Effective Utilization of Renewable Energy Sources in Fog Computing Environment via Frequency and Modulation Level Scaling

Aref Karimiafshar, Massoud Reza Hashemi, Mohammad Reza Heidarpour, and Adel N. Toosi, *Member, IEEE*

***Abstract*—Fog computing introduces a distributed processing capability close to end-users. The proximity of computing to end-users leads to lower service time and bandwidth requirements. Energy consumption is a matter of concern in such a system with a large number of computing nodes. Renewable energy sources can be utilized to lessen the burden on the main power grid and reduce the carbon footprint, but due to fluctuations, effective utilization of renewable energy sources needs proper resource management. In this paper, we deal with properly managing the resources in a fog environment where the fog nodes are equipped with on-site renewable energy. This paper aims to design an efficient mechanism to dynamically dispatch requests among computing nodes and scale frequency and modulation level, based on current workload and the availability of renewable energy sources, to minimize the service time while keeping the renewable energy utilization and stability at a satisfactory level. We state the problem as the design of a controller for a system with time-varying nonlinear state equations. Accordingly, we borrow the Lyapunov optimization technique from the control theory to design the request dispatching mechanisms and prove its asymptotic optimality. We perform extensive simulations to evaluate the effectiveness of the proposed method. Simulation results demonstrate that our proposed method outperforms native time aware baseline scheme up to 26% and 39%, respectively, in terms of service time and renewable energy utilization.**

***Index Terms*—fog Computing, DVFS, DMS, Lyapunov optimization technique, request dispatching, renewable energy.**

I. Introduction

THE number of connected Internet-of-Things (IoT) devices is predicted to reach 50 billion by the end of 2020 [1] and will increase to 75.4 billion in 2025 [2]. It is expected that more than two Exabyte of data will be generated by such devices, on a daily basis [1]. On the other hand, the current trend suggests that, by 2020, data-centers will consume 140 billion Kilowatt-hours to process this huge amount of data, annually [3]. Therefore, without any countermeasure, the share of information and communication technology (ICT) industry in the global CO_2 emission will be more than 2% [4]. Accordingly, immediate actions should be taken to deal with this pressing issue.

The concept of Fog Computing (FC) has been recently proposed as an architectural shift in the service provides' networks in order to support 5G killing applications such as augmented reality, and Industry 4.0, among others [5]. In contrast to the traditional cloud-only paradigm in which requests are directly sent to the cloud to be processed, FC paradigm introduces a three-layer structure (IoT-Fog-Cloud), Fig. 1. In FC, the "Fog Nodes" (FNs) are placed at the edge of the network, in order to preliminary process data close to end-users (IoT devices), instead of remote data-centers, but the cloud yet remains as a possible option. As a result, FC feeds short response time and context-aware demands of many emerging applications. On the other hand, using distributed FNs enables on-site harvesting of renewable (green) energy [6]. However, according to IEEE Computer Society, fog and edge computing itself requires innovation in energy harvesting [7]. Because Green Energy (GE) is unpredictable and fluctuating, there is a need for an effective utilization scheme of the resources to be matched with the GE availability.

The fog paradigm in several aspects is different from the cloud. Server nodes in the cloud are often homogeneous and placed in close proximity, and connected to each other through high-speed networks. This enables the cloud-managing platform to perform smart power management techniques such as Virtual Machine (VM) consolidation and switching on/off the servers [8]. However, the FNs are heterogeneous, highly distributed, and small in size and power, with respect to cloud. As a result, the cloud-specific solutions such as VM consolidation and switching on/off the servers do not work well here [6]. Therefore, we resort our proposed request dispatching mechanism upon two other modern power management schemes which are suitable for FC environment, called DVFS (Dynamic Voltage and Frequency Scaling) and DMS (Dynamic Modulation Scaling). DVFS enables processors to run at different frequencies and voltages settings to reduce the power consumption of the processing unit [9]. DMS helps to dynamically scale the modulation level and hence transmit speed to save the communication energy [10]. In contract to VM consolidation and switching on/off the servers, the DVFS and DMS techniques can be effectively deployed in FC environment with less opportunity for resource sharing among FNs [11].

In this paper, considering the FNs are equipped with on-site renewable energy, we deal with request dispatching problem along with frequency and modulation level scaling to shape workload allocation based on the availability of GE. The requests that come from the IoT devices into the system should be properly dispatched among the available FNs to

A. Karimiafshar, M. Hashemi, and M. Heidarpour are with the Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, IR. e-mail: A.Karimiafshar@ec.iut.ac.ir, {hashemim, mrheidar}@cc.iut.ac.ir.

A. N. Toosi is with Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia. e-mail:adel.n.toosi@monash.edu.

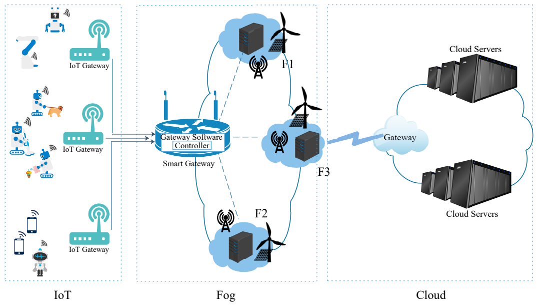


Fig. 1: A fog computing environment consisting of IoT devices, fog nodes, and a cloud at remote data-center.

meet their requirements. Dispatching the requests, considering both service time and energy consumption, is not a trivial task. Indeed, due to the heterogeneity and geographical distribution, the FNs offer a different amount of processing power and GE. Therefore, simply assigning the requests to the FNs which provide the least service time may not be the optimal answer, as the fastest node may be out of GE and/or need to consume very high energy for data processing and communication at the moment. Furthermore, considering GE and its fluctuating nature, and taking the stability into account makes the problem more sophisticated. First, we formulate the problem as an offline stochastic optimization problem. Then, Lyapunov Optimization Technique (LOT) [12] is leveraged to simplify the problem and lead to an asymptotically optimal solution. Finally, based on the LOT analysis, a low-complexity algorithm is proposed for the problem.

The main contributions of this paper are summarized as follows:

- Dynamic request dispatching and frequency and modulation level scaling are simultaneously considered to provide efficient power and resource management.
- Based on LOT, an online algorithm is proposed to dynamically dispatch requests among FNs while adaptively adjust frequency and modulation level of the nodes to match with availability of renewable energy sources.
- Using real solar irradiation data, extensive simulations are performed and the Delay-Energy tradeoff relationship is examined.

