Towards Workload Balancing in Fog Computing Empowered IoT

Citation:
Q. Fan and N. Ansari, "Towards Workload Balancing in Fog Computing Empowered IoT," *IEEE Transactions on Network Science and Engineering*, to be published, DOI: 10.1109/TNSE.2018.2852762.

Towards Workload Balancing in Fog Computing Empowered IoT

Qiang Fan, *Student Member, IEEE*, and Nirwan Ansari, *Fellow, IEEE*

Abstract—As latency is the key performance metric for IoT applications, fog nodes co-located with cellular base stations can move the computing resources close to IoT devices. Therefore, data flows of IoT devices can be offloaded to fog nodes in their proximity, instead of the remote cloud, for processing. However, the latency of data flows in IoT devices consist of both the communications latency and computing latency. Owing to the spatial and temporal dynamics of IoT device distributions, some BSs and fog nodes are lightly loaded, while others, which may be overloaded, may incur congestion. Thus, the traffic load allocation among base stations (BSs) and computing load allocation among fog nodes affect the communications latency and computing latency of data flows, respectively. To solve this problem, we propose a workload balancing scheme in a fog network to minimize the latency of data flows in the communications and processing procedures by associating IoT devices to suitable BSs. We further prove the convergence and the optimality of the proposed workload balancing scheme. Through extensive simulations, we have compared the performance of the proposed load balancing scheme with other schemes and verified its advantages for fog networking.

Index Terms—Fog node, Internet of Things (IoT), workload allocation, user association.

1 Introduction

In the past few years, a tremendous number of smart devices and objects, such as smart phones, wearable devices, industrial and utility components, have been equipped with sensors to sense the real-time physical information from the environment [1]. Hence, Internet of Things (IoT) has been introduced as a concept, where various smart devices are connected with each other via the internet and empowered with data analytics. Various IoT applications, such as smart transportation, smart health, smart city and smart home have been widely studied to improve our daily life [2]. Owing to the high volume and fast velocity of data streams generated by IoT devices, the cloud that can provision flexible and efficient computing resources is employed as a smart "brain" to process and store the big data generated from distributed IoT devices [3], [4]. However, as the data streams generated from IoT devices are transmitted to the remote cloud via Internet, the transferred data may consume a huge amount of bandwidth and energy of the core network [5]. On the other hand, since the remote cloud is usually far from IoT devices, the latency for processing data streams may be too long, especially unbearable for many delay sensitive IoT applications [6]. Therefore, fog nodes, which bring computing resources close to IoT devices and IoT users, can be employed to alleviate the traffic load in the core network and minimize the latency for IoT devices [7], [8].

In a fog network, data flows sensed by IoT devices are transmitted to respective BSs and then processed by fog nodes that are co-located with the BSs. Thus, the latency of each data flow consists of both the communications latency towards the corresponding BS and the computing latency incurred by the respective fog node. Regarding mobile networks, the communications latency of IoT devices' data flows is jointly determined by IoT devices' channel conditions and their BSs' traffic workload status. As the traffic load increases, a BS tends to be congested and thus data flows of IoT devices have to wait for more time to be transmitted. As a result, the traffic load allocation among BSs will significantly affect the delivery time (i.e., communications latency) of data flows. On the other hand, at the side of fog nodes, the computing latency of data flows is directly determined by the computing loads allocated to these fog nodes. Specifically, the heavy computing load of a fog node translates to a longer computing latency. Thus, provided with the dynamic distribution of computing workloads, the load allocation among fog nodes critically impacts the computing latency of all data flows in the network. As each fog node is assumed to be attached to a specific BS in this paper, the workload of a fog node is related to the number of IoT devices associated with its corresponding BS. In other words, when one IoT device is associated with one BS, its data flows are also offloaded to the BS's co-located fog node.

Since adjacent BSs always have overlapped coverage areas, IoT devices in these areas can be associated to suitable BSs in order to balance the loads among BSs; this association critically impacts both the traffic loads of BSs and computing loads of fog nodes. As the latency of each data flow consists of the communications latency and computing latency, both the traffic loads of BSs and computing loads of fog nodes should be taken into consideration in the load balancing process, in order to minimize the latency of data flows. Specifically, owing to the dynamic distribution of IoT devices, when some BSs are overloaded, they will become the bottleneck of the fog network, thus making the communications latency the dominating factor of the latency of data flows; in this case, some IoT devices of these

- *Q. Fan and N. Ansari are with Advanced Networking Lab., Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, 07102 USA (Email: {qf4, nirwan.ansari}@njit.edu).* *This work was supported in part by the National Science Foundation (NSF) under Grant CNS-1647170.*

BSs should be offloaded to other neighboring BSs to mitigate their congested traffic loads. Meanwhile, when some fog nodes are congested, the computing load balancing is more critical, and thus some IoT devices of the BSs co-located with these fog nodes can be assigned to neighboring BSs in order to reduce the computing workloads of these fog nodes. In this case, the computing load balancing may increase the traffic loads of the neighboring BSs, which may in turn degrade the communications latency of all data flows to a certain extent.

To solve the above problem, we propose a LoAd Balancing (LAB) scheme for the fog network to minimize the latency of IoT data flows, by taking into account of both the communications latency and computing latency. Below are major contributions of the paper.

- We formulate the problem of minimizing the latency of all data flows by associating IoT devices with different BSs/fog nodes. The models of both the traffic loads at BSs and computing loads at fog nodes are introduced while the latency ratios (i.e., the amount of time that an IoT flow has to wait to obtain a unit service time) of BSs and fog nodes are adopted to reflect the communications latency in BSs and computing latency in fog nodes, respectively. Moreover, we have also analyzed the impact of load balancing on the average latency of IoT flows.
- To solve the load balancing problem in a fog network, we design a distributed IoT device association scheme (LAB) that assigns IoT devices to suitable BSs/fog nodes to reduce the latency of all data flows. In the scheme, each BS iteratively estimates its traffic load and computing load, and then broadcasts this information. Meanwhile, at the side of IoT devices, they can choose the favorable BSs in each iteration based on the estimated traffic loads and computing loads of BSs/fog nodes. Furthermore, we have proved the convergence and the optimality of the proposed algorithm.

The remainder of this paper is organized as follows. In Section 2, we briefly review related works. In Section 3, we illustrate the fog network architecture and describe the system model. In Section 4, we formulate and analyze the load balancing problem. In Section 5, the LAB algorithm is proposed to obtain the optimal solution of the workload balancing problem. Section 6 shows the simulation results, and concluding remarks are presented in Section 7.

2 RELATED WORKS

Owing to the proximity of fog computing resources to IoT devices and IoT users, some studies have focused on integrating IoT with fog computing. Bonomi *et al.* [9] elicited how fog computing may be applied in various IoT applications. Chiang *et al.* [10] summarized the opportunities and challenges of fog computing in the networking context of IoT and advocated that fog computing can fill the technology gaps in IoT. Sun and Ansari [11] proposed the IoT architecture (EdgeIoT) to handle the data streams from IoT devices at the fog node. Moreover, Jutila [12] proposed adaptive fog computing solutions for IoT networking in order to optimize traffic flows and network resources.

