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

Xin Gao, Xi Huang, Ziyu Shao, *Member, IEEE,* Yang Yang, *Fellow, IEEE*

***Abstract*—Fog computing is a promising paradigm that enables Internet-of-Things (IoT) applications with ultra-low latency and intensive computation. However, it is challenging to make efficient online decisions under varying system dynamics and intertwined power-latency tradeoffs. Moreover, the fundamental limits and benefits of predictive offloading in fog computing systems still remains unknown. In this paper, we study the problem of dynamic workload offloading and resource allocation in multi-tiered fog computing systems. By developing a fine-grained queue model and formulate a stochastic network optimization problem, we propose PORA, an efficient scheme that exploits predictive information to solve the problem. Results from our theoretical analysis and simulations show that PORA achieves near-optimal power consumption with low latency. Furthermore, PORA effectively reduces latencies with only mild-value of predictive information and it's robust against prediction errors.**

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

I. Introduction

With the emergence of IoT applications, cloud computing may fall short to fulfill real-time requirements of such applications, largely due to its distance from user-end. By extending cloud to the edge of network, fog computing comes as a promising complement to meet the stringent requirements of low latency with intensive computation [1], [2].

A typical fog computing system consists of a set of geographically distributed fog nodes. Such nodes are deployed at the network periphery with elastic resource provisioning such as storage, computation, and network [3]. 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, resource-limited edge devices at the user-end, when heavily loaded, can upload and delegate part of their workloads via wireless links to fog nodes nearby, *a.k.a.*, *workload offloading*, to reduce power consumption and accelerate 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 tradeoff, 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 decisions are critical yet challenging to make, due to the highly varying system dynamics in wireless environment, uncertainty in the resulting latency incurred by offloading, and unknown traffic statistics.

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

Recently, many works have contributed to develop effective offloading schemes in fog computing under different cases. Mao *et al.* [4] investigates the power-latency tradeoff in the case of workload offloading between multiple users and one fog node. Liu *et al.* [5] considers the offloading between multiple users and multiple fog nodes. Xiao *et al.* [2] studies the fog computing systems where fog nodes can offload workloads to the cloud. Bozorgchenani *et al.* [6] explores the offloading problem in a two-tiered fog computing system. On the other hand, Nan *et al.* [7] and Liu *et al.* [8] 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. A model that accurately characterizes the system and tradeoffs is the key to the fundamental understanding of the design space.
(b) **Efficient and online decision making:** The decision making must be computationally efficient, so as to minimize the overheads. 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 improve quality of service. For example, Netflix preloads videos onto users' devices based on user behavior prediction [9]. 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 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 the best of our knowledge, we are the first to conduct systematic study on predictive scheduling in fog systems. We compare our work with others in TABLE I and show our main contributions 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 consumptions of fog tiers

TABLE I
COMPARISONS OF RELATED WORKS

	IoT-Fog[1]	Fog-Fog[2]	Fog-Cloud[3]	Dynamic	Arrival Distribution	Prediction
[4]	✓	×	×	✓	Arbitrary	×
[2]	✓	✓	✓	×	–	×
[5]	✓	×	×	✓	Arbitrary	×
[6]	×	✓	✓	×	–	×
[7]	×	×	✓	✓	Poisson	×
[8]	✓	×	✓	✓	Poisson	×
Ours	×	✓	✓	✓	Arbitrary	✓

[1,2,3] "IoT-Fog" means offloading from IoT devices to fog "Fog-Fog" means offloading between fog tiers, while "Fog-Cloud" means offloading from fog to cloud.

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 presents the system model and problem formulation. Section IV proposes 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

This section considers a motivating example that shows the potential power-latency tradeoffs in multi-tiered fog computing systems. The objective is to achieve a low power consumption and short average packet latency.

Figure 1 shows a sample time-slotted fog computing system with two fog tiers, *i.e.*, edge fog tier and central fog tier. Within each fog tier resides one fog node, *i.e.*, an edge fog node (EFN) in edge fog tier and an central fog node (CFN) in central fog tier. The EFN connects to the CFN via a wireless link, while the CFN connects to a cloud data center over wired links. Each fog node maintains one queue to store arriving packets. For example, Fig. 1(a) shows that during time slot t_0, both EFN and CFN store 8 packets in their queues.

To handle the packets, each fog node sticks to one policy all the time, either *processing packets locally* or *offloading them to its next tier (or cloud)*. As for local processing, the processing capacities of EFN and CFN are 1 and 8 packets per time slot, respectively. When it comes to offloading, the maximum transmit rate is 4 packets per time slot from EFN to CFN, and 5 packets per time slot from CFN to cloud.

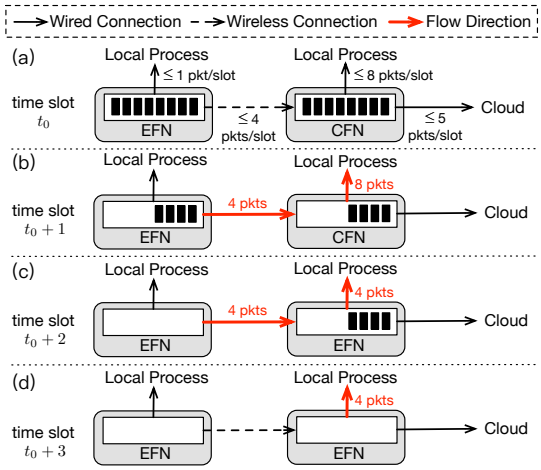


Fig. 1. Motivating example.

The power consumption is assumed linearly proportional to the number of processed/transmitted packets. In particular, processing one packet locally consumes 1 mW power, while transmitting one packet consumes 0.5 mW. Packets that are offloaded to cloud will be processed promptly once delivered.

TABLE II lists the total power consumptions and packets latency of fog nodes under all four possible settings. Figure 1(b)-(d) shows the case when EFN sticks to offloading and CFN sticks to local processing. In time (t_0+1), EFN offloads four packets to CFN at its full transmission rate, while CFN processes all the eight packets locally. In time (t_0+2), EFN offloads the rest four packets to CFN; meanwhile, CFN locally processes the four packets that arrive in previous time slot. In time (t_0+3), CFN finishes processing the rest four packets. In this case, the system consumes 16 mW power in local processing and 4 mW power in transmission, with an average packet latency of 1.75 time slots.

