ABSTRACT

WORKLOAD ALLOCATION IN MOBILE EDGE COMPUTING EMPOWERED INTERNET OF THINGS

by
Qiang Fan

In the past few years, a tremendous number of smart devices and objects, such as smart phones, wearable devices, industrial and utility components, are equipped with sensors to sense the real-time physical information from the environment. Hence, Internet of Things (IoT) is introduced, where various smart devices are connected with each other via the internet and empowered with data analytics. 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 IoT devices. However, since the remote cloud is far from IoT users which send application requests and await the results generated by the data processing in the remote cloud, the response time of the requests may be too long, especially unbearable for delay sensitive IoT applications. Therefore, edge computing resources (e.g., cloudlets and fog nodes) which are close to IoT devices and IoT users can be employed to alleviate the traffic load in the core network and minimize the response time for IoT users.

In edge computing, the communications latency critically affects the response time of IoT user requests. Owing to the dynamic distribution of IoT users (i.e., UEs), drone base station (DBS), which can be flexibly deployed for hotspot areas, can potentially improve the wireless latency of IoT users by mitigating the heavy traffic loads of macro BSs. Drone-based communications poses two major challenges: 1) the DBS should be deployed in suitable areas with heavy traffic demands to serve more UEs; 2) the traffic loads in the network should be allocated among macro BSs and DBSs to avoid instigating traffic congestions. Therefore, a TrAffic Load

baLancing (TALL) scheme in such drone-assisted fog network is proposed to minimize the wireless latency of IoT users. In the scheme, the problem is decomposed into two sub-problems, two algorithms are designed to optimize the DBS placement and user association, respectively. Extensive simulations have been set up to validate the performance of the proposed scheme.

Meanwhile, various IoT applications can be run in cloudlets to reduce the response time between IoT users (e.g., user equipments in mobile networks) and cloudlets. Considering the spatial and temporal dynamics of each application's workloads among cloudlets, the workload allocation among cloudlets for each IoT application affects the response time of the application's requests. To solve this problem, an Application awaRE workload Allocation (AREA) scheme for edge computing based IoT is designed to minimize the response time of IoT application requests by determining the destination cloudlets for each IoT user's different types of requests and the amount of computing resources allocated for each application in each cloudlet. In this scheme, both the network delay and computing delay are taken into account, i.e., IoT users' requests are more likely assigned to closer and lightly loaded cloudlets. The performance of the proposed scheme has been validated by extensive simulations.

In addition, the latency of data flows in IoT devices consist of both the communications latency and computing latency. When some BSs and fog nodes are lightly loaded, other overloaded BSs and fog nodes may incur congestion. Thus, a workload balancing scheme in a fog network is proposed to minimize the latency of IoT data in the communications and processing procedures by associating IoT devices to suitable BSs. Furthermore, the convergence and the optimality of the proposed workload balancing scheme has been proved. Through extensive simulations, the performance of the proposed load balancing scheme is validated.

WORKLOAD ALLOCATION IN MOBILE EDGE COMPUTING EMPOWERED INTERNET OF THINGS

by
Qiang Fan

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology and
Rutgers, The State University of New Jersey – Newark
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Mathematical Sciences

Department of Mathematical Sciences
Department of Mathematics and Computer Science, Rutgers-Newark

May 2019

APPROVAL PAGE

WORKLOAD ALLOCATION IN MOBILE EDGE COMPUTING EMPOWERED INTERNET OF THINGS

Qiang Fan

Nirwan Ansari, Dissertation Advisor Date
Distinguished Professor, Department of Electrical and Computer Engineering, NJIT

Mengchu Zhou, Dissertation Co-Advisor Date
Distinguished Professor, Department of Electrical and Computer Engineering, NJIT

Qing Liu, Committee Member Date
Assistant Professor, Department of Electrical and Computer Engineering, NJIT

Ali Mili, Committee Member Date
Professor, Department of Computer Science, NJIT

Roberto Rojas-Cessa, Committee Member Date
Professor, Department of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Qiang Fan

Degree: Doctor of Philosophy

Date: May 2019

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2019

- Master of Science in Signal and Information Processing,
  Yunnan University of Nationalities, Kunming, P.R. China, 2013

- Bachelor of Science in Applied Physics,
  Suzhou University of Science and Technology, Suzhou, P.R. China, 2009

Major: Electrical Engineering

Presentations and Publications:

Journal articles:

Q. Fan and N. Ansari, "Towards Workload Balancing in Fog Computing Empowered IoT," *IEEE Transactions on Network Science and Engineering*, doi: 10.1109/TNSE.2018.2852762, 2018, early access.

Q. Fan and N. Ansari, "Towards Throughput Aware and Energy Aware Traffic Load Balancing in Heterogeneous Networks with Hybrid Power Supplies," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 4, pp. 890-898, Dec. 2018.

Q. Fan and N. Ansari, "Application Aware Workload Allocation for Edge Computing-Based IoT," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2146-2153, June 2018.

Q. Fan and N. Ansari, "Workload Allocation in Hierarchical Cloudlet Networks," *IEEE Communications Letters*, vol. 22, no. 4, pp. 820-823, April 2018.

Q. Fan and N. Ansari, "On Cost Aware Cloudlet Placement for Mobile Edge Computing," *IEEE/CAA Journal of Automatica Sinica*, accepted, 2019.

Q. Fan, N. Ansari, J. Feng, R. Rojas-Cessa, M. Zhou and T. Zhang, "Reducing the Number of FSO Base Stations with Dual Transceivers for Next-generation Ground-to-Train Communications," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11143-11153, Nov. 2018.

L. Zhang, **Q. Fan** and N. Ansari, "3-D Drone-Base-Station Placement With In-Band Full-Duplex Communications," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1902-1905, Sept. 2018.

Q. Fan and N. Ansari, "Towards Traffic Load Balancing in Drone-assisted Communications for IoT," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2018.2889503, 2018, early access.

Q. Fan, M. Taheri, N. Ansari, J. Feng, R. Rojas-Cessa, M. Zhou, and T. Zhang, "Reducing the Impact of Handovers in Ground-to-Train Free Space Optical Communications," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1292-1301, Feb. 2018.

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, July 2017.

Q. Fan, J. Fan, J. Li, and X. Wang, "A Multi-hop Energy-efficient Sleeping MAC Protocol Based on TDMA Scheduling for Wireless Mesh Sensor Networks," *Journal of Networks*, vol 7, no. 9, pp.1355-1361, Sep, 2012.

Q. Wu, S. Nie, P. Fan, H. Liu, **Q. Fan**, and Z. Li, "A Swarming Approach to Optimize the One-Hop Delay in Smart Driving Inter-Platoon Communications," *Sensors*, vol. 18, no. 10, pp. 3307, 2018.

Q. Wu, S. Xia, P. Fan, **Q. Fan**, and Z. Li, "Velocity Adaptive V2I Fair Access Scheme based on IEEE 802.11 DCF for Platooning Vehicles," *Sensors*, no.18, pp. 4198, 2018.

Q. Wu, H. Liu, C. Zhang, **Q. Fan**, Z. Li, K. Wang, "Trajectory Protection Schemes Based on Gravity Mobility Model in IoT," *Electronics*, vol. 8, no. 2, pp. 148, 2019.

Conference papers

Q. Fan and N. Ansari, "Cost Aware Cloudlet Placement for Big Data Processing at the Edge," *IEEE International Conference on Communications (ICC2017)*, Paris, May 2017, pp. 1-6.

Q. Fan and N. Ansari, "Throughput Aware and Green Energy Aware User Association in Heterogeneous Networks," *IEEE International Conference on Communications (ICC2017)*, May Paris, 2017, pp. 1-6.

Q. Fan and N. Ansari, “Green Energy Aware User Association in Heterogeneous Networks,” *2016 IEEE Wireless Communications and Networking Conference*, Doha, 2016, pp. 1-6.

X. Sun, N. Ansari and **Q. Fan**, “Green Energy Aware Avatar Migration Strategy in Green Cloudlet Networks,” *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, Vancouver, BC, 2015, pp. 139-146.

J. Fan, J. Hao, **Q. Fan**, and H. Wang, “Multi-hop Sleep MAC Protocols for Wireless Mesh Sensor Network Based on Effective Listen Time, ”*the 2011 International Conference on Information and Computer Networks (ICICN 2011)*, Jan, 2011.

Q. Wu, J. Fan, C. Zhang, and **Q. Fan**, “Energy Consumption Balance Tactics in Wireless Mesh Sensor Network Based on Topology Controlling, ” *2010 International Conference on Future Information Technology (ICFIT 2010)*, Dec, 2010.

Patent applications

N. Ansari, M. Zhou, R. Rojas-Cessa, **Q. Fan**, J. Feng, and T. Zhang, “Systems and Methods to Extend Coverage of FSO Base Stations for Ground-to-Train Communications”, Provisional Patent Application no. 62/620,304, filed on Jan. 22, 2018.

J. Feng, M. Taheri, **Q. Fan**, N. Ansari, R. Rojas-Cessa, M. Zhou, G. Chen, D. Wang, J. Tang and T. Zhang, “A Method and System for Railway Communications,” CN107105465, filed on Apr. 21, 2017.

To my beloved parents Zhenjin Fan, Xiuzhen Qin and my sister Li Fan for their unwavering support. Their encouragement, sacrifices, and love have meant more to me than they can imagine. This dissertation is dedicated to them.

ACKNOWLEDGMENT

My deepest gratitude is to my advisor, Dr. Nirwan Ansari. I have been amazingly fortunate to have him giving me the freedom and encouragement to explore research ideas while providing excellent guidance. I also want to thank my co-advisor Dr. Mengchu Zhou. Their persistent support and patience helped me overcome many difficult situations throughout my research. Without their continuous help, this dissertation would not have been possible.

To my committee members, Dr. Qing Liu, and Dr. Ali Mili, and Dr. Roberto Rojas-Cessa. I thank them for their time and advisement.

This dissertation is based upon work supported by the National Science Foundation under Grant No. CNS-1320468 and CNS-1814748.

I want to thank my friends Xiang Sun, Tao Han, Mina Taheri, Xueqing Huang, Abbas Kiani, Xilong Liu, Liang Zhang, Di Wu, Jingjing Yao, Ali Shahini, Shuai Zhang, and many others, who have given me support and encouragement over the last five years. I would like to extend my gratitude to other faculty and staff members of the Department of Electrical and Computer Engineering for their support throughout my doctoral studies.

Finally, thanks to my family for their unwavering support.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

Recently, 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. However, as the data streams generated from IoT devices are transmitted to the remote cloud, the latency for processing data streams may be too long. The concept of edge computing (e.g., cloudlet and fog node) has thus been employed to reduce the network delay by moving the remote cloud resources to the network edge. Since cloudlets and fog nodes are generally placed at access points that are close to IoT devices, IoT devices can access the computing resources with a lower network delay.

In the edge computing empowered IoT, there are several challenging issues to be addressed. As the latency of IoT tasks consists of both the communications latency and computing latency, it is critical to jointly balance the traffic loads at BSs and computing loads at fog nodes to minimize the latency. On the other hand, owing to heterogeneity of various IoT applications, how to allocate various applications' tasks among cloudlets and allocate the computing resources for different applications in each cloudlet remains to be a challenging issue. Meanwhile, given the cloudlet or fog node, the communications latency becomes an important factor. Thus, we can apply drone-mounted base stations (DBSs) to facilitate the data transfer between IoT users and BSs.

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. 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. 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, 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, traffic loads of some IoT devices associated with these 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 design 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.

In addition, in consideration of various IoT applications, when the workload of a cloudlet is too heavy, the computing resources available for an application is limited, and thus the response time of the corresponding tasks is degraded correspondingly. In this case, although the cloudlet in the proximity yields the minimum network delay, the bulk of the response time is attributed to the computing delay. Thus, the workload allocation of different types of requests greatly impacts the response time of requests of user equipments (UEs). On the other hand, for each cloudlet, the resource allocation for different types of applications also affects the computing delay of different types of requests. Since the computing size per request is different for different applications, the computing capacity of a cloudlet should be optimally allocated for different types of applications in order to reduce the computing delay of all Apps of UEs.

To solve the above problem, we design an Application awaRE workload Allocation (AREA) scheme for edge computing based IoT to minimize the total response time of UEs' Apps, where both the network delay and computing delay are taken into account. Below are major contributions of the scheme. Specifically, we formulate the problem of minimizing the average response time of different types of IoT Apps by offloading UEs' different types of requests among distributed cloudlets and allocating optimal computing resources for different applications in each cloudlet. The response time of each type of requests consists of both the network delay and computing delay. On one hand, to reduce the network delay, different types of requests of a UE are favorably assigned to closer cloudlets. On the other hand, each application is assumed to be handled by a dedicated virtual machine in each cloudlet, the capacity of which can be dynamically allocated in each time slot [2]; when a cloudlet is overloaded, the computing resources available for each application are not enough to handle the type of requests, and thus the computing delay becomes the dominating factor of the response time. Hence, different types of requests of a UE should be assigned to other lightly loaded cloudlets to reduce their computing delays.

Moreover, the wireless latency between IoT users (i.e., UEs) and macro base stations (MBSs) where the fog nodes are co-located is a key factor in determining the response time of user requests. Recently, drones have been incorporated into mobile networks to improve the quality of service (QoS) of UEs. Owing to the fast and flexible deployment feature, a DBS can be dynamically placed at hotspot areas as a relay to deliver UEs' IoT tasks to MBSs, and thus improve the channel quality and QoS of UEs.

To improve the wireless latency from UEs to the MBS (i.e., uplink) in the DBS-assisted fog network, several critical issues should be considered. First, as the traffic demands among different locations exhibit spatial and temporal dynamics, the deployment of DBSs to suitable locations critically affects the wireless latency of IoT users. Specifically, if DBSs are placed over areas with higher UE densities, they can provide good channel conditions for more UEs, and thus are more likely to mitigate traffic congestion of the MBS. In contrast, if DBSs are placed over areas with lower UE densities, the traffic loads that can be offloaded from the MBS will be limited (i.e., the utilizations of these DBSs become limited), the wireless latency of all UEs cannot be significantly reduced. Second, since each DBS serves as a relay to deliver IoT requests from UEs to the MBS, the latency of UEs served by the DBS will be determined by both the access link (between UEs and the DBS) and the backhaul link (between the DBS and the MBS). In particular, the favorable channel conditions of the access links may attract a large number of UEs to associate with the DBS; however, if the capacity of the backhaul link of the DBS is limited, the wireless latency of these UEs will be degraded by the traffic congestion of the corresponding backhaul link. Third, the latency of user requests is impacted by UEs' channel conditions and the traffic loads of their BSs simultaneously. Increase in a BS's traffic load (either a DBS or the MBS) tends to congest the BS such that the corresponding IoT requests have to wait for a longer time to be transmitted. In this case, the traffic load allocation among BSs will have a critical impact on the delivery time of IoT requests. To tackle the problem, a TrAffic Load baLancing (TALL) scheme is designed to minimize the communications latency of IoT requests in such DBS-assisted fog network.

The rest of the dissertation is organized as follows. In Chapter 2, we briefly review the related works. In Chapter 3, IoT users are associated to suitable BSs to balance the traffic load at BSs and computing loads at fog nodes simultaneously. In Chapter 4, we design the AREA scheme to assign tasks of different applications among cloudlets and allocate computing resources to various application in each cloudlet to minimize the response time of these IoT tasks. In Chapter 5, we design the TALL scheme to place DBSs and associate IoT users among different DBSs to facilitate the task offloading from IoT users to fog nodes. The simulation results and future work are presented in Chapter 6 and 7, respectively. The conclusion is made in Chapter 8.

CHAPTER 2

RELATED WORK

Owing to the proximity of edge computing resources to IoT devices and IoT users, some studies have focused on integrating IoT with edge computing. Bonomi *et al.* [3] elicited how fog computing may be applied in various IoT applications. Chiang *et al.* [4] 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 [5] designed the IoT architecture (EdgeIoT) to handle the data streams from IoT devices at the fog nodes. Moreover, Jutila [6] proposed adaptive fog computing solutions for IoT networking in order to optimize traffic flows and network resources.

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 [7]. Gu *et al.* [8] integrated 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.* [9] jointly considered the task scheduling and image placement in a fog computing based software-defined embedded system to minimize the response time of task requests. Tong *et al.* [10] 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.* [11] migrated 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 [12] 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 [13, 14] 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.* [15] designed 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 have been 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 [16, 17].

Moreover, many existing works on mobile networks have addressed traffic workload balancing among BSs. Kim *et al.* [18] proposed an iterative distributed user association algorithm to balance the traffic loads among BSs based on different performance metrics. Han and Ansari [19] designed a traffic workload balancing scheme to make a tradeoff between the traffic delivery time and brown energy consumption in a cellular network. Fan *et al.* [20] designed a user association algorithm to improve the flow level throughput and green energy utilization in heterogeneous cellular networks.