II. Related Works

A. Resource and Power Management

Recently, in the context of FC, resource and power management has attracted much attention [13], [14]. There have been extensive studies on the problem of computation offloading, workloads assignment and request dispatching [15]–[17]. For example, Bitam *et al.* [15] studied the problem of job scheduling in FC environment, and proposed a bio-inspired optimization algorithm to optimally distribute a set of tasks among FNs to find a balance between execution time and allocated memory. Also, Ni *et al.* [17] suggested an allocation policy for FC by invoking priced timed Petri nets. They considered both price and time cost of request completion, and, also take credibility evaluation of both FNs and users into account.

On the other hand, there have been some studies focused on energy control and power management [3], [18]–[20].

TABLE I: THE SUMMARY OF MOST RELEVANT WORKS

Reference	Objectives		Techniques		Lyapunov	RNE[3]
	S.T [1]	E.C [2]	DVFS	DMS		
[15]	✓					
[16]	✓					
[17]	✓					
[18]		✓				✓
[19]		✓				✓
[3]		✓				
[20]		✓	✓			
[21]	✓	✓				
[22]	✓	✓				
[23]		✓				✓
[24]	✓	✓				✓
[4]	✓	✓	✓			✓
[25]	✓	✓	✓			
[26]		✓	✓	✓		
[27]	✓	✓	✓			
[9]	✓	✓	✓	✓		
[28]	✓				✓	
[29]	✓	✓			✓	
[5]	✓	✓	✓		✓	
[11]	✓	✓	✓		✓	
This work	✓	✓	✓	✓	✓	✓

[1] S.T: Service Time [2] E.C: Energy Consumption [3] RNE: Renewable Energy

For instance, Mishra *et al.* [3] investigated the scheduling of requests to fog servers as bi-objective minimization problem. They proposed an allocation framework based on metaheuristic techniques. Furthermore, in [20], Naranjo *et al.* studied the minimization of both processing and communication energy consumption. They proposed a bin packing-type heuristic for management of fog resources.

Besides, the works such as [21] and [22] investigated both computation offloading and energy control. For example, Meng *et al.* [21] suggested a computation offloading policy to minimize total energy consumption while completing the requests within a given time constraint. The authors defined the concept of computation energy efficiency and divided the problem into some subproblems which are individually solved. In addition, Liu *et al.* [22] studied the energy consumption in conjunction with execution delay and offloading payment cost in FC. Based on the queuing theory the joint problem of minimizing energy consumption and execution delay is formulated to find the optimum offloading probability for each device.

Previous works explored computation offloading, energy control and power management subject to different objectives such as minimization of service time, energy consumption or price. However, these works have not considered the specific characteristics of the FC, nor the use of DVFS and DMS techniques in their energy control and power management schemes as we do in this paper.

Furthermore, in the context of cloud computing, there have been extensive studies on the scheduling problem with respect to the Green Cloud concept [4], [23], [24]. For example, Li et al. [23] suggested a dynamic VM consolidating method to adjust the number of ON servers in order to match energy consumption with the renewable energy availability. Also, Wu et al. [4] studied green energy-efficient scheduling for cloud computing. They proposed a priority-based job scheduling according to the users' Service Level Agreement (SLA) requirements and calculation of some weights in their own method. However, these works are based on VM consolidation and switching on/off the servers that is not applicable to the

FC, or they just try to reduce the brown energy consumption.

B. DVFS and DMS Techniques

Besides request dispatching, DVFS can help to provide a better power management. Gerards *et al.* in [10] explored the interplay between DVFS and task scheduling. They investigated the problem of simultaneously tuning the frequency and choosing a proper scheduling scheme that together minimize the energy consumption. Furthermore, some works such as [25] and [27] studied the joint problem of computation offloading and choosing the optimum frequency in mobile cloud to reduce the energy consumption.

DMS can be utilized along with DVFS to provide better control on communication energy consumption [9], [26]. For example, Zhang *et al.* in [9], suggested such a method of energy control for wireless sensor network applications.

C. Lyapunov Optimization

LOT has been adopted in several works in the area of resource and power management in FC environment [28], [29]. For instance, Zhang *et al.* in [29] investigated the joint problem of allocation optimization and resource management to minimize average energy consumption while guaranteeing the service delay. The problem has been formulated as a stochastic optimization problem and using LOT an online algorithm is introduced as the solution. Also, Zhao *et al.* [28] explored node assignment and resource allocation in a fog-enabled content delivery wireless network. Based on LOT, they developed a scheduling algorithm to optimize the performance, in terms of service time, queue backlog and network throughput. Although, these papers dealt with service time, energy consumption, or both via LOT, but they did not consider DVFS and DMS techniques nor the use of renewable energy sources.

D. Most Relevant Works

In recent works, Yang *et al.* in [5] developed an analytical framework to balance service delay and energy consumption while taking into account the CPU frequency scaling. They leveraged LOT and proposed an algorithm to reduce both overall energy consumption and service delay. Similarly, in [11], Kwak *et al.* explored the problem of computation offloading in mobile cloud system. They proposed an algorithm by invoking LOT to minimize CPU and network energy consumption while satisfying the delay constraint. Our work in some aspects is similar to [5] and [11]. We also deal with the service time, take the energy consumption into account and use the potential benefits of DVFS. On the other hand, we additionally adopt DMS to control the communication energy and, furthermore, we assume that the FNs can be powered by renewable energy sources. A unique characteristic of our work in this paper is that we designed to shape the electricity demands (adopt workload) of the FNs to match renewable energy supply, considering the specific characteristics of FC. A summary of the most relevant works is presented in TABLE I.

III. System Model

We consider a FC environment consisting of the FNs, a cloud at remote data-center and IoT devices, as depicted in Fig. 1. The set of FNs is denoted by $\mathcal{N} = \{1, 2, \dots, N\}$.