TABLE II
PERFORMANCE UNDER DIFFERENT STRATEGIES

Policy of EFN	Policy of CFN	Total Power Consumptions	Average Packet Latency
Local	Local	16 mW	2.75 slots
Local	Offload	8 mW	2.9375 slots
Offload	Local	20 mW	1.75 slots
Offload	Offload	4 mW	2.125 slots

According to TABLE II, we conclude that: (1) When EFN

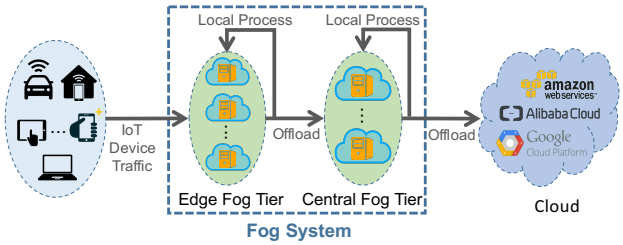


Fig. 2. A sample of Fog computing systems with two fog tiers.

sticks to offloading and CFN sticks to local processing, the system achieves the lowest average packet latency of 1.75 but the maximum power consumption of 20mW. (2) Given the same EFN policy, there is a tradeoff between the total power consumptions and packet latency when CFN sticks to different policies. The reason is that offloading to the cloud can reduce power consumptions but extend latency as well. (3) When CFN sticks to local processing, there is a power-latency tradeoff with different policies at EFN. This is because wireless transmissions consume extra power but processing capacity of CFN is much higher than that of EFN. In summary, there is a potential power-latency tradeoff in both tiers.

III. Model and Problem Formulation

In this section, we consider a hierarchical fog computing system with multiple fog tiers as shown in Figure 2. The system evolves over time slots indexed by $t \in \{0, 1, 2, ...\}$. Each time slot has a length of τ_0. Inside the edge fog tier is a set of edge fog nodes (EFNs) that offer low-latency access to IoT devices. On the other hand, the central fog tier comprises of central fog nodes (CFNs) with greater computing power than EFNs. Due to limited resources, EFNs can offload part of their workloads to CFNs. Likewise, CFNs can also offload to the cloud data center. In our model, we consider only power consumptions and latencies incurred within fog tiers, while the cloud is assumed to have infinite processing capacity with negligible processing latencies. In TABLE III, we summarize the key notations used in this paper.

A. Basic Settings

In general, a 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. All EFNs and CFNs may be distributed in different geographical locations. Under such a restriction, each EFN i can only access to a subset of CFNs, denoted by $\mathcal{M}_i \subset \mathcal{M}$. For each CFN j, we use $\mathcal{N}_j \subset \mathcal{N}$ to denote the set of EFNs that have access to j. Note that for any $i \in \mathcal{N}_j$, we have $j \in \mathcal{M}_i$.

B. Queueing Model for Edge Fog Tier

During time slot t, workloads generated from edge devices arrive at EFN i with an amount of $A_i(t)$ ($\leq A_{\max}$ for some constant $A_{\max}$) such that $\mathbb{E}\{A_i(t)\} = \lambda_i$. We assume that such arrivals are independent over time slots and different EFNs. With the aid of some learning module, each EFN i has access to the future information in a limited lookahead window of size W_i, denoted by $\{A_i(t), A_i(t+1), \ldots, A_i(t+W_i-1)\}$. Besides, it is also assumed applicable to pre-serve [1] workloads before their actual arrivals.

TABLE III
Key notations

Notations	Meanings
τ_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
λ_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
N_0	Power spectral density of the additive white Gaussian noise
$H_{i,j}(t)$	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 consumptions in time slot t

On each EFN, as Figure 3 shows, there are four types of queues during each time slot: Prediction queues with the backlogs as $A_{1,0}(t), ..., A_{i,W_i-1}(t)$, 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 $A_{i,w}(t)$ ($0 \leq w \leq W_i - 1$) stores untreated workloads that will arrive in time slot $(t+w)$. Workloads that actually arrive at EFN i are stored in the arrival queue $A_{i,-1}(t)$, awaiting being forwarded to the local queue $Q_i^{(e,l)}(t)$ or the offloading queue $Q_i^{(e,o)}(t)$. Workloads in $Q_i^{(e,l)}(t)$ will be processed locally by EFN i, while workloads in $Q_i^{(e,o)}(t)$ will be offloaded to CFNs in set $\mathcal{M}_i$.

1) Prediction Queues and Arrival Queues in EFNs: In every time slot, EFN i admits future workloads along with those that actually arrive. We define $\mu_{i,w}(t)$ as the amount of workloads admitted from $A_{i,w}(t)$ in time slot t, for $w \in \{-1, 0, ..., W_i - 1\}$. Such workloads should be dispatched either to the local queue or to the offloading queue. We denote the amounts of workloads dispatched to the local queue and offloading queue as $b_i^{(e,l)}(t)$ and $b_i^{(e,o)}(t)$, respectively, such

[1]Here serving means processing the future workloads locally on edge fog node i or forwarding it to other fog nodes.

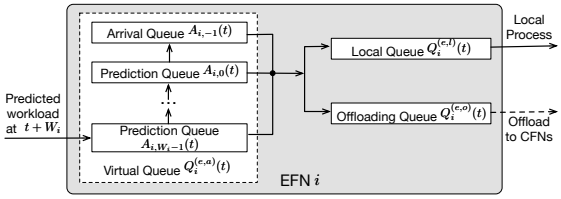


Fig. 3. Prediction queue model: In time slot t, EFN i allocates departure rate among its arrival queue and prediction queues.

that

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

where $b_{i,\max}^{(e,\beta)}$'s are positive constants. As a result, we must have

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

Next, we consider the queueing dynamics for different queues, respectively.

Regarding $A_{i,w}(t)$, it is updated whenever pre-service is finished and the lookahead window moves one slot ahead at the end of each time slot. Therefore, we have

(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 $x \in \mathbb{R}$. In time slot $(t+1)$, the amount of workloads that will arrive after $(W_i - 1)$ time slots is $A_i(t+W_i)$ and it remains unknown until time slot $(t+1)$.

Regarding the arrival queue $A_{i,-1}(t)$, it records the actual backlog of EFN i with the update equation as follows,

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

Note that $\mu_{i,-1}(t)$ denotes the amount of leaving workloads that have already being in the arrival queue $A_{i,-1}(t)$. All the leaving workloads will be dispatched from the queue during time slot t.

Next, we introduce a virtual queue backlog with a size as the sum of all prediction queues and the arrival queue on EFN i, denoted by $Q_i^{(e,a)}(t) \triangleq \sum_{w=-1}^{W_i-1} A_{i,w}(t)$. Under *fully-efficient* [10] service policy, $Q_i^{(e,a)}(t)$ is updated as

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

The input of virtual queue $Q_i^{(e,a)}(t)$ consists of the predicted workloads that will arrive at EFN i in time slot $(t+W_i)$, while its output consists of workloads being dispatched to the local queue and the offloading queue.