Fog computing, which moves computing resources close to IoT devices or mobile users, has been proposed to improve the performance of IoT applications and mobile applications [13], [14], [15]. To optimize different objectives such as latency and energy consumption of the network, many studies have focused on allocating computing workloads among edge computing resources (fog nodes or cloudlets) without considering the traffic load balancing in mobile networks [16]. Gu *et al.* [17] proposed to integrate fog computing and medical cyber-physical system, and then designed a cost efficient resource management scheme by jointly considering BS association, task distribution and virtual machine placement. Zeng *et al.* [18] proposed to jointly consider the task scheduling and image placement in fog computing based software-defined embedded system to minimize the response time of task requests. Tong *et al.* [19] proposed a workload placement algorithm in a hierarchical edge cloud network in order to optimize the response time of all tasks. The algorithm allocates tasks among different tiers of fog nodes and allocates the computing resources of each fog node for their assigned tasks. Fan *et al.* [20] proposed to migrate mobile users' virtual machines (VM) among distributed cloudlets to reduce the brown energy consumption of cloudlets by jointly considering the green energy generation among cloudlets and energy consumption of VM migrations. Fan and Ansari [21] proposed a workload allocation scheme, referred to as WALL, in a hierarchical cloudlet network to optimize the response time of user tasks. This workload allocation scheme assigns user tasks among different tiers of cloudlets and then allocates computing resources of each cloudlet to their associated users. Moreover, some works [22], [23] look into placing a certain number of edge computing resources among a given set of available sites and then assigning workloads to the edge computing resources based on the real-time requirement. Note that all the above works only consider the wired communications latency, where the wireless delay is neglected. In contrast, other works also consider the impact of wireless delay on the latency of tasks while allocating workloads among edge computing resources. Jia *et al.* [24] proposed a model to place cloudlets in the network and realize the load balancing among the cloudlets to minimize the response time of users. In this paper, the wireless delay for each user is assumed to be constant. Some works proposed to control the transmission power of BSs to adjust the data rate of users in the communications links as well as the workloads among edge computing resources, thus reducing the response time of users [25], [26].

Moreover, many existing works on mobile networks have addressed traffic workload balancing among BSs. Kim *et al.* [27] proposed an iterative distributed user association algorithm to balance the traffic loads among BSs based on different performance metrics. Han and Ansari [28] proposed a traffic workload balancing scheme to make a tradeoff between the traffic delivery time and brown energy consumption in the cellular networks. Fan *et al.* [29] proposed a user association algorithm to improve the flow level throughput and green energy utilization in heterogeneous cellular networks.

As we know, the latency of a data flow sensed by a IoT device consists of both the communications latency and computing latency, which are impacted by the traffic loads of BSs and computing loads of fog nodes. As a result, simply balancing the traffic loads or computing loads is not enough to optimize the response time. However, few works have paid attention on balancing the traffic loads and computing loads simultaneously, and this issue remains an open challenge. Therefore, considering the impact of IoT device-BS association on the average communication latency and computing latency, we propose to jointly balance the traffic loads among BSs and computing loads among fog nodes by associating IoT devices to the optimal BSs. When some BSs are congested by heavy traffic loads, these BSs may become the bottleneck of the fog network, i.e., the communications latency is the dominating factor of the latency of IoT devices' data flows. In this case, IoT devices should be released from these overloaded BSs and re-assigned to lightly loaded BSs in order to ease the traffic congestion. On the other hand, if some fog nodes become the bottleneck of the network, some IoT devices located in the coverage of the fog node/BS should be allocated to the neighboring BSs in order to mitigate the congestion of these fog nodes at the expense of increasing the traffic loads of neighboring BSs that may slightly degrade the communications latency of data flows.

3 System Model

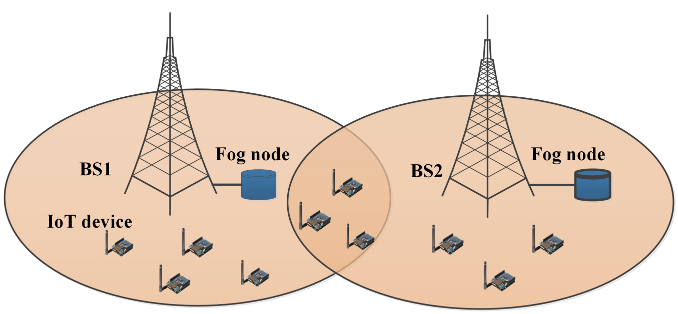


Fig. 1. Fog network architecture.

A fog network architecture is illustrated in Fig. 1, where fog nodes are attached to BSs and neighboring BSs have overlapped coverage areas. Note that all BSs adopt the NB-IoT interface to offer communications services for all IoT devices [11]. In the network, since the workload allocation among fog nodes requires the data flows to go through the mobile cellular core, which incurs additional delay for the IoT flows, the IoT flows are generally preferred to be processed at the local BSs fog node. On the other hand, in the workload allocation among fog nodes, a central controller is required to collect all workload information of both fog nodes and IoT devices in order to execute a centralized algorithm in real time, the complexity of which will be unbearable for large scale networks, e.g., metropolitan area network. Thus, we assume that data flows of an IoT device are processed by the fog node attached to the IoT device's BS instead of other fog nodes. Based on the similar concerns, other existing researches such as [26] also adopt the same assumption. Note that in this case, the computing loads can still be balanced among fog nodes by adjusting IoT device associations among BSs. As the IoT device association is determined by a distributed algorithm run by both the BS and IoT devices, the algorithm has low complexity and is scalable to different networks. Therefore, in this paper, the IoT device association among BSs not only determines the traffic loads among BSs, but also determines the computing loads among fog nodes. Meanwhile, adjacent macrocells employ different frequency spectrum, and thus we do not consider the inter-cell interference [30]. In the fog network, data flows sensed by an IoT device are transmitted to its associated BS, and then processed by the fog node co-located with the BS. Thus, to calculate the latency of data flows, we will focus on the uplink communications of IoT devices and the data processing in fog nodes.

TABLE 1
The Important Notations

Symbol	Definition
$\eta_j(x)$	Binary indicator of location x being associated to BS j.
C_j	Computing capacity of fog node j.
$r_j(x)$	Data rate of an IoT device at location x towards BS j.
$P(x)$	Transmission power of IoT devices at location x.
$\lambda(x)$	The flow arrival rate at location x.
$l(x)$	The average traffic size of a flow at location x.
$\nu(x)$	The average computing size of a flow at location x.
$\mathcal{J}$	Set of BSs/fog nodes.
$\mathcal{A}$	The coverage area of all BSs.
ρ_j	Traffic load of BS j.
$\hat{\rho}_j$	Computing load of fog node j.
μ_j	Communications latency ratio of BS j.
$\hat{\mu}_j$	Computing latency ratio of fog node j.
$L(\boldsymbol{\eta})$	Latency ratio of the fog network.
ρ_{max}	Maximum traffic load threshold of BS j.
$\hat{\rho}_{max}$	Maximum computing load threshold of fog node j.

3.1 Traffic load model

As each BS is assigned with a specific fog node, $\mathcal{J}$ can be used, in this paper, to represent either the set of BSs or the set of fog nodes. Denote $\mathcal{A}$ as the coverage area of all BSs, and x as a location within $\mathcal{A}$. We assume that IoT data flows arrive according to a Poisson Point Process with an average rate per unit area, $\lambda(x)$, at location x. The inhomogeneity results in the spatial variability of traffic loads. Key notations used in this paper are summarized in Table 1.

Denote $P(x)$ as the transmission power of the IoT device at location x, $g_j(x)$ as the uplink channel gain from location x to BS j and σ^2 as the noise power. Then, the signal to noise ratio (SNR) of the IoT device at location x towards BS j can be derived as

$$\gamma_j(x) = \frac{P(x)g_j(x)}{\sigma^2}. \tag{1}$$

Since the uplink data rate of IoT devices depends on the channel condition, IoT devices at different locations may have different data rates. Therefore, if an IoT device at location x is associated with BS j, the capacity of the IoT device (data rate) $r_j(x)$ can be generally expressed as a logarithmic function of its $\gamma_j(x)$, according to the Shannon Hartley theorem,

$$r_j(x) = W_j\ log(1 + \gamma_j(x)), \tag{2}$$

where W_j is the total bandwidth of the jth BS [28].

