IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 20, 2017, accepted October 4, 2017, date of publication October 24, 2017, date of current version November 14, 2017.

*Digital Object Identifier 10.1109/ACCESS.2017.2766165*

INVITED PAPER

# Adaptive Energy-Aware Computation Offloading for Cloud of Things Systems

**YUCEN NAN[1], (Student Member, IEEE), WEI LI[1], (Senior Member, IEEE), WEI BAO[1], (Member, IEEE), FLAVIA C. DELICATO[2], PAULO F. PIRES[2], YONG DOU[3], AND ALBERT Y. ZOMAYA[1], (Fellow, IEEE)**

[1]Centre for Distributed and High Performance Computing, School of Information Technologies, The University of Sydney, NSW 2006, Australia
[2]Department of Computer Science, Federal University of Rio de Janeiro, Rio de Janeiro 20001-970, Brazil
[3]College of Computer Science, University of Defense Technology, Changsha 410073, China

Corresponding author: Yucen Nan (ynan2995@uni.sydney.edu.au)

The work of W. Li was supported in part by the Faculty of Engineering and IT Early Career Researcher scheme, The University of Sydney, and in part by the Faculty of Engineering Information Technologies, The University of Sydney, under the Faculty Research Cluster Program. The work of W. Bao was supported by The University of Sydney DVC Research/Bridging Support. The work of A. Y. Zomaya was supported by an Australian Research Council Discovery under Grant DP130104591. The work F. C. Delicato and P. F. Pires was supported by CNPq Fellowship. The work of Y. Dou was supported by the Natural Science Foundation of China under Grant U1435219.

**ABSTRACT** Cloud computing has become the de facto computing platform for application processing in the era of the Internet of Things (IoT). However, limitations of the cloud model, such as the high transmission latency and high costs are giving birth to a new computing paradigm called edge computing (a.k.a fog computing). Fog computing aims to move the data processing close to the network edge so as to reduce Internet traffic. However, since the servers at the fog layer are not as powerful as the ones in the cloud, there is a need to balance the data processing in between the fog and the cloud. Moreover, besides the data offloading issue, the energy efficiency of fog computing nodes has become an increasing concern. Densely deployed fog nodes are a major source of carbon footprint in IoT systems. To reduce the usage of the brown energy resources (e.g. powered by energy produced through fossil fuels), green energy is an alternative option. In this paper, we propose employing dual energy sources for supporting the fog nodes, where solar power is the primary energy supply and grid power is the backup supply. Based on that, we present a comprehensive analytic framework for incorporating green energy sources to support the running of IoT and fog computing-based systems, and to handle the tradeoff in terms of average response time, average monetary, and energy costs in the IoT. This paper describes an online algorithm, Lyapunov optimization on time and energy cost (LOTEC), based on the technique of Lyapunov optimization. LOTEC is a quantified near optimal solution and is able to make control decision on application offloading by adjusting the two-way tradeoff between average response time and average cost. We evaluate the performance of our proposed algorithm by a number of experiments. Rigorous analysis and simulations have demonstrated its performance.

**INDEX TERMS** Internet of things, fog computing, Lyapunov optimization, green energy.

## I. INTRODUCTION

In the past few years, the emergence of a pervasive and ubiquitous computing paradigm - Internet of Things (IoT) and its applications generate an extensive amount of data. However, the limited computing capability and energy resources of IoT devices prevents the processing of such data by the devices themselves. Therefore, it is natural to utilize the on-demand resourceful cloud computing paradigm to process the data generated from IoT devices. Thus, a new paradigm called cloud of things (CoT) has emerged, which integrates IoT and Cloud Computing in the development of the IoT systems. Such systems comprise of two tiers, the front tier encompassing the Things and the back tier encompassing the Cloud devices. Despite the obvious advantages of using cloud devices to process IoT data, CoT systems still suffer to deliver the needed QoS for delay sensitive IoT applications. The unpredictable communication delay from the Things tier to the Cloud tier become a high risk factor for these applications. Besides, the network bandwidth could be over-utilized when transferring the generated data of IoT applications to the Cloud tier, despite the fact that most of such applications can be processed locally.

As a remedy to the above limitations, a promising computing paradigm called Edge Computing has recently been advocated. The key idea behind Edge Computing is the introduction of an intermediate tier with the data processing capability in between the above two-tiers of CoT systems. Since the edge tier is situated close to the Things tier, the data processing at suhc tier can significantly reduce the communication delay, shortening the response time of IoT applications and generating less traffic over Internet. Fog Computing [1] as a representative paradigm of Edge Computing keeps the data processing close to end users to reduce the communication delay over the Internet and minimize the bandwidth burden by not fully offloading the generated data to the cloud.

Apart from the data offloading issue, the energy-efficiency of fog computing paradigm has become an increasing concern for researchers. This is due to providing reliable grid power supply in some remote areas can be extremely costly or even infeasible [2]. Furthermore, to accommodate the growing demands for ubiquitous information access, the access points and fog computing nodes are becoming increasingly densely deployed. Thus, it is without surprise that the energy consumption of these devices becomes a major portion (60% - 80%) of the energy consumption of CoT systems [3], generating a major source of carbon footprint. In order to reduce the usage of the brown energy resources (e.g. powered by energy produced through fossil fuels), green energy (e.g. powered by solar, wind or geothermal) is embraced as an alternative energy resource to use while reducing the threats of global climate change. The green energy sources can also provide local and affordable energy for urban and rural communities. In spite of the fact that there are lots of advantages of using green energy, the disadvantage of such energy sources is that the energy conversion rate is relative low, and some of the sources are tightly correlated to the weather conditions, so they could vary dramatically leading to a highly unpredictable power generation. To tackle these issues, we employ dual energy sources for supporting the running of the fog computing paradigm in our system, where solar is used as our primary energy supply and grid power as the backup energy supply. The grid power is also used as the primary energy source to support the running of the Cloud data centres.

In this paper, we present a comprehensive analytic framework for not just incorporating green energy sources to support the running of IoT and fog computing-based system, but also enabling an energy-efficient data offloading mechanism to make sure the long-term system cost (measured by the money spending on energy consumption) is minimized and the users would not experience a poor quality of service. In general, such problem can be converted into a constrained stochastic optimization problem. Using the technique of Lyapunov optimization [4], we designed an adaptive decision-making algorithm called LOTEC (Lyapunov Optimization on Time and Energy Cost) for distributing the incoming applications to the corresponding tiers without a priori knowledge of the status of users and system. The main goal of this algorithm is to ensure the data is processed within a certain amount of time meanwhile the availability of the fog servers is still guaranteed, and the cost (measured by the money spending on using grid power) of the whole system is minimized. In addition, our proposed algorithm achieves the average response time arbitrarily close to the theoretical optimum. The performed discrete-event simulation demonstrates that, under this CoT scenario, our proposed framework outperforms the selected benchmarks and provides the best overall performance for the system users.

The rest of the paper is organized as follows: Section 2 introduces the system model including two dimensions: workload model and energy model. Section 3 presents the theoretical solution and the proposed algorithm. Section 4 evaluates the performance of the proposed algorithm. Related work is provided in Section 5, followed by the final remarks in Section 6.