2) Offloading Queues in EFNs: In time slot t, workloads in queue $Q_i^{(e,o)}(t)$ will be offloaded to CFNs in set $\mathcal{M}_i$. The amount of offloaded workload is determined by the transmit power decisions $(p_{i,j}(t))_{j\in\mathcal{M}_i}$, where $p_{i,j}(t)$ is the transmit power from EFN i to CFN j. The transmit power is nonnegative and the total transmit power of each EFN is upper bounded, *i.e.*,

$$p_{i,j}(t) \geq 0,\ \forall i \in \mathcal{N}, j \in \mathcal{M}_i, t, \tag{7}$$

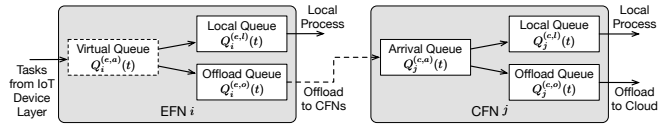


Fig. 4. Queueing model of the system.

$$\sum_{j\in\mathcal{M}_i} p_{i,j}(t) \leq p_{i,\max},\ \forall i \in \mathcal{N}, t. \tag{8}$$

According to Shannon's capacity formula [11], the amount of transmitted workloads from EFN i to CFN j is

$$R_{i,j}(t) \triangleq \hat{R}_{i,j}(p_{i,j}(t)) = \tau_0 B \log_2\left(1 + \frac{p_{i,j}(t) H_{i,j}(t)}{N_0 B}\right), \tag{9}$$

where τ_0 is the length of one time slot, 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. Note that $H_{i,j}(t)$ is an uncontrollable environment state with positive upper bound $H_{\max}$. We assume our system uses Orthogonal Frequency Division Multiplexing (OFDM) technology and the interference among channels is not considered here. Therefore, by adjusting the transmit power $p_{i,j}(t)$, we can offload different amounts of workloads from EFN i to CFN j in time slot t. Accordingly, the update function of offloading 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 arrival may be less than $b_i^{(e,o)}(t)$.

C. Queueing Model for Central Fog Tier

Figure 4 also shows the queueing model on CFN. 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 offloaded from the edge fog tier will be first stored in the arrival queue. Then CFN will dispatch such workloads either to $Q_j^{(c,l)}(t)$ for local processing or to $Q_j^{(c,o)}(t)$, so as to offload them to the cloud later.

1) Arrival Queues in CFNs: The arrivals on CFN j consists of workloads offloaded from all EFNs in the set $\mathcal{N}_j$, with an amount of $\sum_{i\in\mathcal{N}_j} R_{i,j}(t)$. We denote the amounts of workloads dispatched to the local queue and offload queue in time slot t as $b_j^{(c,l)}(t)$ and $b_j^{(c,o)}(t)$, respectively, such that

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

where $b_{j,\max}^{(c,\beta)}$'s are positive constants. Accordingly, $Q_j^{(c,a)}(t)$ is updated as follows

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

2) Offloading Queues in CFNs: For each CFN $j \in \mathcal{M}$, its queue backlog $Q_j^{(c,o)}(t)$ stores the workloads to be offloaded to cloud. We define $D_j(t)$ as the transmission rate of the wired link from CFN j to the cloud during time slot t, which depends

on the network state and satisfies $D_j(t) \leq D_{\max}$ for all j, t. The update function of $Q_j^{(c,o)}(t)$ is thusly

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

Note that the amount of workloads that are actually offloaded to cloud is $\min\{Q_j^{(c,o)}(t), D_j(t)\}$.

D. Local Processing Queues on EFNs and CFNs

Since the local processing queue structures on EFNs and CFNs are identical, we introduce them in a unified way. First, we assume that all fog nodes are able to adjust their CPU frequencies in real time. In practice, such capability can be implemented by applying Dynamic Voltage and Frequency Scaling (DVFS) techniques [12]. Next, we define $L_k^{(\alpha)}$ as the number of CPU cycles that fog node $k \in \mathcal{N} \cup \mathcal{M}$ requires to process one bit of workload, where α is an indicator of the fog node k's category ($\alpha = e$ if k is an EFN, and $\alpha = c$ if k is a CFN). $L_k^{(\alpha)}$ is assumed constant and can be measured offline [13]. Therefore, the local processing capacity of fog node k is $f_k^{(\alpha)}(t)/L_k^{(\alpha)}$. The local processing queue on fog node k evolves as follows,

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

Since CPU-cycle frequencies are nonnegative and finite, we have

$$0 \leq f_k^{(\alpha)}(t) \leq f_{k,\max}^{(\alpha)}, \ \forall k \in \mathcal{N} \cup \mathcal{M}, t \quad (15)$$

where all $f_{k,\max}^{(\alpha)}$'s are positive constants.

E. Power consumptions

We aim at minimizing the total power consumptions $P(t)$ of fog tiers in time slot t, including the power consumption due to local CPU processing and offloading via wireless links. Given a local CPU with frequency f, its power consumption per time slot is ςf^3, where ς is a parameter depending on the hardware architecture that can be measured in practice [14]. Thus $P(t)$ is defined in the following way:

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

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

F. Problem Formulation

We define the long-term time-average expectation of total power consumptions and total queue backlog as follows:

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

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

In this paper, we aim at minimizing $\bar{P}$, while guaranteeing the stability of all queues in the system, *i.e.*, $\bar{Q} < \infty$. Thus the problem is formulated as follows:

$$\begin{aligned} \underset{\boldsymbol{b}(t), \boldsymbol{f}(t), \boldsymbol{p}(t)}{\text{minimize}} \quad & \bar{P} \\ \text{subject to} \quad & (1)(7)(8)(11)(15) \\ & \bar{Q} < \infty \end{aligned} \quad (19)$$

IV. Algorithm Design

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 [15]. Instead of solving the stochastic optimization problem (19) directly, we decouple the problem into a series of per-slot subproblems.

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 workloads assigned to the offloading queue.

We set $b_k^{(\alpha,\beta)}(t)$ 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 workloads as it can from queue $Q_k^{(\alpha,a)}(t)$ to the local/offloading queue $Q_k^{(\alpha,\beta)}(t)$. Otherwise, fog node k will be unwilling to send any workload to the local/offloading queue. The strategy always attempts to achieve the load balancing between the virtual/arrival queue backlog and the local/offloading queue backlog.

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 $f_k^{(\alpha)}(t)$:

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

The optimal solution to (22) is obtained by setting the second derivative of the objective function to zero. Then we set the CPU-cycle frequency of fog node k in time slot t to the optimal solution of (22):

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

Insight: $f_k^{(\alpha)}(t)$ increases proportionally to 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 parameter V. Thus we can increase the value of V when we prefer minimizing power consumption to shortening queueing delay.

We prove the optimality in (23) in Appendix B-A.

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

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

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

We assume that the optimal solution to problem (24) is $\boldsymbol{p}_i^*(t) = (p_{i,j}^*(t))_{j\in\mathcal{M}_i}$. We solve the optimal power allocation decisions using water-filling algorithm.

At first, we solve the equation below

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

for λ^*. 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 λ^*, and ε is the tolerance parameter. Then we obtain 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 $\boldsymbol{p}_i(t) = \boldsymbol{p}_i^*(t)$.