As mentioned above, the traffic (data flows) arrival at location x follows a Poisson distribution with average arrival rate $\lambda(x)$. Assume that the lengths of all data flows follow an exponential distribution with the average value of $l(x)$. Then, the average traffic load density of the IoT device at location x in BS j can be expressed as [31]

$$\varrho_j(x) = \frac{\lambda(x)l(x)\eta_j(x)}{r_j(x)}, \tag{3}$$

where $\eta_j(x)$ is a binary variable indicating whether location x is associated with the jth BS (1 if so; 0, otherwise).

The average traffic load ρ_j of BS j is obtainted by aggregating traffic load densities of all locations covered by BS j. In particular, the value of ρ_j refers to the fraction of time during which BS j is busy (i.e., the utilization of BS j) [27].

$$\rho_j = \sum_{x \in \mathcal{A}} \varrho_j(x). \tag{4}$$

In mobile communications, based on different metrics such as the network capacity and user fairness, various scheduling algorithms have been proposed to help IoT devices properly share the radio resources of a BS [32]. For analytical tractability, in this paper, we assume that IoT devices at different locations associated with a BS can schedule their uplink transmissions in a round-robin fashion, in which multiple IoT devices can access the uplink channel sequentially. In addition, the traffic arrival rate of location x follows the Poisson Process. Meanwhile, since the traffic sizes of data flows follow the exponential distribution while the data rate at each location is given, the service time of data flows at location x satisfies an exponential distribution [28], where the average service time of data flows at location x can be expressed as $s_j(x) = \frac{l(x)}{r_j(x)}$. As a result, the uplink communications of a BS realizes a M/M/1-processor sharing (*PS*) queue [33]. In the model, as different IoT devices have different data rates due to their channel conditions and they will fairly share the ratio resources of a BS, it is a feasible model to emulate the practical data transmission. Moreover, to keep the queue stable, we always need to guarantee that ρ_j is smaller than 1.

Given the M/M/1-processor sharing queue of a BS, the average delivery time of data flows at location x can be expressed as [33]:

$$t_j(x) = \frac{l(x)}{r_j(x)(1 - \rho_j)}. \tag{5}$$

Meanwhile, the average waiting time for each data flow at location x is

$$w_j(x) = t_j(x) - s_j(x) = \frac{\rho_j l(x)}{r_j(x)(1 - \rho_j)}. \tag{6}$$

Denote $\mu_j(x)$ as the latency ratio of the waiting time to the service time in BS j for data flows at location x. Then,

$$\mu_j(x) = \frac{w_j(x)}{s_j(x)} = \frac{\rho_j}{1 - \rho_j}. \tag{7}$$

It is easy to observe that $\mu_j(x)$ is only dependent on the traffic load of BS j. Therefore, all the IoT devices associated with BS j have the same latency ratio. Hence, we define the communications latency ratio of BS j as

$$\mu_j = \frac{\rho_j}{1 - \rho_j}. \tag{8}$$

From Eq. (8), we can see that increasing traffic load ρ_j of BS j will increases μ_j. When μ_j is high, IoT devices associated with BS j have to wait for a longer time to access the transmission channel. Hence, μ_j is used to reflect the average delivery delay of BS j.

3.2 Computing load model

Aside from the communications latency, the latency of data flows in the fog network is also related to the computing latency in the fog nodes. As the flow arrival at location x follows a Poisson process with the average arrival rate of $\lambda(x)$, the flow arrival rate of fog node j, which is the sum of the flow arrivals at different locations covered by fog node j, also constitutes a Poisson process. On the other hand, we assume that the computing sizes of data flows follow an exponential distribution, where the average computing size (in CPU cycles) of a data flow at location x is expressed as $\nu(x)$. Meanwhile, as we are focusing on the coarse grained computing load balancing among fog nodes by IoT device association, we consider a fog node as a computing unit (like a server). Since the computing capacity of a fog node (in CPU cycles per second) is fixed, the service time of a data flow in a fog node, which equals to the computing size of the data flow divided by the capacity of the fog node, also follows an exponential distribution. By considering a fog node as an entity, it is therefore appropriate to model the processing of IoT flows from IoT devices by a fog node as an M/M/1 queueing model.

Denote C_j as the computing capacity (in CPU cycle/second) of fog node j. In fog node i, the average service time of data flows at location x can be expressed as

$$\hat{s}(x) = \frac{\nu(x)}{C_j}. \tag{9}$$

In addition, the average computing load density of data flows at location x in fog node j can be expressed as

$$\hat{\varrho}_j(x) = \frac{\lambda(x)\nu(x)\eta_j(x)}{C_j}. \tag{10}$$

Aggregating the computing load densities at different locations covered by BS j results in the computing load of fog node j:

$$\hat{\rho}_j = \sum_{x \in \mathcal{A}} \hat{\varrho}_j(x). \tag{11}$$

Based on queuing theory regarding the M/M/1 model, the average waiting time of data flows at location x in fog node j can be derived as

$$\hat{w}_j(x) = \frac{\hat{\rho}_j \nu(x)}{C_j(x)(1 - \hat{\rho}_j)}. \tag{12}$$

Denote $\hat{\mu}_j(x)$ as the computing latency ratio, which equals the ratio between the average waiting time and the average service time. In other words, it shows the required waiting time per unit service time in fog node j.

$$\hat{\mu}_j(x) = \frac{\hat{w}_j(x)}{\hat{s}_j(x)} = \frac{\hat{\rho}_j}{1-\hat{\rho}_j}. \tag{13}$$

Since $\hat{\mu}_j(x)$ is only dependent on the computing load of fog node j, all IoT devices have the same latency ratio in fog node j. Hence, we define the computing latency ratio of fog node j as:

$$\hat{\mu}_j = \frac{\hat{\rho}_j}{1-\hat{\rho}_j}. \tag{14}$$

Here, a smaller $\hat{\mu}$ means that fog node j incurs less delay to its associated IoT devices. Hence, $\hat{\mu}_j$ is adopted to reflect the average computing latency in fog node j.

Considering the M/M/1 processor-sharing queue in a BS and M/M/1 queue in the corresponding fog node, we can model the flow processing in a pair of BS and fog node as a queue system as shown in Fig. 2. In order to minimize the latency of IoT devices' data flows in the fog network, we adopt $\mu_j + \hat{\mu}_j$ (latency ratio) to represent the average latency of processing data flows via the pair of BS j and fog node j.

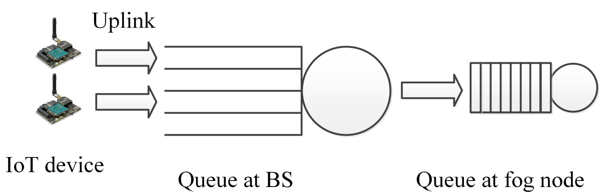


Fig. 2. The queuing system of the fog network.

4 Problem Formulation

In this paper, we aim to improve the latency of all data flows by balancing workloads among BSs/fog nodes. Considering both the communications latency and computing latency, we denote the latency ratio of the fog network as $L(\boldsymbol{\eta}) = \sum_{j\in\mathcal{J}} \mu_j + \hat{\mu}_j$. Our problem is to optimally associate IoT devices to BSs (i.e., balancing loads among BSs/fog nodes) in order to minimize the latency ratio of the fog network. Therefore, the problem can be formulated as follows:

$$P1: \min_{\boldsymbol{\eta}} L(\boldsymbol{\eta}) \tag{15}$$

$$s.t. \quad \sum_{j\in\mathcal{J}} \eta_j(x) = 1, \forall x \in \mathcal{A}; \tag{16}$$

$$0 \le \rho_j \le \rho_{\max}, \forall j \in \mathcal{J}; \tag{17}$$

$$0 \le \hat{\rho}_j \le \hat{\rho}_{\max}, \forall j \in \mathcal{J}; \tag{18}$$

$$\eta_j(x) \in \{0,1\}, \forall x \in \mathcal{A}, \forall j \in \mathcal{J}. \tag{19}$$

Here, Constraint (16) indicates that each location can be associated with only one BS. Constraint (17) imposes the traffic load in BS j not to exceed the maximum load threshold of the BS. Constraint (18) imposes the computing load in fog node i to be less than the maximum load threshold of the fog node.

