, "QoS-aware joint BBU-RRH mapping and user association in Cloud-RANs," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 4, pp. 881–889, Dec. 2018.

[12] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[13] P. Lu, Q. Sun, K. Wu, and Z. Zhu, "Distributed online hybrid cloud management for profit-driven multimedia cloud computing," *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1297–1308, Aug. 2015.

[14] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 790–808, Nov. 2012.

[15] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proceedings of 12th annual international conference on Mobile systems, applications, and services*, 2014, pp. 68–81.

[16] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing based IoT," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2146–2153, Jun. 2018.

[17] J. Yao and N. Ansari, "Caching in energy harvesting aided internet of things: A game-theoretic approach," *IEEE Internet of Things Journal*, 2018, DOI: 10.1109/JIOT.2018.2880483, early access.

[18] M. B. Yassein, S. Aljawarneh, and W. Al-Sarayrah, "Mobility management of internet of things: Protocols, challenges and open issues," in *2017 International Conference on Engineering MIS (ICEMIS)*, May 2017, pp. 1–8.

[19] X. Sun and N. Ansari, "Avaptive avatar handoff in the cloudlet network," *IEEE Transactions on Cloud Computing*, DOI: 10.1109/TCC.2017.2701794, early access.

[20] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[21] M. Bouet and V. Conan, "Mobile edge computing resources optimization: A geo-clustering approach," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 787–796, Jun. 2018.

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

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

[24] N. Ansari and X. Sun, "Mobile edge computing empowers internet of things," *(Invited Paper) IEICE Transactions on Communications*, vol. E101-B, no. 3, pp. 604–619, Mar. 2018.

[25] O. Skarlat, S. Schulte, M. Borkowski, and P. Leitner, "Resource provisioning for IoT services in the fog," in *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)*, Nov. 2016, pp. 32–39.

[26] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han, "Joint radio and computational resource allocation in IoT fog computing," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2018.

[27] M. Kamoun, W. Labidi, and M. Sarkiss, "Joint resource allocation and offloading strategies in cloud enabled cellular networks," in *2015 IEEE International Conference on Communications (ICC)*, Jun. 2015, pp. 5529–5534.

[28] P. Mach and Z. Becvar, "Cloud-aware power control for real-time application offloading in mobile edge computing," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 5, pp. 648–661, 2016.

[29] S. Li, N. Zhang, S. Lin, L. Kong, A. Katangur, M. K. Khan, M. Ni, and G. Zhu, "Joint admission control and resource allocation in edge computing for internet of things," *IEEE Network*, vol. 32, no. 1, pp. 72–79, Jan. 2018.

[30] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, *A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing*. Springer Berlin Heidelberg, 2010.

[31] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[32] L. Kleinrock, *Queueing Systems: Computer Applications*. Hoboken, NJ, USA: Wiley-Interscience, 1976.

[33] D. Wu and R. Negi, "Effective capacity: a wireless link model for support of quality of service," *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 630–643, Jul. 2003.

[34] H. Khazaei, J. Misic, and V. B. Misic, "Performance of cloud centers with high degree of virtualization under batch task arrivals," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 12, pp. 2429–2438, Dec. 2013.

[35] J. B. Rosen, "The gradient projection method for nonlinear programming." *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, no. 4, pp. 514–532, 1961.

[36] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[37] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.

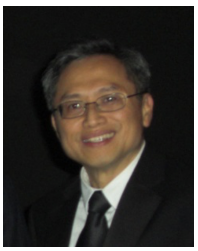
[38] A. Fréville, "The multidimensional 0–1 knapsack problem: An overview," *European Journal of Operational Research*, vol. 155, no. 1, pp. 1–21, 2004.

[39] D. P. H. Kellerer, U. Pferschy, *Knapsack Problems*. Springer Berlin Heidelberg, 2004.

[40] S. Sesia, M. Baker, and I. Toufik, *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.

[41] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 4–4.

Jingjing Yao (S17) received the B.E. degree in information and communication engineering from Dalian University of Technology (DUT) and the M.E. degree in information and communication engineering from University of Science and Technology of China (USTC). She is currently working towards the Ph.D. degree in Computer Engineering at the New Jersey Institute of Technology (NJIT), Newark, New Jersey. Her research interests include cloud computing, cloud radio access networks, and Internet of Things.



Nirwan Ansari (S78-M83-SM94-F09) is Distinguished Professor of Electrical and Computer Engineering at the New Jersey Institute of Technology (NJIT). He has also been a visiting (chair) professor at several universities.

He authored Green Mobile Networks: A Networking Perspective (IEEE-Wiley, 2017) with T. Han, and co-authored two other books. He has also (co-)authored more than 550 technical publications, over 250 published in widely cited journals/magazines. He has guest-edited a number of special issues covering various emerging topics in communications and networking. He has served on the editorial/advisory board of over ten journals including as Senior Technical Editor of IEEE Communications Magazine. His current research focuses on green communications and networking, cloud computing, and various aspects of broadband networks.

He was elected to serve in the IEEE Communications Society (ComSoc) Board of Governors as a member-at-large, has chaired some ComSoc technical and steering committees, has been serving in many committees such as the IEEE Fellow Committee, and has been actively organizing numerous IEEE International Conferences/Symposia/Workshops. He has frequently been delivering keynote addresses, distinguished lectures, tutorials, and invited talks. Some of his recognitions include IEEE Fellow, several Excellence in Teaching Awards, a few best paper awards, the NCE Excellence in Research Award, the ComSoc TC-CSR Distinguished Technical Achievement Award, the ComSoc AHSN TC Technical Recognition Award, the IEEE TCGCC Distinguished Technical Achievement Recognition Award, the NJ Inventors Hall of Fame Inventor of the Year Award, the Thomas Alva Edison Patent Award, Purdue University Outstanding Electrical and Computer Engineer Award, and designation as a COMSOC Distinguished Lecturer. He has also been granted 37 U.S. patents.

He received a Ph.D. from Purdue University—West Lafayette, IN, an MSEE from the University of Michigan—Ann Arbor, MI, and a BSEE (summa cum laude with a perfect GPA) from NJIT—Newark, NJ.