Algorithm 1 Bisection Method for λ^*

1: Initialize $\lambda_{\min} = 0$, $\lambda_{\max} = \max_{j\in\mathcal{M}_i} m_{i,j}(t) l_{i,j}(t) - V$.
2: **while** 1 **do**
3: Set $\lambda^* = (\lambda_{\min} + \lambda_{\max})/2$.
4: **if** $\lambda_{\max} - \lambda_{\min} \le \varepsilon$ **then**
5: Return λ^*.
6: **else**
7: **if** $\sum_{j\in\mathcal{M}_i} \left[\frac{m_{i,j}(t)}{V+\lambda^*} - \frac{1}{l_{i,j}(t)}\right]^+ > p_{i,\max}$ **then**
8: Set $\lambda_{\max} = \lambda^*$.
9: **else**
10: Set $\lambda_{\min} = \lambda^*$.
11: **end if**
12: **end if**
13: **end while**

We prove the optimality of the solution in Appendix B-B. Note that the resulting λ^* converges asymptotically to its optimal value λ^{opt} as ε approaches 0, and the difference between them satisfies $|\lambda^* - \lambda^{\text{opt}}| \le \varepsilon/2$.

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 receive a 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.

We show the pseudocode of PORA in Algorithm 2. Note that α 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-latency 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 workload to CFN j.

b) Power-latency Tradeoffs: First, we consider the case when there is no offloaded workload 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-latency 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, leading to the decrease of total power consumptions. 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, leading to the increase of $Q_j^{(c,a)}(t)$.

B. Computational Complexity of PORA

During each time slot, part of the computational complexity concentrates on the calculation for making the 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 17-25), 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 $O(M \times N)$.

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.

Algorithm 2 Predictive Offloading and Resource Allocation (PORA) in One Time Slot

Initialize $\boldsymbol{b}(t) = \mathbf{0}$, $\boldsymbol{f}(t) = \mathbf{0}$, $\boldsymbol{p}(t) = \mathbf{0}$.
for each fog node $k \in \mathcal{N} \cup \mathcal{M}$ **do**
  %%Make Offloading Decisions
  if $Q_k^{(\alpha,a)}(t) > Q_k^{(\alpha,l)}(t)$ **then**
    Set $b_k^{(\alpha,l)}(t) = b_{k,\max}^{(\alpha,l)}$.
  end if
  if $Q_k^{(\alpha,a)}(t) > Q_k^{(\alpha,o)}(t)$ **then**
    Set $b_k^{(\alpha,o)}(t) = b_{k,\max}^{(\alpha,o)}$.
  end if
  %%Local CPU Resource Allocation
  Set $f_k^{(\alpha)}(t) = \min\{\sqrt{Q_k^{(\alpha,l)}(t)/3V\varsigma L_k^{(\alpha)}}, f_{k,\max}^{(\alpha)}\}$.
end for
%%Transmit Power Allocation
for each EFN $i \in \mathcal{N}$ **do**
  Set $\lambda_{\min} = 0$, $\lambda_{\max} = \max_{j\in\mathcal{M}_i} m_{i,j}(t) l_{i,j}(t) - V$.
  while $\lambda_{\max} - \lambda_{\min} > \varepsilon$ **do**
    Set $\lambda^* = (\lambda_{\min} + \lambda_{\max})/2$
    Set $p_{i,j}(t) = B\left[\frac{Q_i^{(e,o)}(t) - Q_j^{(c,a)}(t)}{V+\lambda^*} - \frac{N_0}{H_{i,j}(t)}\right]^+$.
    if $\sum_{j\in\mathcal{M}_i} p_{i,j}(t) > p_{i,\max}$ **then**
      Set $\lambda_{\max} = \lambda^*$.
    else
      Set $\lambda_{\min} = \lambda^*$.
    end if
  end while
end for
Operate according to offloading decision $\boldsymbol{b}(t)$, CPU-cycle frequency $\boldsymbol{f}(t)$, transmit power allocation $\boldsymbol{p}(t)$.

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 arrival 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 when there is no prediction ($W_i = 0$ for all $i \in \mathcal{N}$). Furthermore, the average power consumption $\bar{P}$ asymptotically approaches the optimal value as V increases to infinity. Next, we can see how the barrier of above power-delay tradeoff is broken by prediction.

2) Latency Reduction: We consider the latency reduction incurred by PORA under perfect prediction compared to the non-predictive case. 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 on 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 $d = \sum_{i\in\mathcal{N}} \lambda_i \sum_{w\geq 1} w\pi_{i,w} / \sum_{i\in\mathcal{N}} \lambda_i$.

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

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

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

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

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 latencies of arrival queues in EFT approaches 0 as prediction window sizes go 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

We conduct extensive simulations to evaluate PORA under various settings. This section presents the key results and the corresponding analysis of the simulations. First, we evaluate the performance of PORA under perfect prediction. Specifically, we show how tradeoff parameter V and prediction window size W influence backlogs and power consumptions, explore how the performance of PORA degrades under low sampling overheads, and compare PORA with some baseline policies. Furthermore, we investigate the performance of PORA under imperfect prediction.

TABLE IV
Simulation Settings

Parameter	Value
B	2 MHz
$H_{i,j}(t), \forall i \in \mathcal{N}, j \in \mathcal{M}$	$24\log_{10} d_{i,j} + 20\log_{10} 5.8$+60 [a][5]
N_0	-174 dBm/Hz [5]
$P_{i,\max}, \forall i \in \mathcal{N}$	500 mW
$L_i^{(e)} \forall i \in \mathcal{N}$, $L_j^{(c)} \forall j \in \mathcal{M}$	297.62 cycles/bit [16]
$f_{i,\max}^{(e)}, \forall i \in \mathcal{N}$	4 G cycles/s
$f_{j,\max}^{(c)}, \forall j \in \mathcal{M}$	8 G cycles/s
ς	10^{-27} W·s^3/cycle^3 [5]
$b_{i,\max}^{(e,l)}, b_{i,\max}^{(e,o)}, \forall i \in \mathcal{N}$	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

We prototype 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 by setting

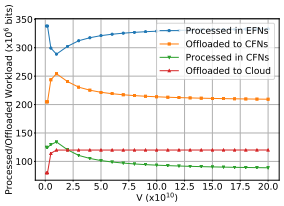
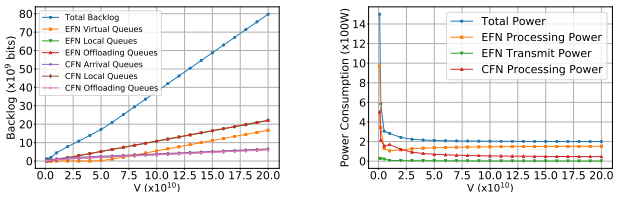
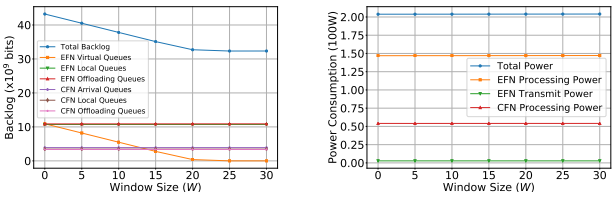


Fig. 5. Offloading decisions vs. V when $W = 10$.