Meanwhile, drone based communications provisions many advantages over current terrestrial wireless communications, such as flexible deployment, flexible reconfiguration, and better channel conditions for user equipments. Many studies have been done to deploy the DBS in the network and improve the QoS of UEs. Sun and Ansari [21] designed a heuristic two-dimensional DBS placement algorithm to deploy a DBS in the network and improve the downlink communications of UEs. Bor-Yaliniz *et al.* [22] designed a 3-D placement algorithm in order to cover as many UEs as possible. Fotouhi *et al.* [23] proposed to place the DBS to increase its spectral efficiency. Al-Hourani *et al.* [24] designed an analytical approach to derive the optimal altitude of a DBS to maximize its coverage. Lyu *et al.* [25] designed a DBS placement algorithm to cover a certain area with the minimum number of DBSs. Wang *et al.* [26] optimally deployed DBSs in order to minimize the transmission power required to serve UEs. Zeng *et al.* [27] introduced the network architecture and challenges of UAV-aided wireless communications. Shi *et al.* [28] optimized the drone-cell deployment to maximize the user coverage while keeping the channel qualities of backhaul links.

CHAPTER 3

WORKLOAD BALANCING IN FOG COMPUTING EMPOWERED IOT

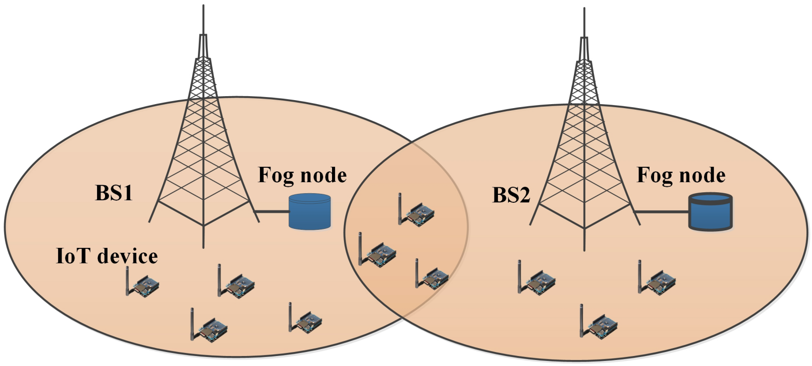

Figure 3.1 Fog network architecture.

A fog network architecture is illustrated in Figure 3.1, 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 [5]. 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 BS's 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 [17] 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 chapter, 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 [29]. 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.

3.1 Traffic Load Model

As each BS is assigned with a specific fog node, $\mathcal{J}$ can be used, in this chapter, 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 traffic loads are spatially dynamic. Key notations used in this chapter are summarized in Table 3.1.

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

$$\gamma_j(x) = \frac{P(x)g_j(x)}{\sigma^2}. \tag{3.1}$$

Since the uplink data rate of an IoT device 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

Table 3.1 List of Symbols in Workload Balancing in Fog Computing

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.

Shannon Hartley theorem,

$$r_j(x) = W_j \ log(1 + \gamma_j(x)), \tag{3.2}$$

where W_j is the total bandwidth of the jth BS [19].

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 [30]

$$\varrho_j(x) = \frac{\lambda(x)l(x)\eta_j(x)}{r_j(x)}, \tag{3.3}$$

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) [18].

$$\rho_j = \sum_{x \in \mathcal{A}} \varrho_j(x). \tag{3.4}$$

In mobile communications, based on different metrics such as the network capacity and user fairness, various scheduling algorithms have been designed to help IoT devices properly share the radio resources of a BS [31]. For analytical tractability, in this chapter, 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 [19], 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 [32]. In the model, as different IoT devices have different data rates due to their channel conditions and they will fairly share the radio 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 [32]:

$$t_j(x) = \frac{l(x)}{r_j(x)(1-\rho_j)}. \tag{3.5}$$

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

$$w_j(x) = t_j(x) - s_j(x) = \frac{\rho_j l(x)}{r_j(x)(1-\rho_j)}. \tag{3.6}$$

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,

$$\mu_j(x) = \frac{w_j(x)}{s_j(x)} = \frac{\rho_j}{1-\rho_j}. \tag{3.7}$$

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

$$\mu_j = \frac{\rho_j}{1-\rho_j}. \tag{3.8}$$

From Equation (3.8), 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 long 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

$$\hat{s}(x) = \frac{\nu(x)}{C_j}. \tag{3.9}$$

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

$$\hat{\varrho}_j(x) = \frac{\lambda(x)\nu(x)\eta_j(x)}{C_j}. \tag{3.10}$$

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

$$\hat{\rho}_j = \sum_{x \in \mathcal{A}} \hat{\varrho}_j(x). \tag{3.11}$$

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

$$\hat{w}_j(x) = \frac{\hat{\rho}_j \nu(x)}{C_j(x)(1 - \hat{\rho}_j)}. \tag{3.12}$$

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.

$$\hat{\mu}_j(x) = \frac{\hat{w}_j(x)}{\hat{s}_j(x)} = \frac{\hat{\rho}_j}{1 - \hat{\rho}_j}. \tag{3.13}$$

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:

$$\hat{\mu}_j = \frac{\hat{\rho}_j}{1 - \hat{\rho}_j}. \tag{3.14}$$

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 queuing system as shown in Figure 3.2. 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.

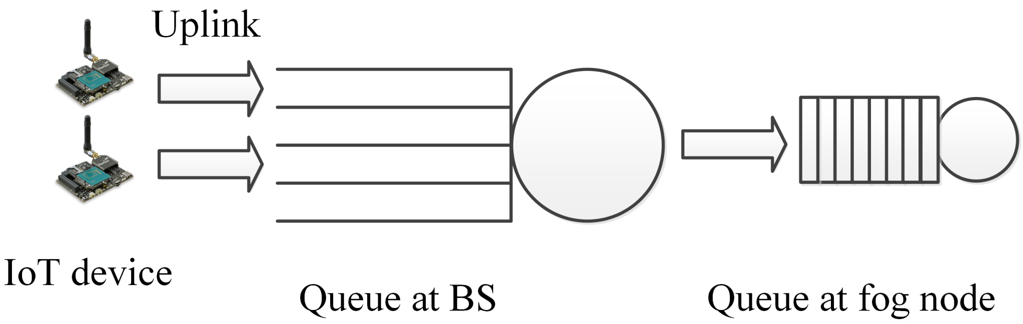

Figure 3.2 Queuing system of the fog network.

3.3 Problem Formulation

In this chapter, 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 $L(\boldsymbol{\eta}) = \sum\limits_{j \in \mathcal{J}} \mu_j + \hat{\mu}_j$. 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:

$$P1 : \min_{\boldsymbol{\eta}} L(\boldsymbol{\eta}) \tag{3.15}$$

$$s.t. \quad \sum_{j \in \mathcal{J}} \eta_j(x) = 1, \forall x \in \mathcal{A}; \tag{3.16}$$

$$0 \le \rho_j \le \rho_{\max}, \forall j \in \mathcal{J}; \tag{3.17}$$

$$0 \le \hat{\rho}_j \le \hat{\rho}_{\max}, \forall j \in \mathcal{J}; \tag{3.18}$$

$$\eta_j(x) \in \{0, 1\}, \forall x \in \mathcal{A}, \forall j \in \mathcal{J}. \tag{3.19}$$

Here, Constraint (3.16) indicates that each location can be associated with only one BS. Constraint (3.17) imposes the traffic load in BS j not to exceed the maximum load threshold of the BS. Constraint (3.18) 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 new 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, our 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.

3.4 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 LAB 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.

3.4.1 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

$$\frac{\partial L(\eta)}{\partial \eta_j(x)} = \lambda(x)\frac{C_j l(x)(1-\hat{\rho}_j(k))^2 + r_j(x)\nu(x)(1-\rho_j(k))^2}{C_j r_j(x)(1-\hat{\rho}_j(k))^2(1-\rho_j(k))^2}. \tag{3.20}$$

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

$$p^k(x) = \arg\max_{j\in\mathcal{J}} C_j r_j(x)\phi_j(k), \tag{3.21}$$

where

$$\phi_j(k) = \frac{(1-\hat{\rho}_j(k))^2(1-\rho_j(k))^2}{C_j l(x)(1-\hat{\rho}_j(k))^2 + r_j(x)\nu(x)(1-\rho_j(k))^2}. \tag{3.22}$$

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

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

3.4.2 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:

$$\tilde{\eta}_j^{k+1}(x) = (1-\beta)\eta_j^k(x) + \beta\tilde{\eta}_j^k(x), \tag{3.23}$$

where $0 \leq \beta \leq 1$ 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

$$\rho_j(k+1) = \int_{x \in \mathcal{A}} \frac{\lambda(x) l(x) \tilde{\eta}_j^{k+1}(x)}{r_j(x)} dx. \tag{3.24}$$

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

$$\hat{\rho}_j(k+1) = \int_{x \in \mathcal{A}} \frac{\lambda(x) \nu(x) \tilde{\eta}_j^{k+1}(x)}{C_j(x)} dx. \tag{3.25}$$

