# PORA: Predictive Offloading and Resource Allocation in Dynamic Fog Computing Systems

Xin Gao, Xi Huang, and Ziyu Shao, *Member, IEEE,*

***Abstract*—Nowadays, fog computing has become a promising paradigm to enable computation-intensive and delay-intensive Internet-of-Things (IoT) applications. To avoid the limitations of cloud computing such as high transmission latency and insufficient network bandwidth, fog computing brings the cloud to the edge of the network. In fog computing systems, workloads of IoT applications are offloaded to fog nodes, while workloads in fog nodes are further offloaded to cloud whenever fog nodes are not sufficiently powerful. In this paper, we study the dynamic offloading and resource allocation problem in a hierarchal fog computing system with multiple fog tiers, take the average power consumption and latency of the fog system into account. One of our contribution is that we propose a predictive offloading and resource allocation (PORA) algorithm which needs no prior system state statistics. Furthermore, we incorporate the prediction of future workload arrivals into dynamic offloading decisions. We characterize the tradeoff between average power consumption and latency by analyzing the performance bounds of PORA, and show that the tradeoff can be adjusted by some parameter $V$. Furthermore, we propose theoretical analysis of the fundamental limitation of predictive offloading under perfect prediction. Trace-driven simulations demonstrate our theoretical analysis, and prove the effectiveness of PORA in keeping low average power consumption and latency. The simulations also show the robustness of PORA under imperfect prediction.**

***Index Terms*—Fog computing, workload offloading, resource allocation, Lyapunov optimization, prediction.**

## I. Introduction

Cloud computing is viewed as a promising paradigm to meet the explosive computation demands of IoT applications in the past decades due to its efficient resources provisioning and low cost [1], [2]. However, with the fast increase of the number of IoT devices, the volume of data streams generated by IoT applications grows high. It is expected that the number of IoT devices will increase to 50 billion by 2020 [3]. With the emergence of IoT applications, cloud computing may fall short to fulfill real-time requirements of such applications largely due to intolerable high network latency and insufficient network bandwidth [4].

By extending cloud to the edge of network, fog computing comes as a promising complement to meet their stringent requirements of low latency with intensive computation [5], [6]. A typical fog computing system consists of a set of geographically distributed fog nodes, which can be access points, routers, gateways, and mobile devices that have excessive computing resources and can offer services to IoT devices [7]. Such nodes are deployed at the network periphery with elastic resource provisioning such as storage, computation, and network [8].

X. Gao, X. Huang and Z. Shao are with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China.

Depending on their distance to user-end of IoT applications, fog nodes are often organized in a hierarchical manner, with each layer as a *fog tier*. Meanwhile, edge devices at the user-end are often resource-limited. When heavily loaded, they can upload and delegate part of their workloads via wireless links to fog nodes nearby, *a.k.a.*, *workload offloading*, to reduce power consumption while accelerating processing. Likewise, each fog node can offload part of its workloads to nodes in its upper fog tier with more powerful processing capacities. However, along with all the benefits comes the downside of extended latency and extra power consumption. Given such a power-latency trade-off, two interesting questions arise. One is to decide *when* and *how much* to offload between successive fog tiers. The other is to decide *resource allocation* for processing and offloading. Such decision making is critical yet challenging to make, due to the highly varying system dynamics in wireless environment, uncertainty in the resulting latency incurred by offloading, and latent and unknown traffic statistics.

Recently, many works have contributed to develop effective offloading schemes in fog computing under different cases. Mao *et al.* [9] investigates the power-delay tradeoff in the case of workload offloading between multiple users and one fog node. Liu *et al.* [10] considers the offloading between multiple users and multiple fog nodes. Xiao *et al.* [6] studies the fog computing systems where fog nodes can offload workloads to the cloud. Bozorgchenani *et al.* [11] explores the offloading problem in a two-tiered fog computing system. On the other hand, Nan *et al.* [12] and Liu *et al.* [13] focus on developing energy-aware offloading schemes with Poisson traffic, though no empirical evidence shows that traffic arrivals follow Poisson process in fog computing systems.

Nonetheless, following challenges still remain unaddressed:
(a)**Characterization of system dynamics and power-latency tradeoff:** Existing works only focus on special cases of the system with flat or two-tiered architecture. However, in practice, a fog system often consists of multiple tiers, with complex interplays between fog tiers and the cloud, not to mention the constantly varying dynamics and intertwined power-latency tradeoffs therein. An exquisite model that accurately characterizes the system and tradeoffs is the key to the fundamental understanding of our design space.
(b)**Efficient and online decision making:** The decision making must be computationally efficient, so as to minimize the overheads brought in. The difficulties often come from the uncertainties of traffic statistics, online nature of workload

arrivals, and intrinsic complexity of the problem.

(c)**Understanding the benefits of prediction:** One natural extension to online decision making is to employ predictive offloading to further reduce latencies and promote quality of service. For example, Netflix preloads videos onto users' devices based on user behavior prediction [14]. Despite the wide applications of such approaches, the fundamental limits of predictive offloading in fog computing still remains unknown.

Different from previous works, in this paper, we focus on a general multi-tiered fog systems. We overcome the above difficulties by developing a fine-grained queue model that accurately depicts general multi-tiered fog systems, and proposing an efficient online scheme that performs offloading on a time-slot basis. To our best knowledge, we are the first to systematically study predictive scheduling in fog systems. We compare our work with others in TABLE I and our main contributions are as follows:

**Problem Formulation:** We formulate the problem of dynamic offloading and resource allocation as a stochastic optimization problem, aiming at minimizing long-term time-average expectation of total power consumption of fog tiers with queue stability guarantee.

**Algorithm Design:** Through a non-trivial transformation, we decouple the problem into a series of subproblems over time slots. By exploiting their unique structures, we propose PORA, an efficient scheme that exploits predictive scheduling to make decisions in an online manner.

**Theoretical Analysis and Experimental Verification:** We conduct theoretical analysis and trace-driven simulations to evaluate PORA. Our results show that PORA achieves a tunable power-latency tradeoff, while effectively reducing the average latency with only mild-value of predictive information, even in the presence of prediction errors.

**New Degree of Freedom in Fog Computing:** We systematically investigate the fundamental limits and benefits of predictive offloading in fog computing systems. With both theoretical analysis and numerical results, we show the effectiveness of our algorithm in the presence of prediction error.

We organize the rest of the paper as follows. Section II provides a motivating example for dynamic offloading and resource allocation in fog computing. Section III then presents the system model and problem formulation. Section IV aims at the algorithm design of PORA, followed by its performance and complexity analysis. Section V discusses results from trace-driven simulations, while Section VI concludes the paper.

## II. Motivating Example

In this section, we use a motivating example to show the potential tradeoff between latency and power consumption in fog tiers. We consider a time slotted fog computing system consisting of one EFN and one CFN. Packets in EFN can be locally processed or offloaded to CFN through wireless connection. Packets in CFN can be locally processed or offloaded to cloud through wired connection. As shown in Fig. 1(a), each fog node maintains only one queue in this simple example since we assume each fog node can choose only one policy during the whole process, *i.e.*, always process locally or always offload. The maximal local processing rate of EFN and CFN are 1 and 8 packets per time slot respectively. The maximal transmit rate from EFN to CFN is 4 packets per time slot, and the maximal transmit rate from CFN to cloud is 5 packets per time slot. We assume the power consumption is linear to the number of processed/transmitted packets. Processing one packet locally costs 1 mW power, and transmitting one packet over the wireless link costs 0.5 mW. Packets offloaded to cloud will be immediately processed once arrive at the cloud.

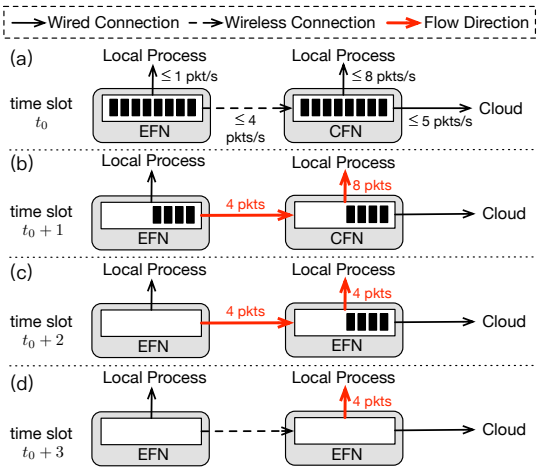



Fig. 1. Motivating example.

As illustrated in Fig. 1(a), there are already 8 packets in each queue at the initial time slot $t_0$. TABLE II lists the total power consumption of fog nodes and the total packets latency under different policies. We take the case when EFN chooses to offload and CFN chooses to process locally as an example. In time slot $t_0+1$, the 8 packets in CFN are processed locally, and 4 of 8 packets in EFN are offloaded to CFN according to the wireless transmission capacity. In time slot $t_0+2$, the 4 packets offloaded to CFN are locally processed, and the rest 4 packets in EFN are offloaded to CFN. In time slot $t_0+3$, all packets are processed by CFN. In this case, the total power consumption includes 16 mW local processing power and 4 mW transmit power, and the total latency of all packets is 28.

From TABLE II we observe: (1) When the EFN chooses to offload and the CFN chooses to process locally, the total packet latency is minimized. However, the strategy leads to most power consumption. (2) Under the same EFN policy, there is a tradeoff between the total power consumption and packet latency when CFN chooses different policies. The reason is that offloading to the cloud can save power but incur extra transmission latency. (3) Given that the CFN chooses to process locally, there is a power-delay tradeoff when EFN chooses different policies since wireless transmissions cost extra power but the processing latency of CFN is much smaller than EFN. In summary, there are complex tradeoffs between the power consumption and latency.

## III. Model and Problem Formulation

As shown in Fig. 2, we consider a hierarchical fog computing system with two fog tiers. The edge fog tier contains a

TABLE I
COMPARISONS OF RELATED WORKS

| | Fog-Fog[1] | Fog-Cloud[2] | Dynamic | Arrival Distribution | Communication Mode | Prediction |
|---|---|---|---|---|---|---|
| [9] | × | × | ✓ | Arbitrary | Many-to-One | × |
| [10] | × | × | ✓ | Arbitrary | Many-to-Many | × |
| [6] | ✓ | ✓ | × | – | Many-to-Many | × |
| [11] | ✓ | × | × | – | One-to-One | × |
| [12] | × | ✓ | ✓ | Poisson | One-to-One | × |
| [13] | × | ✓ | ✓ | Poisson | Many-to-One | × |
| Ours | ✓ | ✓ | ✓ | Arbitrary | Many-to-Many | ✓ |

[1,2] "Fog-Fog" means offloading between fog tiers, while "Fog-Cloud" means offloading from fog to cloud.

TABLE II
PERFORMANCE UNDER DIFFERENT STRATEGIES

| Policy of EFN | Policy of CFN | Total Power Consumption | Total Packet Latency |
|---|---|---|---|
| Local | Local | 16 mW | 44 slots |
| Local | Offload | 8 mW | 47 slots |
| Offload | Local | 20 mW | 28 slots |
| Offload | Offload | 4 mW | 34 slots |

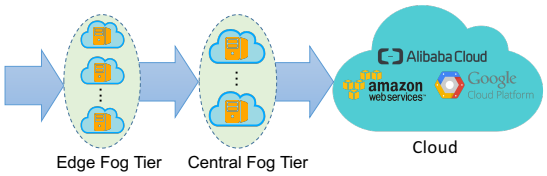



Fig. 2. Fog computing system with two fog tiers.

set of EFNs that that offer low access latency to IoT devices. The central fog tier comprises CFNs that offer medium access latency. CFNs usually have higher computation capacities than EFNs and can compensate for the low processing ability of edge fog tier. To avoid long processing latency, EFNs offload part of their workloads to the central fog tier. Workloads in CFNs can be further offloaded to the cloud. There are sufficient computation resources in cloud and the processing latency in it can be ignored. However, large transmission latency will incur if we offload too much workload to cloud. In TABLE III, we summarize the key notations used in this paper.

We assume the system is time-slotted and the length of each time slot is $\tau_0$ seconds. The fog computing system consists of $N$ EFNs in edge fog tier and $M$ CFNs in central fog tier. Let $\mathcal{N}$ be the set of EFNs and $\mathcal{M}$ be the set of CFNs. For geometric reason, each EFN $i$ can only access to a subset $\mathcal{M}_i \subset \mathcal{M}$ of CFNs. For each CFN $j$, denote the set of EFNs that can access to it as $\mathcal{N}_j \subset \mathcal{N}$. Note that for any $i \in \mathcal{N}_j$, there holds $j \in \mathcal{M}_i$.

### A. Queueing Model of Edge Fog Tier

Let $A_i(t)$ denotes the amount of new workload arrive to EFN $i$ in time slot $t$. We assume $A_i(t)$ is independent over different time slot with $\mathbb{E}\{A_i(t)\} = \lambda_i$ and has an upper bound $A_{\max}$ for all $i \in \mathcal{N}$ and $t \in \{0, 1, 2, ...\}$. We assume the future arrivals of every EFN $i$ in the next $W_i$ time slots can be predicted and pre-served. In other words, EFN $i$ has access to the future information in the lookahead window $\{A_i(t), A_i(t+1), ..., A_i(t+W_i-1)\}$ and can serve[1] them before they actually arrive.