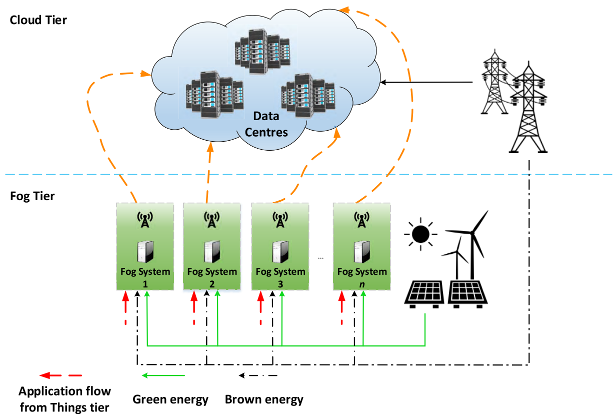


**FIGURE 1.** The system architecture of the CoT.

## II. SYSTEM MODELS

### A. THE OVERVIEW OF THE THREE-TIER CLOUD OF THINGS

As shown in Fig. 1, our CoT system mainly focuses on two tiers in this paper, namely Fog tier and Cloud tier. The Things tier is assumed to be our data source in this study, so it is not explicitly included in the figure. The interested readers are referred to our previous works [5] [6] for further information.

The Fog tier encompasses many distributed fog systems, each one associated with IP addresses from different subnets. Within each fog system, we consider the exisence of a server and a gateway, which are physically co-located. The fog server receives and processes data generated by the IoT devices within the same subnet. The number of newly arrival applications in a unit time slot (termed as arrival rate) may be higher than the processing capacity of the fog server (termed as service rate). Therefore, this will lead to an increment of queue length or even an excess of queue length in the fog server. To avoid overload in fog server,

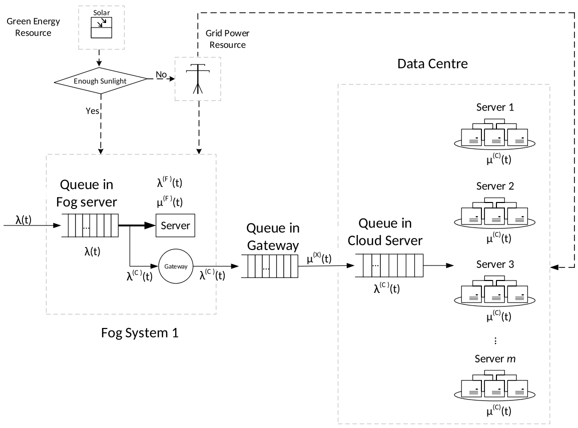


**FIGURE 2.** Application processing flow.

the applications can also be processed in the Cloud tier that comprises typical (macro) data centres. Unlike directly sending data generated by IoT devices to the fog server, some applications on the fog server will then be sent to the Cloud tier via the gateway for their processing. The transmission delay from the fog server to the gateway is negligible due to the physical proximity. The service rates of the servers located in the cloud tier are generally higher than the fog servers, but the data transmission from fog to cloud causes extra delay and power consumption. In our study, we assume that the Cloud is completely supported by the grid power with certain monetary cost while the fog server can be partially supported by the green energy without any monetary cost. Fig. 2 shows the application processing flow in our system. For the sake of the simplicity of analysis, we use only one fog system to demonstrate how our proposed algorithm works, but the approach also applies to the case of multiple fog systems. Please note that, the cooperation between fog systems is out of the scope of this paper. In Table 1, we provide the definitions of the important parameters used in our paper.

### B. WORKLOAD MODEL

We consider a discrete-time model by dividing the operation period into coarse-grained time slots of equal length indexed by $t = 0, 1, \ldots$, in accordance with the length of the long-term-ahead grid market, e.g., days or hours.

#### 1) FOG SERVER IN FOG SYSTEM

During the $t$th time slot, the arrival rate of applications in the fog system is $\lambda(t)$ and the service rate of the server in this fog system is $\mu^{(F)}(t)$. Since the fog server has limited computation capacity, $\lambda(t)$ may be greater than $\mu^{(F)}(t)$, we allow a portion of the applications to be offloaded to the Cloud tier. Let $\eta^{(F)}(t)$ denote the portion of the applications processed locally, and $\eta^{(C)}(t)$ denote the portion offloaded to the cloud server. Let $\lambda^{(F)}(t) = \eta^{(F)}(t)\lambda(t)$ denote the equivalent local arrival rate of the applications at the server in the fog system, and the remaining applications which denoted as $\lambda^{(C)}(t)$ will be offloaded to the cloud server for processing

**TABLE 1.** Key parameters.

| Notation | Definition |
|---|---|
| $t$ | The $t$th time slot for a long-term period. |
| $\lambda(t)$ | Arrival rate of applications at the fog system at $t$. |
| $\lambda^{(F)}(t)$ | Arrival rate of applications processing in the fog server at $t$. |
| $\lambda^{(C)}(t)$ | Arrival rate of applications in cloud servers at $t$. |
| $\mu^{(F)}(t)$ | Service rate of the fog server at $t$. |
| $\mu^{(X)}(t)$ | Transmission rate of the gateway at $t$. |
| $\mu^{(C)}(t)$ | Service rate of the cloud server at $t$. |
| $\tau^{(F)}(t)$ | Average response time of processing applications in the fog server at $t$. |
| $\tau^{(X)}(t)$ | Average response time of transmitting applications from fog system to the Cloud tier at $t$. |
| $\tau^{(C)}(t)$ | Average response time of processing applications in cloud server at $t$. |
| $\tau^{'}(t)$ | A propagation delay for transmitting via gateway at $t$. |
| $\overline{\tau}(t)$ | Overall average response time of processing applications at $t$. |
| $S(t)$ | Number of running server in the Cloud tier at $t$. |
| $\omega_s^{(F)}$ | The unit power consumption for the fog server. |
| $\omega_b^{(F)}$ | The unit power consumption for the gateway in the fog system. |
| $\omega^{(C)}$ | The unit power consumption for the cloud server. |
| $E_{in}^{F}(t)$ | The total volume of green energy obtained at $t$. |
| $E_{out}^{F}(t)$ | The total energy consumption in the fog system at $t$. |
| $p_e$ | The unit price of the electricity produced by gird system. |
| $M(t)$ | The total cost of the whole system at $t$. |
| $D$ | Required mean response time |

after transmitting via gateway. We have

$$\lambda(t) = \lambda^{(F)}(t) + \lambda^{(C)}(t). \tag{1}$$

We model the server in the fog system as an $M/M/1$ queue [7]. According to the Little's Law, the average local response time is

$$\tau^{(F)}(t) = \frac{1}{\mu^{(F)}(t) - \lambda^{(F)}(t)}, \tag{2}$$

note that we must satisfy

$$\mu^{(F)}(t) > \lambda^{(F)}(t). \tag{3}$$

#### 2) TRANSMISSION FROM FOG TIER TO CLOUD TIER

As we discussed before, the transmission delay from the fog server to the gateway can be ignored due to the physical proximity. During the $t$th time slot, the arrival rate to the gateway equals to the arrival rate of the Cloud, which is denoted as $\lambda^{(c)}(t)$. The transmission rate in communication channel is $\mu^{(X)}(t)$. We model the process of transmission in gateway as an M/M/1 queue, and the average transmission time from the gateway to the cloud can be denoted as:

$$\tau^{(X)}(t) = \frac{1}{\mu^{(X)}(t) - \lambda^{(C)}(t)}, \tag{4}$$

note that we must satisfy

$$\mu^{(X)}(t) < \lambda^{(C)}(t). \tag{5}$$

Furthermore, if the gateway is used, there will be a propagation delay $\tau^{'}(t)$. In summary, if the application will be sent to the Cloud, the average response time for sending applications to the cloud server from the fog system is

$$\tau^{(X)}(t) = \tau^{(X)}(t) + \tau^{'}(t). \tag{6}$$

#### 3) SERVERS IN THE CLOUD TIER

In this paper, we assume that the Cloud data centre is able to provide auto-scale services to the users, which means the number of cloud servers can be automatically varied by monitoring some selected metrics in the centre. In this study, we choose the length of waiting queue in the Cloud data centre as the metric to monitor, which can be denoted as $\gamma(t)$. Correspondingly, the threshold of scaling up or down the numbers of servers can be denoted by $\gamma_{up}$ and $\gamma_{down}$. After that, the action of scaling can be formulated as following:

$$f(t) = \begin{cases} y & \gamma(t) \geq \gamma_{up} \\ 0 & \gamma_{down} \leq \gamma(t) \leq \gamma_{up} \\ -z & \gamma(t) < \gamma_{down}, \end{cases}$$

where the number of the servers can be varied once scaling up or down is triggered, which can be denoted as $y$ and $z$ respectively. When $\gamma(t)$ exceeds the $\gamma_{up}$, the cloud data centre will use $y$ more servers to process applications. On the contrary, when $\gamma(t)$ drops below $\gamma_{down}$, the cloud data centre will switch off $z$ running servers. Moreover, when $\gamma(t)$ lies in the range of $[\gamma_{dowm}, \gamma_{up}]$, the number of servers in the cloud data centre remains the same. As mentioned before, the number of the running servers in the cloud data centre during the $t$th time slot can be denoted as following:

$$S(t) = S(t-1) + f(t-1). \tag{7}$$

The arrival rate to a cloud server is $\lambda^{(C)}(t)$. The service rate of a cloud server is $\mu^{(C)}(t)$. Again, an M/M/1 queue is used to evaluate the average response time in the cloud data center, and can be denoted as

$$\tau^{(C)}(t) = \frac{1}{S(t)\mu^{(C)}(t) - \lambda^{(C)}(t)}, \tag{8}$$

note that we must have

$$S(t)\mu^{(C)}(t) > \lambda^{(C)}(t). \tag{9}$$

### C. ENERGY COST MODEL

In order to save money as much as possible and try to be environmental friendly, we use solar as our primary green energy source to supply the fog system. In a given time slot, the volume of electricity converted from solar energy can be denoted as $E_{in}^{(F)}(t)$, which is a stochastic value depending on weather conditions. We assume that no monetary cost will be raised by using the electricity converted from the green energy. However, the green energy is unpredictable and unstable. In the fog system, if the demand of the electricity exceeds that green energy can supply, we need to use traditional grid power to maintain the system functions properly. According to the well-known physical formulations, we can derive the energy consumed in fog system as following:

$$E_{\mu_1}^{(F)}(t) = \tau^{(F)}(t)\omega_s^{(F)}, \tag{10}$$

$$E_{\mu_2}^{(F)}(t) = \tau^{(X)}(t)\omega_b^{(F)}, \tag{11}$$

where $E_{\mu_1}^{(F)}(t)$ is the energy demand of the fog server and $\omega_s^{(F)}$ is the unit power consumption of the fog server. Similarly, $E_{\mu_2}^{(F)}(t)$ denote the energy demand of the gateway and $\omega_b^{(F)}$ is the unit power consumption of the gateway. Then, we can obtain the total energy consumption in the fog system:

$$E_{out}(t) = E_{\mu_1}^{(F)}(t) + E_{\mu_2}^{(F)}(t). \tag{12}$$

Regarding the cost of using grid power, the price is based on time period, which can be divided into: peak hour, shoulder hour and off-peak hour [8]. If we use grid power in the peak hour, the price of electricity can be denoted as $p_{onp}$, and the price of shoulder hour and off-peak hour can be denoted as $p_{shd}$ and $p_{ofp}$ respectively.

$$p_e(x) = \begin{cases} p_{onp} & peak \\ p_{shd} & shoulder \\ p_{ofp} & off-peak, \end{cases}$$

note that, in this paper, we must satisfy

$$p_{onp} > p_{shd} > p_{ofp}. \tag{13}$$

By multiplying the unit price of the electricity in different time period and the amount of electricity that exceeds the green energy can support, we obtain the cost of running the fog system:

$$M^{(F)}(t) = p_e max\left[E_{out}(t) - E_{in}(t), 0\right]. \tag{14}$$

On the contrary, the grid power is used to support the servers in the Cloud. Then we can obtain the cost of running the cloud servers accordingly:

$$M^{(C)}(t) = p_e S(t)\tau^{(C)}(t)\omega^{(C)}. \tag{15}$$

### D. DATA OFFLOADING FOR COST-EFFECTIVE PROCESSING

In order to balance the response time and cost, as we introduced previously, we presume that there is no cost if the applications are processed in the fog system by using green energy. However, as discussed in Section II-C, if there is not enough green energy to use, the applications will be either processed in the Fog system with grid power or sent out to the Cloud for their processing. Moreover, if the applications are sent to the Cloud, it will introduce transmission cost by using grid power. The overall cost at $t$ is

$$\begin{aligned} M(t) &= M^{(F)}(t) + M^{(C)}(t) \\ &= p_e\left(\tau^{(F)}(t)\omega_s^{(F)} + \tau^{(X)}(t)\omega_b^{(F)} - E_{in}(t)\right) \\ &\quad + p_e\left(\tau^{(C)}(t)\omega^{(C)}\right) \end{aligned} \tag{16}$$

On the other hand, as discussed in Section II-B, the total average response time at $t$ is

$$\begin{aligned} \overline{\tau}(t) = &\frac{\lambda^{(F)}(t)}{\lambda(t)}\tau^{(F)}(t) \\ &+ \frac{\lambda^{(C)}(t)}{\lambda(t)}\left[\tau^{(X)}(t) + \tau^{(C)}(t)\right]. \end{aligned} \tag{17}$$

Then our aim is to minimize the long-term average cost

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[M(t)\right] \tag{18}$$

and satisfy the following requirements about response time and energy consumption:

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\overline{\tau}(t)\right]\leq D. \tag{19}$$

## III. PROBLEM SOLUTION THROUGH LYAPUNOV OPTIMIZATION

In order to satisfy the response time constraint (19), we need to introduce a virtual queue $Q(t)$, which are used to accumulate the part of the response time that exceeds the expected finish time, and we define $Q(0) = 0$. $Q(t)$ evolves as follows

$$Q(t+1)=\max[Q(t)-D,0]+\overline{\tau}(t) \tag{20}$$

*Lemma 1:* If $Q(t)$ is mean rate stable [4], then (19) is satisfied.

### A. LYAPUNOV OPTIMIZATION FORMULATION

Next, we define a Lyapunov function as a scalar measure of response time and energy consumption in the system as follows

$$L\left(Q(t)\right)\triangleq\frac{1}{2}Q^2(t), \tag{21}$$

Then we define the conditional unit-slot Lyapunov drift as follows

$$\Delta\left(Q(t)\right)=L\left(Q(t+1)\right)-L\left(Q(t)\right). \tag{22}$$

### B. BOUNDING UNIT-SLOT LYAPUNOV DRIFT

Our primary aim is to optimize the upper bound of the overall response time which is the upper bound of $\Delta\left(Q(t)\right)$.