In the load balancing process, the traffic load allocation and computing load allocation may affect each other. When the heavy workloads of some BSs are the main constraints of the fog network, the proposed scheme pays more attention on balancing the traffic loads among BSs. As a result, the potential traffic congestions in the overloaded BSs will be mitigated, thus reducing the latency of data flows. However, in the above process, IoT devices are allocated to balance the traffic loads among BSs that may incur the uneven computing loads among the fog nodes to a certain extent. In contrast, when some fog nodes become the bottleneck due to their heavy computing loads, the computing latency becomes the dominating factor of data flows' latency. Hence, the proposed scheme will focus on balancing the computing loads among fog nodes by adjusting the IoT device associations among BSs. In this case, although the communications latency may increase owing to the uneven traffic load allocations, the significant reduction of computing latency can still improve the latency of all data flows in the fog network.

5 LAB: A Distributed IoT Device Association Scheme

In this section, we present the LAB scheme, where the communications latency in BSs and the computing latency in fog nodes are taken into account simultaneously. The proposed scheme consists of a BS side algorithm and an IoT device side algorithm. The former one iteratively estimates the traffic loads of BSs and the computing loads of fog nodes, and then broadcasts them to IoT devices. In the latter algorithm, each IoT device selects the suitable BS based on both the updated advertised load information and its uplink data rates towards different BSs such that the latency ratio of the fog network $L(\boldsymbol{\eta})$ is minimized.

5.1 The IoT device side algorithm

At the beginning of the kth iteration, all BSs broadcast their estimated traffic loads ρ_j and computing loads $\tilde{\rho_j}$ to IoT devices. Based on the definition of $L(\boldsymbol{\eta})$, we have

$$\frac{\partial L(\eta)}{\partial \eta_j(x)} = \lambda(x)\frac{C_j l(x)(1-\hat{\rho}_j(k))^2 + r_j(x)\nu(x)(1-\rho_j(k))^2}{C_j r_j(x)(1-\hat{\rho}_j(k))^2(1-\rho_j(k))^2}. \tag{20}$$

Based on the broadcast message, each IoT device can select the suitable BS by

$$p^k(x) = \arg\max_{j\in\mathcal{J}} C_j r_j(x)\phi_j(k), \tag{21}$$

where

$$\phi_j(k) = \frac{(1-\hat{\rho}_j(k))^2(1-\rho_j(k))^2}{C_j l(x)(1-\hat{\rho}_j(k))^2 + r_j(x)\nu(x)(1-\rho_j(k))^2}. \tag{22}$$

Here, $p^k(x)$ is the index of the BS selected by the user at location x, and thus

$$\eta_j^k(x) = \begin{cases} 1, \text{ if } j = p^k(x), \forall x \in \mathcal{A} \\ 0, \text{ if } j \ne p^k(x), \forall x \in \mathcal{A}. \end{cases}$$

5.2 The BS side algorithm

At the side of a BS, it needs to estimate its traffic load and the computing load of its corresponding fog node in each iteration. Thus, it has to estimate an intermediate IoT association $\tilde{\eta}_j^k(x)$ for each IoT device in the iteration. Then, based on the estimated load information among BSs, IoT devices select their BSs/fog nodes by the IoT device side algorithm, and then the current IoT device association in the kth iteration becomes $\eta_j^k(x)$. Therefore, based on the intermediate $\tilde{\eta}_j^k(x)$ (estimated by a BS) and the current IoT device association $\eta_j^k(x)$ (decided by IoT devices) in the kth iteration, BS j can estimate the intermediate IoT association $\tilde{\eta}_j^{k+1}(x)$ for the IoT device at location x in the next iteration as follows:

$$\tilde{\eta}_j^{k+1}(x) = (1-\beta)\eta_j^k(x) + \beta\tilde{\eta}_j^k(x), \tag{23}$$

where $0 \le \beta \le 1$ is a system parameter. Consequently, with the intermediate IoT device association in iteration $k+1$, the advertised traffic load of BS j can be estimated as

$$\rho_j(k+1) = \int_{x\in\mathcal{A}} \frac{\lambda(x)l(x)\tilde{\eta}_j^{k+1}(x)}{r_j(x)}dx. \tag{24}$$

Similarly, the next advertised computing load of fog node j can be estimated as

$$\hat{\rho}_j(k+1) = \int_{x\in\mathcal{A}} \frac{\lambda(x)\nu(x)\tilde{\eta}_j^{k+1}(x)}{C_j(x)}dx. \tag{25}$$

The detailed procedure of the BS side algorithm is illustrated in Algorithm 1.

Algorithm 1 The BS side algorithm

Input: IoT devices' BS selection: $p^k(x), \forall x \in \mathcal{A}$. The intermediate IoT device association vector $\tilde{\eta}^k$ in the kth iteration.

Output: The estimated traffic loads of BSs $\boldsymbol{\rho}(k+1)$ and the estimated computing loads of fog nodes $\hat{\boldsymbol{\rho}}(k+1)$ in the $(k+1)$th iteration.

1: Update the intermediate IoT device association for different locations based on: $\tilde{\eta}_j^{k+1}(x) = (1-\beta)\eta_j^k(x) + \beta\tilde{\eta}_j^k(x), x \in \mathcal{A}, j \in \mathcal{J}$;
2: Calculate $\rho_j(k+1)$ and $\hat{\rho}_j(k+1)$ based on Eq. (24) and (25);
3: **return** $\boldsymbol{\rho}(k)$ and $\hat{\boldsymbol{\rho}}(k+1)$.

As we know, the feasible set of Problem P1 can be expressed as

$$F = \{\boldsymbol{\eta} | \rho_j = \int_{x\in\mathcal{A}} \frac{\lambda(x)l(x)\eta_j(x)}{r_j(x)}dx, \quad \eta_j(x) \in \{0,1\}, 0 \le \rho_j \le \rho_{\max}, \quad \sum_{j\in\mathcal{J}} \eta_j(x) = 1, \forall j \in \mathcal{J}, \forall x \in \mathcal{A}\}. \tag{26}$$

As $\eta_j(x) \in \{0,1\}$, F is not a convex set. In order to derive suitable intermediate IoT associations to gradually reduce the average latency ratio $L(\boldsymbol{\eta})$ in each iteration, we first relax the constraint to make $0 \le \boldsymbol{\eta}^k \le 1$, and then prove that the traffic load and computing load vectors can finally converge in the feasible set. Then, the relaxed feasible set of Problem P1 can be expressed as:

$$\hat{F} = \{\boldsymbol{\eta} | \rho_j = \int_{x\in\mathcal{A}} \frac{\lambda(x)l(x)\eta_j(x)}{r_j(x)}dx, \quad 0 \le \eta_j(x) \le 1, 0 \le \rho_j \le \rho_{\max}, \quad \sum_{j\in\mathcal{J}} \eta_j(x) = 1, \forall j \in \mathcal{J}, \forall x \in \mathcal{A}\}. \tag{27}$$

Lemma 1. *The relaxed feasible set $\hat{F}$ is a convex set.*

Proof: Since the set $\hat{F}$ includes any convex combination of $\boldsymbol{\eta}$, it is a convex set. □

Lemma 2. *The objective function $L(\boldsymbol{\eta})$ is a convex function of η, when η is defined in $\hat{F}$.*

Proof: This lemma can be easily proved by showing that $\nabla^2 L(\eta) > 0$ when η is defined in $\hat{F}$. □

5.3 Analysis of the algorithm

In this section, we will analyze the convergence and optimality of the LAB scheme in the feasible set of Problem P1.