TABLE III
KEY NOTATIONS

| Notations | Meanings |
|---|---|
| $\tau_0$ | Length of one time slot |
| $\mathcal{N}$, $N$ | $\mathcal{N}$ is the set of EFNs, and $N$ is the number of EFNs |
| $\mathcal{M}$, $M$ | $\mathcal{M}$ is the set of CFNs, and $N$ is the number of CFNs |
| $\mathcal{N}_j$ | Set of EFNs that CFN $j$ can access |
| $\mathcal{M}_i$ | Set of CFNs that EFN $i$ can access |
| $A_i(t)$ | Amount of workload arrive to EFN $i$ in time slot $t$ |
| $\lambda_i$ | Average workload arrival rate to EFN $i$, $\lambda_i = \mathbb{E}\{A_i(t)\}$ |
| $W_i$ | Prediction window size of EFN $i$ |
| $Q_i^{(e,a)}(t)$ | Prediction queue backlog of EFN $i$ in time slot $t$ |
| $Q_i^{(e,l)}(t)$ | Local queue backlog of EFN $i$ in time slot $t$ |
| $Q_i^{(e,o)}(t)$ | Offloading queue backlog of EFN $i$ in time slot $t$ |
| $b_i^{(e,l)}(t)$ | Amount of workload to be sent to $Q_i^{(e,l)}(t)$ in time slot $t$ |
| $b_i^{(e,o)}(t)$ | Amount of workload to be sent to $Q_i^{(e,o)}(t)$ in time slot $t$ |
| $f_i^{(e)}(t)$ | CPU-cycle frequency of EFN $i$ in time slot $t$ |
| $B$ | Channel bandwidth |
| $H_{i,j}(t)$ | Wireless channel gain between EFN $i$ and CFN $j$ |
| $p_{i,j}(t)$ | Transmit power from EFN $i$ to CFN $j$ in time slot $t$ |
| $R_{i,j}(t)$ | Transmit rate from EFN $i$ to CFN $j$ in time slot $t$ |
| $Q_j^{(c,a)}(t)$ | Arrival queue backlog of EFN $i$ in time slot $t$ |
| $Q_j^{(c,l)}(t)$ | Local queue backlog of EFN $i$ in time slot $t$ |
| $Q_j^{(c,o)}(t)$ | Offloading queue backlog of EFN $i$ in time slot $t$ |
| $b_j^{(c,l)}(t)$ | Amount of workload to be sent to $Q_j^{(c,l)}(t)$ in time slot $t$ |
| $b_j^{(c,o)}(t)$ | Amount of workload to be sent to $Q_j^{(c,o)}(t)$ in time slot $t$ |
| $f_j^{(c)}(t)$ | CPU-cycle frequency of CFN $j$ in time slot $t$ |
| $P(t)$ | Total power consumption in time slot $t$ |

As shown in Figure 3, each EFN $i \in \mathcal{N}$ maintains four kinds of queues: prediction queues $A_{1,0}(t), ..., A_{i,W_i-1}(t)$, arrival queue $A_{i,-1}(t)$, local queue $Q_i^{(e,l)}(t)$, and offloading queue $Q_i^{(e,o)}(t)$. In time slot $t$, prediction queue $A_{i,w}(t)$ ($w \in \{0, 1, ..., W_i - 1\}$) stores residual arrivals that will arrive in time slot $t+w$ after going through a series of pre-services before time slot $t$. In time slot $t$, workload actually arrives to EFN $i$ is stored in the arrival queue $A_{i,-1}(t)$ and wait for being distributed to the local queue $Q_i^{(e,l)}(t)$ and offloading

[1]Here the serve means serve the future tasks locally in edge fog node $i$ or transmit it to other fog nodes.

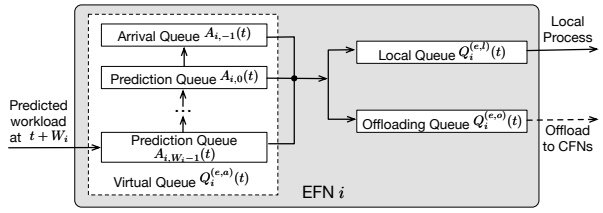



Fig. 3. Prediction queue model: In time slot $t$, EFN $i$ allocates departure rate among its arrival queue $A_{i,-1}(t)$ and prediction queues $A_{i,0}(t)$, $A_{i,W_i-1}(t)$. Note that data in prediction queues can either departure through local process or offloading, or both of them.

queue $Q_i^{(e,o)}(t)$. Workload in $Q_i^{(e,l)}(t)$ will be processed by the local CPU of EFN $i$, and workload in $Q_i^{(e,o)}(t)$ will be offloaded to the central fog tier.

*1) Prediction Queues and Arrival Queues in EFNs:* In every time slot, EFN $i$ decides the amounts of workload sent to the local queue and the offloading queue. We denote these two decisions as $b_i^{(e,l)}(t)$ and $b_i^{(e,o)}(t)$ respectively. They satisfy the following constraints

$$0 \leq b_i^{(e,\beta)}(t) \leq b_{i,\max}^{(e,\beta)},\ \forall \beta \in \{l,o\} \tag{1}$$

where $b_{i,\max}^{(e,\beta)} > 0$ are positive constants. $b_i^{(e,l)}(t)$ and $b_i^{(e,o)}(t)$ are composed of departures from prediction queues and the arrival queue. Denote the amount of workload departs from $A_{i,w}(t)$ in time slot $t$ as $\mu_{i,w}(t)$, where $w \in \{-1,0,...,W_i-1\}$. We have

$$\sum_{w=-1}^{W_i-1} \mu_{i,w}(t) = b_i^{(e,l)}(t) + b_i^{(e,o)}(t). \tag{2}$$

Now we consider the update of queue $A_{i,w}(t)$. The prediction queues of EFN $i$ evolve as

(i) If $w = W_i - 1$, then

$$A_{i,W_i-1}(t+1) = A_i(t+W_i). \tag{3}$$

(ii) If $0 \leq w \leq W_i - 2$, then

$$A_{i,w}(t+1) = [A_{i,w+1}(t) - \mu_{i,w+1}(t)]^+. \tag{4}$$

where $[x]^+ \triangleq \max\{x,0\}$ for any real number $x$. Note that in time slot $t+1$, the amount of workload that will arrive after $w = W_i - 1$ time slots is $A_i(t+W_i)$ and it is not predicted until time slot $t+1$. The arrival queue $A_{i,-1}(t)$ records the true data backlog of fog node $i$ and it evolves as

$$A_{i,-1}(t+1) = [A_{i,-1}(t) - \mu_{i,-1}(t)]^+ + [A_{i,0}(t) - \mu_{i,0}(t)]^+. \tag{5}$$

Note that $\mu_{i,-1}(t)$ is the amount of workload that is already in the arrival queue and will leave the arrival queue $A_{i,-1}(t)$ in time slot $t$.

We define a virtual queue $Q_i^{(e,a)}(t) \triangleq \sum_{w=-1}^{W_i-1} A_{i,w}(t)$ whose backlog is the sum of all prediction queues and the arrival queue of EFN $i$. Under *fully-efficient* [15] predictive algorithm, queue $Q_i^{(e,a)}(t)$ update as

$$Q_i^{(e,a)}(t+1) = [Q_i^{(e,a)}(t) - (b_i^{(e,l)}(t) + b_i^{(e,o)}(t))]^+ + A_i(t+W_i) \tag{6}$$

by update equations (3)(4)(5). The arrival of virtual queue $Q_i^{(e,a)}(t)$ is the predicted workload that will arrive to EFN $i$ in time slot $t+W_i$. The output is the total workload distributed to the local queue and the offloading queue.

*2) Offloading Queues in EFNs:* In time slot $t$, workload in offloading queue $Q_i^{(e,o)}(t)$ will be transmitted to CFNs in set $\mathcal{N}_i$. The amount of transmitted workload is determined by the transmit power $(p_{i,j}(t))_{j\in\mathcal{M}_i}$, where $p_{i,j}(t)$ is the transmit power from node $i$ to node $j$. According to Shannon-Hartley theorem, the amount of workload being transmitted from EFN $i$ to CFN $j$ is

$$R_{i,j}(t) \triangleq \hat{R}_{i,j}(p_{i,j}(t)) = \tau_0 B \log_2\left(1 + \frac{p_{i,j}(t)H_{i,j}(t)}{N_0 B}\right) \tag{7}$$

where $B$ is the channel bandwidth, $H_{i,j}(t)$ is the wireless channel gain between EFN $i$ and CFN $j$, and $N_0$ is the system power spectral density of the additive white Gaussian noise. Adjusting the transmit power $p_{i,j}(t)$, we can change the amount of workload offloaded from EFN $i$ to CFN $j$ in time slot $t$. Note that $H_{i,j}(t)$ is an uncontrollable environment state which varies over time, it is assumed to satisfy $H_{i,j}(t) \leq H_{\max}$ for some positive constant $H_{\max}$. We assume that our system uses Orthogonal Frequency Division Multiplexing (OFDM) technology, thus the interference among channels is not considered here. Also, we assume the transmission power can not be negative and the total transmit power of fog node $i$ is upper bounded by a positive constant $p_{i,\max}$, *i.e.*,

$$p_{i,j}(t) \geq 0,\ \forall i \in \mathcal{N}, j \in \mathcal{M}_i, t, \tag{8}$$

$$\sum_{j\in\mathcal{M}_i} p_{i,j}(t) \leq p_{i,\max},\ \forall i \in \mathcal{N}, t. \tag{9}$$

According to above description, the update function of offload queue $Q_i^{(e,o)}(t)$ is

$$Q_i^{(e,o)}(t+1) \leq [Q_i^{(e,o)}(t) - \sum_{j\in\mathcal{M}_i} R_{i,j}(t)]^+ + b_i^{(e,o)}(t), \tag{10}$$

where $\sum_{j\in\mathcal{M}_i} R_{i,j}(t)$ is the total allocated transmission rate to EFN $i$ in time $t$. We use inequality here for the reason that the actually arrived workload amount can be less than $b_i^{(e,o)}(t)$.

### *B. Queueing Model of Central Fog Tier*

As shown in Fig. 4, each CFN $j \in \mathcal{M}$ maintains three queues: an arrival queue $Q_j^{(c,a)}(t)$, a local queue $Q_j^{(c,l)}(t)$, and an offloading queue $Q_j^{(c,o)}(t)$. Similar to EFNs, workloads from the edge fog tier will be first stored in the arrival queue and waiting for being distributed to the local queue and the offloading queue. Workload in the local queue will be locally processed, and workload in the offloading queue will be offloaded to the cloud.

*1) Arrival Queues in CFNs:* Arrival of CFN $j \in \mathcal{M}$ is composed of workloads transmitted from EFNs in the set $\mathcal{N}_j$, *i.e.*, $\sum_{i\in\mathcal{N}_j} R_{i,j}(t)$. We denote the amount of workload distributed to the local queue and offload queue in time slot $t$ as $b_i^{(e,l)}(t)$ and $b_i^{(e,o)}(t)$ respectively, and they satisfy the following constraints

$$0 \leq b_j^{(c,\beta)}(t) \leq b_{j,\max}^{(c,\beta)},\ \forall \beta \in \{l,o\} \tag{11}$$

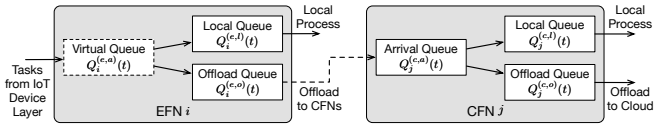



Fig. 4. Queueing model of the system.

where $b_{j,\max}^{(c,\beta)} > 0$ are positive constants. The update function of queue $Q_j^{(c,a)}(t)$ is

$$Q_i^{(c,a)}(t+1) \leq [Q_i^{(c,a)}(t) - (b_i^{(c,l)}(t) + b_i^{(c,o)}(t))]^+ + \sum_{i\in\mathcal{N}_j} R_{i,j}(t). \quad (12)$$

*2) Offloading Queues in CFNs:* For each CFN $j \in \mathcal{M}$, offloading queue $Q_j^{(c,o)}(t)$ stores workload to be offloaded. We refer to $D_j(t)$ as the transmission rate of the wired link from CFN $j$ to the cloud in time slot $t$, which is determined by the network state and satisfies $D_j(t) \leq D_{\max}$ for all $j$, $t$. The update function of $Q_j^{(c,o)}(t)$ in time slot $t$ is

$$Q_j^{(c,o)}(t+1) = [Q_j^{(c,o)}(t) - D_j(t)]^+ + b_j^{(c,o)}(t). \quad (13)$$

### C. Local Queues in EFNs and CFNs

Since the local queues in EFNs and CFNs have the same structure, we introduce them together. Suppose all fog nodes have DVFS (Dynamic Voltage and Frequency Scaling) capability, then each fog node can adjust its local CPU-cycle frequency [16]. Let $L_k^{(\alpha)}$ be the number of CPU cycles needed to process one bit of workload in fog node $k$. It can be measured offline [17]. Note that $\alpha = e$ if fog node $k$ is an EFN, and $\alpha = c$ if fog node $k$ is a CFN. The local service rate of fog node $k$ is $f_k^{(\alpha)}(t)/L_k^{(\alpha)}$, and the local queue of fog node $k$ updates as

$$Q_k^{(\alpha,l)}(t+1) = [Q_k^{(\alpha,l)}(t) - \tau_0 f_k^{(\alpha)}(t)/L_k^{(\alpha)}]^+ + b_k^{(\alpha,l)}(t). \quad (14)$$

We assume that the local CPU-cycle frequency satisfies the bound constraint

$$0 \leq f_k^{(\alpha)}(t) \leq f_{k,\max}^{(\alpha)}, \quad (15)$$

in which $f_{k,\max}^{(\alpha)} > 0$ is a constant. The reason is that the CPU-cycle frequency must be nonnegative and finite.