*Lemma 2:* For any $t\in\{0, T-1\}$, given any possible control decision, the Lyapunov drift $\Delta\left(Q(t)\right)$ can be deterministically bounded [4] as follows

$$\Delta(Q(t))\leq H+Q(t)\mathbb{E}\left[-D+\overline{\tau}(t)|Q(t)\right], \tag{23}$$

where $H=\frac{1}{2}\left[\max\mathbb{E}\left(\overline{\tau}(t)^2\right)+D^2\right]$.

### C. MINIMIZING THE DRIFT-PLUS-COST PERFORMANCE

Defining the same Lyapunov function $L\left(Q(t)\right)$ as in (21), and letting $\Delta\left(Q(t)\right)$ represents the conditional Lyapunov drift at $t$. While taking actions to minimize a bound on $\Delta\left(Q(t)\right)$ every time slot would stabilize the system, the resulting cost might be unnecessarily large. In order to avoid this, we minimize a bound on the following drift-plus-penalty expression instead of minimizing a bound on $\Delta\left(Q(t)\right)$

$$\Delta\left(Q(t)\right)+V\mathbb{E}\left[M(t)|Q(t)\right], \tag{24}$$

where $V\geq 0$ is a parameter that represents an "important weight" on how much we emphasize cost minimization. We add a penalty term to both sides of (23), yielding a bound on the drift-plus-penalty

$$\begin{aligned}\Delta\left(Q(t)\right)&+V\mathbb{E}\left[M(t)|Q(t)\right]\\ &\leq H+\mathbb{E}\left[Q(t)\overline{\tau}(t)|Q(t)\right]\\ &\quad-DQ(t)+V\mathbb{E}\left[M(t)|Q(t)\right].\end{aligned} \tag{25}$$

At each time slot, we are motivated to minimize the following term.

$$\min_{\lambda^{(F)}(t),\lambda^{(C)}(t)}\mathbb{E}\left[Q(t)\overline{\tau}(t)|Q(t)\right]+V\mathbb{E}\left[M(t)|Q(t)\right], \tag{26}$$

substituting (26), we have the following one-time slot optimization problem $C1$. (26) is minimized if we opportunistically minimize (27) as follows at each step [4].

$$\begin{aligned}\min_{\lambda^{(F)}(t),\lambda^{(C)}(t)} &Q(t)\left(\frac{\lambda^{(F)}(t)}{\lambda(t)}\tau^{(F)}(t)+\frac{\lambda^{(C)}(t)}{\lambda(t)}\left(\tau^{(X)}(t)+\tau^{(C)}(t)\right)\right)\\ &+Vp_e\left(\tau^{(F)}(t)\omega_s^{(F)}+\tau^{(X)}(t)\omega_b^{(F)}-E_{in}+\tau^{(C)}(t)\omega^{(C)}\right),\end{aligned} \tag{27}$$

$$\text{subject to } \lambda^{(F)}(t)\leq\mu^{(F)}(t), \tag{28}$$

$$\lambda^{(C)}(t)\leq\min(\mu^{(C)}(t),\mu^{(X)}(t)). \tag{29}$$

In the next subsection, we show that if (26) is minimized at each time slot, we can achieve quantified near optimal solution.

### D. OPTIMALITY ANALYSIS

Let † denote any S-only offloading policy[1], and $\overline{\tau}^{\dagger}(t)$ and $M^{\dagger}(t)$ denote the average response time and average cost at $t$ based on policy †.

$$\begin{aligned}&\Delta(Q(t))+V\mathbb{E}\left[M(t)|Q(t)\right]\\ &\quad\leq qH+Q(t)\mathbb{E}\left[\overline{\tau}(t)-D|Q(t)\right]+V\mathbb{E}\left[M(t)|Q(t)\right] &(30)\\ &\quad\leq H+Q(t)\mathbb{E}\left[\overline{\tau}^{\dagger}(t)-D|Q(t)\right]+V\mathbb{E}\left[M^{\dagger}(t)|Q(t)\right] &(31)\\ &\quad= H+Q(t)\mathbb{E}\left[\overline{\tau}^{\dagger}(t)-D\right]+V\mathbb{E}\left[M^{\dagger}(t)\right]. &(32)\end{aligned}$$

Now we assume that there exists $\delta > 0$ such that $\mathbb{E}\left[\overline{\tau}^{\dagger}(t)\right]\leq D-\delta$ can be achieved by an S-only policy [9], and among all feasible S-only policies, $\overline{M}^{*}(\delta)$ is the optimal average cost. We have

$$(31)\leq H-Q(t)\delta+V\overline{M}^{*}(\delta). \tag{33}$$

Taking expectations of (33), we have

$$\begin{aligned}\sum_{t=0}^{T-1}&\mathbb{E}\left[\Delta(Q(t))\right]+V\sum_{t=0}^{T-1}\mathbb{E}\left[\mathbb{E}(M(t)|Q(t))\right]\\ &\leq TH-\sum_{t=0}^{T-1}\mathbb{E}\left[Q(t)\right]\delta+VT\overline{M}^{*}(\delta).\end{aligned}$$

[1] S-only offloading policy means that the decision on $\lambda^{(F)}(t),\lambda^{(C)}(t)$ depends only on the system state at $t$ (i.e., $\lambda(t),\mu^{(F)}(t),\mu^{(X)}(t),\mu^{(C)}(t),\tau(t),M(t)$), but does not depend on $Q(t)$.

Using the law of iterated expectations as before yield and summing the above over $t \in [0, T-1]$ for some positive integer T yield, we have

$$\mathbb{E}[L(Q(T))] - \mathbb{E}[L(Q(0))] + V\sum_{t=0}^{T-1}\mathbb{E}[M(t)] \leq TH - \sum_{t=0}^{T-1}\mathbb{E}[Q(t)]\,\delta + VT\overline{M}^{*}(\delta). \quad (34)$$

Rearranging the terms in the above and neglecting non-negative quantities where appropriate yields the following two inequalities

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[Q(t)] \leq \frac{H}{\delta} + \frac{V\overline{M}^{*}(\delta) - \frac{V}{T}\sum_{t=0}^{T-1}\mathbb{E}[M(t)]}{\delta} + \frac{\mathbb{E}[L(Q(0))]}{T\delta} \quad (35)$$

and

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[M(t)] \leq \frac{H}{V} + \overline{M}^{*}(\delta) + \frac{\mathbb{E}[L(Q(0))]}{VT}, \quad (36)$$

where the first inequality follows by dividing (34) by $VT$ and the second follows by dividing (34) by $T\delta$. Taking limits as $T \to \infty$ shows that

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[Q(t)] \leq \frac{H + V\left[\overline{M}^{*}(\delta) - \overline{M}^{*}\right]}{\delta}, \quad (37)$$

where $\overline{M}^{*}$ is the optimal long-term average cost achieved by any policy. We also have

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{T_k=0}^{T-1}\mathbb{E}[M(t)] \leq \frac{H}{V} + \overline{M}^{*}(\delta). \quad (38)$$

The bounds (37) and (38) demonstrate an $[O(V), O(1/V)]$ tradeoff between average response time and average cost. We can use an arbitrarily large $V$ to make $\frac{H}{V}$ arbitrarily small, so that the inequality (38) illustrates that with the increasing of parameter $V$, the money cost is closer to the $M^*$. However, when $V$ is too large, the data queue is not stable, which means the response time will exceeded the predefined system expected finish time and not satisfying the constraint (19) anymore. It is obvious that tuning the parameter $V$ can minimize the response time and money cost at the same time. This comes with a tradeoff: the average response time bound in inequality (37) is $O(V)$.