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 $k$th 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 Equations (3.24) and (3.25);
        **return** $\boldsymbol{\rho}(k)$ and $\hat{\boldsymbol{\rho}}(k+1)$.
```

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

$$\begin{aligned} F = \{\boldsymbol{\eta} | \rho_j = \int_{x \in \mathcal{A}} \frac{\lambda(x) l(x) \eta_j(x)}{r_j(x)} dx, \\ \eta_j(x) \in \{0,1\}, 0 \leq \rho_j \leq \rho_{\max}, \\ \sum_{j \in \mathcal{J}} \eta_j(x) = 1, \forall j \in \mathcal{J}, \forall x \in \mathcal{A}\}. \end{aligned} \tag{3.26}$$

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 $0 \leq \boldsymbol{\eta}^k \leq 1$, 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:

$$\begin{aligned}\hat{F} = \{\boldsymbol{\eta} | \rho_j = \int_{x \in \mathcal{A}} & \frac{\lambda(x) l(x) \eta_j(x)}{r_j(x)} dx, \\ 0 \leq \eta_j(x) & \leq 1, 0 \leq \rho_j \leq \rho_{\max}, \\ \sum_{j \in \mathcal{J}} \eta_j(x) & = 1, \forall j \in \mathcal{J}, \forall x \in \mathcal{A}\}.\end{aligned} \tag{3.27}$$

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 $\hat{F}$. □

3.4.3 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 $0 \leq \tilde{\eta}_j^k(x) \leq 1$, $L(\tilde{\boldsymbol{\eta}})$ 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

$$\begin{aligned}&\left\langle \nabla L(\tilde{\boldsymbol{\eta}}^k), \tilde{\boldsymbol{\eta}}^{k+1} - \tilde{\boldsymbol{\eta}}^k \right\rangle \\ &= \int_{x \in \mathcal{A}} \sum_{j \in \mathcal{J}} \lambda(x) v(x) \frac{\tilde{\eta}_j^{k+1}(x) - \tilde{\eta}_j^k(x)}{C_j r_j(x) \phi_j(k)} \\ &= \int_{x \in \mathcal{A}} \lambda(x) v(x) \sum_{j \in \mathcal{J}} \frac{\tilde{\eta}_j^{k+1}(x) - \tilde{\eta}_j^k(x)}{C_j r_j(x) \phi_j(k)}.\end{aligned} \tag{3.28}$$

Based on Equation (3.23), we have

$$\tilde{\eta}_j^{k+1}(x) - \tilde{\eta}_j^k(x) = (1-\beta)(\eta_j^k(x) - \tilde{\eta}_j^k(x)). \tag{3.29}$$

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

$$\sum_{j\in\mathcal{J}}(1-\beta)\frac{\eta_j^k(x) - \tilde{\eta}_j^k(x)}{C_j r_j(x)\phi_j(k)} \leq 0. \tag{3.30}$$

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

$$\sum_{j\in\mathcal{J}}(1-\beta)\frac{\eta_j^k(x) - \tilde{\eta}_j^k(x)}{C_j r_j(x)\phi_j(k)} < 0. \tag{3.31}$$

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 to minimize the latency ratio of the network 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 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., $\left\langle \nabla L(\boldsymbol{\eta}^{k}), \boldsymbol{\eta}^{'} - \boldsymbol{\eta}^{k} \right\rangle \geq 0$.

$$\begin{aligned}
&\left\langle \nabla L(\boldsymbol{\eta}^{k}), \boldsymbol{\eta}^{'} - \boldsymbol{\eta}^{k} \right\rangle \qquad (3.32)\\
&= \int_{x\in\mathcal{A}} \sum_{j\in\mathcal{J}} \lambda(x)\nu(x)(\eta_j^{'}(x) - \eta_j^{k}(x)) \frac{1}{C_j r_j(x)\phi_j(k)} dx\\
&= \int_{x\in\mathcal{A}} \lambda(x)\nu(x) \sum_{j\in\mathcal{J}} (\eta_j^{'}(x) - \eta_j^{k}(x)) \frac{1}{C_j r_j(x)\phi_j(k)} dx.
\end{aligned}$$

Since

$$p^{k}(x) = \arg\max_{j\in\mathcal{J}} C_j r_j(x)\phi_j(k), \tag{3.33}$$

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

Then, we have

$$\sum_{j\in\mathcal{J}} \eta_j^{'}(x) \frac{1}{C_j r_j(x)\phi_j(k)} \geq \sum_{j\in\mathcal{J}} \eta_j^{k}(x) \frac{1}{C_j r_j(x)\phi_j(k)}. \tag{3.34}$$

Hence, $\left\langle \nabla L(\boldsymbol{\eta}), \boldsymbol{\eta}^{'} - \boldsymbol{\eta}^{k} \right\rangle \geq 0$. Therefore, $\boldsymbol{\eta}^{k}$ is the 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 LAB 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 Equations (3.24) and (3.25), 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.*

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 [19].

CHAPTER 4

APPLICATION AWARE EDGE COMPUTING FOR IOT

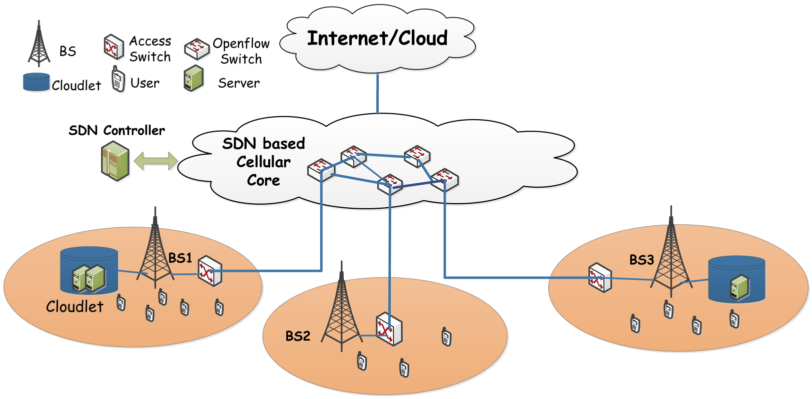

Figure 4.1 Cloudlet network architecture.

A distributed cloudlet network architecture is illustrated in Figure 4.1, where cloudlets are co-located with some base stations (BSs). The software defined network (SDN), which consists of a SDN controller and open flow switches, is employed as the cellular core network, thus enabling flexible routing and communications resource among BSs. All BSs are equipped with two interfaces (i.e., NB-IoT and LTE) to offer the seamless coverage for both IoT devices and IoT users (UEs). Thus, the sensed data of IoT devices can be stored at their closest cloudlets and the remote cloud, which act as brokers. Meanwhile, a Resource Directory (RD) is located at the SDN controller to help each IoT application discover the location of its required IoT data. On the other hand, each UE can access different cloudlets through its BS and the SDN based cellular core network. Within one cloudlet, we assume that each virtual machine (VM) only processes the workloads of one application, i.e., each application is mapped to a dedicated VM. Note that each IoT application has only one VM in a cloudlet. Considering the diversity of applications, the computing capacities of VMs are heterogeneous in a cloudlet and can be adjusted dynamically [2]. We define an

IoT App as the software program running on a UE that requests the specific type of application service. As a UE may run multiple IoT Apps, each type of application requests of the UE can be offloaded to a cloudlet having the corresponding type of VMs. Thus, when an application VM in a cloudlet receives an application request, it quickly retrieves the required IoT data from other brokers under the direction of RD and then processes the request to get the result.

Note that each UE may have several types of IoT Apps. As each App in a UE is assigned to only one cloudlet individually, the size of the set of Apps in the network can be derived as: $|\mathcal{Z}| = \sum\limits_{j\in\mathcal{J}} |\mathcal{K}_j|$, in which the variables are defined in the list of symbols shown in Table 4.1.

4.1 System Model

4.1.1 Computing Delay

Assume that type k requests of UE j are generated according to a Poisson Process with the average arrival rate λ_{jk}. Thus, the workload of type k VM in cloudlet i can be expressed as:

$$\lambda_{ik} = \sum_{j\in J} x_{ijk}\lambda_{jk}, \tag{4.1}$$

and it also follows a Poisson Process. On the other hand, the computing capacity (in terms of CPU cycles per second) of type k VM in cloudlet i (i.e., μ_{ik}) is fixed in each time slot; the computing size of a type k application request (in terms of the CPU cycles) follows an exponential distribution with the average value of l_k. Thus, we can derive the service time for type k requests running in a cloudlet's VM as l_k/μ_{ik}, which also follows an exponential distribution. Since the arrival rate of each VM of a cloudlet follows a Poisson Process while the corresponding service time follows an exponential distribution, each VM of a cloudlet can form an M/M/1 queuing model to process its corresponding application requests. Note that to keep the queue stable, the average

Table 4.1 List of Symbols in Application Aware Edge Computing for IoT

Symbol	Definition
$\mathcal{I}$	Set of distributed cloudlets.
$\mathcal{J}$	Set of UEs.
$\mathcal{K}$	Set of different IoT applications.
$\mathcal{R}$	Set of BSs.
x_{ijk}	Binary indicator of UE j's App k being assigned to cloudlet i.
y_{rj}	Binary indicator of UE j being covered by BS r.
$\mathcal{K}_j$	Set of Apps run by UE j.
μ_{ik}	Computing capacity of type k VM in cloudlet i.
τ_{ri}	E2E delay between BS r and cloudlet i.
λ_{jk}	Average request arrival rate of type-k App in UE j.
λ_{ik}	Average request arrival rate of type k VM in cloudlet i.
l_k	Average computing size of a type-k request.
d_{ij}	Network delay between UE j and cloudlet i.
D_k	Maximum allowed computing delay of Application k.
$\mathcal{Z}$	Set of Apps of all UEs.
j_z	Index of the UE where App $z \in Z$ is located.
d_{iz}	Network delay between App z and cloudlet i.

arrival rate of the VM (i.e., λ_{ik}) should be smaller than its average service rate (i.e., μ_{ik}/l_k), and thus we can derive that $\mu_{ik}/l_k - \lambda_{ik} > 0$. We define the computing delay of type k requests in cloudlet i, t_{ik}, as the average system delay of type k VM's queue (i.e., including the waiting delay and service time):

$$t_{ik} = \frac{1}{\mu_{ik}/l_k - \sum\limits_{j \in J} x_{ijk}\lambda_{jk}}, \forall i \in \mathcal{I}, k \in \mathcal{K}. \tag{4.2}$$

4.1.2 Network Delay

When a request of a UE is sent to a cloudlet, the request goes through its BS and the SDN-based cellular core network. Therefore, the E2E delay between a UE's App and its cloudlet consists of two parts: first, the E2E delay between the UE and its associated BS, i.e., the wireless delay; second, the E2E delay between its BS and its assigned cloudlet. However, the cloudlet selection for a UE does not affect its wireless delay, which only depends on the UE's service plan and the mobile provider's bandwidth allocation strategy [31]. Thus, we just consider the E2E delay between the BS and cloudlet. Denote τ_{ri} as the E2E delay between BS r and cloudlet i, and $\mathcal{Y}$ as a given indicator matrix to reflect the UE-BS association at the beginning of each time slot, in which $y_{rj} \in \mathcal{Y}$ represents whether UE j is covered by BS r or not. Note that the value of τ_{ri} can be measured and recorded by the SDN controller [33, 34]. Thus, the network delay between UE j and cloudlet $i \in \mathcal{I}$ can be expressed as

$$d_{ij} = \sum_{r \in \mathcal{R}} y_{rj}\tau_{ri}, \ \forall i \in \mathcal{I}, j \in \mathcal{J}. \tag{4.3}$$

4.2 Problem Formulation

The response time of a UE's App consists of both the computing delay and network delay. In the workload allocation, both of them should be taken into account. On one hand, owing to the dynamic distribution of workloads among different cloudlets, the overloaded cloudlets incur remarkably higher computing delay than other lightly loaded cloudlets. Thus, if the closest cloudlet of a UE is overloaded, the requests of each App of the UE should be allocated to alternative cloudlets to reduce the response time. On the other hand, offloading an App's requests from its closest cloudlet to other cloudlets will increase the network delay. The main goal is to minimize the response time of all IoT Apps in the network by assigning the requests of each App among cloudlets and flexibly allocating the computing resource of each cloudlet to different types of VMs to serve the assigned Apps. Thus, we can formulate the application aware workload allocation problem in each time slot as follows:

$$P1: \min_{x_{ijk},\mu_{ik}} \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}_j} x_{ijk}\left(d_{ij}+\frac{1}{\mu_{ik}/l_k-\sum\limits_{j\in\mathcal{J}} x_{ijk}\lambda_{jk}}\right) \tag{4.4}$$

$$s.t. \sum_{k\in\mathcal{K}} \mu_{ik} \leq C_i, \forall i \in \mathcal{I}, \tag{4.5}$$

$$\sum_{i\in\mathcal{I}} x_{ijk} = 1, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}_j, \tag{4.6}$$

$$x_{ijk}\left(\frac{1}{\mu_{ik}/l_k-\sum\limits_{j\in\mathcal{J}} x_{ijk}\lambda_{jk}}\right) \leq x_{ijk}D_k, \tag{4.7}$$

$$\forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}_j,$$

$$\mu_{ik}/l_k - \sum_{j\in\mathcal{J}} x_{ijk}\lambda_{jk} > 0, \forall i \in I, \forall k \in \mathcal{K}, \tag{4.8}$$

$$x_{ijk} \in \{0,1\}, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}_j, \tag{4.9}$$

$$\mu_{ik} \in \left[0, C_i\right], \forall i \in \mathcal{I}, \forall k \in \mathcal{K}. \tag{4.10}$$

Here, the objective function is to minimize the total response time of UEs Apps in the network. C_i is the computing capacity of cloudlet i and D_k is the maximum allowed computing delay of application k. Constraint (4.5) indicates that the aggregated computing resources of all VMs in a cloudlet should be no larger than the cloudlet's computing capacity. Constraint (4.6) ensures that each App of a UE is assigned to only one cloudlet. Constraint (4.7) imposes the computing delay for each UE's type k APP to meet the QoS requirement of the application in terms of the maximum allowed computing delay D_k. Constraint (4.8) imposes the average service rate of VM k in a cloudlet to be smaller than the VM's average task arrival rate, in order to keep the queue of the VM stable.

Lemma 5. *The problem of application aware workload allocation (i.e., P1) is NP-hard.*

Proof. Suppose there is only one IoT application; the capacity of VM k equals to the capacity of a cloudlet, i.e., $\mu_{ik} = C_i$. Meanwhile, we assume that the computing delay threshold $D_k = +\infty$. Therefore, both Constraints (4.5) and (4.7) can be relaxed from P1. Then, to prove that P1 is a NP-hard problem, we can demonstrate that its corresponding decision problem is NP-complete. The decision problem of P1 can be expressed as: given a positive value of b, is it possible to find a feasible solution $X = \{x_{ijk} | i \in \mathcal{I}, j \in \mathcal{J}\}$ such that $\sum\limits_{i \in I} \sum\limits_{j \in \mathcal{J}} x_{ijk} \left(d_{ij} + \frac{1}{\mu_{ik}/l_k - \sum\limits_{j \in \mathcal{J}} x_{ijk} \lambda_{jk}} \right) \leq b$, and Constraints (4.6), (4.8) and (4.9) are satisfied?

In order to prove that the above decision problem is NP-complete, only two cloudlets are considered and the average service rate of either cloudlet is set to be the same, i.e., $\mu_1/l_k = \mu_2/l_k = \frac{1}{2} \sum\limits_{j \in \mathcal{J}} \lambda_{jk} + \epsilon$, where ϵ is a very small positive value, i.e., $\epsilon \ll \frac{1}{2} \min\{\lambda_{jk} | j \in \mathcal{J}\}$. Moreover, assume that $b \to +\infty$. Thus, $\sum\limits_{i \in \mathcal{I}} \sum\limits_{j \in \mathcal{J}} x_{ijk} \left(d_{ij} + \frac{1}{\mu_{ik}/l_k - \sum\limits_{j \in \mathcal{J}} x_{ijk} \lambda_{jk}} \right) \leq b$ is always satified for all solutions of $\mathcal{X}$ and can be relaxed. To satisfy Constraint (4.8) (i.e., $\mu_{ik}/l_k - \sum\limits_{j \in \mathcal{J}} x_{ijk} \lambda_{jk} > 0, \forall i \in \mathcal{I}$), we need to guarantee that $\sum\limits_{j \in \mathcal{J}} \lambda_{jk} x_{1jk} = \sum\limits_{j \in \mathcal{J}} \lambda_{jk} x_{2jk} = \frac{1}{2} \sum\limits_{j \in \mathcal{J}} \lambda_{jk}$. Consequently, the decision problem can be transformed into a partition problem, i.e., whether the UEs can be partitioned into two sets to make the average request arrival rates of the two sets the same. Hence, the partition problem is reducible to the decision problem of P1. As the partition problem is a well-known NP-complete problem, the decision problem of P1 is also NP-complete, and thus P1 is NP-hard.

□

4.3 The AREA Algorithm

Since P1 is NP-hard, which is challenging to achieve the optimal solution, we design the heuristic Application awaRE workload Allocation (AREA) algorithm to effectively allocate different types of workloads among cloudlets as well as flexibly allocate computing resources for different VMs in each cloudlet, with low computational complexity. Note that the major challenge of solving P1 is that μ_{ik} depends on the App assignment x_{ijk}. To solve P1 more efficiently, we decompose the original problem into two sub-problems: the App assignment sub-problem and the resource allocation sub-problem. We will first assign different types of Apps among cloudlets (i.e., determining x_{ijk}), and then try to optimally allocate the computing resources to different types of VMs in each cloudlet (i.e., μ_{ik}) based on the given x_{ijk}.

4.3.1 App Assignment

When assigning Apps' workloads among cloudlets, the priority of assigning each App to its closest cloudlets should be considered to reduce the total network delay. Therefore, we will initialize the App assignment by allocating all Apps to their closest cloudlets; then, the algorithm will iteratively select a suitable App with the highest response time and reallocate it to an alternative cloudlet which minimizes its response time, until each App cannot find a better cloudlet.

Given the capacities of cloudlets, the initial App assignment is determined by the network delay between UEs that host Apps and cloudlets, and thus can be obtained by solving the following problem, which aims to minimize the total network delay between UEs' Apps and their cloudlets:

$$P2: \min_{x_{ijk}} \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}_j} x_{ijk} d_{ij} \tag{4.11}$$

$$s.t. \sum_{i\in\mathcal{I}} x_{ijk} = 1, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}_j, \tag{4.12}$$

$$\sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}_j} \lambda_{jk} l_k x_{ijk} \leq C_i, \forall i \in \mathcal{I} \tag{4.13}$$

$$x_{ijk} \in \{0,1\}, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}. \tag{4.14}$$

As each App of a UE is assigned among cloudlets individually, we denote $\mathcal{Z}_1$ as the set of Apps of all UEs which are waiting to be assigned among cloudlets, and $\mathcal{I}_1$ as the set of cloudlets which have excess computing resources. At the beginning, all UEs' Apps have not be assigned and are included in $\mathcal{Z}_1$ (i.e., $\mathcal{Z}_1 = \mathcal{Z}$), while all cloudlets are empty without any assigned Apps, i.e., all cloudlets are included in $\mathcal{I}_1$. Denote d_{iz} as the network delay between an App z (i.e., $z \in \mathcal{Z}_1$) and cloudlet i, j_z as the UE where App z is located. Hence, we have $d_{iz} = d_{ij_z}, \forall i \in \mathcal{I}, \forall z \in \mathcal{Z}_1$.

In the initialization, for App z, the optimal cloudlet $i^* \in \mathcal{I}_1$ is the one that incurs the lowest network delay, i.e., $i^* = \arg\min\{d_{iz}|i \in \mathcal{I}_1\}$; the suboptimal cloudlet $i^{'}$ is the one that incurs the second lowest network delay among the cloudlets in $\mathcal{I}_1$, i.e., $i^{'} = \arg\min_i\{d_{iz}|i \in \{\mathcal{I}_1 \backslash i^*\}\}$.

As shown in P2, the capacity of each cloudlet is limited, and thus it is impossible to allocate all Apps to their corresponding optimal cloudlets. The basic idea of the initialization is to iteratively select a suitable App, whose suboptimal cloudlet $i^{'}$ incurs a significant network delay degradation as compared to the optimal cloudlet i^*, and then allocate the App into its optimal cloudlet. It is easy to observe that the network delay degradation incurred by the suboptimal cloudlet determines the priority of assigning App z to its optimal cloudlet. For example, if App z's suboptimal cloudlet B leads to a remarkably higher delay than its optimal cloudlet A as compared to other Apps, assigning App z to the suboptimal cloudlet will significantly impact the total network delay of all Apps. In this case, App z is given a higher priority than other Apps to be assigned into its optimal cloudlet A.

Denote Δd_z as the network delay degradation by allocating App z from the optimal cloudlet i^* to the suboptimal cloudlet $i^{'}$, i.e.,

$$\Delta d_z = d_{i^{'}z} - d_{i^*z}, \forall z \in \mathcal{Z}_1. \tag{4.15}$$

Thus, as shown in Algorithm 1, in each iteration of the initialization, the algorithm will select and allocate a suitable App z, which has the highest network delay degradation (i.e., $z = \arg\max\{\Delta d_z | z \in \mathcal{Z}_1\}$), to its optimal cloudlet. Afterwards, if the workload of a cloudlet exceeds its capacity, the cloudlet is removed from $\mathcal{I}_1$. Note that once $\mathcal{I}_1$ is updated, the algorithm has to recalculate i^*, $i^{'}$ and Δd_z for each App $z \in \mathcal{Z}_1$. The above procedure is repeated until all Apps are assigned among cloudlets, i.e., $\mathcal{Z}_1 = \emptyset$.

Lemma 6. *Algorithm 1 terminates after a finite number of iterations, yielding a feasible IoT App assignment.*

Proof. Let $\xi = |\mathcal{I}_1| = N$ initially, i.e., $\xi > 0$. Then, for each iteration, since the algorithm chooses a suitable App z, where $z = \arg\max_z\{\Delta d_z | z \in \mathcal{Z}_1\}$, and allocates it to its optimal cloudlet i^* (i.e., $i^* = \arg\min_i\{d_{iz} | i \in \mathcal{I}_1\}$), ξ is decremented by one. As a result, ξ will become zero after a finite number of iterations, and thus $\mathcal{I}_1 = \emptyset$. □

As shown in Algorithm 1, the complexity of Step 2 is $|\mathcal{Z}|$. After Step 2, the complexity of Steps 4-5 is $O(|\mathcal{Z}| + |\mathcal{I}|)$ in the worst case; as they repeat for $|\mathcal{Z}|$ times, the corresponding complexity is $O(|\mathcal{Z}|(|\mathcal{Z}| + |\mathcal{I}|))$. Meanwhile, as Steps 9-10 repeat for at most $|\mathcal{I}|$ times, the corresponding complexity is $O((|\mathcal{Z}| + 1)|\mathcal{I}|)$. Aggregating all these steps, the complexity of Algorithm 1 becomes $O(|\mathcal{Z}|(|\mathcal{Z}| + |\mathcal{I}|))$.

Algorithm 2

Input: The UE-BS association vector $\mathcal{Y} = \{y_{rj}|r \in \mathcal{R},\ j \in \mathcal{I}\}$. The matrix of E2E delay between BSs and cloudlets $\mathcal{T} = \{\tau_{ri}|r \in \mathcal{R},\ i \in \mathcal{I}\}$. The vector of the average task arrival rate for UEs' Apps $\Lambda = \{\lambda_{jk}|j \in \mathcal{J}, j \in \mathcal{K}_j\}$.

Output: The initial App assignment matrix, i.e., $\mathcal{X} = \{x_{ijk}|i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{K}_j\}$.