Lemma 3. *When $\tilde{\boldsymbol{\eta}}^{k+1} \neq \tilde{\boldsymbol{\eta}}^k$, $\tilde{\boldsymbol{\eta}}^{k+1}$ provides a descent direction for $L(\tilde{\boldsymbol{\eta}})$ at $\tilde{\boldsymbol{\eta}}^k$.*

Proof: As $0 \le \tilde{\eta}_j^k(x) \le 1$, $L(\tilde{\boldsymbol{\eta}})$ is defined in $\hat{F}$. As shown in Lemma 2, $L(\tilde{\boldsymbol{\eta}})$ is a convex function of $\tilde{\boldsymbol{\eta}}$, and thus we need to prove $\left\langle \nabla L(\tilde{\boldsymbol{\eta}}^k), \tilde{\boldsymbol{\eta}}^{k+1} - \tilde{\boldsymbol{\eta}}^k \right\rangle < 0$. Thus, we have

$$\begin{aligned}
&\left\langle \nabla L(\tilde{\boldsymbol{\eta}}^k), \tilde{\boldsymbol{\eta}}^{k+1} - \tilde{\boldsymbol{\eta}}^k \right\rangle \\
&= \int_{x\in\mathcal{A}} \sum_{j\in\mathcal{J}} \lambda(x)v(x) \frac{\tilde{\eta}_j^{k+1}(x) - \tilde{\eta}_j^k(x)}{C_j r_j(x)\phi_j(k)} \\
&= \int_{x\in\mathcal{A}} \lambda(x)v(x) \sum_{j\in\mathcal{J}} \frac{\tilde{\eta}_j^{k+1}(x) - \tilde{\eta}_j^k(x)}{C_j r_j(x)\phi_j(k)}.
\end{aligned} \tag{28}$$

Based on Eq. (23), we have

$$\tilde{\eta}_j^{k+1}(x) - \tilde{\eta}_j^k(x) = (1-\beta)(\eta_j^k(x) - \tilde{\eta}_j^k(x)). \tag{29}$$

As we know,

$$\eta_j^k(x) = \begin{cases} 1, \text{ if } j = p^k(x) \\ 0, \text{ if } j \neq p^k(x). \end{cases}$$

Owing to the BS selection rule at the user side in the kth iteration, i.e., $p^k(x) = \arg\max_{j\in\mathcal{J}} C_j r_j(x)\phi_j(k)$, we can derive

$$\sum_{j\in\mathcal{J}} (1-\beta)\frac{\eta_j^k(x) - \tilde{\eta}_j^k(x)}{C_j r_j(x)\phi_j(k)} \le 0. \tag{30}$$

Since $\tilde{\boldsymbol{\eta}}^{k+1} \neq \tilde{\boldsymbol{\eta}}^k$,

$$\sum_{j\in\mathcal{J}} (1-\beta)\frac{\eta_j^k(x) - \tilde{\eta}_j^k(x)}{C_j r_j(x)\phi_j(k)} < 0. \tag{31}$$

Hence, we have proved $\left\langle \nabla L(\tilde{\boldsymbol{\eta}}^k), \tilde{\boldsymbol{\eta}}^{k+1} - \tilde{\boldsymbol{\eta}}^k \right\rangle < 0$. □

Meanwhile, as the LAB scheme is executed iteratively, we will also analyze if the BS selection rule at the IoT device side in each iteration is the best option by proving the following theorem.

Theorem 1. *Given the advertised traffic loads of BSs and computing loads of fog nodes, the optimal IoT device association rule at the IoT device side is:*
$p^k(x) = \arg\max_{j\in\mathcal{J}} C_j r_j(x)\phi_j(k)$.

Proof: In the kth iteration, $\boldsymbol{\eta}^k$ is the IoT device association achieved by the proposed IoT device side algorithm: $p^k(x) = \arg\max_{j\in\mathcal{J}} C_j r_j(x)\phi_j(k)$. Meanwhile, let $\boldsymbol{\eta}^{'}$ denote any other possible IoT device association vector in the iteration. Thus, to prove this theorem, we just need to prove that $\boldsymbol{\eta}^{'}$ cannot reduce $L(\boldsymbol{\eta})$ any more as compared to $\boldsymbol{\eta}^k$, i.e., $\left\langle \nabla L(\boldsymbol{\eta}^k), \boldsymbol{\eta}^{'} - \boldsymbol{\eta}^k \right\rangle \geq 0$.

$$\begin{aligned}&\left\langle \nabla L(\boldsymbol{\eta}^k), \boldsymbol{\eta}^{'} - \boldsymbol{\eta}^k \right\rangle \qquad (32)\\ &= \int_{x\in\mathcal{A}} \sum_{j\in\mathcal{J}} \lambda(x)\nu(x)(\eta_j^{'}(x) - \eta_j^k(x)) \frac{1}{C_j r_j(x)\phi_j(k)} dx\\ &= \int_{x\in\mathcal{A}} \lambda(x)\nu(x) \sum_{j\in\mathcal{J}} (\eta_j^{'}(x) - \eta_j^k(x)) \frac{1}{C_j r_j(x)\phi_j(k)} dx.\end{aligned}$$

Since

$$p^k(x) = \arg\max_{j\in\mathcal{J}} C_j r_j(x)\phi_j(k), \tag{33}$$

$$\eta_j^k(x) = \begin{cases} 1, \text{ if } j = p^k(x) \\ 0, \text{ if } j \neq p^k(x). \end{cases}$$

Then, we have

$$\sum_{j\in\mathcal{J}} \eta_j^{'}(x) \frac{1}{C_j r_j(x)\phi_j(k)} \geq \sum_{j\in\mathcal{J}} \eta_j^k(x) \frac{1}{C_j r_j(x)\phi_j(k)}. \tag{34}$$

Hence, $\left\langle \nabla L(\boldsymbol{\eta}), \boldsymbol{\eta}^{'} - \boldsymbol{\eta}^k \right\rangle \geq 0$. Therefore, $\boldsymbol{\eta}^k$ is an optimal IoT device association in the kth iteration. □

As we know, all BSs will estimate and broadcast the traffic load vector $\boldsymbol{\rho}$ and the compuitng load vector $\hat{\boldsymbol{\rho}}$ iteratively, which can be employed by IoT devices to select the suitable BSs. Thus, we need to prove the convergence of $\boldsymbol{\rho}$ and $\hat{\boldsymbol{\rho}}$ for the proposed scheme.

Theorem 2. *At the BS side, the estimated traffic load vector $\boldsymbol{\rho}$ and computing load vector $\hat{\boldsymbol{\rho}}$ converge to the optimal load vectors $\boldsymbol{\rho}^*$ and $\hat{\boldsymbol{\rho}}^*$, respectively, such that $L(\tilde{\boldsymbol{\eta}})$ is minimized.*

Proof: As shown in Lemma 3, $\tilde{\boldsymbol{\eta}}^{k+1} - \tilde{\boldsymbol{\eta}}^k$ provides a decent direction of $L(\tilde{\boldsymbol{\eta}})$ at $\tilde{\boldsymbol{\eta}}^k$, and hence $L(\tilde{\boldsymbol{\eta}})$ gradually decreases in each iteration. Since $L(\tilde{\boldsymbol{\eta}}) > 0$, $\tilde{\boldsymbol{\eta}}$ will eventually converge when $L(\tilde{\boldsymbol{\eta}})$ is minimized.