### E. LOTEC OPTIMIZATION

In this subsection, we need to further solve the problem $P1$ in (27)-(29) and the pseudo code of our proposed algorithm LOTEC is given in Algorithm 1.

**Algorithm 1** LOTEC Optimization Algorithm

**for** each *Application* $\in \mathcal{N}$ **do**
  Calculating (27) under the constraints (28) and (29). Derive the optimal value of (27) by selecting Fog or Cloud respectively, which are denoted by $F_{\text{fog}}$ and $F_{\text{cloud}}$
  **if** $F_{\text{fog}} \leq F_{\text{cloud}}$ **then**
    Set $\mathcal{N}_{\text{fog}} = \mathcal{N}_{\text{fog}} + 1$,
    Set $\lambda^{(F)*} = \lambda^{(F)}(t)$
  **else**
    Set $\mathcal{N}_{\text{cloud}} = \mathcal{N}_{\text{cloud}} + 1$
    Set $\lambda^{(C)*} = \lambda^{(C)}(t)$
  **end if**
**end for**
The optimal solution is $\lambda^{(F)}(t) = \lambda^{(F)*}$, and $\lambda^{(C)}(t) = \lambda^{(C)*}$

## IV. PERFORMANCE EVALUATION

In this section, we first provide the details of the simulations, then we investigate the performance of our proposed algorithm by evaluating the tradeoff between average response time and average cost. After that, we investigate how different amounts of solar energy will impact the system performance. Lastly, we present three selected algorithms implemented as benchmarks to compare with our proposed algorithm.

### A. SIMULATION ENVIRONMENT SETTING

In order to evaluate the performance of our proposed algorithm, we carried out a discrete event simulation (DES) making use of the SimPy [10].

We first established the three-tier CoT system as shown in Fig. 2 in our simulations. There were totally N applications generated by the IoT devices in a time slot and they would be directly sent to the fog server in the fog system, which a Poisson-distributed service rate. Meanwhile, there is a gateway located in the fog system which is used to sent the selected applications to the servers in Cloud tier with a Poisson-distributed transmission rate. To address different workloads on the Cloud, we assumed that the number of servers in the Cloud is scalable depending on the backlog. When the backlog in Cloud tier exceeds the preset upper threshold (set to 100 in our experiment), the number of servers will be scaled up to 4 servers (2 servers is the default value) to speed up the applications processing. However, when the backlog is lower than the lower threshold (set to 50 in our experiment), the number of Cloud server will be scaled down to 1 to save energy. In addition, regarding to our proposed algorithm, there is a penalty term $V$ to show how much we emphasize on the average cost.

In our experiments, we consider each time slot as 1 day (24 hours) and divided it into three parts with different electricity price, namely peak hour (2pm-8pm), shoulder hour (7am-2pm and 8pm-10pm) and off-peak hour (10pm-7am) [8]. In peak hour, the price of traditional grid electricity is set to \$0.22 per kilowatt, the price of shoulder hour is set to \$0.06 per kilowatt, and the price of off-peak hour is set to \$0.02 per kilowatt [11]. Except the traditional grid power, we use solar energy (with no monetary cost) as our primary power supply for the fog system, and the period of sunshine duration is randomly generated in each time slot, ranging from 4 hours to 12 hours.

Since the servers in fog tier are not as powerful as servers in Cloud tier, the unit power consumed by fog server is set to 41.6 Kw/h while power consumption of cloud server is set to 277.8 Kw/h in our experiments. In addition, the unit power consumed by gateway is set to 2.7Kw/h and the solar energy is the first choice to power Fog tier (fog server and gateway).

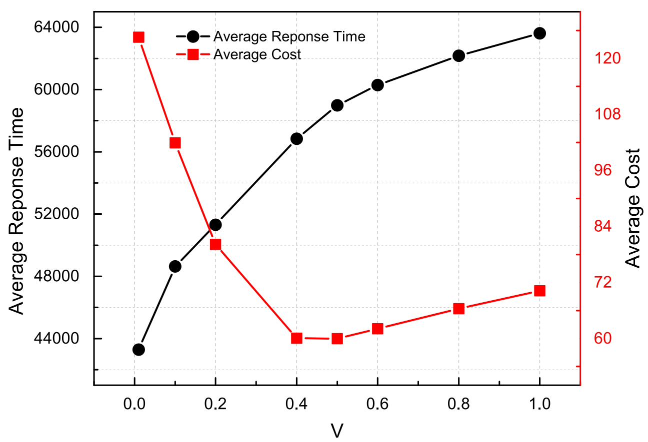


**FIGURE 3.** The influence of penalty term in LOTEC optimization algorithm.

### B. THE IMPACT OF DIFFERENT PENALTY TERMS ON THE PERFORMANCE OF THE PROPOSED ALGORITHM

To investigate the effects of the penalty term V in our proposed algorithm, we varied its value and observed how the average cost and average response time change accordingly. In this experiment, we first fixed the available amount of solar energy and then varied the value of V from 0.01 to 1, which represents the importance of the average cost spending will affect the system performance. As depicted in Fig. 3, by setting the value of V to a small one, we observed that the average response time remains small while the average cost remains large. This is due to the penalty of cost spending is low in the system, so that the applications are preferred to be processed at cloud in order to get a better response time. By setting V to a large value, the significant increment of the average response time can be observed in the figure. However, the change of average cost is not as simple as average response time. Two phrases of the change of the average cost are shown in the figure. When the value of V is lower than 0.5, the average cost is dropping from 125 to around 60. When V keeps increasing from 0.5 to 1, the average cost stops decreasing, instead, it starts increasing slowly. This is because when the green energy is exhausted, the fog server will use brown energy to support its running, but its low service rate results in a longer response time, increasing the monetary cost. It is clear that when V is set to 0.5, the average cost of the system is minimized.

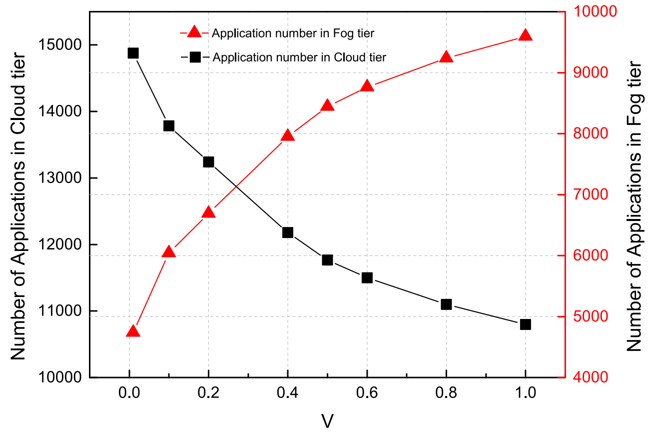


**FIGURE 4.** The influence of penalty term in application offloading.

Apart from studying the impact of the change of V on average cost and average response time, we also study how the application offloading in the entire system is affected. In Fig. 4, the red curve depicts the number of applications in the Fog tier and the black curve depicts the number of applications in the Cloud tier. As shown in the Figure, the number of applications in the Fog tier is increasing along with the value of V is changing from 0.01 to 1. It indicates that we care more about monetary cost, so the application are preferred to be processed in the Fog tier. On the contrary, the number of the applications sent to cloud for their processing is decreasing in the same time since the Cloud tier is fully supported by gird power with more cost. In conclusion, the larger V is given, the more applications will be processed in the Fog tier.