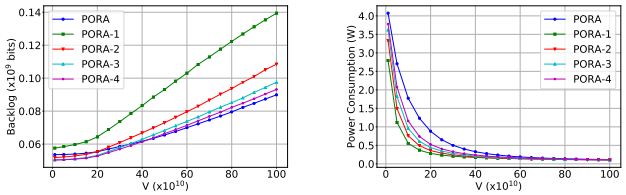
(a) Queue backlog. (b) Power consumption.

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



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

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



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

Fig. 8. Performance of variants of PORA.

$W = 0$, we are actually simulating the case without prediction. For each EFN i, its reachable CFN set $\mathcal{M}_i$ is chosen uniformly randomly from the 20 CFNs with size $|\mathcal{M}_i| = 5$. We then set the time slot length $\tau_0 = 1$ second. During each time slot, workloads arrive to the system in the form of packets, each with a fixed size of 4096 bits. Next, we conduct trace-driven simulations with packet arrival statistics drawn from previous measure work [17], whereby the distribution of flow arrivals has a mean of 538 flows per second, and the distribution of flow size has a mean of 13 Kb. Given these settings, the average arrival rate is about 7 Mb/s. We list the rest parameters in TABLE IV. All simulation results excluding part V-B4 are averaged over 50000 time slots.

B. Evaluation with Perfect Prediction

Under the perfect prediction setting, we evaluate how trade-off parameter V and prediction window size W influence the performance of PORA, respectively, while comparing PORA against its variants and baseline policies.

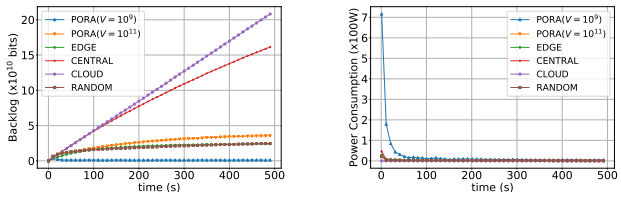
1) System Performance vs. V: Figure 5 shows the impact of parameter V on the offloading decisions of PORA: When $V = 10^{10}$, the time-average amount of locally processed workloads on EFNs reaches the bottom of the curve, while other offloading decisions induce the peak workloads. The reason is that the offloading decisions are not only determined by V, but also influenced by queue backlogs, as analyzed previously. Figure 6 presents the impact of parameter V on the different categories of queues and power consumptions in the system. As V increases, all categories of backlog sizes rise, whereas the local processing power consumption and the transmit power consumption decrease. This conforms to the power-latency tradeoff that we shown in *Theorem 1*.

2) System Performance vs. Prediction Window Size: Figure 7(a) and 7(b) show how the system performance varies as prediction window size W rises from 0 to 30. With perfect prediction, PORA effectively incurs a reduction in the average latency of EFN arrival queues – eventually close to zero with no additional power consumption. Figure 7(a) further shows that PORA reduces the average queueing latency to nearly optimal with only a mild-value of prediction window size ($W = 20$ in this case).

3) PORA vs. PORA-d (Low-Sampling Variants): In practice, PORA may incur considerable sampling overheads since it requires to sample system dynamics across various fog nodes. Therefore, we propose variants of PORA algorithm named PORA-ds. By adopting the idea of randomized load balancing techniques [18], PORA-d reduces the sampling overheads by choosing d CFNs uniformly randomly for each EFN from its accessible CFN set upon decision making. However, for EFN i, if the size of $\mathcal{M}_i$ is smaller than d, then EFN i just samples the whole $\mathcal{M}_i$ set.

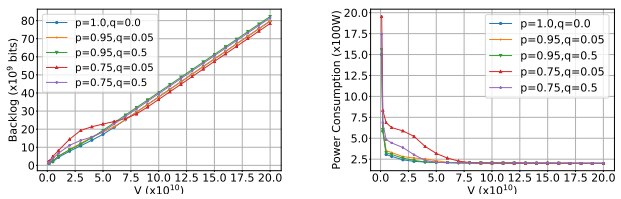
Figure 8 compares the performance of PORA with PORA-ds. We observe that when V is small ($\leq 2 \times 10^{11}$), PORA-2, PORA-3, and PORA-4, achieve smaller backlog sizes and power consumptions than PORA. Once parameter V exceeds 4.5×10^{11}, PORA achieves a minimal queue backlog size and similar power consumption as PORA-ds and the gap of their backlog increases linearly. However, we notice that the percentage-point gap is small: When $V = 10^{12}$, PORA-4 achieves 3.3% more backlog than PORA, and PORA-3 achieves 8.9% more backlog than PORA. In summary, variants PORA-ds (when $d = 2, 3, 4$) effectively decrease the sampling overheads under endurable performance degradations.

4) PORA vs. Benchmark Policies: We introduce four benchmark policies to evaluate the performance of PORA: (1)EDGE: All nodes in the edge fog tier process the workloads locally. (2)CENTRAL: All workloads are offloaded to the central fog tier and processed therein. (3)CLOUD: All workloads are offloaded to the cloud. (4)RANDOM: Each EFN and CFN randomly choose to offload each packet to its upper layer or process it locally with equal chance. Note that all the above policies are also assumed capable of pre-serving future workloads in the prediction window. Figure 9 compares the instant total queue backlog size and power consumptions under the five policies (PORA, EDGE, CENTRAL, CLOUD, RANDOM) in each time slot, where we set $W = 10$ and



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

Fig. 9. Comparison between PORA and other policies.



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

Fig. 10. Performance of PORA under imperfect prediction.

choose $V \in \{10^9, 10^{11}\}$.

We observe that CLOUD achieves the minimum power consumption, but incurs ever-increasing queue backlog size over time. As Fig. 9 illustrates, PORA achieves the maximum power consumption but the smallest backlog size at $V = 10^9$. When PORA converges, the power consumptions under all these policies reach the same level, but the differences between backlogs become more obvious: PORA ($V = 10^9$) reduces 96% of the queue backlog when compared with policy EDGE and RANDOM. In summary, the simulation results demonstrate that with appropriate value of V, PORA can guarantee less latency than the four benchmark policies under the same power consumption when V is chosen appropriately.

C. Evaluation with Imperfect Prediction

Previously, we show results and conduct analysis for cases with perfect prediction. But in practice, prediction errors are usually inevitable. In this part, we investigate the performance of PORA against prediction errors in the cases with imperfect prediction [19].

Particularly, we focus on two kinds of arrivals: true-positive arrivals and false-positive arrivals. A packet is true-positive if it's predicted to arrive and eventually it arrives indeed. A packet is false-positive if it is predicted to arrive but not arrive actually. For EFN i, given the number of its actual arrivals in time t as $A_i(t)$, we assume that a fraction q_i ($0 \le q_i \le 1$) of actual arrivals are missed to predict. Then the number of true-positive arrivals is $(1-q_i)A_i(t)$. Moreover, we assume that a fraction p_i ($0 \le p_i \le 1$) 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 $(1-q_i)(1-p_i)A_i(t)/p_i$. Note that the setting ($q_i = 0$, $p_i = 1$) corresponds to the case when the prediction is perfect.