TABLE II: THE SUMMARY OF KEY SYMBOLS

Symbol	Definition
t	Index of time slots
$\mathcal{F}$	The set of frequency levels
$\mathcal{Z}$	The set of modulation levels
$\mathcal{P}$	The set of computing nodes
$\mathcal{K}^{\mathrm{t}}$	The set of requests at time slot t
f_m	Each frequency level in the set of $\mathcal{F}$
z_j	Each modulation level in the set of $\mathcal{Z}$
$A(t)$	Request arrival rate into the system
i	Index of computing nodes
$R_i(t)$	Request arrival rate into the computing node i
$B_i(t)$	Service rate of the computing node i
$\mathbf{c}(t)$	Vector of control decision
$Q_i(t)$	Queue backlog related to the computing node i
$e_i(t)$	Energy consumption of the computing node i
$e_i^P(t)$	Energy consumed for processing
$e_i^C(t)$	Energy consumed for communication
k	Index of requests
S_k	Processing requirement of the request k
D_k	Communication requirement of the request k
$u_{k,i}$	Uploading time for the request k and the computing node i
$w_{t,i}$	Waiting time for the request k and the computing node i
$p_{k,i}$	Processing time for the request k and the computing node i
$d_{k,i}$	Time to return back the result of request k from the computing node i
$L(t)$	Lyapunov function
$\Delta(\boldsymbol{\theta}(t))$	Drift in Lyapunov function regarding the vector $\boldsymbol{\theta}(t)$

The FNs are located in different places and communicate with each other through radio communication. It is assumed that the FNs are equipped with DVFS capable CPUs and DMS capable radio peripherals. The DVFS-enabled CPUs support M discrete frequency levels, $\mathcal{F} = \{f_1, f_2, \dots, f_M\}$, and the DMS-enabled radios have J modulation levels, $\mathcal{Z} = \{z_1, z_2, \dots, z_J\}$.

The FNs are equipped with on-site renewable energy sources (e.g. solar panels) and configurable switches to control the connection of the nodes to either main power grid (PG), on-site renewable energy source, or both. Whenever the GE is available the FNs are powered by on-site resources, and whenever the GE is not enough or available at all, the FNs can consume electricity from the main PG network. Since FNs are connected to PG as backup, we consider no battery or storage devices to store the energy for later uses. Indeed, energy harvesting can come along with or without energy storage devices (batteries) [23], [30]. Using batteries has its own drawbacks, such as energy losses in batteries, other kinds of pollution that batteries have, or cost of providing and maintaining the batteries which is dominating in many cases [24]. On the other hand, when there are no devices to store the extra energy, the controller should put all its efforts to maximize the GE utilization at each moment.

We assume the system operates in a time-slotted manner, indexed by $t \in \{0, 1, 2, \dots\}$. The system parameters such as the number of Computing Nodes[1] (CNs) and quality of wireless links, remain fixed during each time slot but can vary from one slot to the other. Moreover, this is at the beginning of each time slot that the requests are assigned to CNs. The CNs, under the assistance of a controller, accept and execute incoming requests. The controller, at each time slot, conducts resource discovery and makes a control decision $\mathbf{c}(t) = \{(k, i, f, z) | k \in \mathcal{K}^{\mathrm{t}}, i \in \mathcal{P}, f \in \mathcal{F} \ and \ z \in \mathcal{Z}\}$, where $\mathcal{K}^{\mathrm{t}}$ denotes the set of requests at time slot t.

[1]Including the set of FNs and the cloud, $\mathcal{P} = \mathcal{N} \cup \{C\}$. Therefore, we have $|\mathcal{P}| = N + 1$.

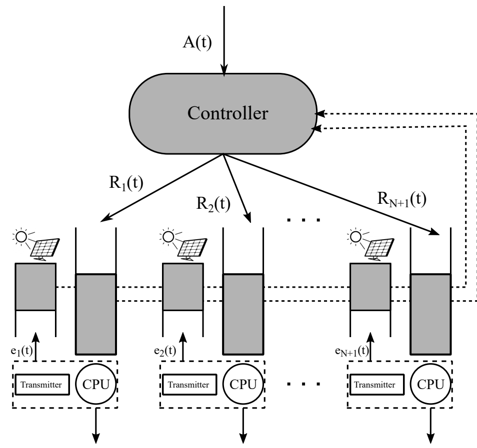


Fig. 2: Queue Model of The System.

Notations: We use the following notations throughout the paper. Vectors are specified in boldface letter such as $\mathbf{Q}$. Sets are specified by fancy letters such as $\mathcal{P}$. We use $\mathbb{E}$ to present expectation. Furthermore, the summary of key symbols is presented in TABLE II.

The requests come into the controller from IoT devices or other FNs. It is assumed that the request arrival rate, $A(t)$, is independent and identically distributed (i.i.d). The requests have different computational and communication demands. The controller, based on available resources and system conditions, dispatches the requests among the CNs and scales their frequency and modulation levels. The requests that are dispatched to each CN, enter into the queue and may wait for some time to get processed. We denote the arrival rate into each CN by $R_i(t)$ that is a function of control decision $\mathbf{c}(t)$, with the constraint of $0 \leq R_i(t) \leq R_i^{max}$ for all $i \in \mathcal{P}$. The requests get processed with service rate $B_i(t)$ at each CN, Fig. 2. The service rate is proportional to the processing capability of the CN which generally can be converted to $B_i(t) = f_m/\vartheta$, where ϑ is in terms of number of cycles/request [25]. At each time slot we have $A(t) = \sum_{i=1}^{N+1} R_i(t)$ and $A(t) \leq \sum_{i=1}^{N+1} B_i(t)$. Therefore, the dynamic equation for each CN can be written as,

$$Q_i(t+1) = \max[Q_i(t) - B_i(t), 0] + R_i(t), \tag{1}$$

where $Q_i(t)$ is the queue backlog of i^{th} CN. Equation (1) expresses that, at each time slot, some of the requests (up to the service rate) can be served and the remainder (if there are any) have to wait in the queue for the next time slots.

A. *Energy Consumption Models*

The CNs consume energy for either processing or communication. Thus, we can write the energy consumption of the CN i as the sum of the energy consumed for processing a request, e_i^P, and the energy consumed for the reception/transmission of the input/output data, e_i^C, in the form of,

$$e_i(t) = e_i^P(t) + e_i^C(t). \tag{2}$$

1) Processing Energy Consumption Model: The DVFS-enabled CPU can be configured on one of M frequency levels. The processing energy for each CN is a function of the CPU frequency f_m and the supply voltage V_m in the form of $f_m V_m^2$. However, the voltage is approximately a linear function of the frequency when the CPU is working at the low voltage limits [31]. Therefore, the energy consumption is given by,

$$e_i^P(t) = \sum_{k=1}^{K_c^t} (\alpha_i f_m^3 + e_i^{P,ind})(\frac{S_k}{f_m}), \tag{3}$$

where α_i is a constant which represents the CPU switching capacitance of CN i, $e_i^{P,ind}$ is the speed-independent energy consumption of the CPU, K_c^t denotes the number of requests assigned to CN i, and S_k represents the computing requirement of request k at time slot t.

