Spring 5-31-2018

From geographically dispersed data centers towards hierarchical edge computing

Abbas Kiani
New Jersey Institute of Technology

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

FROM GEOGRAPHICALLY DISPERSED DATA CENTERS TOWARDS HIERARCHICAL EDGE COMPUTING

by
Abbas Kiani

Internet scale data centers are generally dispersed in different geographical regions. While the main goal of deploying the geographically dispersed data centers is to provide redundancy, scalability and high availability, the geographic dispersity provides another opportunity for efficient employment of global resources, e.g., utilizing price-diversity in electricity markets or utilizing locational diversity in renewable power generation. In other words, an efficient approach for geographical load balancing (GLB) across geo-dispersed data centers not only can maximize the utilization of green energy but also can minimize the cost of electricity. However, due to the different costs and disparate environmental impacts of the renewable energy and brown energy, such a GLB approach should tap on the merits of the separation of green energy utilization maximization and brown energy cost minimization problems. To this end, the notion of green workload and green service rate, versus brown workload and brown service rate, respectively, to facilitate the separation of green energy utilization maximization and brown energy cost minimization problems is proposed. In particular, a new optimization framework to maximize the profit of running geographically dispersed data centers based on the accuracy of the G/D/1 queueing model, and taking into consideration of multiple classes of service with individual service level agreement deadline for each type of service is developed. A new information flow graph based model for geo-dispersed data centers is also developed, and based on the developed model, the achievable tradeoff between total and brown power consumption is characterized.

Recently, the paradigm of edge computing has been introduced to push the computing resources away from the data centers to the edge of the network, thereby reducing the communication bandwidth requirement between the sources of data and the data centers. However, it is still desirable to investigate how and where at the edge of the network the computation resources should be provisioned. To this end, a hierarchical Mobile Edge Computing (MEC) architecture in accordance with the principles of LTE Advanced backhaul network is proposed and an auction-based profit maximization approach which effectively facilitates the resource allocation to the subscribers of the MEC network is designed. A hierarchical capacity provisioning framework for MEC that optimally budgets computing capacities at different hierarchical edge computing levels is also designed. The proposed scheme can efficiently handle the peak loads at the access point locations while coping with the resource poverty at the edge. Moreover, the code partitioning problem is extended to a scheduling problem over time and the hierarchical mobile edge network, and accordingly, a new technique that leads to the optimal code partitioning in a reasonable time even for large-sized call trees is proposed. Finally, a novel NOMA augmented edge computing model that captures the gains of uplink NOMA in MEC users' energy consumption is proposed.

FROM GEOGRAPHICALLY DISPERSED DATA CENTERS TOWARDS HIERARCHICAL EDGE COMPUTING

by
Abbas Kiani

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Helen and John C. Hartmann Department of
Electrical and Computer Engineering

May 2018

APPROVAL PAGE

FROM GEOGRAPHICALLY DISPERSED DATA CENTERS TOWARDS HIERARCHICAL EDGE COMPUTING

Abbas Kiani

Dr. Nirwan Ansari, Dissertation Advisor — Date
Distinguished Professor, Helen and John C. Hartmann
Department of Electrical and Computer Engineering, NJIT

Dr. Ashutosh Dutta, Committee Member — Date
Principal Member of Technical Staff, AT&T

Dr. Edwin Hou, Committee Member — Date
Professor, Helen and John C. Hartmann
Department of Electrical and Computer Engineering, NJIT

Dr. Abdallah Khreishah, Committee Member — Date
Associate Professor, Helen and John C. Hartmann
Department of Electrical and Computer Engineering, NJIT

Dr. Mengchu Zhou, Committee Member — Date
Distinguished Professor, Helen and John C. Hartmann
Department of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Abbas Kiani

Degree: Doctor of Philosophy

Date: May 2018

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2018
- Master of Science in Electrical Engineering, Communications,
  Shahed University, Tehran, Iran, 2011
- Bachelor of Science in Electrical Engineering,
  Imam Khomeini International University, Qazvin, Iran, 2008

Major: Electrical Engineering

Presentations and Publications:

Journal articles:

A. Kiani, N. Ansari and Abdallah Khreishah, "Hierarchical Capacity Provisioning for Fog Computing," *IEEE/ACM Transactions on Networking*, in review.

A. Kiani and N. Ansari, "Edge Computing Aware NOMA for 5G Networks," *IEEE Internet of Things Journal*, Vol. 5, No. 2, April 2018.

A. Kiani and N. Ansari, "Towards Hierarchical Mobile Edge Computing: An Auction-Based Profit Maximization Approach," *IEEE Internet of Things Journal*, Vol. 4, No. 6, December 2017.

A. Kiani and N. Ansari, "Optimal Code Partitioning Over Time and Hierarchical Cloudlets," *IEEE Communications Letters*, Vol. 22, No. 1, January 2018.