```
Set $\mathcal{Z}_1 = \mathcal{Z}$ and $\mathcal{I}_1 = \mathcal{I}$ based on their definitions;
$\forall z \in \mathcal{Z}_1$, calculate $\Delta d_z$ based on Equation (4.15);
while $\mathcal{Z}_1 \neq \emptyset$ do
    Find App $z$, where $z = \arg\max_{z}\{\Delta d_z|z \in \mathcal{Z}_1\}$;
    Allocate App $z$ to its optimal cloudlet $i^*$ (i.e., $i^* = \arg\min_{i}\{d_{ij}|i \in \mathcal{I}_1\}$);
    Let $x_{ij_zk_z} = 1$;
    Update the App set $\mathcal{Z}_1$, i.e., $\mathcal{Z}_1 = \mathcal{Z}_1 \backslash z$ .
    if cloudlet $i^*$ is full then
        Remove $i^*$ from $\mathcal{I}_1$, i.e., $\mathcal{I}_1 = \mathcal{I}_1 \backslash i^*$;
        $\forall z \in \mathcal{Z}_1$, recalcuate $\Delta d_z$ based on Equation (4.15);
    end if
end while
    return $\mathcal{X}$.
```

After the initialization, the AREA algorithm, as shown in Algorithm 2, iteratively selects a suitable App with the highest response time, and reallocates it to an alternative cloudlet. At the beginning, all Apps are unmarked and we define $\mathcal{Z}_2$ as the set of unmarked Apps. Then, in each iteration, the AREA algorithm finds the App with the highest response time among all unsigned Apps, and searches for a new cloudlet for the App to minimize its response time. Note that in each iteration, the computing resource for each application in a cloudlet is determined by the percentage of the application's workload in the total workloads in the cloudlet, and thus we can derive the response time of Apps in different cloudlets. If a new cloudlet is found, AREA proceeds to the next iteration. Otherwise, the algorithm marks the App (i.e., removing the App from $\mathcal{Z}_2$) and continues to the next iteration. The AREA algorithm repeats the iterations until $\mathcal{Z}_2 = \emptyset$.

We now analyze the computational complexity of Algorithm 2. In each iteration, the algorithm checks cloudlets for an App, and the number of related cloudlets can be $|\mathcal{I}|$ in the worst case. Therefore, the complexity of each iteration is $O(|\mathcal{I}|)$. Then, we analyze the required number of iterations for the algorithm to optimally place all Apps among the cloudlets. Each App has a choice of up to $|\mathcal{I}|$ cloudlets. In each cloudlet, the App can have at most $|\mathcal{Z}|$ different response times owing to the different number of Apps allocated to the cloudlet. As a result, the number of improvements for the App is limited by $|\mathcal{I}||\mathcal{Z}|$. Thus, considering the number of Apps is $|\mathcal{Z}|$, the total number of iterations in the worst case is $|\mathcal{I}||\mathcal{Z}|^2$. So, the computational complexity of Algorithm 2 is $O(|\mathcal{I}|^2|\mathcal{Z}|^2)$. When we fix the number of cloudlets $|\mathcal{I}|$, the complexity of Algorithm 2 is polynomial with respect to the number of the Apps.

Algorithm 3

```
Initialize App assignment by Algorithm 1 and obtain $\mathcal{X}$;
Set $Z_2$ based on its definition, i.e., $\mathcal{Z}_2 = \{z|z \in \mathcal{Z}\}$
while $\mathcal{Z}_2 \neq \emptyset$ do
    Find App $z \in \mathcal{Z}_2$ with the highest response time;
    Obtain the current cloudlet $i$ of App $z$;
    Find the suitable cloudlet $i^*$ for App $z$, i.e., $i^* = \arg\min \left( d_{ij} + \frac{1}{\mu_{ik}/l_k - \sum\limits_{j \in \mathcal{J}} x_{ijk}\lambda_{jk}} \right)$;
    if $i^* \neq i$ then
        Assign App $z$ to the new cloudlet $i^*$ and update $\mathcal{X}$;
    else
        Mark App $z$ and let $\mathcal{Z}_2 = \mathcal{Z}_2 \backslash z$;
    end if
end while
    return $\mathcal{X}$.
```

4.3.2 Resource Allocation

After all UEs' Apps are assigned to different cloudlets, the primary problem P1 can be transformed into a resource allocation problem for each cloudlet i as follows:

$$P3 : \min_{\mu_{ik}} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} x_{ijk} \left(d_{ij} + \frac{1}{\mu_{ik}/l_k - \sum\limits_{j \in \mathcal{J}} x_{ijk}\lambda_{jk}} \right) \tag{4.16}$$

$$s.t. \;\; Constraints(4.5), (4.7), (4.8), (4.10).$$

We can then prove the following lemma:

Lemma 7. *When each x_{ijk} is determined, $P3$ is a convex optimization problem.*

Proof. For brevity, let $f = \sum\limits_{j \in \mathcal{J}} \sum\limits_{k \in \mathcal{K}} x_{ijk} \left(d_{ij} + \frac{1}{\mu_{ik}/l_k - \sum\limits_{j \in \mathcal{J}} x_{ijk}\lambda_{jk}} \right)$, and we use μ_k to substitue μ_{ik} in cloudlet i. Thus, we have

$$\frac{\partial^2 f}{\partial \mu_k \partial \mu_{k'}} = \begin{cases} \sum\limits_{j \in \mathcal{J}} 2x_{ijk} l_k^{-2} (\mu_k/l_k - \sum\limits_{j \in \mathcal{J}} x_{ijk}\lambda_{jk})^{-3}, \text{if} k = k', \\ 0, \text{ otherwise.} \end{cases} \tag{4.17}$$

Since $(\mu_k/l_k - \sum\limits_{j \in \mathcal{J}} x_{ijk}\lambda_{jk}) > 0$, the Hessian matrix $H = \frac{\partial^2 f}{\partial \mu_k \partial \mu_{k'}}$ of f is a positive definite matrix. As a result, function f is convex. Moreover, since Constraints (4.5), (4.7), (4.8), (4.10) are linear, the optimization problem $P3$ is a convex optimization problem.

□

As $P3$ is a convex problem, we can derive the optimal solution of $P3$ by solving the KKT condition of $P3$ [35]. Therefore, the computing resource of each cloudlet is optimally allocated to different VMs to minimize the response time. Consequently, the suboptimal solution of P1 is achieved.

CHAPTER 5

LOAD BALANCING IN DRONE-ASSISTED COMMUNICATIONS FOR IOT

5.1 System Model

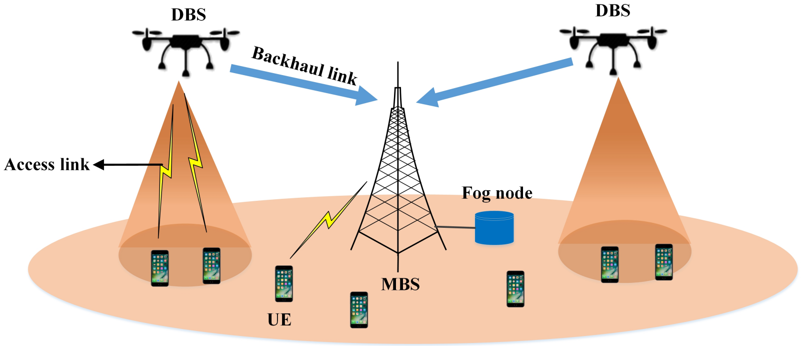

Figure 5.1 DBS-assisted edge computing architecture.

In the network as shown in Figure 5.1, each MBS is attached with a fog node and two interfaces (i.e., LTE and NB-IoT) that offer seamless coverage for IoT users and IoT devices. Considering the large number of heterogeneous IoT devices, IoT devices can employ NB-IoT interfaces to communicate with the MBS. Hence, the data of IoT devices can be transfered to and stored at their local fog nodes, which work as brokers. Meanwhile, a Resource Directory (RD) is deployed at the mobile core network [36]. Upon receiving an IoT request, the IoT application in the fog node can promptly process the request by retrieving and operating on data from other brokers under the supervision of RD. In addition, as IoT requests are processed in their local fog nodes, DBSs can be placed over particular hotpot areas in the coverage region of the MBS to reduce the latency of delivering IoT requests from IoT users to the fog node (i.e., uplink). For each DBS, both the access link and backhaul link share the same in-band frequency spectrum.

In this chapter, the whole radio coverage region of an MBS is divided into a number of locations, each with a small coverage. Denote $\mathcal{I}$ as the set of all of these locations, and i as the index of a location within $\mathcal{I}$. Denote $\mathcal{J}$ as the set of potential locations for BSs (note that $\mathcal{J} \subset \mathcal{I}$), in which $s \in \mathcal{J}$ is the predefined location of the MBS and $\mathcal{J}\backslash s$ represents the potential locations that DBSs can be placed. Note that if $y_j = 0$, η_{ij} will always be zero. We assume that IoT requests arrive in each location according to a Poisson Point Process having an average arrival rate λ_i at location i. The key notations used in this chapter are listed in Table 5.1.

Table 5.1 List of Symbols in Drone-assisted Communications for IoT

Symbol	**Definition**
η_{ij}	Indicator of UEs at location i being assigned to BS j.
y_j	Indicator of a DBS being placed at candidate location $j \in \{\mathcal{J}\backslash s\}$
p^{los}	Probability of LoS channel.
φ_{ij}^{los}	Path loss of the LoS channel between location i and DBS j.
φ_{ij}^{nlos}	Path loss of the NLoS channel between location i and DBS j.
r_{ij}	Data rate of a UE at location i towards DBS j.
ρ_j	BS j's traffic load.
μ_j	Communications latency ratio of BS j.
N	Number of DBSs that can be placed in the network.
λ_i	Average request arrival rate at location i.
ρ_{max}	Maximum allowed traffic load of each BS.

5.1.1 Communications Model

The Average Path Loss between UEs and A DBS The communications channel between a DBS and UEs at location i is assumed to be a probabilistic Line of Sight (LoS) channel, where the probability of the LoS channel between them is [24]

$$p^{los} = \frac{1}{1 + ae^{-b(\theta_{ij} - a)}}. \tag{5.1}$$

Here, a and b are dependent on the specific environment (rural, urban, etc.) and are constant parameters that can be measured proactively. Meanwhile, θ_{ij} is the elevation angle (in degree) between DBS j and location i, and can be expressed as $\theta_{ij} = \arctan\left(\frac{h_d}{\delta_{ij}}\right)$ in which h_d is the DBS's altitude and δ_{ij} is the horizontal distance between the DBS and location i. Note that we assume that the altitudes of DBSs are predefined (i.e., h_d).

Denote φ_{ij}^{los} and φ_{ij}^{nlos} as the path loss between UEs at location i and DBS j with the LoS connection and non-LoS (NLoS) connection, respectively [37].

$$\varphi_{ij}^{los} = \xi^{los} + \tau^{los}\log_{10}\left(\sqrt{(\delta_{ij})^2 + (h_d)^2}\right), \tag{5.2}$$

$$\varphi_{ij}^{nlos} = \xi^{nlos} + \tau^{nlos}\log_{10}\left(\sqrt{(\delta_{ij})^2 + (h_d)^2}\right). \tag{5.3}$$

Here, ξ^{los} and ξ^{nlos} indicate the path loss at the reference distance for the LoS and NLoS connections; τ^{los} and τ^{nlos} represent the path loss exponents for the LoS and NLoS connections, respectively. Note that the parameters can be measured in specific areas. Moreover, the 3D distance between DBS j and location i is calculated by $\sqrt{(\delta_{ij})^2 + (h_d)^2}$. Therefore, we can derive the average path loss between UEs at location i and DBS j as:

$$\varphi_{ij} = p^{los}\varphi_{ij}^{los} + (1 - p^{los})\varphi_{ij}^{nlos}. \tag{5.4}$$

The Average Path Loss from DBS to MBS Assuming the altitude of a DBS is high enough to enable the LoS channel from the DBS to the MBS, the path loss between DBS j and the MBS can be expressed as

$$\varphi_{sj} = \xi^{los} + \tau^{los}\log_{10}\left(\sqrt{(\delta_{sj})^2 + (h_d - h_s)^2}\right), \tag{5.5}$$

where h_s is the altitude of the MBS and δ_{sj} is the horizontal distance from DBS j to the MBS.

Data Rate of UEs Denote g_{ij} as the uplink channel gain from location i to BS j and P_i as the transmission power of the UE at location i. Let σ^2 be the noise power. Hence, we can model the signal to noise ratio (SNR) of location i towards BS j (the access link) as $\gamma_{ij} = \frac{P_i g_{ij}}{\sigma^2}, j \in \mathcal{J}$, where $g_{ij} = 10^{\frac{-\varphi_{ij}}{10}}$. Therefore, the data rate of the access link at location i can be modeled as

$$r_{ij} = W_j \; log(1 + \gamma_{ij}), \tag{5.6}$$

where W_j is the bandwidth exclusively used by BS j [38].

Meanwhile, denote $g_j^d s$ as the channel gain from DBS j to the MBS and P_j^d as the transmission power of DBS j. Thus, the signal to noise ratio (SNR) of the backhaul link from DBS j to MBS s is

$$\gamma_{js}^d = \frac{P_j^d g_{js}^d}{\sigma^2}, j \in \mathcal{J} \backslash s. \tag{5.7}$$

When a DBS is used as a relay between the MBS and UEs, either the Decode-and-Forward (DF) [39] or Amplify-and-Forward (AF) [40] cooperative communication mode can be adopted. We assume that each DBS employs the DF cooperative communication mode to relay the data towards the MBS. In the DF mode, the time domain for a UE is divided into two parts (two slots). In the first slot, the UE broadcasts its data, and thus both the DBS and MBS act as receivers. Then, in the second time slot, the DBS decodes the received data and forwards it to the MBS [41]. Based on the DF mode, considering both the access and backhaul link, the data rate of UEs at location x towards the MBS via the DBS $j \in \mathcal{J}\backslash s$ is expressed as [42]

$$r_{ij}^{d} = \frac{W_j}{2} \min(\log_2(1 + \gamma_{ij}), \log_2(1 + \gamma_{js}^{d} + \gamma_{is})). \tag{5.8}$$

Thus, the data rate of a UE can be summarized as

$$r_{ij} = \begin{cases} r_{is}, \text{ if } j = s \\ r_{ij}^{d}, \text{ if } j \neq s. \end{cases} \tag{5.9}$$

5.1.2 Traffic Load Model

We assume that IoT requests arrive at location i based on a Poisson Process having the average request arrival rate λ_i. The traffic sizes of all IoT requests follow an exponential distribution with the average value of l_i. Therefore, the traffic load density of location i in BS j can be derived as [38]

$$\varrho_{ij} = \frac{\lambda_i l_i \eta_{ij}}{r_{ij}}, \tag{5.10}$$

where η_{ij} is a binary indicator representing whether location i is associated with BS j.

The average traffic load ρ_j of BS i can be expressed as the sum of traffic load densities of its associated locations. In particular, ρ_j represents the utilization of the BS (i.e., how much time BS j is busy): $\rho_j = \sum\limits_{i \in \mathcal{I}} \varrho_{ij}$.

For the uplink channel, many scheduling algorithms have been designed to enable UEs to properly share ratio resources of a BS. To be analytically tractable, we assume that UEs of a BS schedules the transmissions of its associated UEs in a round robin fashion, i.e., different UEs can access the uplink channel sequentially. Meanwhile, as mentioned above, the request arrival rate of each location satisfies a Poisson Process, and thus the aggregated request arrival rate of a BS also satisfies a Poisson Process. In addition, the traffic size of IoT requests follows a general distribution such that the service time of IoT requests also satisfies a general distribution. The average service time of location i's IoT requests can be expressed $\tau_{ij} = \frac{l_i}{r_{ij}}$. Thus, based on queuing theory, the uplink communications of a BS can realize an M/G/1 processor sharing queue [30]. Note that each BS's traffic load should always be smaller than 1 to maintain stability of the queuing system.

With the M/G/1 processor sharing queue of a BS, we can model the average delivery time of IoT requests at location i as [32]: $t_{ij} = \frac{l_i}{r_{ij}(1-\rho_j)}$. At the same time, the average waiting time for each IoT request at location i is expressed as

$$w_{ij} = t_{ij} - \tau_{ij} = \frac{\rho_j l_i}{r_{ij}(1-\rho_j)}. \tag{5.11}$$

Let μ_{ij} be the latency ratio of the waiting time to the service time in BS j for IoT requests at location i:

$$\mu_{ij} = \frac{w_{ij}}{\tau_{ij}} = \frac{\rho_j}{1-\rho_j}. \tag{5.12}$$

It is obvious that μ_{ij} is determined by the traffic load of BS j. Hence, all locations covered by BS j will have the same latency ratio. Therefore, the communications latency ratio of BS j can be defined as

$$\mu_j = \frac{\rho_j}{1-\rho_j}. \tag{5.13}$$

It can be seen from Equation (5.13) that when traffic load ρ_j of BS j increases, μ_j also increases. Increasing μ_j implies that it takes a longer time for UEs at locations covered by BS j to access the transmission channel. Hence, μ_j is used to reflect the average delivery delay of BS j.

5.2 Problem Formulation

The goal of this is to improve the latency of all UEs by placing DBSs to suitable locations and balancing the traffic loads among BSs. Let the latency ratio of the network be $L = \sum\limits_{j \in \mathcal{J}} \frac{\rho_j}{1-\rho_j}$. Thus, the problem is to optimally place DBSs in the network and associate UEs to BSs so as to minimize the latency ratio of the network. Therefore, we can formulate the problem as follows

$$P1: \min_{y_j, \eta_{ij}} \sum_j \frac{\rho_j}{1-\rho_j} \tag{5.14}$$

$$s.t. \sum_j \eta_{ij} = 1, \forall i \in \mathcal{I}, \tag{5.15}$$

$$0 \leq \rho_j \leq \rho_{max}, \tag{5.16}$$

$$\eta_{ij} \leq y_j, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \tag{5.17}$$

$$\sum_j y_j = N, \forall j \in \mathcal{J} \backslash s, \tag{5.18}$$

$$y_j \in \{0, 1\}, y_s = 1, \eta_{ij} \in \{0, 1\}, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}. \tag{5.19}$$

The objective of this problem is to minimize the latency ratio of the network. ρ_{max} is the maximum allowed traffic load of each BS and N is the number of DBSs that can be placed in the network. Constraint (5.15) imposes each location to be associated to only one BS. Constraint (5.16) imposes the traffic load of each BS not to be larger than the maximum allowed traffic load ρ_{max}. Constraint (5.17) represents that IoT requests at location i can be assigned to a DBS at location j only if the DBS has been placed at location j in advance. Constraint (5.18) indicates that the number of DBSs is N.

5.3 The TALL Scheme

Since P1 is an interger non-linear programming problem which is challenging to solve, we design the TrAffic Load baLancing (TALL) scheme to effectively tackle the problem. Note that the user association η_{ij} is dependent on the DBS placement y_j. Consequently, the original problem is decomposed into two sub-problems: the DBS placement and the user association.

5.3.1 DBS Placement

In the DBS placement, DBSs are preferred to be placed over locations with high user densities so that they can provide LoS channels for more UEs and offload more traffic loads from the MBS. Thus, we will select some locations for DBSs such that the total distance between UEs and BSs is minimized. Then, the DBS placement problem is formulated as follows:

$$P2 : \min_{y_j, \eta_{ij}} \sum_j \sum_i \lambda_i \eta_{ij} \delta_{ij} \tag{5.20}$$

$$s.t. \sum_{j \in \mathcal{J}} \eta_{ij} = 1, \forall i \in \mathcal{I}, \tag{5.21}$$

$$\eta_{ij} \leq y_j, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \tag{5.22}$$

$$\sum_j y_j = N, \forall j \in \mathcal{J} \backslash s, \tag{5.23}$$

$$y_j \in \{0, 1\}, y_s = 1, \eta_{ij} \in \{0, 1\}, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \tag{5.24}$$

where δ_{ij} is the distance between location i and DBS j and λ_i is the weight of the distance. For simplicity, let $d_{ij} = \lambda_i \delta_{ij}$ be the weighed distance between DBS j and location i.

Lemma 8. *The DBS deployment problem P2 is NP-hard.*

Proof. Suppose the bandwidth allocated to the MBS is zero; all UEs are served by the DBSs. In this case, P2 becomes a p-median problem which is a classical NP-hard problem. Since p-median problem is reducible to P2, the DBS placement problem P2 is also NP-hard. □

To solve the DBS placement problem, we design a heuristic algorithm to obtain the sub-optimal solution. We will initialize the DBS deployment by placing DBSs sequentially, and then adjust the locations of DBSs iteratively until each DBS cannot find a better location.

In the initialization, we will place a new DBS in each iteration until all DBSs are placed. Denote $\mathcal{J}_1$ as the set of selected locations for DBSs, and $\mathcal{J}_2$ as the set of remaining candidate locations (i.e., $\mathcal{J}_2 = \mathcal{J} \backslash \mathcal{J}_1$). At the beginning, let $\mathcal{J}_1 = \emptyset$ and $\mathcal{J}_2 = \mathcal{J}$. In the nth iteration, C_j is the total wighted distance between UEs and DBSs (i.e., the objective function of P2) if the new DBS is placed at the candidate location j; thus, we have $C_j = \min\limits_{\eta_{ij}} \sum\limits_{k \in \mathcal{J}_1 \cup j} \sum\limits_{i} \eta_{ik} d_{ik}, \forall j \in \mathcal{J}_2$. As shown in Algorithm 1, the basic idea of the initialization is to iteratively choose a suitable location $j \in \mathcal{J}_2$ for the new DBS such that C_j is minimized in each iteration. Specifically, in the nth iteration, $(n-1)$ DBSs have already be placed; thus, we need to select the candidate location with the minimum C_j for the nth DBS.

After the initialization, all DBSs are deployed in the network; then, the algorithm will iteratively adjust the locations of DBSs to approach the optimal solution. Denote j_i^1 as the optimal (closest) BS for UEs at location i, and d_i^1 as the corresponding weighted distance between location i and BS j_i^1. Denote j_i^2 as the sub-optimal BS for UEs at location i, and d_i^2 as the sub-optimal weighted distance. Denote $\mathcal{I}_j^1$ as the set of locations whose optimal DBS is j (i.e., $\{i | i \in \mathcal{I}, j_i^1 = j\}$). Denote $j' \in \mathcal{J}_2$ as a candidate location for DBSs. Then, the interchange benefit by placing DBS $j \in \mathcal{J}_1$ to location $j' \in \mathcal{J}_2$ can be derived as

$$\Delta C_{jj'} = \sum_{i \in \mathcal{I} \backslash \mathcal{I}_j^1} \max(0, (d_i^1 - d_{ij'})) - \sum_{i \in \mathcal{I}_j^1} (min(d_i^2, d_{ij'}) - d_i^1),$$

$$\forall j \in \mathcal{J}_1, \forall j' \in \mathcal{J}_2, \tag{5.25}$$

where the first term represents the distance reduction incurred by moving DBS j to location j' while the second term is the distance increment due to the redeployment.

As shown in Algorithm 1, from Step 8, the DBS placement algorithm iteratively selects a suitable DBS and moves it to a better location such that the interchange benefit is maximized. When all candidate locations of $\mathcal{J}_2$ have been checked or each DBS cannot find a better location, the algorithm stops.

The complexity of Algorithm 1 is analyzed as follows. The complexity of Steps 4-6 is $|\mathcal{J}|+2$; as they repeat for N times, the corresponding complexity is $O(|\mathcal{J}|N)$. In addition, the complexity of Step 10 is $|\mathcal{J}_1||\mathcal{J}|$ (i.e., $|\mathcal{J}_1| = N$) while Step 11 has the same complexity. As Steps 10-17 repeat for at most $|\mathcal{J}|$ times, the corresponding complexity is $O(2N|\mathcal{J}|^2 + 2|\mathcal{J}|)$. Summarizing all the steps, the total complexity of Algorithm 1 can be expressed as $O(N|\mathcal{J}|^2)$.

5.3.2 User Association

After DBSs have been deployed in the coverage area of the MBS, the locations of all DBSs are determined. Denote the set of both the MBS and DBSs as $\mathcal{J}_0$. Then, problem P1 can be transformed into:

$$P3: \min_{\eta_{ij}} \sum_{j \in \mathcal{J}_0} \frac{\rho_j}{1-\rho_j} \tag{5.26}$$

$$s.t. \sum_{j \in \mathcal{J}_0} \eta_{ij} = 1, \forall i \in \mathcal{I}, \tag{5.27}$$

$$0 \leq \rho_j \leq \rho_{\max}, \forall j \in \mathcal{J}_0. \tag{5.28}$$

In this section, we design a user association algorithm to enable all BSs to iteratively estimate their traffic loads until the latency ratio of the network is minimized. At the beginning, all UEs report their data rates towards BSs to the MBS; then the MBS will execute the algorithm to achieve the optimal user association.

Algorithm 4 The DBS placement algorithm