In our 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 $(0.75, 0.5)$. Figure 10 presents the simulation results when prediction window size $W = 10$. The figure shows that when $V \le 6\times10^{10}$, the average backlog size and power consumption reach the peak value under setting $(p, q) = (0.75, 0.05)$. It is even worse than the case of $(p, q) = (0.75, 0.5)$, with the same correct rate but worse miss rate. The reason is that the amount of predicted arrivals and the amount of falsely predicted arrivals under setting $(p, q) = (0.75, 0.05)$ are both the largest when comparing with other four settings. Thus when $(p, q) = (0.75, 0.05)$, 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 to predicted arrivals, thus the performances under five settings are similar. Notice that when V is large enough ($V \ge 7.5 \times 10^{10}$ in this simulation), our predictive algorithm PORA is robust against 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-latency 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 [15] $L(\boldsymbol{Q}(t))$ as

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

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

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

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

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

Substituting (32) into (30), we obtain

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

Thus we obtain the upper bound of $\Delta_V L\left(\boldsymbol{Q}\left(t\right)\right)$ by (9) and (16):

$$\begin{aligned}
&\Delta_V L\left(\boldsymbol{Q}\left(t\right)\right)\leq M\\
&+\sum_{i\in\mathcal{N}}Q_i^{(e,a)}\left(t\right)\mathbb{E}\left\{A_i\left(t+W_i\right)|\boldsymbol{Q}\left(t\right)\right\}\\
&+\sum_{i\in\mathcal{N}}\mathbb{E}\left\{\left(Q_i^{(e,l)}\left(t\right)-Q_i^{(e,a)}\left(t\right)\right)b_i^{(e,l)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\}\\
&+\sum_{i\in\mathcal{N}}\mathbb{E}\left\{\left(Q_i^{(e,o)}\left(t\right)-Q_i^{(e,a)}\left(t\right)\right)b_i^{(e,o)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\}\\
&+\sum_{i\in\mathcal{N}}\mathbb{E}\left\{V\tau_0\varsigma\left(f_i^{(e)}\left(t\right)\right)^3-\frac{\tau_0Q_i^{(e,l)}\left(t\right)}{L_i^{(e)}}f_i^{(e)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\}\\
&+\sum_{j\in\mathcal{M}}\mathbb{E}\left\{\left(Q_j^{(c,l)}\left(t\right)-Q_j^{(c,a)}\left(t\right)\right)b_j^{(c,l)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\}\\
&+\sum_{j\in\mathcal{M}}\mathbb{E}\left\{\left(Q_j^{(c,o)}\left(t\right)-Q_j^{(c,a)}\left(t\right)\right)b_j^{(c,o)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\}\\
&+\sum_{j\in\mathcal{M}}\mathbb{E}\left\{V\tau_0\varsigma\left(f_j^{(c)}\left(t\right)\right)^3-\frac{\tau_0Q_j^{(c,l)}\left(t\right)}{L_j^{(c)}}f_j^{(c)}\left(t\right)|\boldsymbol{Q}\left(t\right)\right\}\\
&+\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}_i}\mathbb{E}\left\{V\tau_0p_{i,j}\left(t\right)\right.\\
&\qquad\left.-\tau_0m_{i,j}\left(t\right)\log_2\left(1+l_{i,j}\left(t\right)p_{i,j}\left(t\right)\right)|\boldsymbol{Q}\left(t\right)\right\}\\
&-\sum_{j\in\mathcal{M}}Q_j^{(c,o)}\left(t\right)\mathbb{E}\left\{D_j\left(t\right)|\boldsymbol{Q}\left(t\right)\right\}
\end{aligned}\tag{34}$$

where $m_{i,j}(t)\triangleq(Q_{i,o}(t)-Q_j(t))B$ and $l_{i,j}(t)\triangleq\frac{H_{i,j}(t)}{N_0B}$ for all $i\in\mathcal{N},j\in\mathcal{M}_i$.

To solve problem (19), we should 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:

$$\begin{aligned}
\underset{\boldsymbol{b},\boldsymbol{f},\boldsymbol{p}}{\text{Minimize}}\ &\sum_{i\in\mathcal{N}}\left(Q_i^{(e,l)}\left(t\right)-Q_i^{(e,a)}\left(t\right)\right)b_i^{(e,l)}\\
&+\sum_{i\in\mathcal{N}}\left(Q_i^{(e,o)}\left(t\right)-Q_i^{(e,a)}\left(t\right)\right)b_i^{(e,o)}\\
&+\sum_{i\in\mathcal{N}}V\tau_0\varsigma\left(f_i^{(e)}\right)^3-\frac{\tau_0Q_i^{(e,l)}\left(t\right)}{L_i^{(e)}}f_i^{(e)}\\
&+\sum_{j\in\mathcal{M}}\left(Q_j^{(c,l)}\left(t\right)-Q_j^{(c,a)}\left(t\right)\right)b_j^{(c,l)}\\
&+\sum_{j\in\mathcal{M}}\left(Q_j^{(c,o)}\left(t\right)-Q_j^{(c,a)}\left(t\right)\right)b_j^{(c,o)}\\
&+\sum_{j\in\mathcal{M}}V\tau_0\varsigma\left(f_j^{(c)}\right)^3-\frac{\tau_0Q_j^{(c,l)}\left(t\right)}{L_j^{(c)}}f_j^{(c)}\\
&+\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}}V\tau_0p_{i,j}-\tau_0m_{i,j}\left(t\right)\log_2\left(1\right.\\
&\qquad\left.+l_{i,j}\left(t\right)p_{i,j}\right)\\
\text{Subject to}\ &(1)(7)(8)(11)(15)
\end{aligned}\tag{35}$$

Note that Problem (35) can be decomposed into subproblems. ■

Appendix B
Proof of Algorithm

A. Proof of local CPU-cycle frequency setting

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

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

Its first- and second-order derivatives are as follows:

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

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

From the above two derivatives, we 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 $\sqrt{\frac{Q_k^{(e,a)}(t)}{3V\varsigma L_k^{(\alpha)}}}$. Thus the minimal point of $F_i^{(\alpha,t)}(\cdot)$ in region $[0, f_{k,\max}^{(\alpha)}]$ is $\min\left\{\sqrt{\frac{Q_k^{(\alpha,a)}(t)}{3V\varsigma L_k^{(\alpha)}}}, f_{k,\max}^{(\alpha)}\right\}$.