A. Kiani and N. Ansari, "On The Fundamental Energy Trade-offs of Geographical Load Balancing," *IEEE Communications Magazine*, Vol. 55, No. 5, May 2017.

A. Kiani and N. Ansari, "A Fundamental Tradeoff Between Total and Brown Power Consumption in Geographically Dispersed Data Centers," *IEEE Communications Letters*, Vol. 20, No. 10, October 2016.

A. Kiani and N. Ansari, "Profit Maximization for Geographical Dispersed Green Data Centers," *IEEE Transactions on Smart Grids*, Vol. 9, No. 2, March 2018.

A. Kiani and N. Ansari, "Towards Low-Cost Workload Distribution for Integrated Green Data Centers," *IEEE Communications Letters*, Vol. 19, No. 1, January 2014.

A. Kiani and S. Akhlaghi, "A Non-MDS Erasure Code Scheme For Storage Applications," *Journal of Communications Engineering (JCE)*, Vol. 2, No. 3, June 2013.

A. Kiani and S. Akhlaghi, "Selective Regenerating Codes," *IEEE Communications Letters*, Vol. 15, No. 8, August 2011.

S. Akhlaghi, **A. Kiani** and M. Ghanavati, "Cost-Bandwidth Tradeoff In Distributed Storage Systems," *Computer Communications (Elsevier)*, Vol. 33, No. 17, November 2010.

Conference papers

A. Kiani, S. Akhlaghi and M. Ghanavati "Distributed Storage Systems And Connection to Network Coding,"(Invited Paper) *IEEE 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)*, Rome, Italy, November 2010.

S. Akhlaghi, **A. Kiani** and M. Ghanavati, "A Fundamental Trade-off Between the Download Cost and Repair Bandwidth in Distributed Storage Systems," *IEEE International Symposium on Network Coding (NetCod)*, Toronto, Canada, June 2010.

If, our fellow student, thou remain, wash white the leaves;
For, in the book, love's art is not.

Hãfez Shirãzi

Dedicated to my inspiring parents, brother and sisters.

For their endless love, support and encouragement

ACKNOWLEDGMENT

My deepest gratitude is to my advisor, Dr. Nirwan Ansari. I have been amazingly fortunate to have him give me the freedom and encouragement to explore research ideas while providing excellent guidance. His persistent support and patience helped me overcome many difficult situations throughout my research. Without his continuous help, this dissertation would not have been possible.

To my committee members, Dr. Ashutosh Dutta, Dr. Edwin Hou, Dr. Abdallah Khreishah, and Dr. Mengchu Zou, I thank them for their time and advisement.

I want to thank my friends Tao Han, Yan Zhang, Mina Taheri, Thomas Lo, Xueqing Huang, Xiang Sun, Xilong Liu, Qiang Fan, Ali Shahini, Liang Zhang, Di Wu, Jingjing Yao, 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.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

The demand for online services including web search, online gaming, distributed file systems such as Google File System (GFS), and distributed Storage System such as BigTable and MapReduce is growing exponentially. This explosion of demand for online services has led to a multitude of challenges in Data Center Networks (DCNs) from DCN architecture design, congestion notification, TCP Incast, virtual machine migration, to routing in DCNs [67].

Most importantly, data centers electric power usage is growing at a rapid pace. In 2013, U.S. data centers consumed an estimated 91 billion kilowatt-hours of electricity, and as the fastest growing consumer of electricity, they are estimated to consume roughly 140 billion kilowatt-hours in 2020 which will cost $13 billion in electricity bill and emit 100 million metric tons of carbon pollution [25]. This huge average annual electricity consumption is due not only to the the continuing explosion of Internet traffic but also to the gravity of preparing DCNs as a scalable and reliable computing infrastructure. Online services run on hundreds of thousands of servers spread across server farms provisioned for the peak load. In fact, to assure the user demands satisfaction, the servers run 24/7 and in vast underutilization the majority of the time. To put this in perspective, the total power consumption at a data center includes the Base Load and Proportional Load. The base load indicates the power consumption even when some of the turned on servers are idle. On the other hand, the proportional load is the extra power consumption which is proportional to the CPU utilization of the servers and accordingly to the load. Therefore, even being idle, servers draw the base load power, thus incurring a substantial amount of annual energy use. However, in the past few years, more server capacities have

been virtualized to facilitate multiple Virtual Machines (VMs) being run on a single Physical Machine (PM).