```
Start the initializaion; set $\mathcal{J}_1 = \emptyset$; let $n = 0$;
Place the MBS to its predefined location $s$;
while ($n \leq N$) do
    Set $n = n + 1$;
    Find the candidate location $j \in \mathcal{J}_2$ with the minimum $C_j$ for the new DBS;
    Let $\mathcal{J}_1 = \mathcal{J}_1 \cup j$ and $\mathcal{J}_2 = \mathcal{J}_2 \backslash j$
end while
Start the deployment adjustment; set $\mathcal{J}_2 = J \backslash \mathcal{J}_1$;
while $\mathcal{J}_2 \neq \emptyset$ do
    Calculate $\Delta C_{jj'}, \forall j \in \mathcal{J}_1, \forall j' \in \mathcal{J}_2$;
    Find $j$, $j'$ and $\Delta C^*_{jj'}$ by $\{j, j'\} = \underset{j \in J_1, j' \in \mathcal{J}_2}{\arg\max} \Delta C_{jj'}$;
    if $\Delta C^*_{jj'} > 0$ then
        Let $\mathcal{J}_1 = \mathcal{J}_1 \backslash j$ and $\mathcal{J}_1 = \mathcal{J}_1 \cup j'$;
        Let $\mathcal{J}_2 = \mathcal{J}_2 \backslash j'$;
    else
        break;
    end if
end while
     return $\mathcal{J}_1$.
```

Specifically, in each iteration, the algorithm running in the MBS consists of two parts: the BS selection for UEs at different locations and traffic load estimation for BSs.

The BS selection At the beginning of the kth iteration, the algorithm selects the optimal BS for each UE based on estimated traffic loads of BSs and the UEs' data rates towards BSs. Specifically, according to the definition of L, we have

$$\frac{\partial L(\rho)}{\partial \rho_j} = (\phi_j(k))^{-1} = \frac{1}{(1-\rho_j)^2}. \tag{5.29}$$

Consequently, the suitable BS for UEs at location i is

$$p_i{}^k = \arg\max_{j \in \mathcal{J}_0} r_{ij}\phi_j(k), \tag{5.30}$$

where p_i^k is the BS's index selected by UEs at location i.

The traffic load estimation Once the BS for each UE is selected in iteration k, the perceived traffic load of BS is

$$\rho_j^k = min(\sum_{i \in \mathcal{I}} \frac{\lambda_i l_i \eta_{ij}^k}{r_{ij}}, \rho_{max}). \tag{5.31}$$

Denote $\tilde{\rho}_j^k$ as the estimated traffic load of BS j in the kth iteration. After obtaining the perceived traffic loads, the user association algorithm estimates the traffic load of each BS in the next iteration as:

$$\tilde{\rho}_j^{k+1} = (1-\beta(k))\rho_j^k + \beta(k)\tilde{\rho}_j^k, \tag{5.32}$$

where $\beta(k)$ is a system parameter to enable

$$L(\tilde{\rho}^{k+1}) \leqslant L(\tilde{\rho}^k) + \zeta(1-\beta(k)) \sum_{j \in J_0} \phi_j(k)^{-1}(\rho_j^k - \tilde{\rho}_j^k). \tag{5.33}$$

Here, $0 \leq \zeta \leq 0.5$ is a constant. The detailed procedure of the user association algorithm is illustrated in Algorithm 2.

Note that the feasible set of P3 is:

$$\begin{aligned} F = &\{\boldsymbol{\rho} | \rho_j = \sum_{i \in \mathcal{I}} \frac{\lambda_i l_i \eta_{ij}}{r_{ij}}, \eta_{ij} \in \{0,1\}, 0 \leq \rho_j \leq \rho_{\max}, \\ &\sum_{j \in \mathcal{J}_0} \eta_{ij} = 1, \forall j \in \mathcal{J}_0, \forall i \in \mathcal{I}\}. \end{aligned}$$

Algorithm 5 The user association algorithm