B. Proof of offloading decisions for edge fog tier

We denote the objective function of subproblem (24) by $G_i^{(t)}(\boldsymbol{p}_i)$, defined as

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

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 $[0,\infty)$ by $\tilde{p}_{i,j}^{(t)}$, *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

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

Thus we can obtain $\tilde{p}_{i,j}^{(t)}$ by calculating 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 have $\boldsymbol{p}_i^*(t) = \tilde{\boldsymbol{p}}_i^{(t)}$ 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 $\theta_3 \in (0, \min(\theta_1,\theta_2)]$. Here $\boldsymbol{p}_i^0(t)$ 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 that $\tilde{p}_{i,j'}^{(t)}$ minimizes $G_{i,j'}^{(t)}(\cdot)$ and that $G_{i,j'}^{(t)}(\cdot)$ is either a convex or strictly increasing function (depending 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

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

which contradicts to the fact that $\boldsymbol{p}_i^*(t)$ is the optimal solution to (24). Thus α must be 0. ■

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

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 exist a feasible solution $\boldsymbol{p}_i^1(t)$ such that $p_{i,j}^1(t) = p_{i,j}^*(t)$ for all $j \in \mathcal{M}_i/j'$ and $p_{i,j'}^1(t) = 0$, where

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

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

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

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

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

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 obtain a 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. Furthermore, we conclude that $p_{i,j}^{*}(t) = 0$ whenever $\tilde{p}_{i,j}^{(t)} = 0$. ■

We define $\mathcal{M}_i^{+} \triangleq \left\{ j | j \in \mathcal{M}_i, m_{i,j}(t) > \frac{V}{l_{i,j}(t)} \right\}$. By applying 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 $(p_{i,j}^{*}(t))_{j\in\mathcal{M}_i^{+}}$:

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

Since $(p_{i,j}^{*}(t))_{j\in\mathcal{M}_i^{+}}$ is optimal for problem (53), it must satisfy KKT conditions such that

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

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

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

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

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

where λ^{*} and $(\mu_j^{*})_{j\in\mathcal{M}_i^{+}}$ are the corresponding optimal dual variables. By multiplying both sides of (54) with $p_{i,j}^{*}(t)$, we have

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

Per (55), it follows that

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

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

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

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

1) When $\lambda^{*} < m_{i,j}(t)\, l_{i,j}(t) - V$, then (61) holds only if $p_{i,j}^{*}(t) > 0$. By (60) it implies that

$$\lambda^{*} = \frac{m_{i,j}(t)\, l_{i,j}(t)}{1 + l_{i,j}(t)\, p_{i,j}^{*}(t)} - V. \tag{62}$$

Solving with $p_{i,j}^{*}(t)$ we conclude that $p_{i,j}^{*}(t) = \frac{m_{i,j}(t)}{V+\lambda^{*}} - \frac{1}{l_{i,j}(t)}$.

2) When $\lambda^{*} \geq m_{i,j}(t)\, l_{i,j}(t) - V$, $p_{i,j}^{*}(t) > 0$ can't be true, since it implies that $\left(V - \frac{m_{i,j}(t) l_{i,j}(t)}{1 + l_{i,j}(t) p_{i,j}^{*}(t)} + \lambda^{*} \right) p_{i,j}^{*}(t) > 0$, violating condition (60). Thus $p_{i,j}^{*}(t) = 0$ if $\lambda^{*} \geq m_{i,j}(t)\, l_{i,j}(t) - V$.

In conclusion, we have

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

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

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

Note that above expression is also applicable to the case where $m_{i,j}(t) \leq \frac{V}{l_{i,j}(t)}$. By substituting (64) into (57), we obtain

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

The left-hand-side is a piecewise-linear decreasing function of λ^{*}, with each breakpoint at $(m_{i,j}(t)\, l_{i,j}(t) - V)$. Therefore, 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 the minimum time-average expectation of power consumption P_W^{*} under system stability condition. Hence, for all t,

$$\mathbb{E}\left\{ \tilde{b}_i^{(e,a)*}(t) \right\} \geq \mathbb{E}\left\{ A_i(t + W_i) \right\}, \ \forall i \in \mathcal{N}; \tag{66}$$

$$\mathbb{E}\left\{ \tau_0 \frac{\tilde{f}_i^{(e)}(t)}{L_i^{(e)}} \right\} \geq \mathbb{E}\left\{ \tilde{b}_i^{(e,l)}(t) \right\}, \ \forall i \in \mathcal{N}; \tag{67}$$

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

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

$$\forall j \in \mathcal{M}; \tag{70}$$

$$\mathbb{E}\left\{ \tau_0 \frac{\tilde{f}_i^{(e)}(t)}{L_i^{(e)}} \right\} \geq \mathbb{E}\left\{ \tilde{b}_i^{(e,l)}(t) \right\}, \ \forall j \in \mathcal{M}; \tag{71}$$

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

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

Since the predictive algorithm derived from Lyapunov optimization techniques minimizes the right-hand-side of (33), we have the following result using the above inequalities,

$$\begin{aligned}
&\Delta_V L\left(\boldsymbol{Q}\left(t\right)\right) \leq M + V\mathbb{E}\left\{\hat{P}\left(\tilde{\boldsymbol{f}}\left(t\right), \tilde{\boldsymbol{p}}\left(t\right)\right)\right\} \\
&+ \sum_{i\in\mathcal{N}} Q_i^{(e,a)}\left(t\right)\mathbb{E}\left\{A_i\left(t+W_i\right) - \tilde{b}_i^{(e,a)}\left(t\right)\right\} \\
&+ \sum_{i\in\mathcal{N}} Q_i^{(e,l)}\left(t\right)\mathbb{E}\left\{\tilde{b}_i^{(e,l)}\left(t\right) - \tau_0\frac{\tilde{f}_i^{(e)}\left(t\right)}{L_i^{(e)}}\right\} \\
&+ \sum_{i\in\mathcal{N}} Q_i^{(e,o)}\left(t\right)\mathbb{E}\left\{\tilde{b}_i^{(e,o)}\left(t\right) - \sum_{j\in\mathcal{M}_i}\hat{R}_{i,j}\left(\tilde{p}_{i,j}\left(t\right)\right)\right\} \\
&+ \sum_{j\in\mathcal{M}} Q_j^{(c,a)}\left(t\right)\mathbb{E}\left\{\sum_{i\in\mathcal{N}_j}\hat{R}_{i,j}\left(\tilde{p}_{i,j}\left(t\right)\right) - \tilde{b}_j^{(c,l)}\left(t\right)\right. \\
&\qquad\qquad \left. - \tilde{b}_j^{(c,o)}\left(t\right)\right\} \\
&+ \sum_{j\in\mathcal{M}} Q_j^{(c,l)}\left(t\right)\mathbb{E}\left\{\tilde{b}_j^{(c,l)}\left(t\right) - \tau_0\frac{\tilde{f}_j^{(c)}\left(t\right)}{L_j^{(c)}}\right\} \\
&+ \sum_{j\in\mathcal{M}} Q_j^{(c,o)}\left(t\right)\mathbb{E}\left\{\tilde{b}_j^{(c,o)}\left(t\right) - D_j\left(t\right)\right\} \\
&\leq M + VP_W^*
\end{aligned} \tag{74}$$