### C. THE IMPACT OF DIFFERENT AMOUNT OF SOLAR ENERGY ON THE PERFORMANCE OF THE LOTEC ALGORITHM

As mentioned before, the usage of solar energy has zero cost and it acts as the primary power supply for the Fog tier. In real world scenarios, the power amount converted from solar energy generally varies in different days, which are caused by different natural environmental factors such as sunshine durations. Therefore, it is also important for us to investigate the system performance to be managed by LOTEC algorithm under different amount of green energy supply. To study the impact of various energy loads on system performance, the V value is kept as a fixed value.

Fig. 5 depicts the changes of application offloading and how their average response time varies accordingly with different amount of solar energy during the fixed length time slots and penalty terms. In this experiment, we sample ten fixed length time slots (24 hours per slot) loaded with different solar energy supply. As shown in the figure, the average response time of our proposed algorithm increases along with the increment of solar energy amount at a moderate rate. This is due to more applications to be processed in the fog tier powered by the cost free solar energy. As a result, the total

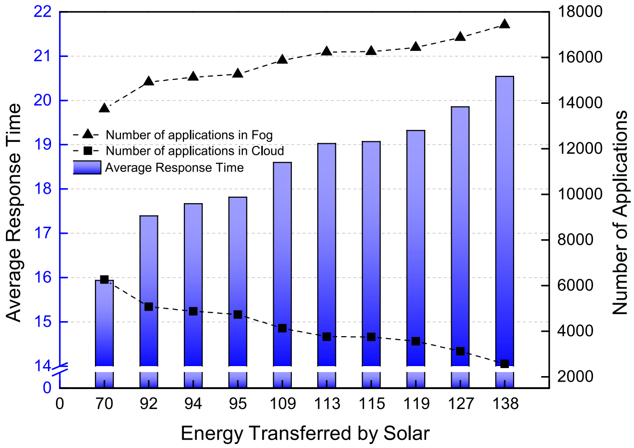


**FIGURE 5.** Average response time and data offloading in different amount of solar energy.

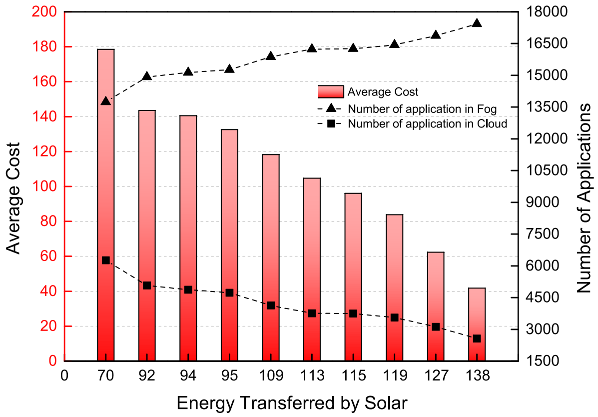


**FIGURE 6.** Average Cost and data offloading in different amount of solar energy.

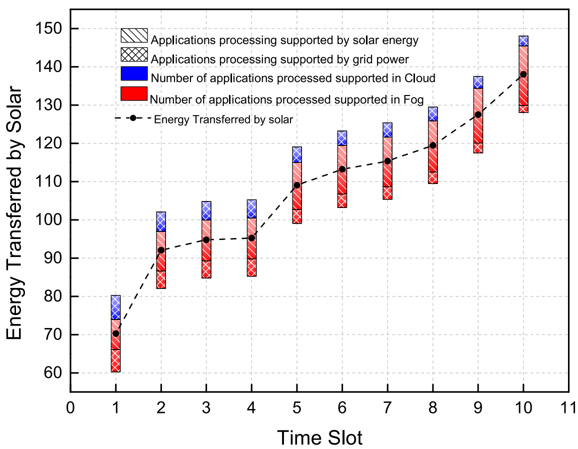


**FIGURE 7.** Offloading proportion in different amount of solar energy.

cost is spontaneously reduced as showed in Fig. 6. This indicates LOTEC is capable of taking full advantages of green energy without significantly increasing the response time.

Fig. 7 detailed the percentage of application offloading in the corresponding ten time slots with increasing amount of solar energy load (shown as the black square in this figure). Each floating bar is composed of three different parts, including the numbers of applications processed at Cloud tier powered solely by grid electricity (in blue, slashed), the numbers of applications processed at Fog tier powered by grid electricity (in red, slashed) and the numbers of applications processed at Fog tier powered by the cost free solar energy (in red, cross slashed). This is a much clearer observation that the LOTEC algorithm can effectively allocate more applications to be processed at the Fog tier powered by the cost free solar energy when the solar energy supply becomes more sufficient.

### *D. PERFORMANCE EVALUATION OF DIFFERENT ALGORITHMS*

To further evaluate the performance of our proposed algorithm, we implemented Fog-Only, Cloud-Only and Round Robin algorithms as our benchmarks. In the Fog-Only algorithm, all newly arrival applications are sent to fog sever for their processing. In the Cloud-Only algorithm, the applications allocation strategy is opposite to the Fog-Only algorithm, where all newly arrival applications are sent to the Cloud for their processing. In the Round Robin algorithm, the odd-indexed applications stay in the fog tier for their processing while the even-indexed applications are sent to the Cloud for their processing.

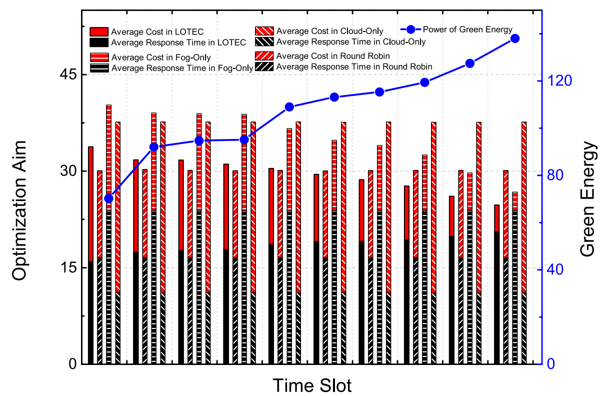


**FIGURE 8.** The performance of processing applications in different algorithms.

Fig. 8, shows the variation of average cost and average response time of these four different algorithms displayed as the upper and lower parts of each bar respectively with increasing amount of solar energy applied. It is obvious that the Fog-Only algorithm delivered the slowest services however cost less than any of the other there algorithms. The Cloud-Only algorithm presents the opposite features with significant less response time but much higher cost than the Fog-Only algorithm. Both two algorithms failed to provide a cost/service trade-off under all ranges of solar energy supplies in the system. It is quite interesting that with low solar energy supply, (e.g. the 1st, 2nd selected time slots), the Round-Robin algorithm showed a good performance with similar response time but lower cost than LOTEC, which we believe this could be caused by the nature of its fair applications distribution. However, as the solar energy supply increases, the superiority of LOTEC emerges, with significant decrease in the cost and well controlled increase in the response time. It is clearly that when the total solar energy supply reaches 109.05 Kwh, the values of the overall optimization aims of LOTEC is already much lower than the Round-Robin one. This indicates that LOTEC has better optimization ability when solar energy is universally applied as a large potion in the power supply system, which is in coincidence of current energy development strategy worldwide.