```
Initialize the estimated traffic loads $\tilde{\rho}_j, \forall j \in \mathcal{J}_1$;
Let $k = 0$;
**while** (1) **do**
    Set $k = k + 1$;
    Find the suitable BS for all UEs based on:
$p_i^k = \arg\max_{j \in \mathcal{J}_0} C_j r_{ij} \phi_j(k)$;
    Calculate the perceived traffic loads $\rho_j, \forall j \in \mathcal{J}_1$ based on Equation (5.31);
    **if** $L(\boldsymbol{\rho}^k) - L(\boldsymbol{\rho}^{k-1}) \leq \epsilon$ **then**
        break;
    **end if**
    Assign $\beta(k) = 0$;
    **while** (33) is not true **do**
        $\beta(k) = 1 - \xi(1 - \beta(k))$, where $0 \leq \xi \leq 1$;
    **end while**
    Update the estimated traffic load for each BS based on:
$\tilde{\rho}_j^{k+1} = (1 - \beta)\rho_j^k + \beta\tilde{\rho}_j^k, j \in \mathcal{J}_0$;
**end while**
    **return** $\boldsymbol{\rho}$.
```

As $\eta_{ij} \in \{0,1\}$, $\boldsymbol{\rho}$ is not continuous such that F is not a convex set. To gradually decrease the average latency ratio $L(\boldsymbol{\rho}^k)$ by estimating the optimal traffic loads in each iteration, we relax the constraint to make $0 \leq \boldsymbol{\rho}^k \leq 1$, and then show that the traffic load vector can finally converge in the feasible set. Thus, the relaxed feasible set of P3 is expressed as:

$$\begin{aligned}\hat{F} =&\{\boldsymbol{\rho}|\rho_j = \sum_{i\in\mathcal{I}} \frac{\lambda_i l_i \eta_{ij}}{r_{ij}}, 0 \leq \eta_{ij} \leq 1, 0 \leq \rho_j \leq \rho_{\max},\\ &\sum_{j\in\mathcal{J}_0} \eta_{ij} = 1, \forall j \in \mathcal{J}_0, \forall i \in \mathcal{I}\}.\end{aligned}$$

Lemma 9. *The objective function $L(\boldsymbol{\rho})$ is convex, when ρ is defined in $\hat{F}$.*

Proof. The proof of this lem can be easily made by showing that $\nabla^2 L(\boldsymbol{\rho}) > 0$ where $\boldsymbol{\rho}$ is defined in $\hat{F}$. □

Analysis of the algorithm In this section, the convergence and optimality of the user association algorithm in the feasible set F is analyzed.

Lemma 10. *When $\boldsymbol{\rho}^k \neq \tilde{\boldsymbol{\rho}}^k$, $\boldsymbol{\rho}^k$ provides a descent direction for $L(\tilde{\boldsymbol{\rho}})$ at $\tilde{\boldsymbol{\rho}}^k$.*

Proof. As $0 \leq \tilde{\eta}_j^k(x) \leq 1$, $L(\tilde{\boldsymbol{\rho}})$ is defined in $\hat{F}$. As shown in lem 2, $L(\tilde{\boldsymbol{\rho}})$ is a convex function of $\tilde{\boldsymbol{\rho}}$. Therefore, we need to prove $\left\langle \nabla L(\tilde{\boldsymbol{\rho}}^k), \boldsymbol{\rho}^k - \tilde{\boldsymbol{\rho}}^k \right\rangle < 0$. Hence, we have

$$\begin{aligned}&\left\langle \nabla L(\tilde{\boldsymbol{\rho}}^k), \boldsymbol{\rho}^k - \tilde{\boldsymbol{\rho}}^k \right\rangle \\ &= \sum_{i\in\mathcal{I}} \lambda_i l_i \sum_{j\in\mathcal{J}_0} \frac{\eta_{ij}^k - \tilde{\eta}_{ij}^k}{r_{ij}\phi_j(k)}\end{aligned} \tag{5.34}$$

Note that $\eta_{ij}^* = \begin{cases} 1, \text{ if } j = p^k(x) \\ 0, \text{ if } j \neq p^k(x). \end{cases}$

Considering the BS selection rule at the user side in the kth iteration, i.e., $p_i^k = \arg\max_{j\in\mathcal{J}_0} r_{ij}\phi_j(k)$, we can derive

$$\sum_{j\in\mathcal{J}_0} \frac{\eta_{ij}^k - \tilde{\eta}_{ij}^k}{r_{ij}\phi_j(k)} \leq 0. \tag{5.35}$$

Since $\boldsymbol{\rho}^k \neq \tilde{\boldsymbol{\rho}}^k$,

$$\sum_{i\in\mathcal{I}} \lambda_i l_i \sum_{j\in\mathcal{J}_0} \frac{\eta_{ij}^k - \tilde{\eta}_{ij}^k}{r_{ij}\phi_j(k)} < 0. \tag{5.36}$$

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

Theorem 3. *The estimated traffic load vector $\tilde{\boldsymbol{\rho}}$ converges to the optimal load vectors $\tilde{\boldsymbol{\rho}}^* \in F$.*

Proof. As shown in lem 4, $\boldsymbol{\rho}^k$ provides a descent direction for $L(\tilde{\boldsymbol{\rho}})$ at $\tilde{\boldsymbol{\rho}}^k$ when $\boldsymbol{\rho}^k \neq \tilde{\boldsymbol{\rho}}^k$, and hence $L(\tilde{\boldsymbol{\rho}}^{k+1}) < L(\tilde{\boldsymbol{\rho}}^k)$ in each iteration. Since $L(\tilde{\boldsymbol{\rho}}) > 0$, $\tilde{\boldsymbol{\rho}}$ will eventually converge to $\tilde{\boldsymbol{\rho}}^*$ when $L(\tilde{\boldsymbol{\rho}})$ is minimized. Considering

$$\tilde{\rho}^{k+1} = \beta\tilde{\rho}^k + (1-\beta)\rho^k = \tilde{\rho}^k + (1-\beta)(\rho^k - \tilde{\rho}^k), \tag{5.37}$$

$\boldsymbol{\rho}$ and $\tilde{\boldsymbol{\rho}}$ will converge to $\tilde{\boldsymbol{\rho}}^*$. As $\boldsymbol{\rho}$ is obtained by user association (i.e., $\eta_{ij} = \{0, 1\}$), $\tilde{\boldsymbol{\rho}}^*$ is in the feasible set F. □

Theorem 4. *Suppose the traffic loads of BSs converge to $\boldsymbol{\rho}^*$, the user association corresponding to $\boldsymbol{\rho}^*$ minimizes $L(\boldsymbol{\rho})$.*

Proof. Denote $\boldsymbol{\eta}^*$ as the user association for the traffic load vector $\boldsymbol{\rho}^*$. Meanwhile, let $\boldsymbol{\eta}^{'}$ be the user association corresponding to any other possible traffic load vector $\boldsymbol{\rho}^{'}$. Therefore, we just need to prove that $\boldsymbol{\rho}^{'}$ cannot get a smaller $L(\boldsymbol{\rho})$ than $\boldsymbol{\rho}^*$, i.e., $\left\langle \nabla L(\boldsymbol{\rho}^*), \boldsymbol{\rho}^{'} - \boldsymbol{\rho}^* \right\rangle \geq 0$.

$$\begin{aligned} &\left\langle \nabla L(\boldsymbol{\rho}^*), \boldsymbol{\rho}^{'} - \boldsymbol{\rho}^* \right\rangle \\ &= \sum_{i\in\mathcal{I}} \lambda_i l_i \sum_{j\in\mathcal{J}_0} (\eta_{ij}^{'} - \eta_{ij}^*) \frac{1}{r_{ij}\phi_j(k)} dx. \end{aligned} \tag{5.38}$$

Since $p_i^k = \arg\max_{j\in\mathcal{J}} r_{ij}\phi_j(k)$,

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

Then, we have

$$\sum_{j \in \mathcal{J}_0} \eta_{ij}^{'} \frac{1}{r_j(x)\phi_j(k)} \geq \sum_{j \in \mathcal{J}_0} \eta_{ij}^{*} \frac{1}{r_{ij}\phi_j(k)}. \tag{5.39}$$

Hence, $\left\langle \nabla L(\boldsymbol{\rho}), \boldsymbol{\rho}^{'} - \boldsymbol{\rho}^{*} \right\rangle \geq 0$. □

As the user association algorithm is a gradient algorithm (i.e., a classic algorithm to solve convex problems), the number of iterations used to achieve convergence of $L(\boldsymbol{\rho})$, which reflects the computational complexity, has been provided by other existing works [38]. Following the same procedure outlined in Ref. [38], the required number of iterations is at most $\left\lceil \frac{\log((L(\boldsymbol{\rho}(1)) - L(\boldsymbol{\rho}^{*}))/\varepsilon)}{\log 1/z} \right\rceil$, where $\boldsymbol{\rho}(1)$ is the initial traffic load vector, $L(\boldsymbol{\rho}^{*})$ is the optimal solution, and $\varepsilon > 0$ is a small real number. Since $L(\boldsymbol{\rho})$ is a convex function, there exist q and Q such that $qI \leqslant \nabla^2 L(\boldsymbol{\rho}) \leqslant QI$. Thus, $z = 1 - \min\{2q\zeta, 2q\zeta\xi/Q\} < 1$.

CHAPTER 6

SIMULATION RESULTS

6.1 Performance of the LAB Algorithm

In this section, we set up simulations of the LAB scheme to evaluate its performance. We select two other algorithms for comparison: α-distributed algorithm [18] 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.

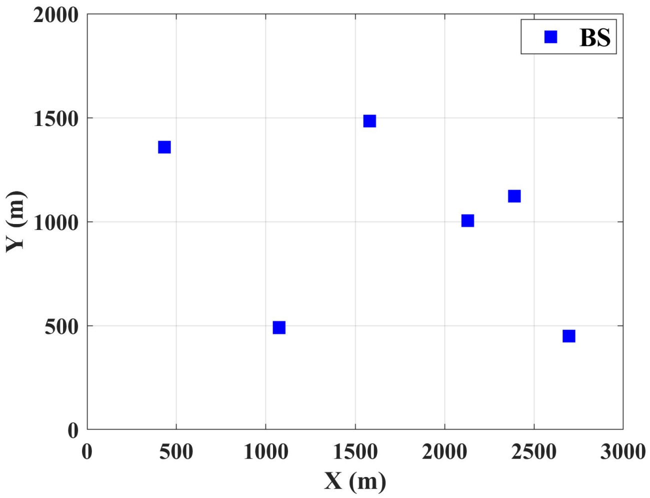

Figure 6.1 Network topology.

In the simulation, six BSs are randomly deployed in a 3000×2000 m^2 area as shown in Figure 6.1. 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 Equation (3.4) and (3.11), 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 [43] 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.

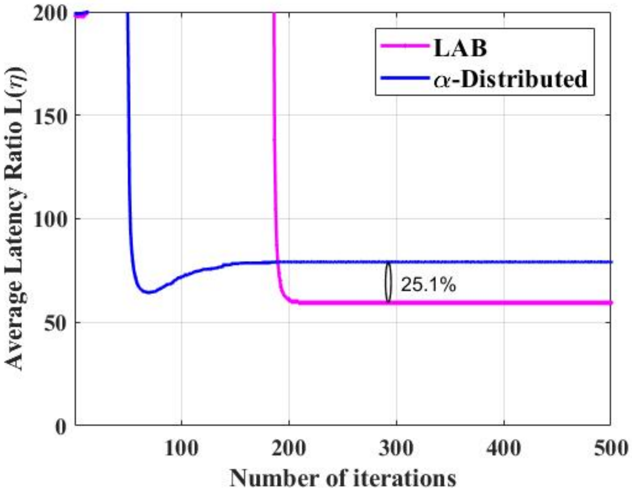

Figure 6.2 Average latency ratio $L(\boldsymbol{\eta})$ with respect to the number of iterations ($\lambda = 0.5$, $C_i = 7.1 * 10^6$).

As shown in Figure 6.2, the average latency ratios of both LAB and α-distributed algorithms do converge. Meanwhile, Figure 6.3 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

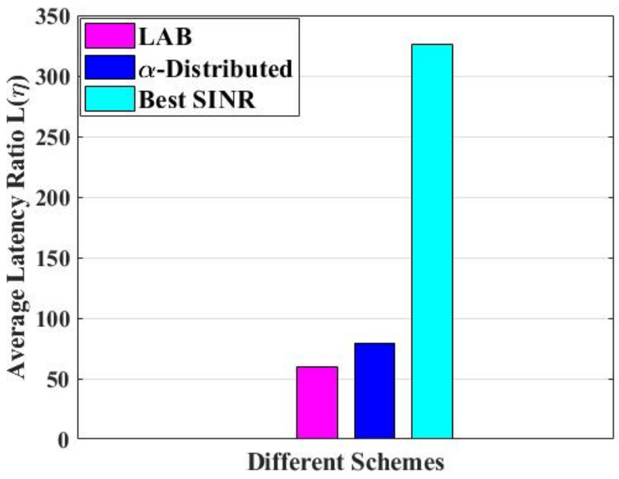

Figure 6.3 Average latency ratio $L(\boldsymbol{\eta})$ for different algorithms ($\lambda = 0.5$, $C_i = 7.1 * 10^6$).

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, 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 Figure 6.4, 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,

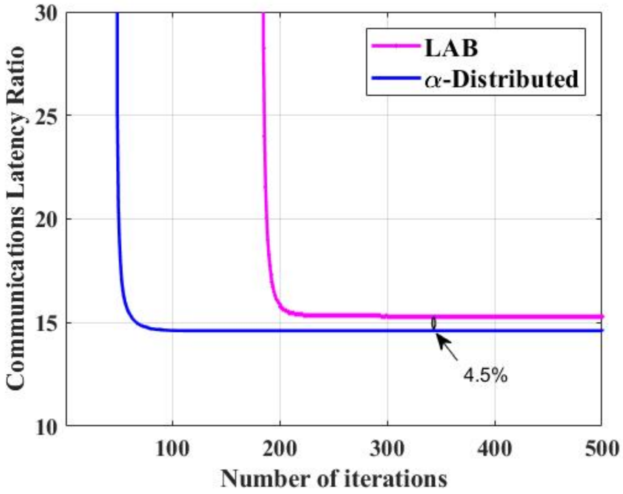

Figure 6.4 Average communications latency ratio with respect to the number of iterations ($\lambda = 0.5$, $C_i = 7.1 * 10^6$).

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.

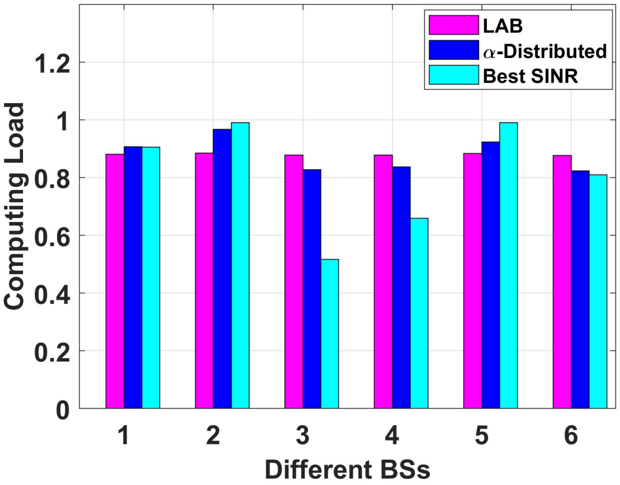

Figure 6.5 Computing loads of different fog nodes.

To further study the load balancing process in the fog network, we also compare the computing loads among fog nodes and the traffic loads among BSs for different

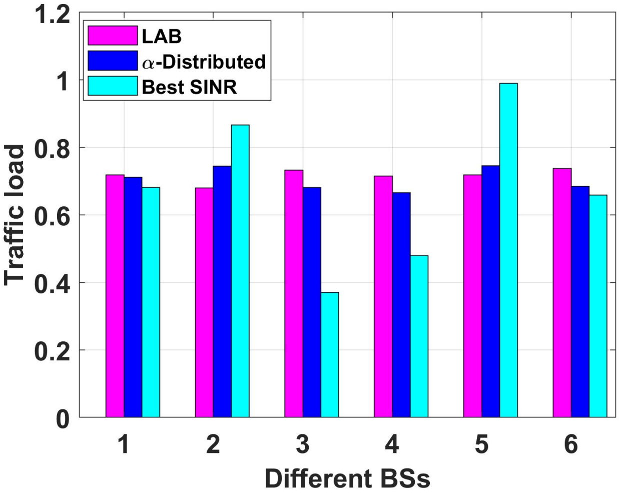

Figure 6.6 Traffic loads of different BSs.

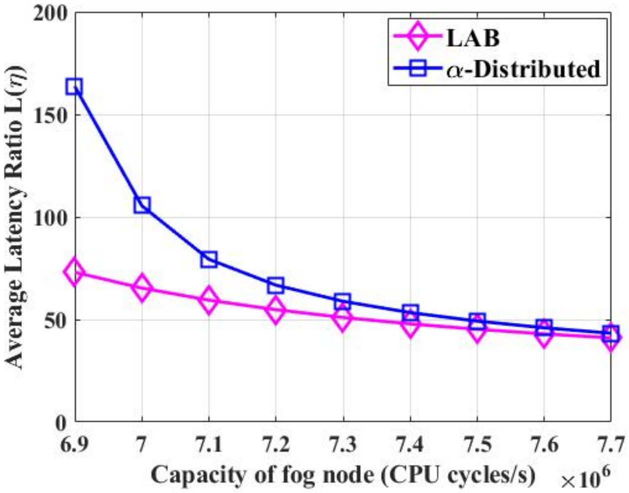

Figure 6.7 Average latency ratio with respect to the capacity of each fog node ($\lambda = 0.5$).

schemes. Figure 6.5 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, Figure 6.6 shows the

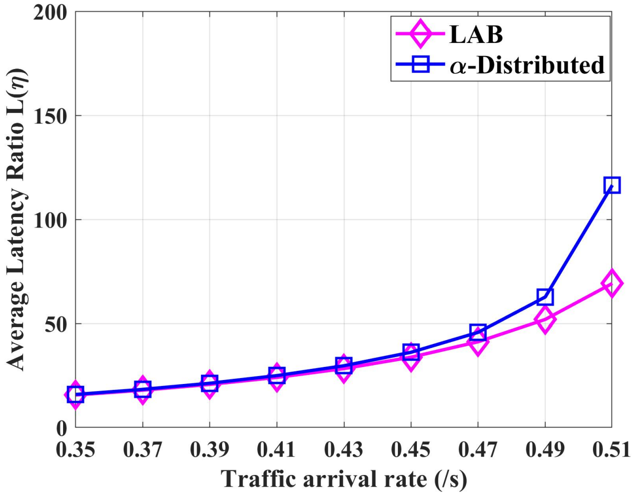

Figure 6.8 Average latency ratio with respect to flow arrival rate $\lambda(x)$ ($C_i = 7.1*10^6$).

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 Equation (3.10), 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 Figure 6.7, 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 Figure 6.8, 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.

6.2 Performance of the AREA Algorithm

In this section, we set up simulations of the AREA algorithm to evaluate its performance. We select two other workload allocation strategies for comparison: the density-based clustering (DBC) strategy [44] and the latency-based strategy [45]. The basic idea of DBC is to offload UEs' workloads to suitable cloudlets until the workloads of the cloudlets exceed the average workload among cloudlets. On the other hand, the latency-based strategy is to minimize the network delay between Apps and cloudlets by assigning Apps to suitable cloudlets. In the above two strategies, the computing resource of each cloudlet is allocated to different types of VMs according to the percentage of different types of workloads in the cloudlet.

The simulation environment consists of 25 BSs within an area of 25 km^2, where the coverage of each BS is 1 km^2 and each BS is attached with a cloudlet. Meanwhile, 1000 UEs are uniformly distributed among the BSs and assumed to be associated with their closest BSs. There are 10 types of IoT applications in the cloudlet network, and we randomly choose three types of Apps for each UE (i.e., the total number of Apps in the network is 3000). The length of each time slot is set as 5 mins. As each App's task arrival rate follows a Poisson distribution, we randomly choose the average task arrival rate of each App between 0 and λ_{max}. As the computing sizes of application k's requests follow an exponential distribution with the average value of l_k, the average size of different types of requests is chosen according to the Normal distribution with an average of 10^6 CPU cycles and a variance of $2 * 10^5$ cycles, i.e., $N(10^6, 2 * 10^5)$. Moreover, we assume the network delay between a BS and a cloudlet is a linear function of the distance between them [21, 46], i.e., $\tau_{ri} = \alpha \times d + \beta$, where d is the distance between BS r and cloudlet i, and α and β are set as 5 and 22.3, respectively. In addition, the maximum allowed computing delay for different types of applications is chosen according to $N(60, 20)$ (ms).

Figure 6.9 shows the average response time per App, in which AREA achieves lower response time as compared to the other two strategies. Specifically, the latency-based strategy always assigns Apps' requests to their closest cloudlets without considering the workload in each cloudlet; DBC assigns Apps to the closest cloudlets until the workload of each cloudlet exceeds the average workload among cloudlets, without considering the diversity of applications in each cloudlet. Thus, both DBC

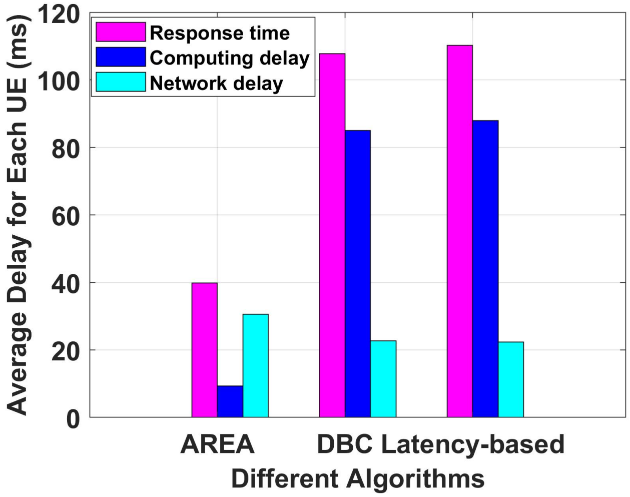

Figure 6.9 Average performance of an App for different algorithms ($\lambda_{max} = 1.5$, $C_i = 2 * 10^8$).

and the latency-based strategy lead to a lower network delay and a higher computing delay than AREA. AREA considers both the network delay of each App and the different types of workloads for each cloudlet in the workload allocation. To reduce the computing delay of all Apps, it tends to assign Apps with small computing sizes to the lightly loaded cloudlets. Furthermore, it also optimally allocates computing resources for different types of VMs based on their corresponding workloads, and thus significantly reduces the average response time per App. Meanwhile, as shown in Figure 6.10, the average response time for different types of applications in AREA is significantly smaller than those of DBC and the latency-based strategy.

We further analyze how the workloads of Apps affect the performance of the three algorithms. Note that the value of λ_{max} reflects the workloads of Apps, i.e., increasing λ_{max} increases workloads of Apps. As shown in Figure 6.11, with the increase of λ_{max}, the average response time of the three algorithms increases gradually. However, the average response time of AREA is much lower and increases more slowly as compared to those of the other two algorithms. When the workloads of Apps are heavy, AREA can always offload the App with the highest response time

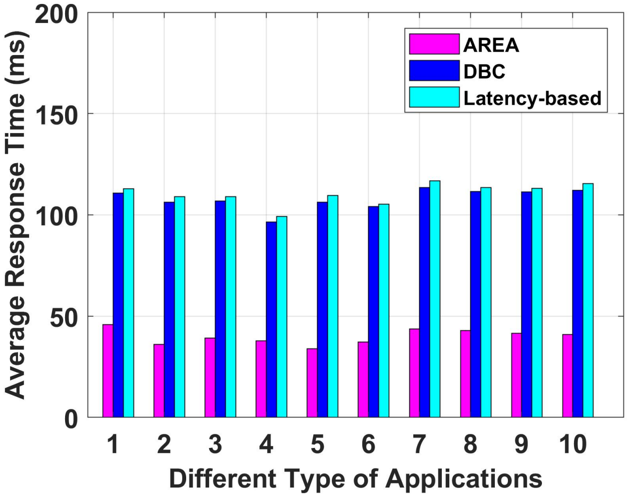

Figure 6.10 Average response time for different types of IoT applications ($\lambda_{max} = 1.5$, $C_i = 2 * 10^8$).

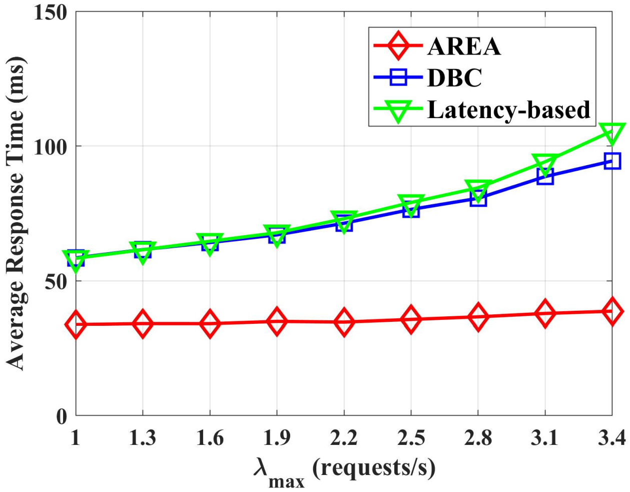

Figure 6.11 Average response time with respect to λ_{max} ($C_i = 3.8 * 10^8$).

to an alternative cloudlet, and thus iteratively minimize the maximum response time among Apps. Meanwhile, AREA also optimally allocates the computing resources of each cloudlet to different types of applications based on their workloads and their corresponding computing sizes, and thus further reduces the computing delay.

Moreover, we investigate the impact of cloudlets' capacities on the average response time. Figure 6.12 shows that the response time of the three algorithms when the capacities of cloudlets increase. It can be seen that AREA achieves much lower