### D. Total Power Consumption

The total power consumption of the fog tiers in time slot $t$ is denoted by $P(t)$. It includes the power consumption of local CPUs and wireless transmissions. The power consumption of a local CPU with CPU-cycle frequency $f$ can be calculated as $\kappa f^3$, where $\kappa$ is a parameter depending on the hardware architecture can be really measured [18]. Then $P(t)$ can be expressed as

$$P(t) \triangleq \hat{P}(\boldsymbol{f}(t), \boldsymbol{p}(t)) = \sum_{i\in\mathcal{N}} \tau_0 \kappa (f_i^{(e)}(t))^3 + \sum_{j\in\mathcal{M}} \tau_0 \kappa (f_j^{(c)}(t))^3 + \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{M}_i} \tau_0 p_{i,j}(t), \quad (16)$$

where $\boldsymbol{f}(t) \triangleq ((f_i^{(e)}(t))_{i\in\mathcal{N}}, (f_j^{(c)}(t))_{j\in\mathcal{M}})$ is the vector of CPU-cycle frequencies of all fog nodes, and $\boldsymbol{p}(t) \triangleq (\boldsymbol{p}_i(t))_{i\in\mathcal{N}}$ is the vector of transmit power allocation of all EFNs. $\boldsymbol{p}_i(t) = (p_{i,j}(t))_{j\in\mathcal{M}_i}$ is the transmit power allocation of EFN $i$.

### E. Problem Formulation

At first, we define the long-term time average expectation of the total power consumption and total queue backlog as

$$\bar{P} \triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{P(\tau)\}, \quad (17)$$

$$\bar{Q} \triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{\alpha\in\{a,l,o\}} \left( \sum_{i\in\mathcal{N}} \mathbb{E}\{Q_i^{(e,\alpha)}(\tau)\} + \sum_{j\in\mathcal{M}} \mathbb{E}\{Q_j^{(c,\alpha)}(\tau)\} \right). \quad (18)$$

In this paper, we aim at minimizing $\bar{P}$, while guarantee system stability. Here the stability refers to the strong stability defined in [19], and it means $\bar{Q} < \infty$ in our case. Thus he problem can be formulated as

$$\begin{aligned} \underset{\boldsymbol{b}(t),\boldsymbol{f}(t),\boldsymbol{p}(t)}{\text{minimize}} \quad & \bar{P} \\ \text{subject to} \quad & (1)(15)(8)(9)(11) \\ & \bar{Q} < \infty \end{aligned} \quad (19)$$

## IV. Predictive Algorithm

### A. Predictive Algorithm

In this section, we develop a predictive offloading and resource allocation algorithm PORA based on the queueing model using Lyapunov optimization methods [19]. Instead of solving the stochastic optimization problem (19) directly, we decouple the problem into a series of subproblem over time slots.

The following parts are all new:

*1) Offloading Decision:* In each time slot, fog node $k \in \mathcal{N} \cup \mathcal{M}$ makes offloading decisions $b_k^{(\alpha,l)}(t)$ and $b_k^{(\alpha,o)}(t)$ by solving the following subproblem:

$$\min_{0\leq b_k^{(\alpha,\beta)} \leq b_{k,\max}^{(\alpha,\beta)}} \left( Q_k^{(\alpha,\beta)}(t) - Q_k^{(\alpha,a)}(t) \right) b_k^{(\alpha,\beta)} \quad (20)$$

where $\alpha \in \{e, c\}$ and $\beta \in \{l, o\}$. Note that when $k \in \mathcal{N}$ is an EFN, $\alpha = e$; When $k \in \mathcal{M}$ is a CFN, $\alpha = c$. Moreover, when $\beta = l$ we are determining the amount of workload assigned to the local queue; When $\beta = c$ we are determining the amount of workload assigned to the offloading queue. We update $b_k^{(\alpha,\beta)}(t)$ to the optimal solution of (20):

$$b_k^{(\alpha,\beta)}(t) = \begin{cases} b_{k,\max}^{(\alpha,\beta)}, & \text{if } Q_k^{(\alpha,\beta)}(t) < Q_k^{(\alpha,a)}(t), \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

***Insight:*** If the virtual/arrival queue backlog $Q_k^{(\alpha,a)}(t)$ is larger than the local queue backlog $Q_k^{(\alpha,l)}(t)$, fog node $k$ will send as much workload from queue $Q_k^{(\alpha,a)}(t)$ to the local/offloading queue $Q_k^{(\alpha,\beta)}(t)$ as it can. Otherwise, fog node $k$

will be unwilling to send any workload to the local/offloading queue. The strategy tries to achieve the load balancing between the virtual/arrival queue and the local/offloading queue.

*2) Local CPU-Cycle Frequency Setting:* Each fog node $k \in \mathcal{N} \cup \mathcal{M}$ solves the following subproblem to determine its local CPU-cycle frequency $f_k^{(\alpha)}(t)$:

$$\min_{0 \le f_k^{(\alpha)} \le f_{k,\max}^{(\alpha)}} V\kappa (f_k^{(\alpha)})^3 - Q_k^{(\alpha,l)}(t) f_k^{(\alpha)} / L_k^{(\alpha)} \tag{22}$$

The optimal solution of (22) is the $f_k^{(\alpha)}$ makes the second derivative of the objective function equal to 0. We update the CPU-cycle frequency of fog node $k$ in time slot $t$ by setting it to the optimal solution of (22):

$$f_k^{(\alpha)}(t) = \min \left\{ \sqrt{Q_k^{(\alpha,l)}(t) / 3V\kappa L_k^{(\alpha)}}, f_{k,\max}^{(\alpha)} \right\}. \tag{23}$$

***Insight:*** $f_k^{(\alpha)}(t)$ increases with the increase of local queue backlog $Q_k^{(\alpha,l)}(t)$, which shows that the policy tries to avoid explosive queue backlog. On the other hand, $f_k^{(\alpha)}(t)$ decreases with the the increase of tradeoff parameter $V$. Thus we can increase the value of $V$ when we value low power consumption more than small queueing delay.

The proof for the optimal solution (23) is shown in Appendix B-A.

*3) Power allocations for EFNs:* Each EFN $i \in \mathcal{N}$ solves the following subproblem to determine the transmission power allocation $\boldsymbol{p}_i(t)$ in time slot $t$:

$$\begin{aligned} \min_{\boldsymbol{p}_i} \ & \sum_{j \in \mathcal{M}_i} V p_{i,j} - m_{i,j}(t) \log_2 (1 + l_{i,j}(t) p_{i,j}) \\ \text{s.t.} \ & \sum_{j \in \mathcal{M}_i} p_{i,j} \le p_{i,\max}, \\ & p_{i,j} \ge 0, \ \forall j \in \mathcal{M}_i, \end{aligned} \tag{24}$$

where $m_{i,j}(t) = (Q_i^{(e,o)}(t) - Q_j^{(c,a)}(t))B$, $l_{i,j}(t) = \frac{H_{i,j}(t)}{N_0 B}$.

Assume the optimal solution of problem (24) is $\boldsymbol{p}_i^*(t) = (p_{i,j}^*(t))_{j \in \mathcal{M}_i}$. We solve for the optimal power allocation decisions using water-filling algorithm. At first, we solve the equation

$$\sum_{j \in \mathcal{M}_i} [m_{i,j}(t) / (V + \lambda^*) - 1/l_{i,j}(t)]^+ = p_{i,\max} \tag{25}$$

for $\lambda^*$. To solve (25), we use bisection method shown in Algorithm 1, where $\lambda_{\min}$ and $\lambda_{\min}$ are the lower bound and upper bound of $\lambda^*$, and $\varepsilon$ is the tolerance parameter. Then we get the optimal power allocation decision

$$p_{i,j}^*(t) = [m_{i,j}(t)/(V + \lambda^*) - 1/l_{i,j}(t)]^+. \tag{26}$$

We choose the power allocation decision as $\boldsymbol{p}_i(t) = \boldsymbol{p}_i^*(t)$.

The proof for the optimal solution is shown in Appendix B-B.

***Insight:*** EFN $i$ is more willing to allocate transmit power to the CFN with larger arrival queue $Q_j^{(c,a)}(t)$ for load balancing. On the other hand, CFN with better wireless channel conditions to EFN $i$ (larger $H_{i,j}(t)$) is easier to get larger transmit power allocation considering power efficiency. Moreover, by setting a larger $V$ we can save more transmit power, but the queueing latency will increase. Until here.

**Algorithm 1** Bisection Method for $\lambda^*$

```
Initialize λ_min = 0, λ_max = max_{j∈M_i} m_{i,j}(t) l_{i,j}(t) − V.
while 1 do
    Set λ* = (λ_min + λ_max)/2.
    if λ_max − λ_min ≤ ε then
        Return λ*.
    else
        if Σ_{j∈M_i} [ m_{i,j}(t)/(V+λ*) − 1/l_{i,j}(t) ]^+ > p_{i,max} then
            Set λ_max = λ*.
        else
            Set λ_min = λ*.
        end if
    end if
end while
```

We show the pseudocode of PORA in Algorithm 2. Note that $\alpha$ indicates each fog node's type. Specifically, for any fog node $k$, $\alpha = e$ if $k$ is an EFN and $\alpha = c$ if $k$ is a CFN. Next, we discuss some notable features of PORA, including load balancing and tunable power-delay tradeoffs in each fog tiers.

*a) Load Balancing:* Upon making offloading decisions for fog node $k$, PORA adjusts the load balancing dynamically between its virtual/arrival queue and local/offloading queue. In particularly, whenever node $k$'s virtual/arrival queue backlog $Q_k^{(\alpha,a)}(t)$ is greater than its local/offloading queue $Q_k^{(\alpha,\beta)}(t)$, it will admit as much workloads as it can from $Q_k^{(\alpha,a)}(t)$ to the local/offloading queue $Q_k^{(\alpha,\beta)}(t)$. Otherwise, no workload will be admitted. Besides, PORA also avoids overloading node $k$'s local queue backlog by increasing its CPU frequency as its local queue backlog $Q_k^{(\alpha,l)}(t)$ increases. In addition, PORA achieves load balancing between the edge fog tier and the central fog tier. Under PORA, EFN $i$ is more willing to allocate transmit power to the CFN $j$ with smaller arrival queue $Q_j^{(c,a)}(t)$ in set $\mathcal{M}_i$. If the arrival queue backlog of CFN $j$ is greater than the offloading queue on EFN $i$, EFN $i$ will transmit no workloads to CFN $j$.

*b) Power-Delay Tradeoffs:* First, we consider the case when there is no offloaded workloads from edge fog tier to central fog tier. As $V$ increases, the CPU frequency $f_k^{(\alpha)}(t)$ on node $k$ tends to decrease, resulting in the growth of all its queue backlogs. Thus there is a power-delay tradeoff across all fog nodes, including EFNs and CFNs. Next, we consider the case with offloaded workloads from edge fog tier to central fog tier. As $V$ increases, the CPU frequency $f_i^{(e)}(t)$ and transmit power $p_{i,j}(t)$ of EFN $i$ tend to decrease. As $p_{i,j}(t)$ decreases, the offloading rate to CFN $j$ also tends to decrease, leading to the decrease of arrival queue $Q_j^{(c,a)}(t)$ on CFN $j$ as well. However, the reduction in $Q_j^{(c,a)}(t)$ will promote the willingness of EFN $i$ to offload workloads to CFN $j$, thusly leading to the increase of $Q_j^{(c,a)}(t)$.

### *B. Computational Complexity of PORA*

During each time slot, the first part of complexity comes from calculating its CPU frequency allocation and offloading

decisions. Since the calculation (line 5-12) requires only constant time for each fog node, the total complexity of these steps is $O(N+M)$. Next, each EFN $i$ makes transmit power allocation decision by applying the bisection method (line 16-18), with a complexity of $O\left(\log_2\left(\frac{\lambda_{\max}-\lambda_{\min}}{\varepsilon}\right)+|\mathcal{M}_i|\right)$. After that, EFN $i$ determines the transmit power to each CFN in the set $\mathcal{M}_i$. In the worst case, each EFN could be potentially connected to all CFNs. Thus the total complexity of PORA algorithm is $O\left(M\times N\right)$.