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

$$\begin{aligned}
&\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(t+1\right)\right)\right\} - \mathbb{E}\left\{L\left(\boldsymbol{Q}\left(t\right)\right)\right\} + \mathbb{E}\left\{P\left(t\right)\right\} \\
&\qquad\qquad \leq M + VP_W^*.
\end{aligned} \tag{75}$$

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

$$\begin{aligned}
&\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(t\right)\right)\right\} - \mathbb{E}\left\{L\left(\boldsymbol{Q}\left(0\right)\right)\right\} + \sum_{\tau=0}^{t-1}\mathbb{E}\left\{P\left(\tau\right)\right\} \\
&\qquad\qquad \leq \left(M + VP_W^*\right)t.
\end{aligned} \tag{76}$$

Dividing both sides by t, as t approaches to infinity, we have

$$\begin{aligned}
&\limsup_{t\to\infty}\frac{\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(t\right)\right)\right\}}{t} - \limsup_{t\to\infty}\frac{\mathbb{E}\left\{L\left(\boldsymbol{Q}\left(0\right)\right)\right\}}{t} \\
&\qquad + \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{P\left(\tau\right)\right\} \leq M + VP_W^*.
\end{aligned} \tag{77}$$

The inequality can be further relaxed,

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

Based on 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

$$\bar{P} \leq M + VP_W^* \leq M + VP^*. \tag{79}$$

B. Proof of Time Average Expectation of Power Consumption

Given *slackness assumption*, there must exist an S-only predictive algorithm $(\check{\boldsymbol{f}}, \check{\boldsymbol{P}})$ such that

$$\mathbb{E}\left\{\check{b}_i^{(e,a)*}\left(t\right)\right\} \geq \mathbb{E}\left\{A_i\left(t+W_i\right)\right\} + \epsilon,\ \forall i\in\mathcal{N}; \tag{80}$$

$$\mathbb{E}\left\{\tau_0\frac{\check{f}_i^{(e)}\left(t\right)}{L_i^{(e)}}\right\} \geq \mathbb{E}\left\{\check{b}_i^{(e,l)}\left(t\right)\right\} + \epsilon,\ \forall i\in\mathcal{N}; \tag{81}$$

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

$$\forall i\in\mathcal{N}; \tag{83}$$

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

$$+\epsilon,\ \forall j\in\mathcal{M}; \tag{85}$$

$$\mathbb{E}\left\{\tau_0\frac{\check{f}_i^{(e)}\left(t\right)}{L_i^{(e)}}\right\} \geq \mathbb{E}\left\{\check{b}_i^{(e,l)}\left(t\right)\right\} + \epsilon,\ \forall j\in\mathcal{M}; \tag{86}$$

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

$$\mathbb{E}\left\{\hat{P}\left(\check{\boldsymbol{f}}(t), \check{\boldsymbol{p}}(t)\right)\right\} = P_W^{\epsilon}. \tag{88}$$

Since the predictive algorithm derived from Lyapunov optimization techniques minimizes the right-hand-side of (33), the following inequality holds

$$\begin{aligned}
&\Delta_V L\left(\boldsymbol{Q}\left(t\right)\right) \leq M + V\mathbb{E}\left\{\hat{P}\left(\check{\boldsymbol{f}}\left(t\right), \check{\boldsymbol{p}}\left(t\right)\right)\right\} \\
&+ \sum_{i\in\mathcal{N}} Q_i^{(e,a)}\left(t\right)\mathbb{E}\left\{A_i\left(t+W_i\right) - \check{b}_i^{(e,a)}\left(t\right)\right\} \\
&+ \sum_{i\in\mathcal{N}} Q_i^{(e,l)}\left(t\right)\mathbb{E}\left\{\check{b}_i^{(e,l)}\left(t\right) - \tau_0\frac{\check{f}_i^{(e)}\left(t\right)}{L_i^{(e)}}\right\} \\
&+ \sum_{i\in\mathcal{N}} Q_i^{(e,o)}\left(t\right)\mathbb{E}\left\{\check{b}_i^{(e,o)}\left(t\right) - \sum_{j\in\mathcal{M}_i}\hat{R}_{i,j}\left(\check{p}_{i,j}\left(t\right)\right)\right\} \\
&+ \sum_{j\in\mathcal{M}} Q_j^{(c,a)}\left(t\right)\mathbb{E}\left\{\sum_{i\in\mathcal{N}_j}\hat{R}_{i,j}\left(\check{p}_{i,j}\left(t\right)\right) - \check{b}_j^{(c,l)}\left(t\right)\right. \\
&\qquad\qquad \left. - \check{b}_j^{(c,o)}\left(t\right)\right\} \\
&+ \sum_{j\in\mathcal{M}} Q_j^{(c,l)}\left(t\right)\mathbb{E}\left\{\check{b}_j^{(c,l)}\left(t\right) - \tau_0\frac{\check{f}_j^{(c)}\left(t\right)}{L_j^{(c)}}\right\} \\
&+ \sum_{j\in\mathcal{M}} Q_j^{(c,o)}\left(t\right)\mathbb{E}\left\{\check{b}_j^{(c,o)}\left(t\right) - D_j\left(t\right)\right\} \\
&\leq M + VP_{\max} \\
&- \epsilon\sum_{i\in\mathcal{N}}\left(Q_i^{(e,a)}\left(t\right) + Q_i^{(e,l)}\left(t\right) + Q_i^{(e,o)}\left(t\right)\right) \\
&- \epsilon\sum_{j\in\mathcal{M}}\left(Q_j^{(c,a)}\left(t\right) + Q_j^{(c,l)}\left(t\right) + Q_j^{(c,o)}\left(t\right)\right)
\end{aligned} \tag{89}$$

Taking expectation on both sides and substituting definitions

(30), we have

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

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

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

Dividing both sides by t, as t approaches to infinity, with (17) and (18), we have

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

Furthermore, the inequality can be relaxed as

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

Now that $\boldsymbol{Q}(0)$ is bounded, $L(\boldsymbol{Q}(t))\geq 0$, and $P(\tau)\geq 0$. Dividing both sides by ϵ, we have

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

Appendix D
Proof of Theorem 2

By applying the Corollary 1 in [10], 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 size of EFN i under predictive algorithm is

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

The average total arrival queue backlog size of all EFNs 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 predictive scheduling 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

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

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

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

■

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

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

On the other hand, we have

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

By 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't be negative. Substitute (104) into (102), we have

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

■

REFERENCES

[1] 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.

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

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] R. G. Gallager, *Principles of digital communication*. Cambridge University Press Cambridge, UK:, 2008, vol. 1.

[12] 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.

[13] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proceedings of ACM HotCloud*, 2010.

[14] 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.

[15] 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.

[16] 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.

[17] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of ACM IMC*, 2010.

[18] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1094–1104, 2001.

[19] K. Chen and L. Huang, "Timely-throughput optimal scheduling with prediction," in *Proceedings of IEEE INFOCOM*, 2018.