## V. RELATED WORK

ICT (Information and communication technology) has been recognized as one of the major energy consumers in the world. One of its representatives, cloud datacentres consume a significant amount of the total ICT energy consumption for providing elastic and on-demand ICT services. In many ICT systems designs, energy consumption is not perceived as a critical success factor, and then at some point energy needs become a constraint and thereby degrade the overall system performance. These systems are generally brown-powered (e.g. powered by energy produced through fossil fuels). Such energy production process emits large amount of greenhouse gases emissions and causes significant environmental impact. A number of energy-efficient techniques have thus been proposed to reduce the usage of the carbon intensive energy of ICT systems, while not compromise system robustness and availability.

In recent years, the main focus of developing energy-efficient techniques for ICT systems was on Cloud datacentre since its tremendous amount of energy consumption. The main energy consumers of a datacentre have been identified in [12], and the corresponding energy saving techniques have been well studied. However, the needs [5] of processing the time-sensitive IoT applications and enabling computation intelligence at the network edge drive the emergence of edge computing. The energy-efficiency techniques developed for the edge computing paradigm is still in its infancy age and only a few works address this issue. In [13], the authors compared the energy demands of the cloud computing applications and fog computing applications, and showed that some IoT applications can save more energy if they are run on fog. In [6], the authors studied the resource management issue of cloud of things system with the aim of minimizing the system resource usage (e.g. energy consumption of IoT devices) while meeting the deïĄned QoS requirements. However, no usage of renewable energy is considered in the work. In [14], the authors advocated for leveraging on-site renewable energy production in the different edge computing nodes to enable greener IoT systems while offering improved QoS compared to the core cloud approaches. An analytic model is developed to decide whether to offload a task to the nearby edge node or to the remote cloud datacentres for its processing depending on the renewable energy availability and the desired application QoS.

In dynamic systems, Lyapunov optimization is a promising approach to solve resource allocation, traffic routing, and scheduling problems. It has been applied to the works of traffic routing and scheduling in wireless ad hoc networks [15], [16], but they did not consider processing applications in the processing network environment.

In processing networks, [17] proposed a method on how to offload one application from a mobile device to a cloud datacentre for its processing. [9], [18], [19] studied processing networks with limited numbers of network nodes, while we allow arbitrarily large numbers of things, nodes and datacentres in the system. In [20], the authors focused on studying task processing and offloading in a wired network, but they overlooked to address the potential conflict of performance requirements between them.

## VI. CONCLUSION

In this paper, we studied the problem of providing energy-effective data processing service in a three-tiers Cloud of Things system, and proposed an efficient and effective online algorithm, called LOTEC (Lyapunov Optimization on Time and Energy Cost) for balancing the tradeoff between data processing time and the monetary cost on running the system. Simulation results demonstrated that the our proposed algorithm is promising.

In the future, we will include a more realistic weather forecasting function so as to further improve the performance of our solution. In addition, to investigate how to enable the cooperation between fog systems will be essential to continue exploration of this work.

## APPENDIX A
## PROOF OF THE MEAN RATE STABLE EQUATION

*Proof:*

Due to the definition of $Q(t)$, we have

$$Q(t+1) \geq Q(t) - D + \overline{\tau}(t).$$

Then taking expectation of the above inequality, we have

$$\mathbb{E}\left[Q(t+1)\right] - \mathbb{E}\left[Q(t)\right] \geq -D + \mathbb{E}\left[\overline{\tau}(t)\right].$$

Summing up both sides of the above inequality over $t \in [0, T-1]$ for some positive integer T yields and using the law of iterated, we have

$$\mathbb{E}\left[Q(T)\right] - \mathbb{E}\left[Q(0)\right] \geq -TD + \sum_{t=0}^{T-1} \mathbb{E}\left[\overline{\tau}(t)\right].$$

Then through dividing by $T$, we have

$$\frac{\mathbb{E}\left[Q(T)\right] - \mathbb{E}\left[Q(0)\right]}{T} \geq -D + \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\overline{\tau}(t)\right].$$

Applying $Q(0) = 0$ to the above inequality, we have

$$\frac{\mathbb{E}\left[Q(T)\right]}{T} \geq -D + \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\overline{\tau}(t)\right].$$

Finally, letting $T \to \infty$, we have

$$\limsup_{T\to\infty} \frac{\mathbb{E}\left[Q(T)\right]}{T} \geq -D + \limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\overline{\tau}(t)\right].$$

If $Q(t)$ is mean rate stable, then $\limsup_{T\to\infty} \frac{\mathbb{E}[Q(T)]}{T} = 0$

$$0 \geq -D + \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\overline{\tau}(t)\right].$$

Rearranging terms in the above, we have

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\overline{\tau}(t)\right] \leq D.$$

By introducing the virtual queue $Q(t)$, we are able to convert the constraint (19) into a queue stable problem. If we can guarantee that $Q(t)$ is mean rate stable, we are able to satisfy (19).

■

## APPENDIX B
## PROOF OF THE BOUNDED LYAPUNOV DRIFT EQUATION

*Proof:*

$$\begin{aligned}\Delta\left(Q(t)\right) &= \mathbb{E}\left[L\left(Q(t+1)\right) - L\left(Q(t)\right)|Q(t)\right] \\ &= \frac{1}{2}\mathbb{E}\left[Q^2(t+1) - Q^2(t)|Q(t)\right].\end{aligned}$$

Applying equation (20), we have

$$\Delta(Q(t)) = \frac{1}{2}\mathbb{E}\left[\left[\max\left[Q(t)-D,0\right]+\overline{\tau}(t)\right]^2 - Q(t)^2|Q(t)\right].$$

For any $Q(t) \geq 0$, $D \geq 0$, $\overline{\tau}(t) \geq 0$, we have

$$\begin{aligned}\left[\max[Q(t)-D,0]+\overline{\tau}(t)\right]^2 \leq Q(t)^2 + \overline{\tau}(t)^2 + D^2 \\ + 2Q(t)(\overline{\tau}(t) - D),\end{aligned}$$

then we have

$$\begin{aligned}\Delta(Q(t)) &\leq \frac{1}{2}\mathbb{E}\left[\overline{\tau}(t)^2 + D^2 + 2Q(t)(\overline{\tau}(t)-D)|Q(t)\right] \\ &\leq \mathbb{E}\left[\frac{\overline{\tau}(t)^2 + D^2}{2}|Q(t)\right] \\ &\quad + \mathbb{E}\left[Q(t)\overline{\tau}(t) - Q(t)D|Q(t)\right].\end{aligned}$$