**Algorithm 2** PORA

1: **for** $t=\{0,1,2,\cdots\}$ **do**
2: Initialize $\boldsymbol{b}(t)=\mathbf{0}$, $\boldsymbol{f}(t)=\mathbf{0}$, $\boldsymbol{p}(t)=\mathbf{0}$.
3: **for** each fog node $k\in\mathcal{N}\cup\mathcal{M}$ **do**
4: *%%Make Offloading Decisions*
5: **if** $Q_k^{(\alpha,a)}(t)>Q_k^{(\alpha,l)}(t)$ **then**
6: Set $b_k^{(\alpha,l)}(t)=b_{k,\max}^{(\alpha,l)}$.
7: **end if**
8: **if** $Q_k^{(\alpha,a)}(t)>Q_k^{(\alpha,o)}(t)$ **then**
9: Set $b_k^{(\alpha,o)}(t)=b_{k,\max}^{(\alpha,o)}$.
10: **end if**
11: *%%Local CPU-Cycle Frequency Allocation*
12: Set $f_k^{(\alpha)}(t)=\min\{\sqrt{Q_k^{(\alpha,l)}(t)/3V\kappa L_k^{(\alpha)}},f_{k,\max}^{(\alpha)}\}$.
13: **end for**
14: **for** each EFN $i\in\mathcal{N}$ **do**
15: *%%Transmit Power Allocation*
16: Get $\lambda^*$ from Algorithm 1.
17: **for** each CFN $j\in\mathcal{M}_i$ **do**
18: Set $p_{i,j}(t)=[m_{i,j}(t)/(V+\lambda^*)-1/l_{i,j}(t)]^+$.
19: **end for**
20: **end for**
21: Operate according to offloading decision $\boldsymbol{b}(t)$, CPU-cycle frequency $\boldsymbol{f}(t)$, transmit power allocation $\boldsymbol{p}(t)$.
22: **end for**

### C. Performance Analysis

We conduct theoretical analysis on the upper bounds of the average queue backlog $\bar{P}$ and power consumption $\bar{Q}$ under PORA scheme. Besides, we also analyze the benefits that predictive offloading brings in terms of latency reduction.

*1) Time-average Power Consumption and Queue Backlog:* Let $P^*$ and $P_W^*$ be the achievable minimums of $\bar{P}$ over all feasible non-predictive and predictive polices, respectively. Since any feasible non-predictive policy is also a feasible policy for the predictive system, we have $P_W^*\leq P^*$. Now we have the following theorem:

*Theorem 1:* Assume the system arrivals lies in the interior of the capacity region and $\boldsymbol{Q}(0)<\infty$. Under PORA algorithm, there exists constants $M>0$ and $\epsilon>0$ such that

$$\bar{P}\leq M/V+P^*,\ \bar{Q}\leq(M+VP_{\max})/\epsilon.$$

$\bar{P}$ and $\bar{Q}$ in Theorem 1 are defined in (17) and (18). The proof of Theorem 1 is shown in APPENDIX C.

***Insight:*** By Little's Theorem, the average queue backlog is proportional to the average latency. Thus Theorem 1 implies that by adjusting parameter $V$, PORA achieves an $[O(1/V),O(V)]$ power-delay tradeoff at different levels.

*2) Latency Reduction:* Next, we consider the latency reduction incurred by PORA under perfect prediction compared to non-predictive scheme. We denote the prediction window vector $\boldsymbol{W}$ by $(W_i)_{i\in\mathcal{N}}$ and the corresponding delay reduction by $\eta(\boldsymbol{W})$. For each unit of workload in EFN $i$, let $\pi_{i,w}$ denotes the steady-state probability that it experiences a latency of $w$ slots in $A_{i,-1}(t)$. Without prediction, the average latency of the arrival queues in the edge fog tier is $d=\sum_{i\in\mathcal{N}}\lambda_i\sum_{w\geq1}w\pi_{i,w}/\sum_{i\in\mathcal{N}}\lambda_i$.

*Theorem 2:* Suppose the system steady-state behavior depends only on the statistical behaviors of the arrivals and service processes. Then the latency reduction $\eta(\boldsymbol{W})$ is

$$\eta\left(\boldsymbol{W}\right)=\frac{\sum_{i\in\mathcal{N}}\lambda_i\left(\sum_{1\leq w\leq W_i}w\pi_{i,w}+W_i\sum_{w\geq1}\pi_{i,w+W_i}\right)}{\sum_{i\in\mathcal{N}}\lambda_i}. \tag{27}$$

Further, if $d<\infty$, as $\boldsymbol{W}\to\infty$, *i.e.*, with more and more predictive information, we have

$$\lim_{\boldsymbol{W}\to\infty}\eta\left(\boldsymbol{W}\right)=d. \tag{28}$$

The proof of Theorem 2 is shown in APPENDIX D.

***Insight:*** As the prediction window size increases, the delay reduction offered by PORA increases. Moreover, the average latency approaches 0 as prediction window size goes to infinity. In practice, often times only limited future information is available. However, we show that the average latency can be effectively reduced with only mild-value of such information in simulations.

## V. Numerical Results

TABLE IV
Simulation Settings

| Parameter | Value |
|---|---|
| $B$ | 2 MHz |
| $H_{i,j},\forall i\in\mathcal{N},j\in\mathcal{M}$ | $24\log_{10}d_{i,j}+20\log_{10}5.8$+60 [a][10] |
| $N_0$ | $-174$ dBm/Hz [10] |
| $P_{i,\max},\forall i\in\mathcal{N}$ | 500 mW |
| $R_{j,c},\forall j\in\mathcal{M}$ | 6 Mb/s [20] |
| $L_i^{(e)}\forall i\in\mathcal{N}$, $L_j^{(c)}\forall j\in\mathcal{M}$ | 297.62 cycles/bit [21] |
| $f_{i,\max}^{(e)},\forall i\in\mathcal{N}$ | 4 G cycles/s |
| $f_{j,\max}^{(c)},\forall j\in\mathcal{M}$ | 8 G cycles/s |
| $\kappa$ | $10^{-27}$ W·s$^3$/cycle$^3$ [10] |
| $b_{i,\max}^{(e,l)},b_{i,\max}^{(e,o)},\forall i\in\mathcal{N}$ | 6 Mb/s |
| $b_{j,\max}^{(c,l)},b_{j,\max}^{(c,o)},\forall j\in\mathcal{M}$ | 12 Mb/s |
| $D_j(t),\forall j\in\mathcal{J},t$ | 6 Mb/s |

[a] $d_{i,j}$ is the distance between EFN $i$ and CFN $j$.

### A. Basic Settings

In our simulations, we consider a hierarchal fog computing system with 80 EFNs and 20 CFNs. All EFNs have a uniform prediction window size $W$, varying from 0 to 30. Note that when $W=0$, we are indeed simulating the case when there is no prediction. The reachable CFN set $\mathcal{M}_i$ of each EFN $i$ is chosen randomly from the 20 CFNs with size $|\mathcal{M}_i|=5$. We set the length of each time slot as $\tau_0=1$ second. In

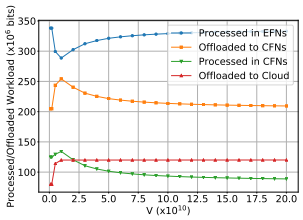
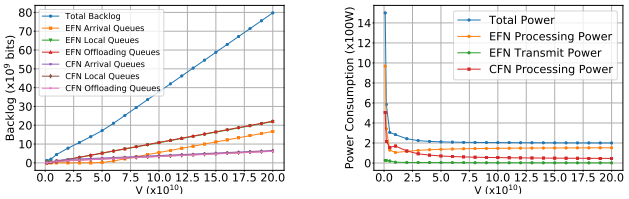
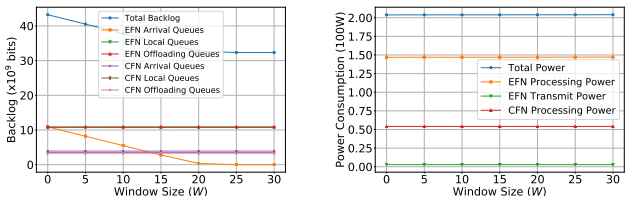



Fig. 5. Offloading decisions vs. $V$ when $W = 10$.



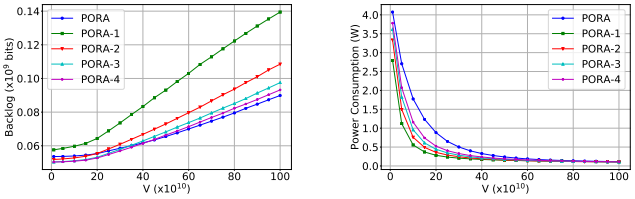

(a) Queue backlog. (b) Power consumption.

Fig. 6. Performance of PORA vs. $V$ when $W = 10$.




(a) Queue backlog . (b) Power consumption.

Fig. 7. Performance of PORA vs. prediction window size when $V = 10^{11}$.




(a) Total queue backlog. (b) Total power consumption.

Fig. 8. Performance of variants of PORA.

each time slot, workloads arrive to the system in the form of packets with a fixed size of 4096 bits. We conduct trace-driven simulations and use arrival properties presented in [22]. We use the distribution of active flow number within one second that has a mean of 538, and choose the flow size distribution which has a mean of 13 Kb. Thus the average arrival rate is about 7 Mb/s. The remaining parameter settings are listed in TABLE IV. All simulation results excluding part V-B4 are averaged over 50000 time slots.

### *B. Evaluation with Perfect Prediction*

In this subsection, we consider how tradeoff parameter $V$ and prediction window size $W$ influence the behavior of PORA under perfect prediction. We also compare PORA with its variants and benchmark policies.

*1) System Performance vs. $V$:* We observe a complex impact of $V$ on the offloading decisions of PORA in Figure 5: When $V = 10^{10}$, the amount of workload processed in EFNs reaches the lowest point, while other offloading decisions reach the highest point. The reason is that the offloading decisions are not only determined by $V$, but also influenced by queue backlogs, as we analyzed previously. Figure 6 shows the impact of $V$ on the different categories of queues and power consumptions in the system. We observe that as $V$ increases, all categories of backlogs increase, but the local processing power consumption and the transit power consumption decrease. This proves the power-delay tradeoff of the system mentioned above.

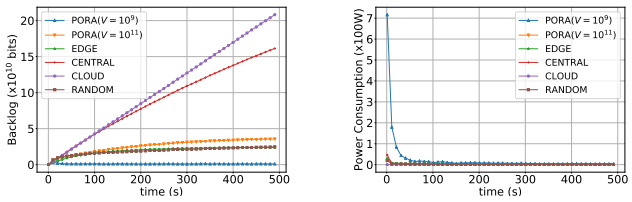
*2) System Performance vs. Prediction Window Size:* In this part we investigate the system performance when the prediction window size $W$ increases from 0 to 30. Figure 7(a) and 7(b) plot the curves of average queue backlog and average power consumption respectively. The result shows that with perfect prediction, we can effectively decrease the average delay of EFN arrival queues to zero with no additional power consumption.

*3) PORA vs. PORA-$d$:* We propose a series of variants of PORA algorithm: PORA-$d$. In PORA-$d$ algorithm, we apply the idea of *The-Power-of-d-Choice* to PORA algorithm to leverage overheads sampling. Particularly, every EFN $i$ chooses $d$ CFNs randomly from the accessible CFN set $\mathcal{M}_i$ to offload. If the size of $\mathcal{M}_i$ is smaller than $d$, then EFN $i$ just chooses all CFNs in $\mathcal{M}_i$.

Figure 8 compares the performance of PORA with PORA-$d$s. We observe that when $V$ is small, PORA-2, PORA-3, and PORA-4 possess smaller backlogs and power consumptions than PORA. However, as $V$ exceeds $4.5 \times 10^{11}$, PORA achieves minimal queue backlog and similar power consumption as PORA-$d$s. Moreover, the gap of their backlog even increases linearly. In summary, in the case of small $V$, PORA-$d$s are better than PORA when considering both the average latency and power consumption, otherwise PORA is the best choice.

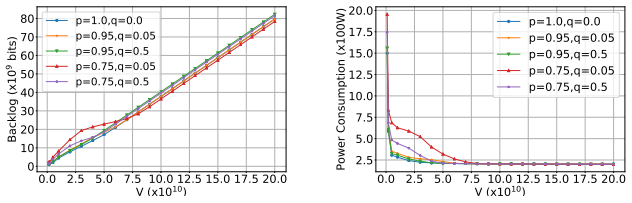
*4) PORA vs. Benchmark Policies:* We introduce four benchmark policies to evaluate the performance of PORA: (1)EDGE: The edge fog tier process all workloads locally. (2)CENTRAL: All workloads are offloaded to the central fog tier for local processing. (3)CLOUD: All workloads are offloaded to the cloud. (4)RANDOM: Each EFN and CFN make offloading decisions randomly for every packets. All these policies pre-serve future workloads in the prediction window. Figure 9 compares the instant total queue backlog and power consumption under the four policies (PORA, EDGE, CENTRAL, CLOUD, RANDOM) in each time slot. Here we set $W = 10$ and choose $V \in \{10^9, 10^{11}\}$.