Complying with all of our online activities but limiting the increasing energy demand in an environmentally friendly manner calls for innovations across different disciplines. Recently, a great deal of research has been done to cut the data center's power consumption and accordingly the cost of electricity. A great part of the studies mainly aims at proposing new power management techniques by investigating the CPU and memory power consumption of the servers. For examples, Dynamic Voltage/Frequency Scale (DVFS) schemes like [53] have been deployed to reduce the CPU power and new techniques such as [28] have been proposed to adjust the power states of the memory devices in order to dynamically limit memory power consumption. However, the data center operators prefer to maintain a high level of reliability and uptime with their less expensive inefficient facilities rather than to install energy efficient devices at the cost of higher upfront price [25].

Opportunities to improve the data centers energy efficiency is not limited to the improvements in computing components. The energy consumption break down of data centers shows that a course of action is required to improve the energy consumption at other components like network equipment, electrical power delivery and conversion, cooling, and lighting. To this end, Power Usage Effectiveness (PUE) metric has been commonly adopted as a measure of data centers efficiency, and is defined as the ratio of the total energy consumed by the data center to that consumed by the Information Technology (IT) equipment (EPA report on server and data center energy efficiency, Final Report to Congress, Aug. 2007). Power delivery and cooling efficiency has been the subject of interest of many recent research papers, and a large number of studies have aimed at innovating networking components and topologies to shave the power consumed by the IT network.

Another approach which addresses the energy consumption in all components is referred to as green data centers. The concept not only tries to cut down the electricity consumption and its cost but also integrates renewable energy resources such as solar panels and wind farms into data centers, thereby promoting sustainability and green energy. The data center operators can assess the sustainability of their data centers using the Carbon Usage Effectiveness (CUE) metric along with PUE. CUE is defined as the ratio of the total CO_2 emissions caused by the total data center energy consumption to that by the IT equipment energy consumption. CUE has the ideal value of 0.0 which indicates no carbon use is associated with the data center operations [9].

Shaving the energy consumption and its cost via load shedding and load shifting ([64] and references therein) is another approach. Load shedding is associated with QoS degradation where data centers based on the Service Level Agreements (SLAs) decide to serve some types of the workload less effectively by utilizing less energy. On the other hand, load shifting algorithms investigate the possibility of shifting the load in time to run when for example cheaper electricity is available.

Moreover, the effectiveness of the geographical load balancing on the energy costs has been demonstrated in some studies. In the so called Geographical Load Balancing (GLB), the workload is distributed among Internet scale data centers spread across geographical diversity [52]. In fact, the Internet scale powerful data centers are few because of the scale and cost of the deployment and operation. These few numbers of data centers are generally dispersed in different geographical regions. The main goal of deploying such geo-dispersed data centers is not only to provide redundancy, scalability and high availability but also to more efficiently employ global resources such as utilizing price-diversity in electricity markets or utilizing locational diversity in renewable power generation [52] (see Figure 1.1). Therefore, the powerful data centers are generally deployed far away from a large majority of

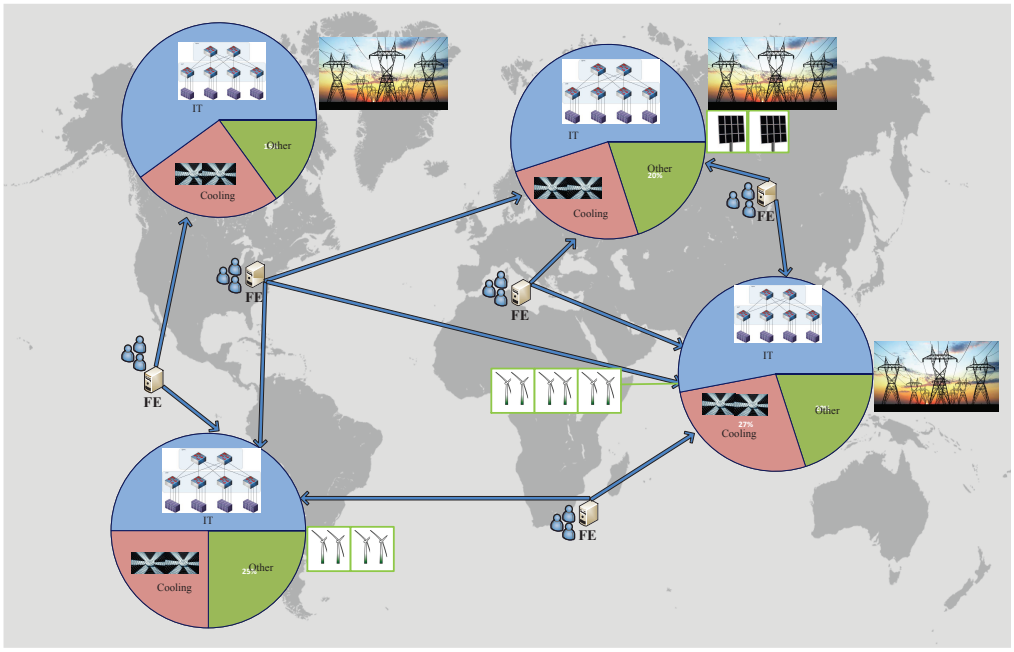


Figure 1.1 Geographical dispersed data centers.

users. To this end, Front-End (FE) servers are co-located with users. Each FE server receives requests from its nearby users and distribute the requests to the back-end servers at geo-dispersed data centers. In fact, each FE server functions as a workload distribution center that manages the workload by distributing the user requests to the appropriate data centers.