Defining $H$ as a finite constant that bounds the first term on the right-hand-side of the above drift inequality, so that for all $t$, all possible $Q(t)$, we have

$$H = \frac{1}{2}\left[\max \mathbb{E}(\overline{\tau}(t)^2) + D^2\right].$$

■

## REFERENCES

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput. (MCC)*, 2012, pp. 13–16. [Online]. Available: http://doi.acm.org/10.1145/2342509.2342513
[2] C. Li *et al.*, "Towards sustainable *in-situ* server systems in the big data era," *ACM SIGARCH Comput. Archit. News*, vol. 43, no. 3, pp. 14–26, 2015.
[3] L. Chen, S. Zhou, and J. Xu. (Jan. 2017). "Energy efficient mobile edge computing in dense cellular networks." [Online]. Available: https://arxiv.org/abs/1701.07405
[4] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synth. Lectures Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010.
[5] Y. Nan, W. Li, W. Bao, F. Delicato, P. Pires, and A. Zomaya, "Cost-effective processing for delay-sensitive applications in cloud of things systems," in *Proc. IEEE 15th Int. Symp. Netw. Comput. Appl. (NCA)*, Oct. 2016, pp. 162–169.
[6] W. Li *et al.*, "System modelling and performance evaluation of a three-tier Cloud of Things," *Future Generat. Comput. Syst.*, vol. 70, pp. 104–125, May 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X16302047
[7] P. P. Bocharov, C. D'Apice, A. V. Pechinkin, and S. Salerno, *Queueing Theory*. Utrecht, The Netherlands: Academic, 2004. [Online]. Available: https://books.google.com.au/books?id=EG-hYILIwPQC
[8] Ausgrid. *Time of Use Pricing*. Accessed: Jun. 28, 2017. [Online]. Available: http://www.ausgrid.com.au/Common/Customer-Services/Homes/Meters/Time-of-use-pricing.aspx#.WWbRIph97IU
[9] Y. Niu, B. Luo, F. Liu, J. Liu, and B. Li, "When hybrid cloud meets flash crowd: Towards cost-effective service provisioning," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 1044–1052.
[10] N. Matloff, "Introduction to discrete-event simulation and the SimPy language," Dept. Comput. Sci., Univ. California, Davis, CA, USA, Tech. Rep., 2008, p. 2009, vol. 2.
[11] Ausgrid. *Electricity Price 2017-2018 in Australia*. Accessed: Jun. 28, 2017. [Online]. Available: http://www.ausgrid.com.au/Common/Industry/Regulation/Network-prices/Price-lists-and-policy.aspx#.WWbR9Zh97IU
[12] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J.-M. Pierson, and A. V. Vasilakos, "Cloud computing: Survey on energy efficiency," *ACM Comput. Surv.*, vol. 47, no. 2, p. 33, 2015.
[13] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1728–1739, May 2016.
[14] 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 *Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2017, pp. 186–195.
[15] M. J. Neely, E. Modiano, and C. P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 396–409, Apr. 2008.
[16] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.
[17] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.
[18] S. Li, Y. Zhou, L. Jiao, X. Yan, X. Wang, and M. R. T. Lyu, "Towards operational cost minimization in hybrid clouds for dynamic resource provisioning with delay-aware optimization," *IEEE Trans. Serv. Comput.*, vol. 8, no. 3, pp. 398–409, May 2015.
[19] Y. Mao, J. Zhang, and K. B. Letaief. (May 2016). "Dynamic computation offloading for mobile-edge computing with energy harvesting devices." [Online]. Available: https://arxiv.org/abs/1605.05488
[20] A. Destounis, G. S. Paschos, and I. Koutsopoulos, "Streaming big data meets backpressure in distributed network computation," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9, doi: 10.1109/INFOCOM.2016.7524388.

**YUCEN NAN** (S'16) received the B.E. degree in computer science engineering from Southwest Jiaotong University, Chengdu, China, in 2015, and the M.Phil. degree from The University of Sydney in 2017. She is currently pursuing the Ph.D. degree with the Center for Distributed and High Performance Computing, School of Information Technology, The University of Sydney. Her current research interests are in the area of cloud computing, Internet of Things, and optimization.

**WEI LI** (SM'16) received the Ph.D. degree from the School of Information Technologies, The University of Sydney. He is currently a Research Associate with the Centre for Distributed and High Performance Computing, The University of Sydney. His current research interests include Internet of Things, task scheduling, energy efficient algorithms, wireless sensor network, and optimization. He is a member of ACM.

**WEI BAO** (M'16) received the B.E. degree in communications engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2009, the M.A.Sc. degree in electrical and computer engineering from The University of British Columbia, Vancouver, Canada, in 2011, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, Canada, in 2016. He is currently a Lecturer with the School of Information Technologies, The University of Sydney, Sydney, Australia. His current research interests are in networked systems and mobile computing. He received best-paper awards at the ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems in 2013, and the IEEE International Symposium on Network Computing and Applications in 2016.

**FLAVIA C. DELICATO** received the Ph.D. degree in electrical and computer engineering from the Federal University of Rio de Janeiro in 2005. She is currently an Associate Professor of computer science with the Federal University of Rio de Janeiro, Brazil. She is the author of over 100 papers and participates in several research projects with funding from international and Brazilian government agencies. Her research interests include wireless sensor networks, the Internet of Things, adaptive systems, middleware, and software engineering techniques applied to ubiquitous systems. She is a level 1 Researcher Fellow of the Brazilian National Council for Scientific and Technological Development.

**PAULO F. PIRES** received the Ph.D. degree in 2002. He is currently an Associate Professor with the Department of Computer Science, Federal University of Rio de Janeiro (UFRJ), Brazil. He leads the Ubicomp Laboratory, UFRJ, where he coordinates researches in cyber physical systems, and software engineering techniques applied in the development of complex distributed systems. He has coordinated several cooperative industry university projects as well as research projects in those areas, with funding from several International and Brazilian government agencies, including CNPq, CAPES, FINEP, FAPERJ, Microsoft, Fundacin Carolina (Spain), and Australian Government (Australian Leadership Awards-ALA). He has published over 100 refereed international journal articles, conference papers, and book chapters. He is currently the Editor in Chief of the International Journal of Semantic and Infrastructure Services. He holds a technological innovation productivity scholarship from the Brazilian Research Council, since 2010. He is a member of the Brazilian Computer Society.

**YONG DOU** was born in Jilin, China, in 1966. He received the Ph.D. degree from the Military Engineering Institute of the PLA, Harbin, in 1995. His major fields of interest are computer architecture and compiler optimization, and reconfigurable computing. Since 1995, he was 13 years with the College of computer Science from the National University of Defense Technology, Changsha. As a Full Professor, he taught and made researches in the field of parallel architecture, reconfigurable array processor design. His research is recently oriented into FPGA pipeline scheduling and loopself pipelining onto reconfigurable array. He is the architecture professional committee of China Computer Federation, committees for international conference APPT.

**ALBERT Y. ZOMAYA** (F'04) is currently the Chair Professor of high performance computing and networking with the School of Information Technologies, University of Sydney, and he also serves as the Director of the Centre for Distributed and High Performance Computing. He has published over 600 scientific papers and articles and is author, co-author or editor of over 20 books. His research interests are in the areas of parallel and distributed computing and complex systems He is the recipient of the IEEE Technical Committee on Parallel Processing Outstanding Service Award (2011), the IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing (2011), and the IEEE Computer Society Technical Achievement Award (2014). He is a Chartered Engineer, a Fellow of AAAS, IEEE, and IET. He is the Founding Editor in Chief of the IEEE Transactions on Sustainable Computing and serves as an Associate Editor for over 20 leading journals. He served as an Editor in Chief for the IEEE Transactions on Computers from 2011 to 2014.