We observe that CLOUD achieves the minimal power consumption, but incurs infinite queue backlog as the time goes by. As illustrated in Fig. 9, PORA achieves the maximal power consumption but the smallest backlog when $V = 10^9$. When PORA converges, the power consumptions of all these policies reach the same level, but the differences between backlogs become more obvious. In summary, the simulation results demonstrate that with appropriate value of $V$, PORA can achieve much better performance than the four benchmark policies when converges.




(a) Total queue backlog. (b) Total power consumption.

Fig. 9. Comparison between PORA and other policies.




(a) Total queue backlog. (b) Total power consumption.

Fig. 10. Performance of PORA under imperfect prediction.

### C. Evaluation with Imperfect Prediction

In previous simulations we assume the prediction is perfect. But in practice, prediction errors are inevitable. In this part, we use the imperfect prediction model in [23] to investigate how PORA react with prediction errors.

In this model, we focus on two classes of arrivals: true-positive arrivals and false-positive arrivals. A packet is true-positive if it is correctly predicted to arrive. A packet is false-positive if it is predicted to arrive but not arrive actually. For EFN $i$, the number of true arrivals during time slot $t$ is $A_i(t)$. We assume that a $q_i$ $(0 \leq q_i \leq 1)$ fraction of actual arrivals are not predicted. Then the number of true-positive arrivals is $(1-q_i)A_i(t)$. Moreover, we assume that a $p_i$ $(0 \leq p_i \leq 1)$ fraction of predicted arrivals are correct. Then the number of predicted arrivals is $(1-q_i)A_i(t)/p_i$, and the number of false-positive arrivals is $(1-q_i)(1-p_i)A_i(t)/p_i$. The ratio of falsely predicted arrivals and correctly predicted arrivals in the prediction window is $(1-p_i)/p_i$. Note that when $q_i = 0$, $p_i = 1$, the prediction is perfect.

In the simulation, we set $q_i = q$, $p_i = p$ for all EFNs, and simulate five choices of $(q, p)$: $(1.0, 0.0)$, $(0.95, 0.05)$, $(0.95, 0.5)$, $(0.75, 0.05)$, and $(0.75, 0.5)$. Moreover, we set the prediction window size $W = 10$. The simulation results are presented in Figure 10. When $V \leq 6 \times 10^{10}$, the curve of $(p = 0.75, q = 0.05)$ has both largest backlog and most power consumption. Its performance is even worse than the curve of $(p = 0.75, q = 0.5)$, which has the same correct rate but worse miss rate. The reason is that when $p = 0.75$ and $q = 0.05$ the amount of predicted arrivals and the amount of falsely predicted arrivals are both the largest among all five simulated cases, thus the system wastes more power on the falsely predicted arrivals in this case. When $V$ is large, PORA is unwilling to allocate extra resources of fog nodes for predicted arrivals. The most important observation is that when $V$ is large enough ($V \geq 7.5 \times 10^{10}$ in this simulation), our predictive algorithm PORA is robust under prediction error.

## VI. Conclusion

In this paper, we studied the problem of dynamic offloading and resource allocation with prediction in a fog computing system with multiple tiers. By formulating it as a stochastic network optimization problem, we proposed PORA, an efficient online scheme that exploits predictive offloading to minimize power consumption with queue stability guarantee. Our theoretical analysis and trace-driven simulations show that PORA achieves a tunable power-delay tradeoff, while effectively reduces latencies with only mild-value of future information, even in the presence of prediction errors.

## Appendix A
## Proof of Upper Bound of Drift-Plus-Penalty

First, we define Lyapunov function [19] $L(\boldsymbol{Q}(t))$

$$L(\boldsymbol{Q}(t)) \triangleq \frac{1}{2}[\sum_{i\in\mathcal{N}}(Q_i^{(e,a)}(t))^2 + (Q_i^{(e,l)}(t))^2 + (Q_i^{(e,o)}(t))^2 + \sum_{j\in\mathcal{M}}(Q_j^{(c,a)}(t))^2 + Q_j^{(c,l)}(t))^2 + (Q_j^{(c,o)}(t))^2] \quad (29)$$

Next, we define the drift-plus-penalty $\Delta_V L(\boldsymbol{Q}(t))$ as

$$\Delta_V L(\boldsymbol{Q}(t)) \triangleq \Delta L(\boldsymbol{Q}(t)) + V\mathbb{E}\{P(t)|\boldsymbol{Q}(t)\} \quad (30)$$

where $V > 0$ is a tradeoff parameter which will influence the tradeoff between queueing delay and power consumption.

According to definition (29), we have

$$\begin{aligned} &L(\boldsymbol{Q}(t+1)) - L(\boldsymbol{Q}(t)) = \\ &\frac{1}{2}\sum_{i\in\mathcal{N}}\sum_{\beta\in\{a,l,o\}}[(Q_i^{(e,\beta)}(t+1))^2 - (Q_i^{(e,\beta)}(t))^2] \\ &+\frac{1}{2}\sum_{j\in\mathcal{M}}\sum_{\beta\in\{a,l,o\}}[(Q_j^{(c,\beta)}(t+1))^2 - (Q_j^{(c,\beta)}(t))^2]. \end{aligned} \quad (31)$$

By the update functions (6)(10)(12)(13) and (14), there exists a positive constant $M > 0$ such that

$$\begin{aligned} &L(\boldsymbol{Q}(t+1)) - L(\boldsymbol{Q}(t)) \\ &\leq M + \sum_{i\in\mathcal{N}} Q_i^{(e,a)}(t)\left(A_i(t+W_i) - b_i^{(e,a)}(t)\right) \\ &+\sum_{i\in\mathcal{N}} Q_i^{(e,l)}(t)\left(b_i^{(e,l)}(t) - \tau_0\frac{f_i^{(e)}(t)}{L_i^{(e)}}\right) \\ &+\sum_{i\in\mathcal{N}} Q_i^{(e,o)}(t)\left(b_i^{(e,o)}(t) - \sum_{j\in\mathcal{M}_i} R_{i,j}(t)\right) \\ &+\sum_{j\in\mathcal{M}} Q_j^{(c,a)}(t)\left(\sum_{i\in\mathcal{N}_j} R_{i,j}(t) - b_j^{(c,l)}(t) - b_j^{(c,o)}(t)\right) \\ &+\sum_{j\in\mathcal{M}} Q_j^{(c,l)}(t)\left(b_j^{(c,l)}(t) - \tau_0\frac{f_j^{(c)}(t)}{L_j^{(c)}}\right) \\ &+\sum_{j\in\mathcal{M}} Q_j^{(c,o)}(t)\left(b_j^{(c,o)}(t) - D_j(t)\right). \end{aligned} \quad (32)$$

Substitute (32) into (30), we have

$$\begin{aligned}
&\Delta_V L\left(\boldsymbol{Q}\left(t\right)\right) \leq M \\
&M+\sum_{i\in\mathcal{N}} Q_i^{(e,a)}\left(t\right)\mathbb{E}\left\{A_i\left(t+W_i\right)-b_i^{(e,a)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\} \\
&+\sum_{i\in\mathcal{N}} Q_i^{(e,l)}\left(t\right)\mathbb{E}\left\{b_i^{(e,l)}\left(t\right)-\tau_0\frac{f_i^{(e)}\left(t\right)}{L_i^{(e)}}|\boldsymbol{Q}\left(t\right)\right\} \\
&+\sum_{i\in\mathcal{N}} Q_i^{(e,o)}\left(t\right)\mathbb{E}\left\{b_i^{(e,o)}\left(t\right)-\sum_{j\in\mathcal{M}_i} R_{i,j}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\} \\
&+\sum_{j\in\mathcal{M}} Q_j^{(c,a)}\left(t\right)\mathbb{E}\left\{\sum_{i\in\mathcal{N}_j} R_{i,j}\left(t\right)-b_j^{(c,l)}\left(t\right)-b_j^{(c,o)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\} \\
&+\sum_{j\in\mathcal{M}} Q_j^{(c,l)}\left(t\right)\mathbb{E}\left\{b_j^{(c,l)}\left(t\right)-\tau_0\frac{f_j^{(c)}\left(t\right)}{L_j^{(c)}}|\boldsymbol{Q}\left(t\right)\right\} \\
&+\sum_{j\in\mathcal{M}} Q_j^{(c,o)}\left(t\right)\mathbb{E}\left\{b_j^{(c,o)}\left(t\right)-D_j\left(t\right)|\boldsymbol{Q}\left(t\right)\right\}
\end{aligned} \tag{33}$$

Then we can obtain the upper bound of $\Delta_V L\left(\boldsymbol{Q}\left(t\right)\right)$ by (7)(16):

$$\begin{aligned}
&\Delta_V L\left(\boldsymbol{Q}\left(t\right)\right) \leq M \\
&+\sum_{i\in\mathcal{N}} Q_i^{(e,a)}\left(t\right)\mathbb{E}\left\{A_i\left(t+W_i\right)|\boldsymbol{Q}\left(t\right)\right\} \\
&+\sum_{i\in\mathcal{N}} \mathbb{E}\left\{\left(Q_i^{(e,l)}\left(t\right)-Q_i^{(e,a)}\left(t\right)\right)b_i^{(e,l)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\} \\
&+\sum_{i\in\mathcal{N}} \mathbb{E}\left\{\left(Q_i^{(e,o)}\left(t\right)-Q_i^{(e,a)}\left(t\right)\right)b_i^{(e,o)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\} \\
&+\sum_{i\in\mathcal{N}} \mathbb{E}\left\{V\tau_0\kappa\left(f_i^{(e)}\left(t\right)\right)^3-\frac{\tau_0 Q_i^{(e,l)}\left(t\right)}{L_i^{(e)}}f_i^{(e)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\} \\
&+\sum_{j\in\mathcal{M}} \mathbb{E}\left\{\left(Q_j^{(c,l)}\left(t\right)-Q_j^{(c,a)}\left(t\right)\right)b_j^{(c,l)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\} \\
&+\sum_{j\in\mathcal{M}} \mathbb{E}\left\{\left(Q_j^{(c,o)}\left(t\right)-Q_j^{(c,a)}\left(t\right)\right)b_j^{(c,o)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\} \\
&+\sum_{j\in\mathcal{M}} \mathbb{E}\left\{V\tau_0\kappa\left(f_j^{(c)}\left(t\right)\right)^3-\frac{\tau_0 Q_j^{(c,l)}\left(t\right)}{L_j^{(c)}}f_j^{(c)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\} \\
&+\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}_i} \mathbb{E}\{V\tau_0 p_{i,j}\left(t\right) \\
&\qquad -\tau_0 m_{i,j}\left(t\right)\log_2\left(1+l_{i,j}\left(t\right)p_{i,j}\left(t\right)\right)|\boldsymbol{Q}\left(t\right)\} \\
&-\sum_{j\in\mathcal{M}} Q_j^{(c,o)}\left(t\right)\mathbb{E}\left\{D_j\left(t\right)|\boldsymbol{Q}\left(t\right)\right\}
\end{aligned} \tag{34}$$

where $m_{i,j}(t) \triangleq (Q_{i,o}(t)-Q_j(t))B$ and $l_{i,j}(t) \triangleq \frac{H_{i,j}(t)}{N_0 B}$ for all $i\in\mathcal{N}, j\in\mathcal{M}_i$.

To develop algorithms that solve problem (19), we need to minimize the upper bound of $\Delta_V L\left(\boldsymbol{Q}\left(t\right)\right)$. However, it is hard to solve the minimization problem with expectation. Thus we instead solve the following deterministic problem in each time slot $t$ and set decisions $(\boldsymbol{b}(t),\boldsymbol{f}(t),\boldsymbol{p}(t))$ to its optimal solution:

$$\begin{aligned}
\min_{\boldsymbol{b},\boldsymbol{f},\boldsymbol{p}} &\sum_{i\in\mathcal{N}}\left(Q_i^{(e,l)}\left(t\right)-Q_i^{(e,a)}\left(t\right)\right)b_i^{(e,l)} \\
&+\sum_{i\in\mathcal{N}}\left(Q_i^{(e,o)}\left(t\right)-Q_i^{(e,a)}\left(t\right)\right)b_i^{(e,o)} \\
&+\sum_{i\in\mathcal{N}} V\tau_0\kappa\left(f_i^{(e)}\right)^3-\frac{\tau_0 Q_i^{(e,l)}\left(t\right)}{L_i^{(e)}}f_i^{(e)} \\
&+\sum_{j\in\mathcal{M}}\left(Q_j^{(c,l)}\left(t\right)-Q_j^{(c,a)}\left(t\right)\right)b_j^{(c,l)} \\
&+\sum_{j\in\mathcal{M}}\left(Q_j^{(c,o)}\left(t\right)-Q_j^{(c,a)}\left(t\right)\right)b_j^{(c,o)} \\
&+\sum_{j\in\mathcal{M}} V\tau_0\kappa\left(f_j^{(c)}\right)^3-\frac{\tau_0 Q_j^{(c,l)}\left(t\right)}{L_j^{(c)}}f_j^{(c)} \\
&+\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}} V\tau_0 p_{i,j}-\tau_0 m_{i,j}\left(t\right)\log_2\left(1\right. \\
&\qquad\qquad \left.+l_{i,j}\left(t\right)p_{i,j}\right) \\
\text{s.t. } &(1)(8)(9)(11)(15)
\end{aligned} \tag{35}$$

Problem (35) can be decomposed into subproblems.