2) Communication Energy Consumption Model: The DMS-enabled radio can be configured based on the network and channel condition to one of J modulation levels from set $\mathcal{Z}$. The modulation level determines the number of bits encoded in each signal symbol, and therefore, can affect transmission rate x, because $x = z_j.r$, where r represents the symbol rate. Therefore, the communication energy can be expressed as [9],

$$e_i^C(t) = \sum_{k=1}^{K_c^t} (\beta_i r(2^{x/r} - 1) + e_i^{C,ind})(\frac{D_k}{x}), \tag{4}$$

where β_i is determined by the parameters such as transmission quality, noise level and etc. $e_i^{C,ind}$ is the modulation-independent energy consumption of the radio and D_k indicates the communication requirement of request k at time slot t.

IV. Problem Statements

In this section, we first deal with green energy utilization and formulate it as a lateral constraint in the form of long term average utilization. Then, we formulate the optimization problem to minimize service time with queue stability satisfaction.

A. *Average Utilization of Green Energy*

The requests are dispatched among the FNs based on the availability of renewable energy sources and current workloads. Where and when the GE is not available nor high enough, the configurable switches are adjusted to compensate for the remainder of the required energy from the main grid. In such a case, the lower energy the FN consumes, the lower brown energy is consumed. Therefore, the frequency and modulation level are configured in the way to reduce energy consumption. On the other hand, when and where the FNs have enough available green energy, the frequency and modulation level can be increased to provide better performance or compensate for the computing requirements.

It is desired to minimize the amount of surplus energy borrowed from PG network. We denote the surplus borrowed energy by $U_i(t)$, which is a function of the control decision $c(t)$, $U_i(t) = \hat{U}_i(\mathbf{c}(t))$. We consider it as the long term average utilization and define it as a constraint in the form of $\bar{U}_i(t) = \limsup_{T\rightarrow\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\{\ U_i(t)\} \leq C_U$, where C_U is a finite value indicating the threshold for the utility function.

B. *Queue Stability*

In the queue model of the system, the *stability* is defined as the condition which all the queues have finite backlogs. We can formally write the definition of the queue stability for CN i, $1 \leq i \leq N+1$, as,

$$\bar{Q}_i(t) = \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{|Q_i(t)|\} < \infty. \tag{5}$$

The definition in (5) states that a queue is stable if not only the queue backlog be finite in the long term but also the queue backlog be always bellow a finite value. This kind of stability is called *strong stability* [12].

C. Offline Problem Formulation

We define the service time, from the moment that the request is sent to the selected CN till the result is sent back to the originating IoT device. Therefore, the service time for the request k, processed within the CN i is the sum of the required time to upload the request to selected node, $u_{k,i}$, the time that the request waits in the queue, $w_{t,i}$, the time that is spent to process the request, $p_{k,i}$, and the time to send back the results, $d_{k,i}$, in the form of $\psi_k(t) = \hat{\psi}(c_k(t)) = u_{k,i}+w_{t,i}+p_{k,i}+d_{k,i}$. The total service time can be written as $\psi(t) = \hat{\psi}(\mathbf{c}(t)) = \sum_{k=1}^{K^t} \hat{\psi}(c_k(t))$. Thus, we can formally define the problem as:

$$\textbf{P1:} \min_{\mathbf{c}(t)} \ (\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{\hat{\psi}(\mathbf{c}(t))\}) \tag{6}$$

$$S.T \ \bar{U}_i(t) \leq C_U \tag{6a}$$

$$\bar{Q}_i(t) \leq \infty \ for \ i \in \mathcal{P}, \tag{6b}$$

P1 states the problem as an energy constrained service time minimization problem along with the stability condition satisfaction. In (6), the constraint (6a) is for the GE utilization and the constraint (6b) imposes the queue stability condition for each CN.

The offline problem P1 is a stochastic optimization problem with various unknown variables. Obtaining the optimal solution for P1 is involved as it requires the (exact) knowledge about the dynamic statistics of the system. Therefore, we leverage LOT to derive an online algorithm as an asymptotically optimal approximation solution to P1 [28].

V. Lyapunov Optimization Based Solution

Using LOT, we can transfer the problem P1 into a pure stability per-slot problem which is deterministic at each time slot. The main idea of LOT is starting from a stable initial condition and preserving the stability by continuously bounding the queue backlog changes. A Lyapunov function $L(t)$ is defined, which presents the congestion in the queues. Small values of L indicate that all the queues are working normally, while large values of L indicate that there is at least one congested queue. We define $L(t)$ as a quadratic function as defined in [5], [28], in the form of,

$$L(t) = \frac{1}{2} \sum_{i=1}^{N+1} Q_i(t)^2, \tag{7}$$

which is a scalar representation of congestion in the system.

Also, *drift in Lyapunov function* is defined as the expectation of difference in Lyapunov function at two consecutive time slots as $\Delta(\boldsymbol{Q(t)}) = \mathbb{E}\{L(t+1) - L(t)|\boldsymbol{Q(t)}\}$, where $\boldsymbol{Q(t)}$ is the vector of concatenation of all the queues, $\boldsymbol{Q(t)} \triangleq [Q_1(t), Q_2(t), ..., Q_{N+1}(t)]$.

Lyapunov drift considers only the queue backlog and does not take into account penalty function. Therefore, Lyapunov drift plus Penalty (LDpP) is defined to take the penalty function into the consideration. LDpP can be written as,

$$\Delta(\boldsymbol{Q(t)}) + \omega\mathbb{E}\{\psi(t)|\boldsymbol{Q(t)}\}, \tag{8}$$

where ω is the scaling factor which regulates the privilege of stability against the service time delay.

Considering the queue dynamic equation (1), the bounds on workloads arrivals, and definition of LDpP, the following proposition holds.

Proposition 1: Under any feasible control decision $\mathbf{c}(t)$, LDpP is upper bounded by,

$$\Delta(\boldsymbol{Q(t)}) + \omega\mathbb{E}\{\psi(t)|\boldsymbol{Q}(t)\} \leq \Upsilon + \omega\mathbb{E}\{\hat{\psi}(\mathbf{c}(t))|\boldsymbol{Q}(t)\} \tag{9}$$
$$- \sum_{i=1}^{N+1} Q_i(t)B_i(t) + \sum_{i=1}^{N+1} Q_i(t)\mathbb{E}\{\hat{R}_i(\mathbf{c}(t))|\boldsymbol{Q}(t)\},$$

where Υ is a constant value.