According to Eq. (24) and (25), the traffic loads of BSs $\boldsymbol{\rho}$ and the computing loads of fog nodes $\hat{\boldsymbol{\rho}}$ are determined by $\tilde{\boldsymbol{\eta}}$. Thus, when the intermediate IoT device association $\tilde{\boldsymbol{\eta}}$ converges, the advertised traffic load vector $\boldsymbol{\rho}$ and computing load vector $\hat{\boldsymbol{\rho}}$ also converge at the same time. □

Lemma 4. *Based on the optimal advertised traffic load vector $\boldsymbol{\rho}$ and computing load vector $\hat{\boldsymbol{\rho}}$, the IoT device side algorithm yields the optimal IoT device association for the load balancing problem in the feasible set F.*

Proof: The proof of this lemma is similar to the proof of Theorem 1. □

As LAB is a gradient algorithm, which is a classic algorithm for convex problems, the number of iterations required to ensure convergence can be found in [28].

6 Numerical Results

In this section, we set up simulations of the proposed scheme to evaluate its performance. We select two other algorithms for comparison: α-distributed algorithm [27] and the Best SNR algorithm. The basic idea of the α-distributed algorithm is to optimally allocate traffic workloads among BSs in order to minimize the communications latency ratio (i.e., $\sum_{j\in\mathcal{J}} \mu_j$) without considering the load distribution of fog nodes. On the other hand, the Best SINR algorithm is to associate IoT devices to the BSs that provide the best channel conditions.

In the simulation, six BSs are randomly deployed in a 3000×2000 m^2 area as shown in Fig. 3. The area is divided into 15,000 locations, where each location represents a 20 m×20 m area. The flow arrival at different locations follows the Poisson point process where the average arrival rate per unit area is set as 0.50 flows/second. As the traffic sizes of data flows follow an exponential distribution, we set the average traffic size as 0.05 Mbits. The computing sizes of data flows also follow an exponential distribution; we set the average computing size of each flow as 5000 CPU cycles. Then, the location-based traffic load density and computing load density can be derived based on Eq. (4) and (11), respectively. Meanwhile, we set the maximum traffic load threshold of each BS as 0.99 and the maximum computing load threshold of each fog node as 0.99. In the simulation, the transmission power of each IoT device is set as 100 mW while the uplink frequency bandwidth of each BS is 10 MHz. We employ COST 231 Walfisch-Ikegami [34] as the propagation model with 9 dB rayleigh fading and 5 dB shadowing fading. The carrier frequency is 2110 MHz, the antenna feeder loss is 3 dB, the transmitter gain is 1 dB, the noise power level is -104 dBm, and the receiver sensitivity is -97 dBm.

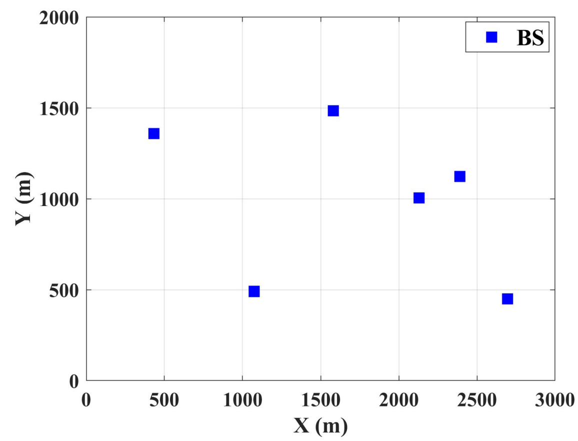


Fig. 3. Network topology.

As shown in Fig. 4, the average latency ratios of both LAB and α-distributed algorithms do converge. Meanwhile, Fig. 5 shows that LAB achieves a much lower average latency ratio than the other two schemes. As we know, the α-distributed algorithm only focuses on the wireless communications latency by allocating the traffic loads among BSs. In this case, the computing loads of fog nodes may be unbalanced (i.e., while some fog nodes are lightly loaded,

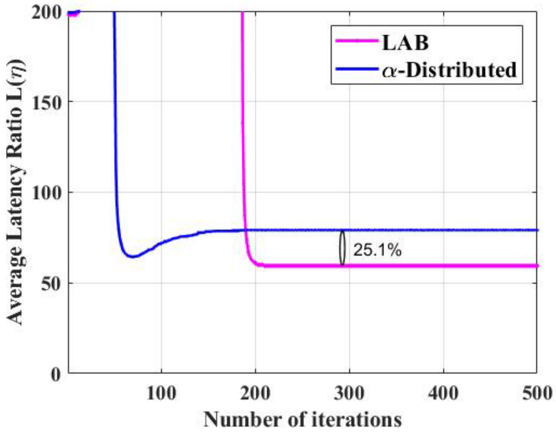


Fig. 4. Average latency ratio $L(\boldsymbol{\eta})$ with respect to the number of iterations ($\lambda = 0.5$, $C_i = 7.1 * 10^6$).

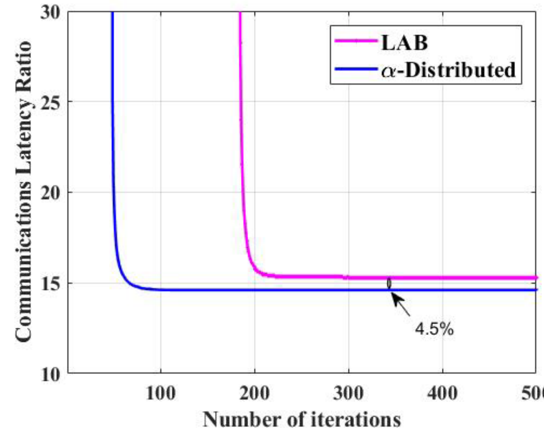


Fig. 6. Average communications latency ratio with respect to the number of iterations ($\lambda = 0.5$, $C_i = 7.1 * 10^6$).

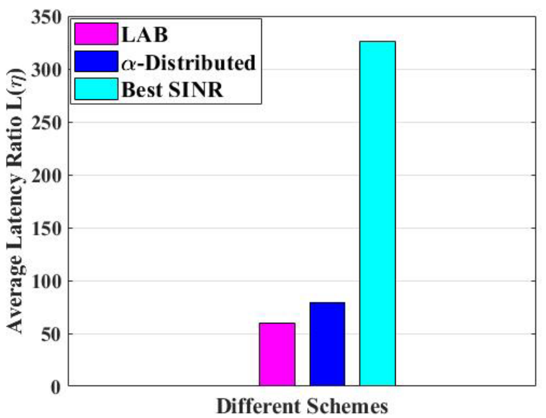


Fig. 5. Average latency ratio $L(\boldsymbol{\eta})$ for different algorithms ($\lambda = 0.5$, $C_i = 7.1 * 10^6$).

other fog nodes are overloaded). Similarly, the Best SINR algorithm aims to assign IoT devices to BSs that provide the best channel conditions, and thus both the traffic loads among BSs and the computing loads among fog nodes may be unbalanced. In contrast, as the latency of a data flow consists of both the communications latency and computing latency, LAB takes into account of both the traffic loads and the computing loads in the load balancing process. As a result, although the communications latency is slightly sacrificed as compared to the α-distributed algorithm, LAB optimizes the average latency ratio of the network by significantly reducing the computing latency in fog nodes.

We also investigate the communications latency of different schemes. From Fig. 6, we can see that LAB incurs a higher average communications latency than the α-distributed algorithm. It is attributed to the fact that the α-distributed algorithm optimally balances the traffic loads among BSs to reduce the communications latency without considering the computing load allocation. In contrast, besides the traffic load balancing, LAB also adjusts the IoT device association to offload the computing loads from overloaded fog nodes to lightly loaded fog nodes. Thus, the adjusted IoT device association cannot guarantee the optimal traffic load balancing, which slightly degrades the performance of communications latency.