## Appendix B
## Proof of Algorithm

### A. Proof of local CPU-cycle frequency setting

To solve for the optimal solution of subproblem (22), we denote its objective function as

$$F_k^{(\alpha,t)}\left(f_k^{(\alpha)}\right) \triangleq V\kappa\left(f_k^{(\alpha)}\right)^3-\frac{Q_k^{(\alpha,a)}\left(t\right)}{L_k^{(\alpha)}}f_k^{(\alpha)}. \tag{36}$$

We can get its first- and second-order derivatives respectively:

$$\frac{dF_k^{(\alpha,t)}\left(f_k^{(\alpha)}\right)}{df_k^{(\alpha)}} = 3V\kappa\left(f_k^{(\alpha)}\right)^2-\frac{Q_i^{(e,a)}\left(t\right)}{L_i^{(\alpha)}}, \tag{37}$$

$$\frac{d^2F_k^{(\alpha,t)}\left(f_k^{(\alpha)}\right)}{\left(df_k^{(\alpha)}\right)^2} = 6V\kappa f_k^{(\alpha)}. \tag{38}$$

From above two derivatives we can conclude that function $F_k^{(\alpha,t)}(\cdot)$ is convex in region $[0, f_{k,\max}]$ since $\frac{d^2F_k^{(\alpha,t)}\left(f_k^{(\alpha)}\right)}{df_i^2} \geq 0$ whenever $f_k^{(\alpha)} \geq 0$. On the other hand, $\frac{dF_k^{(\alpha,t)}(\cdot)}{df_k}=0$ at point $\sqrt{\frac{Q_k^{(e,a)}(t)}{3V\kappa L_k^{(\alpha)}}}$. Thus the minimal point of $F_i^{(\alpha,t)}(\cdot)$ in region $[0, f_{k,\max}^{(\alpha)}]$ is $\min\left\{\sqrt{\frac{Q_k^{(\alpha,a)}(t)}{3V\kappa L_k^{(\alpha)}}}, f_{k,\max}^{(\alpha)}\right\}$.

### B. Proof of offloading decisions for edge fog tier

We denote the objective function of subproblem (24) as $G_i^{(t)}\left(\boldsymbol{p}_i\right)$, and we de note function

$$G_{i,j}^{(t)}\left(p_{i,j}\right) \triangleq Vp_{i,j}-m_{i,j}\left(t\right)\log_2\left(1+l_{i,j}\left(t\right)p_{i,j}\right) \tag{39}$$

for each $j \in \mathcal{M}_i$. Then $G_i^{(t)}(\boldsymbol{p}_i)$ can be expressed as

$$G_i^{(t)}(\boldsymbol{p}_i) = \sum_{j \in \mathcal{M}_i} G_{i,j}^{(t)}(p_{i,j}). \tag{40}$$

We denote the minimizer of function $G_{i,j}^{(t)}(\cdot)$ in region $[0,\infty)$ as $\tilde{p}_{i,j}^{(t)}$, *i.e.*,

$$\tilde{p}_{i,j}^{(t)} \triangleq \arg\min_{p_{i,j} \geq 0} G_{i,j}^{(t)}(p_{i,j}). \tag{41}$$

When $m_{i,j}(t) \leq 0$, $G_{i,j}^{(t)}(\cdot)$ is increasing over region $[0.\infty)$ and $\tilde{p}_{i,j}^{(t)} = 0$. When $m_{i,j}(t) > 0$, $G_{i,j}^{(t)}(\cdot)$ is convex in region $[0,\infty)$ since its second-order derivative satisfies

$$\frac{d^2 G_{i,j}^{(t)}(p_{i,j})}{dp_{i,j}^2} = \frac{m_{i,j}(t)\left(l_{i,j}(t)\right)^2}{\left(1 + l_{i,j}(t)\, p_{i,j}\right)^2} > 0. \tag{42}$$

Thus we can get $\tilde{p}_{i,j}^{(t)}$ by compute its first-order derivative

$$\frac{dG_{i,j}^{(t)}(p_{i,j})}{dp_{i,j}}\Big|_{p_{i,j}=\tilde{p}_{i,j}^{(t)}} = V - \frac{m_{i,j}(t)\, l_{i,j}(t)}{1 + l_{i,j}(t)\, \tilde{p}_{i,j}^{(t)}}. \tag{43}$$

It follows that

$$\tilde{p}_{i,j}^{(t)} = \left[\frac{m_{i,j}(t)}{V} - \frac{1}{l_{i,j}(t)}\right]^+. \tag{44}$$

If $\sum_{j \in \mathcal{M}_i} \tilde{p}_{i,j}^{(t)} \leq p_{i,\max}$, we get $\boldsymbol{p}_i^*(t) = \tilde{\boldsymbol{p}}_i^{(t)}$ directly. Otherwise, we have the following lemma.

*Lemma 1:* If $\sum_{j \in \mathcal{M}_i} \tilde{p}_{i,j}^{(t)} > p_{i,\max}$. $\boldsymbol{p}_i^*(t)$ must satisfy $\sum_{j \in \mathcal{M}_i} p_{i,j}^*(t) = p_{i,\max}$.

*Proof 1:* We prove it using contradiction. Suppose on the contrary that there exists an $\theta_1 > 0$ such that $\sum_{j \in \mathcal{M}_i} p_{i,j}^*(t) + \theta_1 = p_{i,\max}$. Since $\sum_{j \in \mathcal{M}_i} \tilde{p}_{i,j}^{(t)} > p_{i,\max}$, there must exists $j' \in \mathcal{M}_i$ and $\theta_2 > 0$ such that $p_{i,j'}^*(t) < \tilde{p}_{i,j'}^{(t)} - \theta_2$. Now we consider a feasible solution $\boldsymbol{p}_i^0(t)$ of subproblem (24) which satisfies

$$\begin{aligned} p_{i,j'}^0(t) &= p_{i,j'}^*(t) + \theta_3, \\ p_{i,j}^0(t) &= p_{i,j}^*(t), \ \forall j \in \mathcal{M}_i / j', \end{aligned} \tag{45}$$

where $\theta_3 \in (0, \min(\theta_1, \theta_2)]$. Note that $\boldsymbol{p}_i^0(t)$ is feasible because

$$\begin{aligned} \sum_{j \in \mathcal{M}_i} p_{i,j}^0(t) &= \sum_{j \in \mathcal{M}_i} p_{i,j}^*(t) + \theta_3 \\ &\leq \sum_{j \in \mathcal{M}_i} p_{i,j}^*(t) + \theta_1 = p_{i,\max}. \end{aligned} \tag{46}$$

By the definition of $p_{i,j'}^0(t)$ we have

$$p_{i,j'}^*(t) < p_{i,j'}^0(t) < \tilde{P}_{i,j'}^{(t)}. \tag{47}$$

Since $\tilde{p}_{i,j'}^{(t)}$ minimizes $G_{i,j'}^{(t)}(\cdot)$ and $G_{i,j'}^{(t)}(\cdot)$ is either a convex or strictly increasing function (depends on the value of $m_{i,j}(t)$), we have

$$G_{i,j'}^{(t)}\left(p_{i,j'}^*(t)\right) > G_{i,j'}^{(t)}\left(p_{i,j'}^0(t)\right) > G_{i,j'}^{(t)}\left(\tilde{p}_{i,j'}^{(t)}\right). \tag{48}$$

It follows that

$$G_i^{(t)}(\boldsymbol{p}_i^*(t)) > G_i^{(t)}\left(\boldsymbol{p}_i^0(t)\right), \tag{49}$$

which contradict to the fact that $\boldsymbol{p}_i^*(t)$ is the optimal solution of (24). Thus $\alpha$ must be 0. The proof is done.

To find the optimal solution of problem (24) when $\sum_{j \in \mathcal{M}_i} \tilde{p}_{i,j}^{(t)} > p_{i,\max}$, we also need the following lemma.

*Lemma 2:* For any $j \in \mathcal{M}_i$, if $m_{i,j}(t) \leq \frac{V}{l_{i,j}(t)}$, then $p_{i,j}^*(t) = \tilde{p}_{i,j}^{(t)} = 0$.

*Proof 2:* By (44), $\tilde{p}_{i,j}^{(t)} = 0$ if and only if $m_{i,j}(t) \leq \frac{V}{l_{i,j}(t)}$. Now we prove that if there exists a central fog node $j'$ such that $m_{i,j}(t) \leq \frac{V}{l_{i,j}(t)}$, then the optimal $p_{i,j'}^*(t)$ must be 0. We use contradiction to prove the conclusion:

Assume optimal $p_{i,j'}^*(t) > 0$, then there must exists a feasible solution $\boldsymbol{p}_i^1(t)$ in which $p_{i,j}^1(t) = p_{i,j}^*(t)$ for all $j \in \mathcal{M}_i / j'$ and $p_{i,j'}^1(t) = 0$ such that

$$\begin{aligned} G_i^{(t)}(\boldsymbol{p}_i^*(t)) - G_i\left(\boldsymbol{p}_i^1(t)\right) &= V p_{i,j'}^*(t) \\ &\quad - m_{i,j}(t) \log_2\left(1 + l_{i,j}(t)\, p_{i,j'}^*(t)\right). \end{aligned} \tag{50}$$

If $m_{i,j}(t) \leq 0$, then by $p_{i,j'}^*(t) > 0$ we have

$$G_i^{(t)}(\boldsymbol{p}_i^*(t)) - G_i\left(\boldsymbol{p}_i^1(t)\right) > 0. \tag{51}$$

If $0 < m_{i,j'}(t) \leq \frac{V}{l_{i,j'}(t)}$, then we have

$$\begin{aligned} &G_i(\boldsymbol{p}_i^*(t)) - G_i\left(\boldsymbol{p}_i^1(t)\right) \\ &\quad = G_{i,j'}^{(t)}\left(p_{i,j'}^*(t)\right) > G_{i,j'}^{(t)}\left(\tilde{p}_{i,j'}^{(t)}(t)\right). \end{aligned} \tag{52}$$

The inequality holds because $\tilde{p}_{i,j'}^{(t)}(t) = 0 < p_{i,j'}^*(t)$ is the only minimizer of $G_{i,j'}^{(h)}(\cdot)$ over $[0,\infty)$. Thus we get the contradiction, *i.e.*, for any $j$ with $m_{i,j}(t) \leq \frac{V}{l_{i,j}(t)}$, the optimal $p_{i,j}^*(t)$ must be 0. Further, we can get the conclusion that $p_{i,j}^*(t) = 0$ whenever $\tilde{p}_{i,j}^{(t)} = 0$. The proof is done.