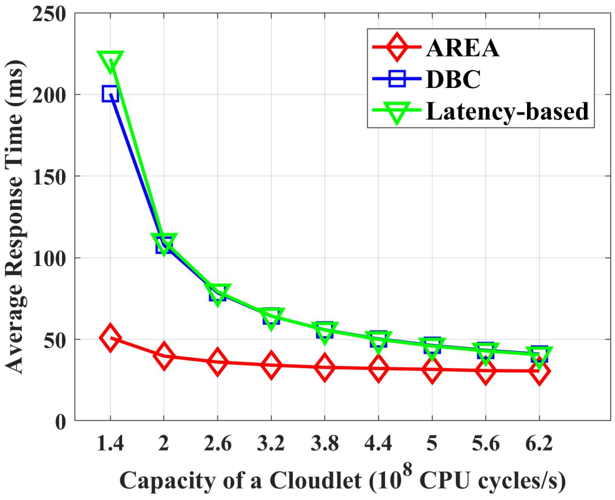

Figure 6.12 Average response time with respect to the capacity of each cloudlet ($\lambda_{max} = 1.5$).

average response time when the capacities of cloudlets change. When the capacities of cloudlets are small, since DBC and the latency-based algorithm do not balance the workloads among cloudlets based on different types of applications (i.e., considering all task requests are homogeneous), AREA leads to a remarkably lower computing delay, and thus incurs lower response time. However, when the capacities of cloudlets are very high, the computing delay is no longer a dominating factor for the average response time, and thus the average response time of DBS and the latency-based algorithm get close to that of AREA.

We also analyze the impact of the number of UEs on the average response time of Apps. As shown in Figure 6.13, the average response time of AREA increases much slower than those of the other two algorithms. Since AREA considers the difference between applications, it tends to assign Apps with smaller task sizes to lightly loaded cloudlets and allocates more computing resources to them, thus minimizing the average response time of all UEs' Apps. Therefore, as the number of UEs increases where the computing delay is the dominating factor in the average response time, AREA is able to achieve a lower computing delay than the other two algorithms, thus improving the performance of the average response time.

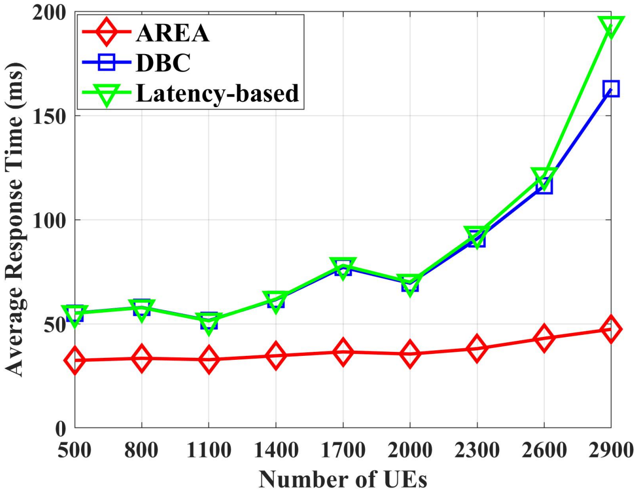

Figure 6.13 Average response time with respect to different number of UEs ($\lambda_{max} = 1.5$, $C_i = 3.8 * 10^8$).

6.3 Performance of the TALL Algorithm

In this section, the performance of the TALL algorithm has been evaluated by simulations. In the simulation, the coverage area of a MBS is 1000×1000 m^2 where the MBS is at the center and three DBSs can be deployed in the MBS's coverage area to facilitate communications. After randomly selecting two locations ($x1$ and $x2$) within this area, we place 180 UEs around the two locations according to the normal distributions $N(x1, 150\ m)$ and $N(x2, 150\ m)$; thus, some hotspot areas are created. The task arrivals of each UE follow a Poison process in which the average task arrival rate is 0.9 requests/s. The traffic sizes of IoT tasks follow the general distribution with the average traffic size equaling to 200 kb. The heights of the MBS and DBSs are set as 10 m and 50 m, respectively. The total bandwidth is 20 MHz in which each BS (either the MBS or a DBS) exclusively utilizes 5 MHz. The transmission powers of a UE and a DBS are set as 200 mW and 2 W, respectively. The maximum allowed traffic load ρ_{max} is 0.99. In addition, ξ^{los} and τ^{los} of the LoS channel are set as 103.4 dB and 24.2 dB/km; ξ^{nlos} and τ^{nlos} of the NLoS channel are set as 131.4 dB and 42.8

dB/km. The parameters of the LoS probability model (a and b) are set as 11.95 and 0.136.

We also consider three other schemes for comparison, i.e., Best SINR scheme [47], Max-Coverage scheme [48], and single MBS (S-MBS) scheme [21]. In Best SINR, DBSs are optimally deployed in hotspot areas with high user densities while UEs are associated to the BS based on the best channel condition of the access link. The basic idea of Max-Coverage is to maximize the number of IoT users covered by DBSs in the DBS deployment, where the coverage ranges of DBSs are given (i.e., 100m). For S-MBS, only the MBS is placed with the whole bandwidth (20 MHz). To express the gap between our scheme and optimal solution, we divide the coverage of the MBS into 36 locations to reduce the running time of the brute-force search. Based on this small-scale network, the optimal solution is obtained through the brute-force search when the average arrival rate is set as 0.95 requests/s. As shown in Figure 6.14, the average latency ratio per BS of TALL is only 0.4% higher than the optimal solution, which demonstrates the effectiveness and efficiency of TALL.

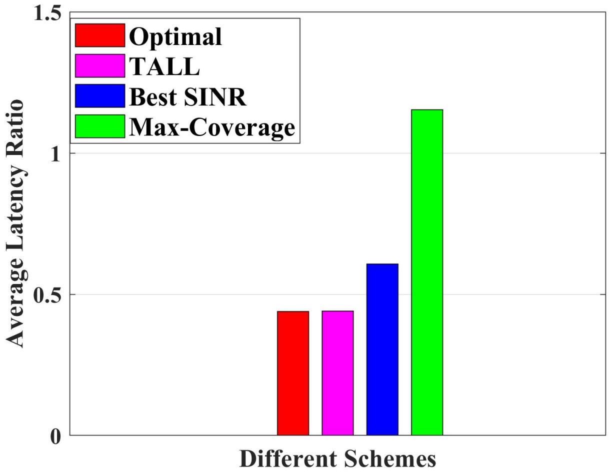

Figure 6.14 Average latency ratio per BS of different schemes.

To better demonstrate the performance of TALL as compared to other schemes, we further divide the MBS's coverage area into 2500 locations, each representing a small area of 20 m × 20 m for the following simulations.

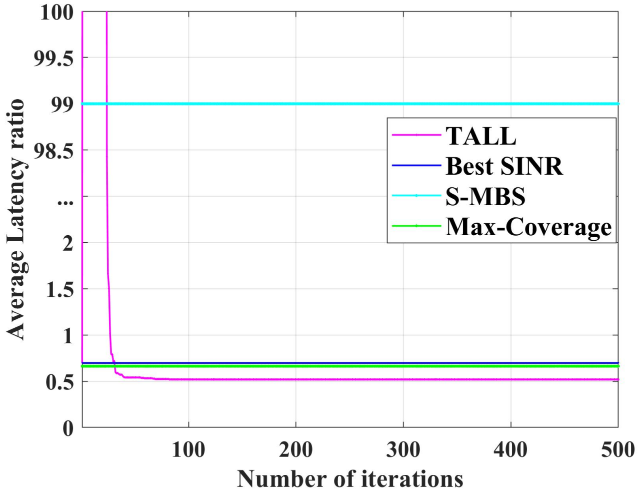

Figure 6.15 Average latency ratio per BS v.s. number of iterations.

In Figure 6.15, the average latency ratio per BS of TALL does converge. Meanwhile, we can see that TALL significantly reduces the average latency ratio per BS as compared to the other three schemes. In S-MBS, only the MBS is considered to deliver traffic, and thus it will always be heavily loaded. For Best SINR, the user association depends on the best channel conditions and thus incurs unbalanced traffic loads among BSs, i.e., while some DBSs are lightly loaded, other DBSs at hotspot areas may be congested owing to the heavy traffic demands around them. As the congested DBSs incur remarkably high latency ratios, the average latency ratio of all DBSs is significantly deteriorated owing to these congested BSs. Meanwhile, in Max-Coverage, UEs are associated to a DBS when they are in the DBS's coverage area (i.e., determined by the downlink), where the channel condition and traffic load balancing among BSs are not considered. In contrast, TALL can optimally allocate traffic load among BSs to reduce their latency ratios.

To better study the traffic load balancing among BSs, we also compare the traffic loads among BSs for different schemes. As shown in Figure 6.16, TALL yields significantly smaller differences of traffic loads among BSs than those by the Best SINR and Max-Coverage. This is attributed to the fact that TALL tries to associate UEs to the suitable BSs in each iteration in order to balance the traffic loads among BSs, and thus improves the latency ratios of BSs. In contrast, Best SINR enables UEs to associate with BSs with best channel conditions without considering the traffic loads of these BSs, and thus incurs the unbalanced traffic loads among BSs owing to the dynamic distribution of UEs. In Max-Coverage, the user association is determined by the locations and downlink coverage areas of DBSs. As the user distribution is dynamic and the coverage ranges of DBSs are given, the workloads among the MBS and DBSs are uneven.

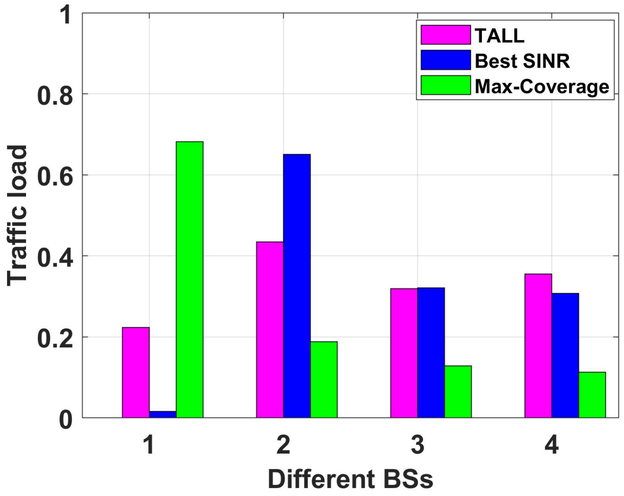

Figure 6.16 Comparison of Traffic load.

The average traffic arrival rate of UEs has an impact on the average latency ratio per BS. In Figure 6.17, as each UE's traffic arrival rate increases, the average latency ratio per BS of the four algorithms increases, where the value of TALL is significantly lower than other schemes. Specifically, when the average traffic arrival rates of UEs are relatively small, the average latency ratio of S-MBS is higher than those of other

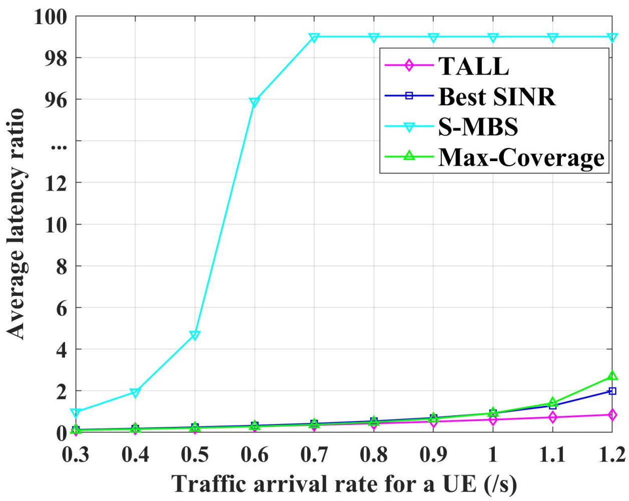

Figure 6.17 Average latency ratio per BS v.s. traffic arrival rate for each UE.

schemes that have similar performance. Meanwhile, when the average traffic arrival rate increases, TALL considers the traffic loads of BSs in the user association, and thus still keeps low average latency even if the traffic load of the network becomes heavy. However, Best SINR only focuses on the channel conditions of UEs and some overloaded DBSs may be congested. Although Max-Coverage focuses on maximizing the number of UEs covered by DBSs, the workload difference between the MBS and DBSs still exacerbates the average latency ratio. Meanwhile, for S-MBS, the traffic load of the MBS increases due to the heavy traffic demands and bad channel, and thus its average latency ratio degrades drastically.

To further study the performance of TALL, we also test the impact of the number of UEs on the average latency ratio per BS. In Figure 6.18, when the number of UEs increases, the average latency ratio per BS of TALL is significantly lower than those of the other three schemes. In TALL, although the number of UEs increases, it can still suitably assign UEs among BSs to avoid the severe congestion. Regarding Max-Coverage and Best SINR, increasing the number of UEs will further degrade the balance of traffic loads among BSs. In S-MBS, the MBS will become congested quickly owing to the increased traffic.

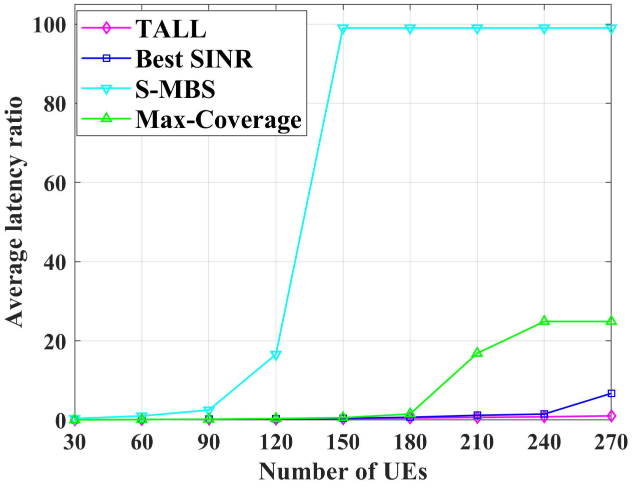

Figure 6.18 Average latency ratio per BS v.s. number of UEs.

CHAPTER 7

OTHER CONTRIBUTIONS

In Chapter 3, to jointly minimize the communications latency and computing latency, IoT devices are associated to suitable BSs to balance the traffic load and computing load simultaneously. In Chapter 4, we have assigned different applications' workloads among different cloudlets and allocated the computing resources for different applications in each cloudlet, thus improving the response time of different application tasks. In Chapter 5, to reduce the wireless latency for IoT tasks especially in hotspot areas, we have placed drone base stations to facilitate the data transfer from IoT users to fog nodes. In this chapter, some other contributions during my doctoral study are introduced. I briefly discuss how to minimize the on-grid energy consumption by migrating virtual machines among cloudlets that are powered by both the green energy and on-grid energy. Meanwhile, I will also introduce how to enhance the performance of the heterogeneous cellular networks by making a tradeoff between the throughput and on-grid energy consumption.

7.1 Energy Driven Avatar Migration in Green Cloudlet Networks

Mobile applications are becoming computation-intensive while the computational capacity of user equipments (*UEs*) remains limited owing to their sizes and battery. Mobile Cloud Computing (*MCC*) enables UEs to offload some tasks to high performance Virtual Machines (*VMs*) in remote clouds, thus reducing the task execution time and energy consumption of UEs. Existing researches mostly consider the remote cloud as the offloading destination, due to its abundant resources. However, long communications delay incurred by transferring data between UEs and remote VMs has a detrimental impact on user experience of applications, such as augmented reality and online gaming, where a short response time is required.

The concept of cloudlet [49] has been proposed to reduce the communications delay between UEs and their VMs. Cloudlets, as tiny versions of data centers, are generally placed at access points in a network that are close to UEs. The physical proximity between UEs and cloudlets leads to a shorter communications delay [44] [50].

To maintain the normal operation of these distributed cloudlets, a large amount of on-grid energy (i.e., brown energy) is consumed, generating tremendous CO_2. As green energy technologies advance, green energy can be readily employed to reduce the on-grid energy cost. Energy generated from solar panels can be used to power distributed cloudlets, with on-grid energy as a backup.

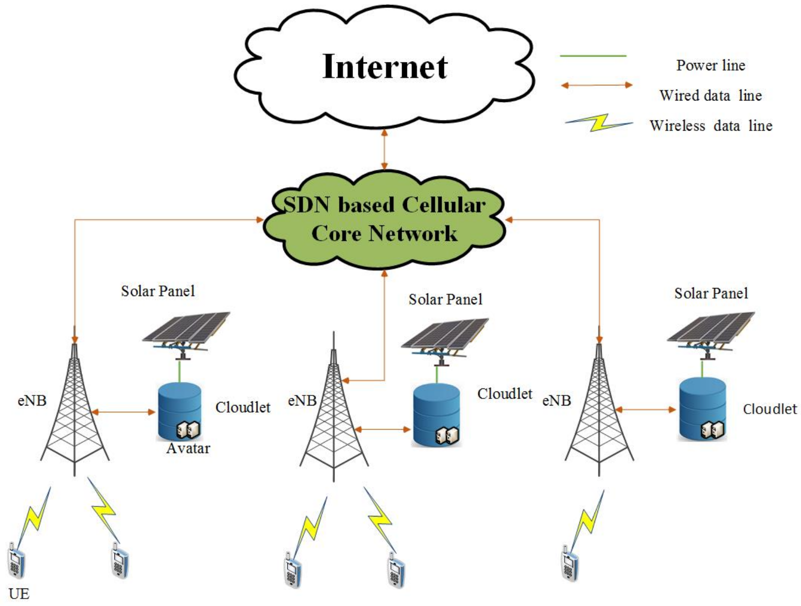

Figure 7.1 GCN architecture

A Green Cloudlet Network (GCN) architecture is illustrated in Figure 7.1 in which each cloudlet is collocated at an eNB, and connects to the eNB via a high speed fiber link. Distributed cloudlets are able to transfer data to each other via the cellular core network and internet. Software Defined Network (SDN) based cellular network is employed to provide efficient and flexible communications paths between eNBs. Meanwhile, LTE providers offer the seamless wireless communications between a UE and its eNB, thereby each UE can connect to a nearby cloudlet to minimize the communications delay. In GCN, each UE can be mapped to a specific Avatar [51], one VM in the cloudlet, to run its offloaded tasks. An Avatar is a software clone of a UE and always offers service to the UE wherever it moves. Moreover, in order to reduce on-grid energy consumption, cloudlets can be equipped with and powered by solar panels. Note that green energy shown in Figure 7.1 can only be utilized by cloudlets. Certainly, eNBs can be equipped with their own solar panel systems. However, since the LTE network provider and cloudlet provider play different roles in the network, they cannot share the same green energy source. In this work, we focus on the available green energy for cloudlet networks.

Since UEs move in the network all the time and the workload of each UE is dynamic, energy demands among different cloudlets are also dynamic. Therefore, some cloudlets may have excessive green energy while the energy demands from their hosting Avatars are light. In contrast, other cloudlets, which have more energy demands and less green energy, have to draw on-grid energy to maintain their Avatars' operation. So, these unbalanced energy demands among cloudlets intensify the on-grid energy consumption of GCNs. In order to reduce the on-grid energy consumption, we need to migrate Avatars from cloudlets being lack of green energy to cloudlets with excessive green energy, thus improving the green energy utilization. However, during an Avatar migration process, the cloudlet provider has to transfer the data of the Avatar from its original cloudlet to its destination, resulting in energy consumption of the cellular core network (i.e., the migration cost), which also contributes to the on-grid energy consumption of GCNs. In order to minimize the on-grid energy consumption of GCNs, we design the Energy driven AvataR migratioN (EARN) scheme in green cloudlet networks to balance the tradeoff relationship between the migration gain (i.e., green energy utilization of cloudlets) and the migration cost (i.e., on-grid energy consumption of the network owing to Avatar migrations) [11]. Moreover, EARN also guarantees the Service Level Agreement

(SLA) of Avatars i.e., the maximum propagation delay between a UE's eNB and its assigned cloudlet.

7.2 Throughput Aware and Energy Aware Traffic Load Balancing in Heterogeneous Networks with Hybrid Power Supplies