Proof: See appendix A. □

VI. Proposed Online Algorithm

In this section we transfer the problem P1 into the Lyapunov optimization framework and present an online algorithm as the solution. In order to use LOT for constrained optimization problems, the first step is to write the constraints in terms of virtual queues. Therefore, in the following, we first introduce a virtual queue for satisfying the green energy utilization constraint. Then, we introduce our online algorithm.

A. Satisfying Green Energy Utilization Constraint

To satisfy constraint (6a), we define a virtual queue G which is updated by,

$$G_i(t+1) = \max[G_i(t) + y_i(t), 0], \tag{10}$$

where, $y_i(t) = U_i(t) - C_U$. Furthermore, $U_i(t)$ is a function of $\boldsymbol{c(t)}$ and is obtained by,

$$\hat{U}_i(\mathbf{c}(t)) = |\min[e_{pro}(t) - e_i(\mathbf{c}(t)), 0]|, \tag{11}$$

where, $e_{pro}(t)$ is the amount of available GE at time slot t, and $e_i(\boldsymbol{c(t)})$ indicates the energy consumption of i^{th} node under control decision $\boldsymbol{c(t)}$ at time slot t, obtained by (2). Defining virtual queue G as in (10) leads to the following proposition.

Proposition 2: If the proposed policy makes the virtual queue G always stable, the constraint (6a) is always satisfied.

Proof: See Appendix B. □

B. Transferring to Lyapunov Optimization Framework

In the following, we transfer the offline problem P1 into Lyapunov optimization framework. We define the vector $\boldsymbol{\theta}(t)$ as the concatenation of real and virtual queues as $\boldsymbol{\theta}(t) \triangleq [\boldsymbol{Q}(t), \boldsymbol{G}(t)]$. Therefore, we rewrite the Lyapunov function as:

$$L(t) = \frac{1}{2}(\sum_{i=1}^{N+1} Q_i(t)^2 + \sum_{i=1}^{N+1} G_i(t)^2). \tag{12}$$

Based on the definition of Lyapunov function in (12), we can write LDpP in the form of $\Delta(\boldsymbol{\theta(t)}) + \omega\mathbb{E}\{\psi(t)|\boldsymbol{\theta}(t)\}$.

In the Lyapunov optimization framework to solve P1, we need to design an algorithm to greedily minimize the upper bound of LDpP at each time slot. According to proposition 1, definition of virtual queue G and the Lyapunov function in (12), and ignoring uncontrollable terms and constant values, the following upper bound for LDpP expression is obtained:

$$\textbf{P2:} \min_{\mathbf{c}(t)} \quad \omega\mathbb{E}\{\hat{\psi}(\mathbf{c}(t))|\boldsymbol{\theta}(t)\} + \sum_{i=1}^{N+1} Q_i(t)\mathbb{E}\{\hat{R}_i(\mathbf{c}(t))|\boldsymbol{\theta}(t)\} + \sum_{i=1}^{N+1} G_i(t)\mathbb{E}\{\hat{y}_i(\mathbf{c}(t))|\boldsymbol{\theta}(t)\}. \quad (13)$$

In order to minimize the expectation expressions in P2, we leverage the concept of opportunistically minimization of an expected value [12]. As a result, at each time slot, we can solve the following problem:

$$\textbf{P3:} \min_{\mathbf{c}(t)} \omega\hat{\psi}(\boldsymbol{c(t)}) + \sum_{i=1}^{N+1} Q_i(t)\hat{R}_i(\boldsymbol{c(t)}) + \sum_{i=1}^{N+1} G_i(t)\hat{y}_i(\boldsymbol{c(t)}). \quad (14)$$

C. An Online Algorithm Based on LOT

In this section, we develop an online algorithm based on the minimization problem P3. This algorithm will be implemented at the controller and performed at the beginning of each time slot. The algorithm takes the update information obtained from resource discovery as the input, solves the minimization problem P3 and produces the control decision $\boldsymbol{c(t)}$ as the output. The algorithm is summarized in Algorithm 1. The algorithm works based on the current system condition and does not need the system dynamics.

Algorithm 1: *FRA*

```
    Input: List of requests, computing nodes and voltage, frequency
    and modulation levels
    Output: control decision c*(t)
1:  Initialization
      Initialize the control parameters in the LOT
2:  While t < t_end, do
    /* Obtaining required parameters and finding the best c*(t) */
    /* c_k(t) ≜ (k, i, f, z)*/
3:    For k = 1 to K^t
4:      For i = 1 to N + 1
5:        For all f in F and all z in Z
6:          LDpP[j] = ω ψ̂(c_k(t))+
          Σ_{i=1}^{N+1} Q_i(t) R̂_i(c_k(t)) + Σ_{i=1}^{N+1} G_i(t) ŷ_i(c_k(t))
          End for
        End for
7:      c*_k(t) = arg min_{c_k(t)} (LDpP)
8:      Logically update the queues based on the model
      End for
9:    c*(t) = ∪_{k=1}^{K^t} c*_k(t)
10:   Dispatch according to  c*(t)
11:   Update the Queues
    End While
```

Line 1: Initialize the parameters in Lyapunov framework and system model, such as the control parameter ω.

Line 3-7: Go through all the requests arrived within the last time slot, search for all the CNs and possible configurations in terms of frequency and modulation level, to find the best node and configuration.

Line 8: Logically update the queues based on the control decision $\mathbf{c}_k^*(t)$ for the selected request and current time slot.

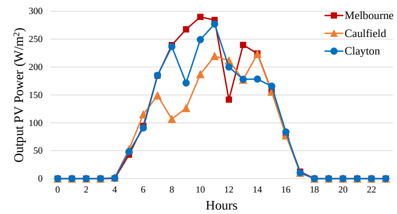


Fig. 3: Output PV Power related to each FN on the 17th of January 2018.

Line 9-11: Make the whole control decision $\mathbf{c}^*(t)$ as the union of all $\mathbf{c}_k^*(t)$, dispatch the requests and update the queues.

Computational complexity: For each request, the algorithm goes through all the CNs to calculate the objective function (Line 4). Considering we have $N+1$ CNs in the system, it takes $N+1$ iterations to obtain all the values. Besides, for each node, to properly scale the frequency and modulation level, the objective function is examined by applying different frequency and modulation levels (Line 5). Considering M frequency and J modulation levels, it takes $M.J = \varphi$ iterations. Therefore, finding the best assignment for each request takes $(N+1).\varphi$ iterations. In regard that the requests are assigned at the beginning of each time slot, and assuming K^t requests arrive to the system at time slot t (Line 3), it takes $K^t.(N+1).\varphi$ iterations to assign all the requests at time slot t.