Denote set $\mathcal{M}_i^+ \triangleq \left\{j | j \in \mathcal{M}_i, m_{i,j}(t) > \frac{V}{l_{i,j}(t)}\right\}$. Based on Lemma 1 and Lemma 2, when $\sum_{j \in \mathcal{M}_i} \tilde{p}_{i,j}^{(t)} > p_{i,\max}$, we just need to solve the following problem for $\left(p_{i,j}^*(t)\right)_{j \in \mathcal{M}_i^+}$:

$$\begin{aligned} \min_{(p_{i,j})_{j \in \mathcal{M}_i^+}} & \sum_{j \in \mathcal{M}_i^+} V p_{i,j} - m_{i,j}(t) \log_2(1 + l_{i,j}(t)\, p_{i,j}) \\ \text{s.t.} & \sum_{j \in \mathcal{M}_i^+} p_{i,j} = P_{i,\max}, \\ & p_{i,j} \geq 0 \ \forall j \in \mathcal{M}_i^+. \end{aligned} \tag{53}$$

Since $\left(p_{i,j}^*(t)\right)_{j \in \mathcal{M}_i^+}$ is the optimal solution of problem (53), it must satisfy its KKT conditions:

$$V - \frac{m_{i,j}(t)\, l_{i,j}(t)}{1 + l_{i,j}(t)\, p_{i,j}^*(t)} + \lambda^* - \mu_j^* = 0, \ \forall j \in \mathcal{M}_i^+, \tag{54}$$

$$\mu_j^* p_{i,j}^*(t) = 0, \ \forall j \in \mathcal{M}_i^+, \tag{55}$$

$$\lambda^*, \mu_j^* \geq 0, \ \forall j \in \mathcal{M}_i, \tag{56}$$

$$\sum_{j \in \mathcal{M}_i} p_{i,j}^*(t) = p_{i,\max}, \tag{57}$$

$$p_{i,j}^*(t) \geq 0. \tag{58}$$

where $\lambda^*$ and $\left(\mu_j^*\right)_{j \in \mathcal{M}_i^+}$ are the corresponding optimal dual variables. Multiply both sides of (54) by $p_{i,j}^*(t)$, we have

$$\left(V-\frac{m_{i,j}\left(t\right)l_{i,j}\left(t\right)}{1+l_{i,j}\left(t\right)p_{i,j}^{*}\left(t\right)}+\lambda^{*}\right)p_{i,j}^{*}\left(t\right) - \mu_{j}^{*}p_{i,j}^{*}\left(t\right)=0. \quad (59)$$

It follows by (55) that

$$\left(V-\frac{m_{i,j}\left(t\right)l_{i,j}\left(t\right)}{1+l_{i,j}\left(t\right)p_{i,j}^{*}\left(t\right)}+\lambda^{*}\right)p_{i,j}^{*}\left(t\right)=0. \quad (60)$$

On the other hand, by (54) and (56), we have

$$\lambda^{*}=\frac{m_{i,j}\left(t\right)l_{i,j}\left(t\right)}{1+l_{i,j}\left(t\right)p_{i,j}^{*}\left(t\right)}-V+\mu_{i}^{*} \geq\frac{m_{i,j}\left(t\right)l_{i,j}\left(t\right)}{1+l_{i,j}\left(t\right)p_{i,j}^{*}\left(t\right)}-V. \quad (61)$$

For any $j\in\mathcal{M}_{i}^{+}$, we consider two cases:

1) If $\lambda^{*}<m_{i,j}\left(t\right)l_{i,j}\left(t\right)-V$, then (61) holds only if $p_{i,j}^{*}(t)>0$. By (60) it implies that
$$\lambda^{*}=\frac{m_{i,j}\left(t\right)l_{i,j}\left(t\right)}{1+l_{i,j}\left(t\right)p_{i,j}^{*}\left(t\right)}-V. \quad (62)$$
Solving for $p_{i,j}^{*}(t)$ we conclude that $p_{i,j}^{*}\left(t\right)=\frac{m_{i,j}(t)}{V+\lambda^{*}}-\frac{1}{l_{i,j}(t)}$.
2) If $\lambda^{*}\geq m_{i,j}\left(t\right)l_{i,j}\left(t\right)-V$, then $p_{i,j}^{*}(t)>0$ is impossible because it would imply $\left(V-\frac{m_{i,j}(t)l_{i,j}(t)}{1+l_{i,j}(t)p_{i,j}^{*}(t)}+\lambda^{*}\right)p_{i,j}^{*}\left(t\right)>0$, which violates condition (60). Thus $p_{i,j}^{*}(t)=0$ if $\lambda^{*}\geq m_{i,j}\left(t\right)l_{i,j}\left(t\right)-V$.

In summary, we have

$$p_{i,j}^{*}\left(t\right)= \begin{cases}\frac{m_{i,j}(t)}{V+\lambda^{*}}-\frac{1}{l_{i,j}(t)}, & \text{if } \lambda^{*}<m_{i,j}\left(t\right)l_{i,j}\left(t\right)-V;\\ 0, & \text{if } \lambda^{*}\geq m_{i,j}\left(t\right)l_{i,j}\left(t\right)-V.\end{cases} \quad (63)$$

More simply, $p_{i,j}^{*}(t)$ can be expressed as

$$p_{i,j}^{*}\left(t\right)=\left[\frac{m_{i,j}\left(t\right)}{V+\lambda^{*}}-\frac{1}{l_{i,j}\left(t\right)}\right]^{+}. \quad (64)$$

Note that above expression also applies to the case when $m_{i,j}(t)\leq\frac{V}{l_{i,j}(t)}$. Thus Substitute (64) into condition (57), we obtain

$$\sum_{j\in\mathcal{M}_{i}}\left[\frac{m_{i,j}\left(t\right)}{V+\lambda^{*}}-\frac{1}{l_{i,j}\left(t\right)}\right]^{+}=p_{i,\max}. \quad (65)$$

The lefthand side is a piecewise-linear decreasing function of $\lambda^{*}$, with breakpoints at each $m_{i,j}\left(t\right)l_{i,j}\left(t\right)-V$, so the equation has a unique solution.

## Appendix C
## Proof of Theorem 1

### A. Proof of System stability

Let $\left(\tilde{\boldsymbol{f}},\tilde{\boldsymbol{b}},\tilde{\boldsymbol{p}}\right)$ be one of the S-only predictive algorithm that achieves minimal time-average expectation of power consumption $P_{W}^{*}$ under system stability. Then we have for all $t$,

$$\mathbb{E}\left\{\tilde{b}_{i}^{(e,a)*}\left(t\right)\right\}\geq\mathbb{E}\left\{A_{i}\left(t+W_{i}\right)\right\},\ \forall i\in\mathcal{N}; \quad (66)$$

$$\mathbb{E}\left\{\tau_{0}\frac{\tilde{f}_{i}^{(e)}\left(t\right)}{L_{i}^{(e)}}\right\}\geq\mathbb{E}\left\{\tilde{b}_{i}^{(e,l)}\left(t\right)\right\},\ \forall i\in\mathcal{N}; \quad (67)$$

$$\mathbb{E}\left\{\sum_{j\in\mathcal{M}_{i}}\hat{R}_{i,j}\left(\tilde{p}_{i,j}\left(t\right)\right)\right\}\geq\mathbb{E}\left\{\tilde{b}_{i}^{(e,o)}\left(t\right)\right\},\ \forall i\in\mathcal{N}; \quad (68)$$

$$\mathbb{E}\left\{\tilde{b}_{j}^{(c,l)}\left(t\right)-\tilde{b}_{j}^{(c,o)}\left(t\right)\right\}\geq\mathbb{E}\left\{\sum_{i\in\mathcal{N}_{j}}\hat{R}_{i,j}\left(\tilde{p}_{i,j}\left(t\right)\right)\right\}, \quad (69)$$

$$\forall j\in\mathcal{M}; \quad (70)$$

$$\mathbb{E}\left\{\tau_{0}\frac{\tilde{f}_{i}^{(e)}\left(t\right)}{L_{i}^{(e)}}\right\}\geq\mathbb{E}\left\{\tilde{b}_{i}^{(e,l)}\left(t\right)\right\},\ \forall j\in\mathcal{M}; \quad (71)$$

$$\mathbb{E}\left\{D_{j}\left(t\right)\right\}\geq\mathbb{E}\left\{\tilde{b}_{j}^{(c,o)}\left(t\right)\right\},\ \forall j\in\mathcal{M}; \quad (72)$$

$$\mathbb{E}\left\{\hat{P}\left(\tilde{\boldsymbol{f}}(t),\tilde{\boldsymbol{p}}(t)\right)\right\}=P_{W}^{*}. \quad (73)$$

Since our predictive algorithm using Lyapunov methods minimizes the right hand side of (33) and using the inequalities above, we have the following result

$$\begin{aligned}&\Delta_{V}L\left(\boldsymbol{Q}\left(t\right)\right)\leq M+V\mathbb{E}\left\{\hat{P}\left(\tilde{\boldsymbol{f}}\left(t\right),\tilde{\boldsymbol{p}}\left(t\right)\right)\right\}\\&+\sum_{i\in\mathcal{N}}Q_{i}^{(e,a)}\left(t\right)\mathbb{E}\left\{A_{i}\left(t+W_{i}\right)-\tilde{b}_{i}^{(e,a)}\left(t\right)\right\}\\&+\sum_{i\in\mathcal{N}}Q_{i}^{(e,l)}\left(t\right)\mathbb{E}\left\{\tilde{b}_{i}^{(e,l)}\left(t\right)-\tau_{0}\frac{\tilde{f}_{i}^{(e)}\left(t\right)}{L_{i}^{(e)}}\right\}\\&+\sum_{i\in\mathcal{N}}Q_{i}^{(e,o)}\left(t\right)\mathbb{E}\left\{\tilde{b}_{i}^{(e,o)}\left(t\right)-\sum_{j\in\mathcal{M}_{i}}\hat{R}_{i,j}\left(\tilde{p}_{i,j}\left(t\right)\right)\right\}\\&+\sum_{j\in\mathcal{M}}Q_{j}^{(c,a)}\left(t\right)\mathbb{E}\left\{\sum_{i\in\mathcal{N}_{j}}\hat{R}_{i,j}\left(\tilde{p}_{i,j}\left(t\right)\right)-\tilde{b}_{j}^{(c,l)}\left(t\right)\right.\\&\qquad\left.-\tilde{b}_{j}^{(c,o)}\left(t\right)\right\}\\&+\sum_{j\in\mathcal{M}}Q_{j}^{(c,l)}\left(t\right)\mathbb{E}\left\{\tilde{b}_{j}^{(c,l)}\left(t\right)-\tau_{0}\frac{\tilde{f}_{j}^{(c)}\left(t\right)}{L_{j}^{(c)}}\right\}\\&+\sum_{j\in\mathcal{M}}Q_{j}^{(c,o)}\left(t\right)\mathbb{E}\left\{\tilde{b}_{j}^{(c,o)}\left(t\right)-D_{j}\left(t\right)\right\}\\&\leq M+VP_{W}^{*}\end{aligned} \quad (74)$$

Taking expectation on both sides and substitute definitions (30), we have

$$\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(t+1\right)\right)\right\}-\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(t\right)\right)\right\}+\mathbb{E}\left\{P\left(t\right)\right\}\leq M+VP_{W}^{*}. \quad (75)$$

Sum over time slots $\{0,1,2,...,t-1\}$, we have

$$\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(t\right)\right)\right\}-\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(0\right)\right)\right\}+\sum_{\tau=0}^{t-1}\mathbb{E}\left\{P\left(\tau\right)\right\}\leq\left(M+VP_{W}^{*}\right)t. \quad (76)$$

Divide both sides by $t$ and let $t$ goes to infinity, we have

$$\limsup_{t\to\infty}\frac{\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(t\right)\right)\right\}}{t}-\limsup_{t\to\infty}\frac{\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(0\right)\right)\right\}}{t}$$

$$+\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{P\left(\tau\right)\right\}\leq M+VP_{W}^{*} \quad (77)$$

and the inequality can be further relaxed:

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{P\left(\tau\right)\right\}\leq M+VP_{W}^{*} \quad (78)$$

since by the assumption in Theorem 1 we have $\boldsymbol{Q}(0)$ is bounded, and by the definition of $L(\boldsymbol{Q}(t))$ we know $L(\boldsymbol{Q}(t))\geq 0$. Then by (17) we have

$$\bar{P}\leq M+VP_{W}^{*}\leq M+VP^{*}. \quad (79)$$

### *B. Proof of Time Average Expectation of Power Consumption*

By Slackness Assumption, there must exist an S-only predictive algorithm $(\check{\boldsymbol{f}},\check{\boldsymbol{P}})$ such that

$$\mathbb{E}\left\{\check{b}_{i}^{(e,a)*}\left(t\right)\right\}\geq\mathbb{E}\left\{A_{i}\left(t+W_{i}\right)\right\}+\epsilon,\ \forall i\in\mathcal{N}; \quad (80)$$

$$\mathbb{E}\left\{\tau_{0}\frac{\check{f}_{i}^{(e)}\left(t\right)}{L_{i}^{(e)}}\right\}\geq\mathbb{E}\left\{\check{b}_{i}^{(e,l)}\left(t\right)\right\}+\epsilon,\ \forall i\in\mathcal{N}; \quad (81)$$

$$\mathbb{E}\left\{\sum_{j\in\mathcal{M}_{i}}\hat{R}_{i,j}\left(\check{p}_{i,j}\left(t\right)\right)\right\}\geq\mathbb{E}\left\{\check{b}_{i}^{(e,o)}\left(t\right)\right\}+\epsilon, \quad (82)$$

$$\forall i\in\mathcal{N}; \quad (83)$$

$$\mathbb{E}\left\{\check{b}_{j}^{(c,l)}\left(t\right)-\check{b}_{j}^{(c,o)}\left(t\right)\right\}\geq\mathbb{E}\left\{\sum_{i\in\mathcal{N}_{j}}\hat{R}_{i,j}\left(\check{p}_{i,j}\left(t\right)\right)\right\} \quad (84)$$

$$+\epsilon,\ \forall j\in\mathcal{M}; \quad (85)$$

$$\mathbb{E}\left\{\tau_{0}\frac{\check{f}_{i}^{(e)}\left(t\right)}{L_{i}^{(e)}}\right\}\geq\mathbb{E}\left\{\check{b}_{i}^{(e,l)}\left(t\right)\right\}+\epsilon,\ \forall j\in\mathcal{M}; \quad (86)$$

$$\mathbb{E}\left\{D_{j}\left(t\right)\right\}\geq\mathbb{E}\left\{\check{b}_{j}^{(c,o)}\left(t\right)\right\}+\epsilon,\ \forall j\in\mathcal{M}; \quad (87)$$

$$\mathbb{E}\left\{\hat{P}\left(\tilde{\boldsymbol{f}}(t),\check{\boldsymbol{p}}(t)\right)\right\}=P_{W}^{\epsilon}. \quad (88)$$