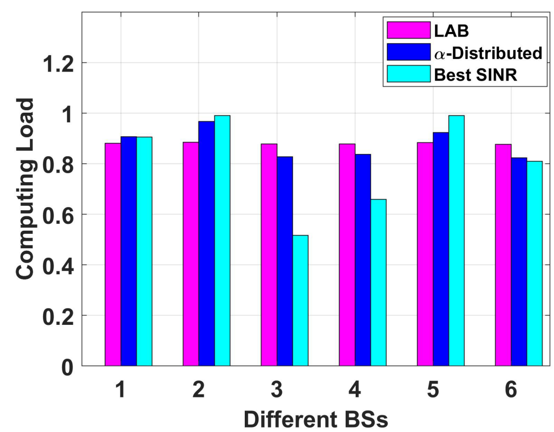


Fig. 7. Computing loads of different fog nodes.

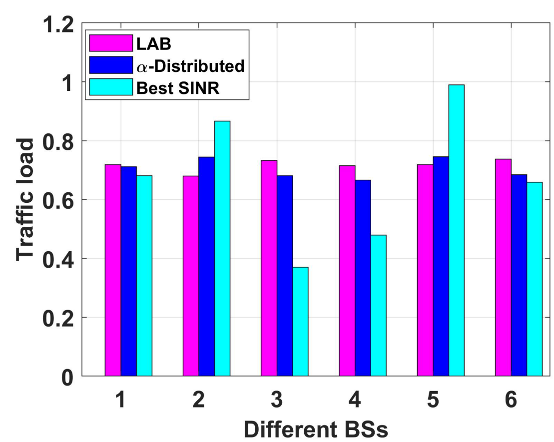


Fig. 8. Traffic loads of different BSs.

To further study the load balancing process in the fog network, we also compare the computing loads among fog

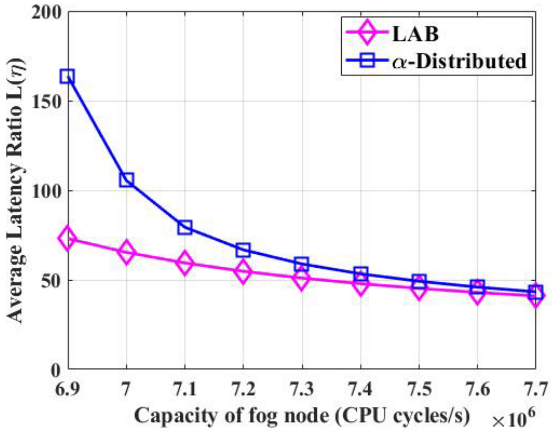


Fig. 9. Average latency ratio with respect to the capacity of each fog node ($\lambda = 0.5$).

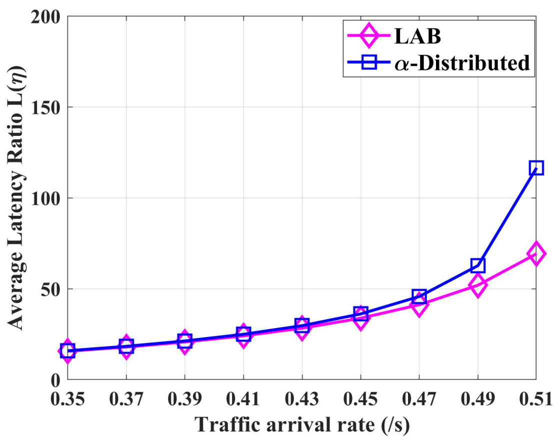


Fig. 10. Average latency ratio with respect to flow arrival rate $\lambda(x)$ ($C_i = 7.1 * 10^6$).

nodes and the traffic loads among BSs for different schemes. Fig. 7 shows that the differences of computing loads among fog nodes achieved by LAB are smaller than those by the α-distributed algorithm and the Best SINR algorithm. While balancing the traffic loads, LAB also balances the computing loads among different fog nodes, thus reducing the computing latency in fog nodes. In contrast, both α-distributed and Best SINR do not consider the computing latency, which is an important factor of the final latency of data flows, and thus incur unbalanced computing loads among fog nodes. Meanwhile, Fig. 8 shows the traffic loads among BSs for different schemes. The differences of traffic loads among BSs for both LAB and α-distributed are smaller than that of the Best SINR algorithm. In other words, the traffic loads of the two schemes are balanced, and thus no BS is congested. Furthermore, since the traffic loads among BSs in LAB and α-distributed are similar, it indicates that LAB only slightly sacrifices the communications latency in the load balancing process, as compared to the α-distributed algorithm.

The capacities of fog nodes can critically impact the computing latency. Specifically, based on Eq. (10), when the capacities of fog nodes increase, the computing load density $\hat{\rho}_j$ will decrease correspondingly. Therefore, we need to study the impact of the capacities of fog nodes on the average latency of all data flows. As shown in Fig. 9, the average latency ratios of both α-distributed and LAB decrease with the increase of fog nodes' capacities. When the capacities of fog nodes are relatively low, LAB achieves a much lower average latency as compared to the α-distributed algorithm because the computing latency becomes the dominating factor of the average latency when fog nodes' capacities are limited. In this case, since LAB can balance the computing loads among fog nodes via the suitable IoT device association, its average latency ratio is remarkably lower than that of the α-distributed algorithm. However, when fog nodes' capacities keep increasing, all fog nodes become lightly loaded and thus the computing latency is no longer the dominating factor of the average latency. In this case, the average latency of the α-distributed algorithm decreases quickly and gets close to that of LAB.

We also investigate the impact of the average traffic arrival rate $\lambda(x)$ on the average latency ratio of the network. As shown in Fig. 10, when the average traffic arrival rate increases, the average latency ratios of both the α-distributed algorithm and LAB increase, where the value of LAB is lower than that of the α-distributed algorithm. When the average arrival rate is relatively low, the average latency ratios of the two schemes are similar because both the BSs and fog nodes in the network are lightly loaded. As a result, the computing load balancing of LAB cannot significantly improve the average latency as compared to the α-distributed algorithm. However, as the average traffic arrival rate increases, the average latency ratio of LAB grows slowly while the performance of the α-distributed algorithm degrades quickly because both the traffic load and computing load in the network become heavy with the increase of the average traffic arrival rate. In this case, the traffic loads among BSs and computing loads among fog nodes jointly impact the average latency ratio. As LAB takes into account of both the traffic load balancing and computing load balancing, it can still maintain low average latency. However, the α-distributed algorithm only focuses on balancing the traffic loads among BSs, in which case some fog nodes are congested especially when the computing loads in the networks are very heavy.

7 Conclusion

In this paper, we have proposed the LoAd Balancing (LAB) scheme for the fog network to minimize the average latency of IoT devices' data flows. Since the latency of a data flow consists of both the communications latency and computing latency, LAB takes into consideration of both the traffic load allocation and computing load allocation by associating IoT devices to suitable BSs/fog nodes. In particular, when the traffic load of the network is heavier than the computing load of the network, the IoT device association focuses on balancing the traffic loads among BSs. Similarly, when the computing load of the network is heavy, i.e., the fog nodes become the bottleneck, the computing latency becomes the dominating factor of the average latency ratio. Nevertheless, LAB can still reduce the average latency by adjusting the IoT device association to balance the traffic load and computing load simultaneously. To solve the problem, we have designed a distributed algorithm to iteratively achieve the optimal solution. Furthermore, we have proved the convergence and optimality of the solution. We have demonstrated the performance of LAB over the α-distributed algorithm and Best SINR algorithm via extensive simulations.