VII. Evaluation and Simulation Results

We performed extensive simulations in our custom-designed simulator in Matlab. In this section, first, we describe the simulation setup. Then, the simulation results are presented and discussed.

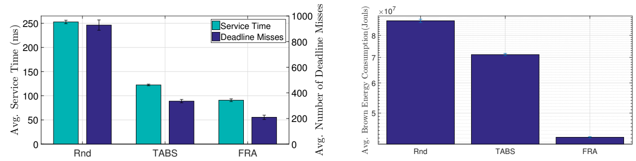
A. Simulation Setup

We build a FC environment with 30 FNs along with a cloud server at remote data-center, and 200 IoT nodes. The processing capabilities of FNs, in terms of "Million Instructions Per Second (MIPS)", have been randomly generated with a uniform distribution over the interval [150, 400]. The processing capability of the cloud server is assumed to be 1600 MIPS. The IoT nodes are specified by their request demands, which follows a Poisson process with different average rates during day (λ_D) and night (λ_N), and transmission rate of the link to the CNs. The fog and IoT nodes are interconnected based on a random topology. We change the transmission channel condition at each time slot to emulate the real environment as much as possible.

The FNs are assumed to be equipped with solar panels, with one of 4, 6 or 8 m^2 size of photovoltaic (PV) modules. The FNs are divided into three domains which are located in different places. The FNs in each domain are scattered in an area of $1Km^2$. The location of each domain, size and configuration of PV modules are summarized in TABLE III.

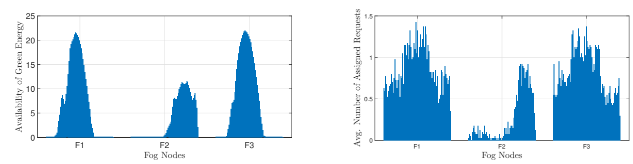
TABLE III: LOCATION AND CONFIGURATION OF THE PV MODULES

Row	Site Place	Latitude	Size(m^2)	Tilt Angle
1	Monash University-Clayton Campus	-37.91	4, 6, 8	37.91°
2	Monash University-Caulfield Campus	-37.87	4, 6, 8	37.87°
3	Melbourne	-37.81	4, 6, 8	37.81°



(a) (b)

Fig. 4: (a) Average Service Time and Average Number of Deadline Misses, and (b) Average Brown Energy Consumption over 100 iterations.



(a) (b)

Fig. 5: (a) Availability of Green Energy and (b) Average Number of Assigned Requests to each Fog Node.

We derived solar irradiation information from Bureau of Meteorology solar Global Horizontal Irradiance (GHI) data. GHI presents solar radiation on horizontal surface. To obtain the effective output power by a PV module, we use the model in [32], with the tilt angle equal to longitude of the location of the sites (the tilt angle for the most efficient performance in course of year, should be equal to longitude of the location [33]) and efficiency of 30% for PV modules. The output PV power for a selected day (on the 17th of January 2018) is depicted in Fig. 3. Also, we derived the frequency levels from the supporting frequency list of the real platform Creator PXA270 [34].

We assume that the incoming requests have different computation, and communication demands follow the exponential process with the average γ_1 and γ_2, respectively. TABLE IV presents the configuration of the system and the model parameters.

TABLE IV: CONFIGURATION OF SYSTEM AND MODEL PARAMETERS

λ_D	λ_N	γ_1	γ_2	ω	C_U
0.2	0.1	0.3	0.1	5×10^5	100

To evaluate the proposed methods, we chose three different metrics, namely average service time, average number of deadline misses and average brown energy consumption.

B. Performance Comparison with Baseline Schemes

In this section, we compare our proposed algorithm with a naive "time-aware baseline scheme", called "TABS". The TABS scheme dispatches the requests to the shortest job queue and scales frequency and modulation level using the time of day as a proxy for GE availability. Also, we compared the results against Rnd scheme to highlight the performance gain of different schemes. In Rnd scheme, the incoming requests are randomly dispatched among the FNs, and the frequency and modulation level are randomly scaled.

Based on the aforementioned setup, we performed the simulation for 1000 time slots and repeated it for 100 runs. The results are reported in the average of 100 runs.

In Fig. 4(a), we demonstrate the performance for the proposed algorithm and the two baseline schemes, in terms of average service time and average number of deadline misses. It can be observed that FRA provides on average 64% and 26% better service time compared to Rnd and TABS, respectively. Also, FRA leads to a lower number of deadline misses, on average 77% and 37%, compared to Rnd and TABS, respectively.

In Fig. 4(b), we evaluate the performance of FRA in terms of brown energy consumption, compared to the two baseline schemes. FRA on average shows 51% and 39% better utilization of green energy than Rnd and TABS schemes, respectively.

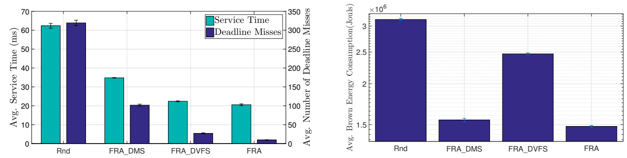
Remark: In practice, the FNs are divided into domains (for instance, a domain of nodes in a factory, university campus, shopping center and etc.) [35]. We may have hundreds or thousands of such domains, each group (cluster) of which under the control of one request dispatching entity. In the simulation, we have considered a cluster consisting of three domains, with 10 FNs in each domain, but the results can be numerically extended to more domains.

C. Impact of the Optimization to the System Metrics

In this section, we further investigate the impact of the optimization to the system metrics. We designed two scenarios to show how the requests are eventually dispatched between the FNs, and how DVFS and DMS can impact dispatching the requests.

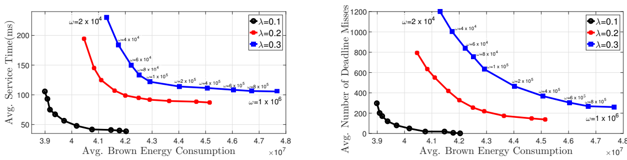
First scenario: We consider a fog network of three FNs (F1, F2 and F3) and a cloud server at a remote data-center. The GE profile of F1 and F3 are set based on the real data presented in Figure 3, but we intentionally set F2's GE profile manually differently to show how the requests are dispatched based on the GE availability. Fig. 5 illustrates the simulation results, presenting both energy profile, Fig. 5(a), and distribution of the requests, Fig. 5(b). It can be observed from Fig. 5 that the distribution of requests follows GE availability. For example, F2 has no GE available at first, therefore less number of requests are assigned to it, but as GE increases more requests are assigned and when F2's GE profile is at its peak the number of requests that are assigned to F2 reaches to its peak.