Since our predictive algorithm using Lyapunov methods minimizes the right hand side of (33), the following inequality holds

$$\begin{aligned}
&\Delta_{V}L\left(\boldsymbol{Q}\left(t\right)\right)\leq M+V\mathbb{E}\left\{\hat{P}\left(\check{\boldsymbol{f}}\left(t\right),\check{\boldsymbol{p}}\left(t\right)\right)\right\}\\
&+\sum_{i\in\mathcal{N}}Q_{i}^{(e,a)}\left(t\right)\mathbb{E}\left\{A_{i}\left(t+W_{i}\right)-\check{b}_{i}^{(e,a)}\left(t\right)\right\}\\
&+\sum_{i\in\mathcal{N}}Q_{i}^{(e,l)}\left(t\right)\mathbb{E}\left\{\check{b}_{i}^{(e,l)}\left(t\right)-\tau_{0}\frac{\check{f}_{i}^{(e)}\left(t\right)}{L_{i}^{(e)}}\right\}\\
&+\sum_{i\in\mathcal{N}}Q_{i}^{(e,o)}\left(t\right)\mathbb{E}\left\{\check{b}_{i}^{(e,o)}\left(t\right)-\sum_{j\in\mathcal{M}_{i}}\hat{R}_{i,j}\left(\check{p}_{i,j}\left(t\right)\right)\right\}\\
&+\sum_{j\in\mathcal{M}}Q_{j}^{(c,a)}\left(t\right)\mathbb{E}\left\{\sum_{i\in\mathcal{N}_{j}}\hat{R}_{i,j}\left(\check{p}_{i,j}\left(t\right)\right)-\check{b}_{j}^{(c,l)}\left(t\right)\right.\\
&\qquad\left.-\check{b}_{j}^{(c,o)}\left(t\right)\right\}\\
&+\sum_{j\in\mathcal{M}}Q_{j}^{(c,l)}\left(t\right)\mathbb{E}\left\{\check{b}_{j}^{(c,l)}\left(t\right)-\tau_{0}\frac{\check{f}_{j}^{(c)}\left(t\right)}{L_{j}^{(c)}}\right\}\\
&+\sum_{j\in\mathcal{M}}Q_{j}^{(c,o)}\left(t\right)\mathbb{E}\left\{\check{b}_{j}^{(c,o)}\left(t\right)-D_{j}\left(t\right)\right\}\\
&\leq M+VP_{\max}\\
&-\epsilon\sum_{i\in\mathcal{N}}\left(Q_{i}^{(e,a)}\left(t\right)+Q_{i}^{(e,l)}\left(t\right)+Q_{i}^{(e,o)}\left(t\right)\right)\\
&-\epsilon\sum_{j\in\mathcal{M}}\left(Q_{j}^{(c,a)}\left(t\right)+Q_{j}^{(c,l)}\left(t\right)+Q_{j}^{(c,o)}\left(t\right)\right)
\end{aligned} \quad (89)$$

Taking expectation on both sides and substitute definitions (30), we have

$$\begin{aligned}
&\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(t+1\right)\right)\right\}-\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(t\right)\right)\right\}+\mathbb{E}\left\{P\left(t\right)\right\}\\
&\leq M+VP_{\max}\\
&-\epsilon\sum_{i\in\mathcal{N}}\left(Q_{i}^{(e,a)}\left(t\right)+Q_{i}^{(e,l)}\left(t\right)+Q_{i}^{(e,o)}\left(t\right)\right)\\
&-\epsilon\sum_{j\in\mathcal{M}}\left(Q_{j}^{(c,a)}\left(t\right)+Q_{j}^{(c,l)}\left(t\right)+Q_{j}^{(c,o)}\left(t\right)\right)
\end{aligned} \quad (90)$$

Sum over time slots $\{0,1,2,...,t-1\}$ and rearrange the terms, we have

$$\begin{aligned}
&\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(t\right)\right)\right\}-\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(0\right)\right)\right\}+\sum_{\tau=0}^{t-1}\mathbb{E}\left\{P\left(\tau\right)\right\}\\
&+\epsilon\sum_{\tau=0}^{t-1}\sum_{i\in\mathcal{N}}\left(Q_{i}^{(e,a)}\left(\tau\right)+Q_{i}^{(e,l)}\left(\tau\right)+Q_{i}^{(e,o)}\left(\tau\right)\right)\\
&+\epsilon\sum_{\tau=0}^{t-1}\sum_{j\in\mathcal{M}}\left(Q_{j}^{(c,a)}\left(\tau\right)+Q_{j}^{(c,l)}\left(\tau\right)+Q_{j}^{(c,o)}\left(\tau\right)\right)\\
&\leq\left(M+VP_{\max}\right)t.
\end{aligned} \quad (91)$$

Divide both sides by $t$ and let $t$ goes to infinity, then by definitions (17) and (18) we have

$$\begin{aligned}
&\limsup_{t\to\infty}\frac{\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(t\right)\right)\right\}}{t}-\limsup_{t\to\infty}\frac{\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(0\right)\right)\right\}}{t}\\
&\qquad+\bar{P}+\epsilon\bar{Q}\leq M+VP_{\max}
\end{aligned} \quad (92)$$

The inequality can be further relaxed:

$$\epsilon \bar{Q} \leq M + VP_{\max} \tag{93}$$

since $\boldsymbol{Q}(0)$ is bounded, $L(\boldsymbol{Q}(t)) \geq 0$, and $P(\tau) \geq 0$. Divide both sides by $\epsilon$ we have

$$\bar{Q} \leq \frac{M + VP_{\max}}{\epsilon}. \tag{94}$$

## APPENDIX D
## PROOF OF THEOREM 2

By the Corollary 1 in [15], we see that the average delay of workload in arrival queue $A_{i,-1}(t)$ of EFN $i$ under our predictive algorithm PORA is

$$d_i^p = \sum_{w \geq 1} w\pi_{i,w+W_i}. \tag{95}$$

According to Little's theorem, the average arrival queue backlog of EFN $i$ under predictive algorithm is

$$q_i^p = \lambda_i d_i^p = \lambda_i \sum_{w \geq 1} w\pi_{i,w+W_i}. \tag{96}$$

Thus the average total arrival queue backlog of all EFNs in the system is

$$q^p = \sum_{i \in \mathcal{N}} q_i^p = \sum_{i \in \mathcal{N}} \lambda_i \sum_{w \geq 1} w\pi_{i,w+W_i}. \tag{97}$$

Similarly, the average total arrival queue backlog of all EFNs without prediction is

$$q = \sum_{i \in \mathcal{N}} q_i^p = \sum_{i \in \mathcal{N}} \lambda_i \sum_{w \geq 1} w\pi_{i,w}. \tag{98}$$

Using (97) and (98), we conclude that

$$\begin{aligned} q - q^p &= \sum_{i \in \mathcal{N}} \lambda_i \left( \sum_{w \geq 1} w\pi_{i,w} - \sum_{w \geq 1} w\pi_{i,w+W_i} \right) \\ &= \sum_{i \in \mathcal{N}} \lambda_i \left( \sum_{w \geq 1} w\pi_{i,w} - \sum_{w \geq 1} (w + W_i)\, \pi_{i,w+W_i} \right. \\ &\quad \left. + \sum_{w \geq 1} W_i \pi_{i,w+W_i} \right) \\ &= \sum_{i \in \mathcal{N}} \lambda_i \left( \sum_{w \geq 1} w\pi_{i,w} - \sum_{w \geq W_i + 1} w\pi_{i,w} \right. \\ &\quad \left. + \sum_{w \geq 1} W_i \pi_{i,w+W_i} \right) \\ &= \sum_{i \in \mathcal{N}} \lambda_i \left( \sum_{1 \leq w \leq W_i} w\pi_{i,w} + W_i \sum_{w \geq 1} \pi_{i,w+W_i} \right) \end{aligned} \tag{99}$$

Divide both sides by $\sum_{i \in \mathcal{N}} \lambda_i$ and using Little's theorem, we have

$$\begin{aligned} dr(\boldsymbol{W}) &= \frac{q}{\sum_{i \in \mathcal{N}} \lambda_i} - \frac{q^p}{\sum_{i \in \mathcal{N}} \lambda_i} \\ &= \frac{\sum_{i \in \mathcal{N}} \lambda_i \left( \sum_{1 \leq w \leq W_i} w\pi_{i,w} + W_i \sum_{w \geq 1} \pi_{i,w+W_i} \right)}{\sum_{i \in \mathcal{N}} \lambda_i}. \end{aligned} \tag{100}$$

Thus (27) is proven.

Now we prove (28). Taking limit $\boldsymbol{W} \to \infty$, we obtain

$$\lim_{\boldsymbol{W} \to \infty} \sum_{i \in \mathcal{N}} \lambda_i \sum_{1 \leq w \leq W_i} w\pi_{i,w} = q. \tag{101}$$

It follows that

$$\lim_{\boldsymbol{W} \to \infty} dr(\boldsymbol{W}) = d + \lim_{\boldsymbol{W} \to \infty} \frac{\sum_{i \in \mathcal{N}} \lambda_i W_i \sum_{w \geq 1} \pi_{i,w+W_i}}{\sum_{i \in \mathcal{N}} \lambda_i}. \tag{102}$$

On the other hand, by definition and Little's theorem we have

$$\begin{aligned} \lim_{\boldsymbol{W} \to \infty} dr(\boldsymbol{W}) &= \frac{q}{\sum_{i \in \mathcal{N}} \lambda_i} - \lim_{\boldsymbol{W} \to \infty} \frac{q^p}{\sum_{i \in \mathcal{N}} \lambda_i} \\ &\leq \frac{q}{\sum_{i \in \mathcal{N}} \lambda_i}, \end{aligned} \tag{103}$$

Combining (102) and (103), we have

$$\lim_{\boldsymbol{W} \to \infty} \frac{\sum_{i \in \mathcal{N}} \lambda_i W_i \sum_{w \geq 1} \pi_{i,w+W_i}}{\sum_{i \in \mathcal{N}} \lambda_i} = 0 \tag{104}$$

since it can not be negative. Substitute (104) into (102), we have

$$\lim_{\boldsymbol{W} \to \infty} dr(\boldsymbol{W}) = d. \tag{105}$$

## REFERENCES


[1] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proceedings of ACM EuroSys*, 2011.

[2] I. Giurgiu, O. Riva, and G. Alonso, "Dynamic software deployment from clouds to mobile devices," in *Proceedings of Springer Middleware*, 2012.

[3] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO white paper*, 2011.

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

[5] V. B. C. d. Souza, W. Ramírez, X. Masip-Bruin, E. Marín-Tordera, G. Ren, and G. Tashakor, "Handling service allocation in combined fog-cloud scenarios," in *Proceedings of IEEE ICC*, 2016.

[6] Y. Xiao and M. Krunz, "Qoe and power efficiency tradeoff for fog computing networks with fog node cooperation," in *Proceedings of IEEE INFOCOM*, 2017.

[7] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[8] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proceedings of IEEE HotWeb*, 2015.

[9] Y. Mao, J. Zhang, S. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proceedings of IEEE GLOBECOM*, 2016.

[10] C.-F. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *Proceedings of IEEE GLOBECOM*, 2017.

[11] A. Bozorgchenani, D. Tarchi, and G. E. Corazza, "An energy and delay-efficient partial offloading technique for fog computing architectures," in *Proceedings of IEEE GLOBECOM*, 2017.

[12] Y. Nan, W. Li, W. Bao, F. C. Delicato, P. F. Pires, Y. Dou, and A. Y. Zomaya, "Adaptive energy-aware computation offloading for cloud of things systems," *IEEE Access*, vol. 5, pp. 23 947–23 957, 2017.

[13] L. Liu, Z. Chang, and X. Guo, "Socially-aware dynamic computation offloading scheme for fog computing system with energy harvesting devices," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1869–1879, 2018.

[14] J. Broughton, "Netflix adds download functionality," https://technology.ihs.com/586280/netflix-adds-download-support, 2016.

[15] L. Huang, S. Zhang, M. Chen, and X. Liu, "When backpressure meets predictive scheduling," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2237–2250, 2016.

[16] 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.
[17] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proceedings of ACM HotCloud*, 2010.
[18] Y. Kim, J. Kwak, and S. Chong, "Dual-side optimization for cost-delay tradeoff in mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1765–1781, 2018.
[19] 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.
[20] M.-H. Chen, M. Dong, and B. Liang, "Resource sharing of a computing access point for multi-user mobile cloud offloading with delay constraints," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2018.
[21] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1594–1608, 2017.
[22] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of ACM IMC*, 2010.
[23] K. Chen and L. Huang, "Timely-throughput optimal scheduling with prediction," in *Proceedings of IEEE INFOCOM*, 2018.