References

[1] S. S. Roy, D. Puthal, S. Sharma, S. P. Mohanty, and A. Y. Zomaya, "Building a sustainable Internet of Things: Energy-efficient routing using low-power sensors will meet the need," *IEEE Consumer Electronics Magazine*, vol. 7, no. 2, pp. 42–49, March 2018.

[2] W. Bao, D. Yuan, Z. Yang, S. Wang, W. Li, B. B. Zhou, and A. Y. Zomaya, "Follow me fog: Toward seamless handover timing schemes in a fog computing environment," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 72–78, Nov. 2017.

[3] H.-L. Truong and S. Dustdar, "Principles for engineering IoT cloud systems," *IEEE Cloud Computing*, vol. 2, no. 2, pp. 68–76, 2015.

[4] L. Wang and R. Ranjan, "Processing distributed internet of things data in clouds," *IEEE Cloud Computing*, vol. 2, no. 1, pp. 76–80, 2015.

[5] M. Ogura and V. M. Preciado, "Stability of spreading processes over time-varying large-scale networks," *IEEE Transactions on Network Science and Engineering*, vol. 3, no. 1, pp. 44–57, Jan 2016.

[6] X. Chen, Q. Shi, L. Yang, and J. Xu, "Thriftyedge: Resource-efficient edge computing for intelligent IoT applications," *IEEE Network*, vol. 32, no. 1, pp. 61–65, Jan 2018.

[7] T. Suganuma, T. Oide, S. Kitagami, K. Sugawara, and N. Shiratori, "Multiagent-based flexible edge computing architecture for IoT," *IEEE Network*, vol. 32, no. 1, pp. 16–23, Jan 2018.

[8] Y. Liu, J. E. Fieldsend, and G. Min, "A framework of fog computing: Architecture, challenges, and optimization," *IEEE Access*, vol. 5, pp. 25 445–25 454, 2017.

[9] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.

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

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

[12] M. Jutila, "An adaptive edge router enabling internet of things," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1061–1069, 2016.

[13] X. Sun, N. Ansari, and Q. Fan, "Green energy aware avatar migration strategy in green cloudlet networks," in *Proceedings - IEEE 7th International Conference on Cloud Computing Technology and Science, (CloudCom' 2015)*, Vancouver, Canada, Nov. 2015.

[14] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2082–2091, Dec 2017.

[15] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing based IoT," *IEEE Internet of Things Journal*, DOI:10.1109/JIOT.2018.2826006, early access, 2018.

[16] X. Sun and N. Ansari, "Green cloudlet network: A sustainable platform for mobile cloud computing," *IEEE Transactions on Cloud Computing*, to be published, DOI:10.1109/TCC.2017.2764463 2017.

[17] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108–119, Jan 2017.

[18] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Transactions on Computers*, vol. 65, no. 12, pp. 3702–3712, Dec 2016.

[19] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *35th Annual IEEE Intl. Conf. on Comp. Comm. (INFOCOM 2016)*, San Francisco, CA, April 2016, pp. 1–9.

[20] Q. Fan, N. Ansari, and X. Sun, "Energy driven avatar migration in green cloudlet networks," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1601–1604, 2017.

[21] Q. Fan and N. Ansari, "Workload allocation in hierarchical cloudlet networks," *IEEE Comm. Letters*, vol. 22, no. 4, pp. 820–823, Apr. 2018.

[22] ——, "Cost aware cloudlet placement for big data processing at the edge," in *2017 IEEE International Conference on Communications (ICC)*, Paris, France, May 2017, pp. 1–6.

[23] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Efficient algorithms for capacitated cloudlet placements," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 10, pp. 2866–2880, 2016.

[24] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 725–737, Oct 2017.

[25] T. G. Rodrigues *et al.*, "Towards a low-delay edge cloud computing through a combined communication and computation approach," in *IEEE 84th Vehicular Technology Conf. (VTC 2016)*, Sept 2016, pp. 1–5.

[26] ——, "Hybrid method for minimizing service delay in edge cloud computing through vm migration and transmission power control," *IEEE Trans. on Computers*, vol. 66, no. 5, pp. 810–819, 2017.

[27] H. Kim, G. de Veciana, X. Yang, and M. Venkatachalam, "Distributed α-Optimal User Association and Cell Load Balancing in Wireless Networks," *IEEE/ACM Trans. on Networking*, vol. 20, no. 1, pp. 177–190, 2012.

[28] T. Han and N. Ansari, "A traffic load balancing framework for software-defined radio access networks powered by hybrid energy sources," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 1038–1051, April 2016.

[29] Q. Fan and N. Ansari, "Throughput aware and green energy aware user association in heterogeneous networks," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

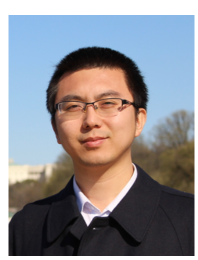
[30] F. Wang, C. Xu, L. Song, and Z. Han, "Energy-efficient resource allocation for device-to-device underlay communication," *IEEE Transactions on Wireless Communications*, vol. 14, no. 4, pp. 2082–2092, April 2015.

[31] N. Sapountzis, T. Spyropoulos, N. Nikaein, and U. Salim, "User association in hetnets: Impact of traffic differentiation and backhaul limitations," *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3396–3410, Dec 2017.

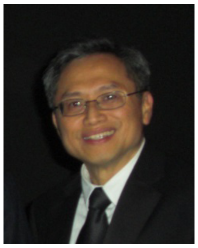
[32] Q. Fan and N. Ansari, "Green energy aware user association in heterogeneous networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC'2016)*, Doha, Qatar, Apr. 2016.

[33] L. Kleinrock, *Queueing Systems: Computer applications*. Wiley-Interscience, 1976.

[34] "Evolution of land mobile radio (including personal) ccommunications: Cost 231." [Online]. Available: http://www.awe-communications.com/Propagation/Urban/COST/



Qiang Fan (S'15) received his M.S. degree in Electrical Engineering from Yunnan University of Nationalities, China, in 2013. He is currently a research assistant and a Ph.D. candidate in the Department of Electrical and Computer Engineering, New Jersey Institute of Technology (NJIT), Newark, New Jersey, USA. His research interests include mobile and cellular networks, mobile cloud computing, Internet of things, and free space optical communications.



Nirwan Ansari (S'78-M'83-SM'94-F'09) received his B.S.E.E. degree from NJIT, M.S.E.E. degree from the University of Michigan, Ann Arbor, and Ph.D. degree from Purdue University, West Lafayette. He is Distinguished Professor of Electrical and Computer Engineering at NJIT. He has also been a visiting (chair) professor at several universities. He recently authored Green Mobile Networks: A Networking Perspective (Wiley-IEEE, 2017) with T. Han, and co-authored two other books. He has also (co-)authored more than 500 technical publications, over 250 in widely cited journals/magazines. He has guest-edited a number of special issues covering various emerging topics in communications and networking. He has served on the editorial/advisory board of over ten journals. His current research focuses on green communications and networking, cloud computing, and various aspects of broadband networks. Some of his recognitions include several Excellence in Teaching Awards, a few best paper awards, the NCE Excellence in Research Award, the ComSoc AHSN TC Technical Recognition Award, the IEEE TCGCC Distinguished Technical Achievement Recognition Award, ComSoC TC-CSR Distinguished Technical Achievement Award, the NJ Inventors Hall of Fame Inventor of the Year Award, the Thomas Alva Edison Patent Award, Purdue University Outstanding Electrical and Computer Engineer Award, and designation as a COMSOC Distinguished Lecturer. He has also been granted 36 U.S. patents.