Second scenario: In this scenario, we investigate the individual role of DVFS and DMS on the system performance. We made changes in the proposed method (FRA) to create two variations, called FRA-DVFS and FRA-DMS. In the FRA-DVFS, we just considered scaling voltage and frequency and did not consider modulation level scaling. On the other hand, in the FRA-DMS, we just considered modulation level scaling and did not consider voltage and frequency scaling. To better present the results we compare the methods against Rnd. Fig. 6(a) and Fig. 6(b) show the simulation results. As it is observed from Fig. 6(a), with respect to Rnd, FRA-DVFS shows 64% and 89%, and FRA-DMS shows 44% and 68% improvement in the course of service time and number of deadline misses, respectively. Therefore, the most of FRA's performance in the course of service time and number of deadline misses is due to use of DVFS technique. On the other hand, as it is observed from Fig. 6(b), FRA-DVFS and FRA-DMS show 22% and 51% improvement in the course



(a) (b)

Fig. 6: (a) Average Service Time (from left) and Average Number of Deadline Misses (from right), and (b) Average Brown Energy Consumption for each method of FRA_DMS and FRA_DVFS compared to FRA and Rnd.



(a) (b)

Fig. 7: (a) Average Service Time and (b) Average Number of Deadline Misses against Average Brown Energy Consumption for different arrival rates and values of control parameter ω.

of GE utilization with respect to Rnd, respectively. Therefore, the most of improvement in the course of GE utilization that FRA presents is obtained by use of DMS technique. These observations also are justifiable in our system model, through equations (3) and (4). Therefore, each of DVFS and DMS has its own positive impact on the performance of the method but using both of them together shows a kind of synergy on the behavior of the method.

D. Delay- Energy Tradeoff

Fig. 7(a) and Fig. 7(b) further investigate the tradeoff relationship between time related performance metrics (i.e., service time and deadline miss) and the brown energy consumption.

In Fig. 7(a), we plot the amount of service time against brown energy consumption for different values of control parameter ω and different workloads. As it can be observed from Fig. 7(a), there is a tradeoff between average service time and average energy consumption depending on the value of ω. Therefore, in practice to balance between these two objectives, a proper value of ω, based on the system characteristics and requirements, should be selected.

Also, Fig. 7(b) depicts the average number of deadline misses against brown energy consumption for different workloads and different values of ω. These results also verify the tradeoff relationship between time related performance metrics and brown energy consumption.

E. Overall Comparison for FRA

Regarding that the problem of request dispatching and dynamically scaling frequency and modulation level is a multi-objective problem, a good design of the controller should provide all the objectives simultaneously. In Fig. 8, we provide an overall comparison between FRA and the two baseline schemes in a Kiviat diagram. The value of each parameter on each axis for different methods are normalized, to be better presented.

Fig. 8 presents the superiority of FRA over the selected baseline schemes, with respect to all the chosen metrics. In

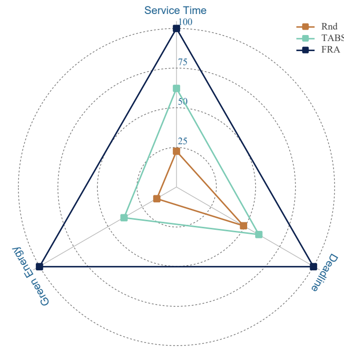


Fig. 8: Over all comparison between FRA and the baseline shemes.

particular, FRA shows 64%, 77% and 51% better performance than Rnd, and 26%, 37% and 39% better performance than TABS, in terms of service time, the number of deadline misses and brown energy consumption, respectively.

VIII. Conclusions

In this paper, we investigated the joint problem of dynamic request dispatching, and frequency and modulation level scaling to provide better utilization of renewable energy sources in fog environment. We first formulated the problem as an offline stochastic optimization problem. Because the offline problem is highly complex to solve, we leveraged Lyapunov optimization technique to provide an online approximation solution. Finally, we came up with an online algorithm as a solution. The proposed algorithm is based on the current system conditions and the queues' backlog information and is independent of the system dynamics. It is a low complexity solution and easy-to-implement. To evaluate the efficiency of the proposed algorithm, we performed extensive simulations using real solar irradiation data. The simulations results proved the superiority of the proposed algorithm, in the course of service time, the number of deadline misses and green energy utilization, over baseline schemes such as Rnd and TABS.

References

[1] E. K. Markakis, K. Karras, N. Zotos, A. Sideris, T. Moysiadis, A. Corsaro, G. Alexiou, C. Skianis, G. Mastorakis, C. X. Mavromoustakis *et al.*, "Exegesis: Extreme edge resource harvesting for a virtualized fog environment," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 173–179, 2017.

[2] statista. Internet of things (iot) connected devices installed base worldwide from 2015 to 2025 (in billions). [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[3] S. K. Mishra, D. Puthal, J. J. Rodrigues, B. Sahoo, and E. Dutkiewicz, "Sustainable service allocation using a metaheuristic technique in a fog server for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4497–4506, 2018.

[4] C.-M. Wu, R.-S. Chang, and H.-Y. Chan, "A green energy-efficient scheduling algorithm using the dvfs technique for cloud datacenters," *Future Generation Computer Systems*, vol. 37, pp. 141–147, 2014.

[5] Y. Yang, S. Zhao, W. Zhang, Y. Chen, X. Luo, and J. Wang, "Debts: Delay energy balanced task scheduling in homogeneous fog networks," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2094–2106, 2018.

[6] F. Jalali, S. Khodadustan, C. Gray, K. Hinton, and F. Suits, "Greening iot with fog: A survey," in *2017 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 2017, pp. 25–31.

[7] Ieee computer society predicts the future of tech: Top 10 technology trends for 2019. [Online]. Available: https://www.computer.org/press-room/2018-news/ieee-cs-top-technology-trends-2019

[8] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755–768, 2012.

[9] B. Zhang, R. Simon, and H. Aydin, "Energy management for time-critical energy harvesting wireless sensor networks," in *Symposium on Self-Stabilizing Systems*. Springer, 2010, pp. 236–251.