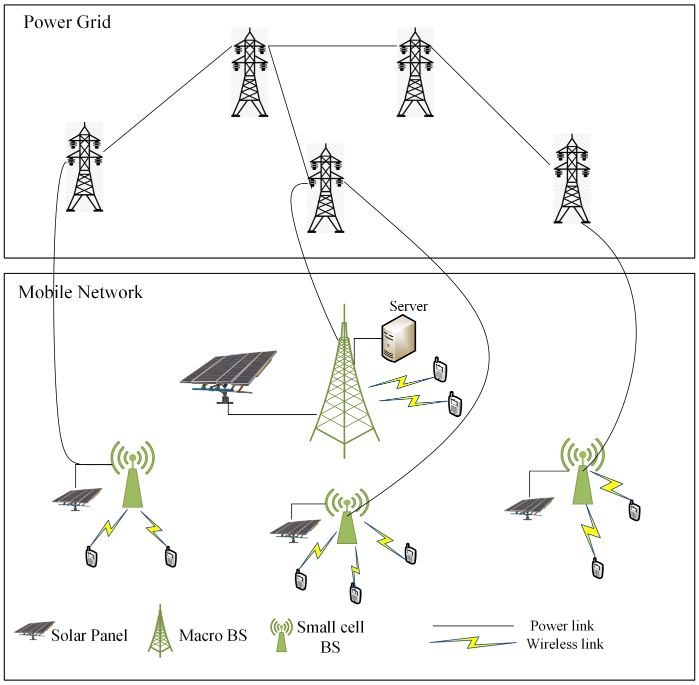

Figure 7.2 Heterogeneous mobile network architecture.

Owing to the direct impact of greenhouse gases on the environment and the climate change, curbing the energy consumption of mobile networks has attracted much attention. Driven by the proliferation of data-hungry devices and applications, mobile data traffic is expected to increase exponentially in the future [13, 52]. In this situation, the increasing traffic not only calls for expansion of network capacity, but also intensifies the energy consumption [53]. Therefore, greening mobile networks is important to mitigate the environmental problems and reduce the operating cost of mobile operators [54], [55]. With the development of green energy technologies, green energy such as solar energy, wind energy and sustainable biofuels is being utilized to power base stations (BSs). However, owing to the unstable generation of green energy, hybrid energy supplies, consisting of both green energy and on-grid power, are a more practical option to power BSs [56]. Thus, green energy can be utilized to reduce the on-grid power consumption and therefore decrease the CO_2 emission, with the on-grid power as a backup power source.

Heterogeneous cellular networks (HCNs), in which the macro cells are overlaid with small cells, are promising to increase the total capacity of cellular networks [57]. Considering the dynamic workload distribution, small cell base stations (SCBSs) are placed in areas with high user density to facilitate more users to connect to a much closer BS, thus improving the channel conditions of users. Meanwhile, as the coverage of each SCBS is very small, the transmission power required by each SCBS is significantly smaller than those of traditional BSs [31], [58]. Therefore, the low power of SCBSs can potentially improve the spectral efficiency and energy efficiency of heterogeneous cellular networks [59].

In a HCN with hybrid power supplies as shown in Figure 7.2, the effective data rate (EDR) of a user's flow is based on both the channel condition of the user towards its BS and the BS's workload status [20]. As the user distribution is dynamic, if a user tends to associate with BSs only based on the channel condition or received power, it may connect to a congested BS, which degrades its EDR. Consequently, some BSs may be congested by the heavy traffic loads while other BSs are lightly loaded. The unbalanced workload distribution among BSs has a negative impact on user Quality-of-Service (QoS) in terms of the EDR. On the other hand, the main operating cost of mobile providers arises from the on-grid energy consumption. Owing to the dynamic traffic workload distribution among BSs, the energy demands of BSs may not match their available green energy, thus incurring the increment of on-grid energy consumption. In other words, while some BSs still have excessive green energy, others have drained their green energy and started to consume on-grid energy. To reduce the operating cost, traffic load balancing can be employed to reduce the gap between the energy demands of BSs and their green energy. Moreover, as mobile providers need to consider the gain of the aggregated EDR (sum of EDRs of all users within the coverage area of a macro BS) and the operating cost in terms of on-grid energy consumption simultaneously, the optimal traffic load balancing strategy should take into consideration of the above two factors. However, in the load balancing process, saving on-grid power is always at the cost of sacrificing an amount of EDR, i.e., the EDR and on-grid energy consumption exhibit a trade-off relationship. How to balance the traffic loads among BSs to optimize the aggregated EDR of the network and on-grid energy consumption still remains to be a critical problem.

To solve the above problem, we design a Throughput aware and Energy Aware (TEA) traffic load balancing scheme for heterogeneous networks to satisfy mobile providers' requirements by balancing traffic loads [60]. The scheme not only optimizes the utilization of green energy in order to reduce the on-grid power consumption, but also optimizes the aggregated EDR of the network. Since the power consumption of a macro BS (MBS) is significantly larger than that of SCBS, associating users with SCBSs may reduce the on-grid power consumption. However, too many users associating with SCBSs may incur traffic congestion in SCBSs and thus degrades the EDRs of their users. The TEA algorithm makes a tradeoff between the aggregated EDR of the network and on-grid energy consumption by assigning users to suitable BSs. Below are the major contributions of this work.

We formulate the problem of making a tradeoff between the aggregated EDR and on-grid energy consumption by balancing traffic workloads among heterogeneous BSs. The mobile providers desire to improve the aggregated EDR while reducing on-grid energy consumption of the network. Since the user association aiming to increase the effective data rate may increase on-grid energy consumption, we need to balance these two factors in the scheme. Thus, we define an energy-throughput coefficient α to make a tradeoff between the aggregated EDR and on-grid energy cost, which can be predefined by each mobile provider based on its practical requirement.

The workload status of a BS has a critical impact on the EDRs of its associated users. To guarantee the user QoS, we assume that the workload of each BS should be smaller than the BS's maximum workload threshold allowed by mobile providers.

To solve the user association problem (i.e., load balancing) in each time slot, we design a heuristic algorithm which iteratively moves users to suitable BSs. Then, we analyze the computational complexity of the algorithm. We also analyze some critical issues of the proposed algorithm in order to facilitate its practical implementation.

CHAPTER 8

CONCLUSION

We have studied the workload allocation in edge computing to optimize the response time of IoT devices and IoT users. First, we have designed the LoAd Balancing (LAB) scheme for the fog network to minimize the average latency of IoT devices' data. Since the latency of IoT data consists of both the communications latency and computing latency, LAB takes into consideration of both the traffic laod allocation and computing load allocation by associating IoT devices to suitable BSs/fog nodes. 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. Second, we have designed an Application awaRE workload Allocation (AREA) scheme for edge computing based IoT that assigns different types of workloads in each IoT user to their corresponding cloudlets and optimally allocates the computing resources of each cloudlet to its application based virtual machines. Third, we have designed TALL scheme to place DBSs to the locations with higher densities, and then allocates the trafic loads among BSs to further minimize the latency ratios of DBSs. Simulation results have verified the performance of these above schemes.

BIBLIOGRAPHY

[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] Y. Wang, R. Yang, T. Wo, W. Jiang, and C. Hu, "Improving utilization through dynamic vm resource allocation in hybrid cloud environment," in *20th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2014)*, 2014, pp. 241–248.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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, 2017.

[9] 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.

[10] 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.

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

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

[13] ——, “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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

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

[20] Q. Fan and N. Ansari, “Throughput aware and green energy aware user association in heterogeneous networks,” in *2017 IEEE International Conference on Communications (ICC)*, Paris, France, May 2017.

[21] X. Sun and N. Ansari, “Latency aware drone base station placement in heterogeneous networks,” in *2017 IEEE Global Communications Conference (GLOBECOM’2017).*, Dec 2017.

[22] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, “Efficient 3-D placement of an aerial base station in next generation cellular networks,” in *2016 IEEE International Conference on Communications (ICC)*, May 2016.

[23] A. Fotouhi, M. Ding, and M. Hassan, “Dynamic base station repositioning to improve spectral efficiency of drone small cells,” in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2017.

[24] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, Dec 2014.

[25] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement optimization of UAV-mounted mobile base stations," *IEEE Communications Letters*, vol. 21, no. 3, pp. 604–607, March 2017.

[26] L. Wang, B. Hu, and S. Chen, "Energy efficient placement of a drone base station for minimum required transmit power," *IEEE Wireless Communications Letters*, doi:10.1109/LWC.2018.2808957, 2018, early access.

[27] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, May 2016.

[28] W. Shi, et al., "Multiple drone-cell deployment analyses and optimization in drone assisted radio access networks," *IEEE Access*, vol. 6, pp. 12 518–12 529, 2018.

[29] 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.

[30] 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.

[31] 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.

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

[33] N. L. Van Adrichem, C. Doerr, and F. A. Kuipers, "Opennetmon: Network monitoring in openflow software-defined networks," in *2014 IEEE Network Ops. and Mgmt. Sym. (NOMS)*, Krakow, Poland, May 2014, pp. 1–8.

[34] C. Yu, C. Lumezanu, A. Sharma, Q. Xu, G. Jiang, and H. V. Madhyastha, "Software-defined latency monitoring in data center networks," in *International Conference on Passive and Active Network Measurement*, vol. 8995, Mar. 2015, pp. 360–372.

[35] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2009.

[36] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing based IoT," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2146–2153, June 2018.

[37] 3GPP, "3GPP Technical Report 36.828 version 11.0.0, release 11: 3rd generation partnership project; further enhancements to LTE time division duplex (TDD) for downlink-uplink (DL-UL) interference management and traffic adaptation," in *3GPP Technical Report*, 2012.

[38] 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, 2016.

[39] Y. G. Kim and N. C. Beaulieu, "Exact BEP of decode-and-forward cooperative systems with multiple relays in rayleigh fading channels," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 823–828, Feb 2015.

[40] Z. Liu, H. Yuan, H. Li, X. Guan, and H. Yang, "Robust power control for amplify-and-forward relaying scheme," *IEEE Communications Letters*, vol. 19, no. 2, pp. 263–266, Feb 2015.

[41] K. R. Liu *et al.*, *Cooperative communications and networking.* Cambridge University Press, 2009.

[42] Z. Bai *et al.*, "Performance analysis of SNR-based incremental hybrid decode-amplify-forward cooperative relaying protocol," *IEEE Transactions on Communications*, vol. 63, no. 6, pp. 2094–2106, June 2015.

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

[44] 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*, DOI: 10.1109/TCC.2015.2449834 2015.

[45] L. Yang, J. Cao, G. Liang, and X. Han, "Cost aware service placement and load dispatching in mobile cloud systems," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1440–1452, 2016.

[46] R. Landa, et al., "The large-scale geography of Internet round trip times," in *IFIP Networking Conference*, Brooklyn, NY, May 2013, pp. 1–9.

[47] L. Zhang, Q. Fan, and N. Ansari, "3-D drone-base-station placement with in-band full-duplex communications," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1902–1905, Sept. 2018.

[48] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 434–437, Aug 2017.

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

[50] X. Sun and N. Ansari, “Green cloudlet network: A distributed green mobile cloud network in future networks,” *IEEE Network*, to appear.

[51] 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.

[52] “Cisco Visual Networking Index: Forecast and Methodology, 2014-2019 White Paper.” [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html

[53] Q. Fan, J. Fan, J. Li, and X. Wang, “A multi-hop energy-efficient sleeping mac protocol based on tdma scheduling for wireless mesh sensor networks.” *Journal of Networks*, vol. 7, no. 9, pp. 1355–1361, Sep. 2012.

[54] N. Ansari and T. Han, *Green Mobile Networks: A Networking Perspective.* Wiley-IEEE Press, ISBN: 978-1-119-12510-5, 2017.

[55] J. Xu, L. Duan, and R. Zhang, “Cost-aware green cellular networks with energy and communication cooperation,” *IEEE Communications Magazine*, vol. 53, no. 5, pp. 257–263, 2015.

[56] N. B. Rached, H. Ghazzai, A. Kadri, and M. S. Alouini, “Energy management optimization for cellular networks under renewable energy generation uncertainty,” *IEEE Transactions on Green Communications and Networking*, vol. 1, no. 2, pp. 158–166, June 2017.

[57] B. Yang, G. Mao, X. Ge, M. Ding, and X. Yang, “On the energy-efficient deployment for ultra-dense heterogeneous networks with nlos and los transmissions,” *IEEE Transactions on Green Communications and Networking*, 2018, early access.

[58] H. S. Dhillon, R. K. Ganti, and J. G. Andrews, “Load-aware modeling and analysis of heterogeneous cellular networks,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1666–1677, April 2013.

[59] C. Liu, B. Natarajan, and H. Xia, “Small cell base station sleep strategies for energy efficiency,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1652–1661, March 2016.

[60] Q. Fan and N. Ansari, “Towards throughput aware and energy aware traffic load balancing in heterogeneous networks with hybrid power supplies,” *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 4, pp. 890–898, Dec 2018.