[10] M. E. Gerards, J. L. Hurink, and J. Kuper, "On the interplay between global dvfs and scheduling tasks with precedence constraints," *IEEE Transactions on Computers*, vol. 64, no. 6, pp. 1742–1754, 2014.

[11] J. Kwak, Y. Kim, J. Lee, and S. Chong, "Dream: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2510–2523, 2015.

[12] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[13] 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.

[14] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2017.

[15] S. Bitam, S. Zeadally, and A. Mellouk, "Fog computing job scheduling optimization based on bees swarm," *Enterprise Information Systems*, vol. 12, no. 4, pp. 373–397, 2018.

[16] A. Brogi and S. Forti, "Qos-aware deployment of iot applications through the fog," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1185–1192, 2017.

[17] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource allocation strategy in fog computing based on priced timed petri nets," *ieee internet of things journal*, vol. 4, no. 5, pp. 1216–1228, 2017.

[18] W. Li, T. Yang, F. C. Delicato, P. F. Pires, Z. Tari, S. U. Khan, and A. Y. Zomaya, "On enabling sustainable edge computing with renewable energy resources," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 94–101, 2018.

[19] Y. Li, A.-C. Orgerie, I. Rodero, M. Parashar, and J.-M. Menaud, "Leveraging renewable energy in edge clouds for data stream analysis in iot," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. IEEE, 2017, pp. 186–195.

[20] P. G. V. Naranjo, E. Baccarelli, and M. Scarpiniti, "Design and energy-efficient resource management of virtualized networked fog architectures for the real-time support of iot applications," *The Journal of Supercomputing*, vol. 74, no. 6, pp. 2470–2507, 2018.

[21] X. Meng, W. Wang, and Z. Zhang, "Delay-constrained hybrid computation offloading with cloud and fog computing," *IEEE Access*, vol. 5, pp. 21 355–21 367, 2017.

[22] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283–294, 2017.

[23] Y. Li, A.-C. Orgerie, and J.-M. Menaud, "Opportunistic scheduling in clouds partially powered by green energy," in *2015 IEEE International Conference on Data Science and Data Intensive Systems*. IEEE, 2015, pp. 448–455.

[24] N. Beldiceanu, B. D. Feris, P. Gravey, S. Hasan, C. Jard, T. Ledoux, Y. Li, D. Lime, G. Madi-Wamba, J.-M. Menaud *et al.*, "Towards energy-proportional clouds partially powered by renewable energy," *Computing*, vol. 99, no. 1, pp. 3–22, 2017.

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

[26] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.

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

[28] S. Zhao, Y. Yang, Z. Shao, X. Yang, H. Qian, and C.-X. Wang, "Femos: Fog-enabled multitier operations scheduling in dynamic wireless networks," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1169–1183, 2018.

[29] H. Zhang, Z. Chen, J. Wu, Y. Deng, Y. Xiao, K. Liu, and M. Li, "Energy-efficient online resource management and allocation optimization in multi-user multi-task mobile-edge computing systems with hybrid energy harvesting," *Sensors*, vol. 18, no. 9, p. 3140, 2018.

[30] Í. Goiri, K. Le, M. E. Haque, R. Beauchea, T. D. Nguyen, J. Guitart, J. Torres, and R. Bianchini, "Greenslot: scheduling energy consumption in green datacenters," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, 2011, pp. 1–11.

[31] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[32] Bureau of meteorology. [Online]. Available: http://www.bom.gov.au/climate/how/newproducts/IDCJAD0111.shtml

[33] Photovoltaic education network. [Online]. Available: http://pveducation.org/

[34] I. Crop, "Intel pxa270 processor electrical mechanical, and thermal specification," 2006.

[35] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing iot service delay via fog offloading," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 998–1010, 2018.

Appendix A
Proof of Proposition 1

Proof: Squaring both sides of (1), and doing some manipulations yields:

$$\frac{Q_i(t+1)^2 - Q_i(t)^2}{2} \leq \frac{B_i(t)^2 + R_i(t)^2}{2} - \tilde{B}_i(t)R_i(t) - Q_i(t)[B_i(t) - R_i(t)], \tag{15}$$

where $\tilde{B}_i(t) = \min[Q_i(t), B_i(t)]$.

Taking conditional expectation from both sides of (16) with respect to $\boldsymbol{Q}(t)$ and then summing over $i \in \{1, 2, \ldots, N+1\}$ yields the following upper bound for $\Delta(\boldsymbol{Q}(t))$,

$$\Delta(\boldsymbol{Q(t)}) \leq \Upsilon - \sum_{i=1}^{N+1} Q_i(t)B_i(t) + \sum_{i=1}^{N+1} Q_i(t)\mathbb{E}\{R_i(t)|\boldsymbol{Q}(t)\}, \tag{16}$$

where $\Upsilon \geq \frac{B_i(t)^2 + R_i(t)^2}{2} + \sum_{i=1}^{N+1} \mathbb{E}\{\tilde{B}_i(t)R_i(t)|\boldsymbol{Q}(t)\}$.

By adding $\omega\mathbb{E}\{\psi(t)|\boldsymbol{Q}(t)\}$ to both sides of (17), the result is proved.

Appendix B
Proof of Proposition 2

Proof: From (11), we can write,

$$G_i(t+1) - G_i(t) \geq y_i(t). \tag{17}$$

Considering any t_1 and t_2 such that $0 \leq t_1 < t_2$, summing both side of (18) and use of telescoping sums, we have:

$$G_i(t_2) - G_i(t_1) \geq \sum_{\tau=t_1}^{t_2-1} y_i(t). \tag{18}$$

By substituting $t_1 = 0$ and $t_2 = t$, then dividing both sides by t, we have:

$$\frac{G_i(t_2)}{t} - \frac{G_i(t_1)}{t} \geq \frac{1}{t}\sum_{\tau=t_1}^{t_2-1} y_i(t). \tag{19}$$

Taking expectation from both sides of (20) and taking $limsup$ while $t \to \infty$ yields:

$$\limsup_{t\to\infty} \frac{\mathbb{E}\{G_i(t)\}}{t} \geq \limsup_{t\to\infty} \bar{y}_i(t). \tag{20}$$

The stability of $G_i(t)$ induces that the left-hand-side of (21) must be equal to Zero, so we have:

$$\limsup_{t\to\infty} \bar{y}_i(t) \leq 0. \tag{21}$$

which shows that the average constraint $\hat{y}(t)$ is satisfied.