The selection of the appropriate data centers can be based on different parameters like server or content availability, the network distance between FE and data center, the efficiency of the data centers, the cost of the electricity, and availability of the renewable energy. Therefore, different workload distribution strategies can be adopted at each FE by considering different objectives like maximizing green energy utilization, minimizing the cost of electricity or maximizing the profit gained by running data center networks. On the other hand, each service request has to be handled within a deadline determined by the Service Level Agreement (SLA). Different parameters like the throughput of the connection between users and FE server, FE server and back end servers at the data center, and the

queuing and processing delay at the data center are contributing to the end-to-end delay of a service request [19]. The QoS at a data center is generally ensured by imposing an upper bound on the queuing delay at the data center which has been commonly modeled as M/GI/1 Processor Sharing (PS) queue or M/M/1 queue [52].

To benefit from the energy efficiency and sustainability advantages of greening, data centers have been recently integrated with a green power source such as wind turbine or solar panel. There are three different ways to green a data center. The first approach, called behind the meter, is to install renewable power generators at the data center location. In this case, the data center operator can own the power generation system itself or a third party can install the system and sell the generated power to the data center. However, the most efficient location to build a renewable power source is not always the same as the best location to build an efficient data center. Therefore, data center operators such as Google choose to either purchase Renewable Energy Certificates (RECs) or make Power Purchase Agreements (PPAs) to procure both power and RECs [5].

To maximize green energy utilization, one FE server can manage the distribution of its incoming workload to different data centers based on the availability of green energy. The available green energy at a data center can be determined by the green energy generation or storage at the data center. The generated on-site green energy at a data center can be predicted by taking into account of weather dependency of green energy. Specifically, when the renewable generator is a wind turbine, the prediction can rely on the foremost forecasting techniques which are based on Numeric Weather Prediction (NWP) of wind speed and power [61]. The prediction may include Very-Short Term Forecasting, Short Term Forecasting, Medium Term Forecasting and Long Term Forecasting techniques. If the case is solar generation, machine learning based prediction techniques can be employed. In the case of purchased green energy, although it is not possible to track the flows of green energy from grid, green energy

generation can be estimated via data center's RECs. Moreover, when extra green energy is available, each data center can store green energy at energy storage devices and draw the energy from the storage device later.

While the data centers operate 24/7, the green energy is not a constant available resource to power them. Therefore, the data centers have to be connected to on-grid brown energy. In this case, we should note that the brown energy is procured in deregulated electricity markets.

Unlike the regulated electricity markets, in deregulated electricity markets such as day-ahead and real-time markets, the electricity prices vary during the day. The final prices are set based on the bidding process between the energy suppliers and consumers. Some studies also suggest that the data centers can participate in the bidding process and procure the electricity directly from the wholesale market [31]. However, the prices are not known to the data centers until the operating time. For example, the day-ahead prices are usually revealed several hours up to one day in advance while the real-time prices are known only a few minutes in advance. Therefore, the electricity price forecasting methods have to be employed when participating in biding process.

GLB can be considered as an opportunity to reduce the cost of electricity by utilizing electricity price diversity at different locations. In other words, in order to minimize the electricity cost, each FE server can mange the workload by sending the requests to the data center locations with cheaper price of electricity.

In the past few years, a small and cohesive body of work investigated workload distribution across multiple data centers and the researchers came up with a variety of policies and algorithms. The social impacts of geographical load balancing is explored in [52] and two distributed algorithms are provided that can be used to compute the optimal routing as well as provisioning decisions for Internet-scale systems. Another couple of research papers approach the problem by employing the mixed integer

programming [51, 54]. Also, Ghamkhari *et al.* [30] addressed the trade-off between minimizing a green data center's energy costs and maximizing its revenue. Also, Zhao *et al.* [69] took into consideration of dynamic VM pricing and designed a new algorithm to maximize the long-term cloud provider's profit.

Recently, fog computing paradigm [12] was introduced by Cisco as a new platform in which the goal is to support the requirements of Internet of Things (IoTs) varying from low latency, mobility, geo-distribution to location awareness [13]. To this end, the fog computing platform was designed as a multi-tiered architecture in which different parts of an IoT application can be deployed on the IoT device, fog platform and a data center as three different tiers. In the past few years, several efforts have developed similar concepts to the fog computing. Most notably, three years before the introduction of fog computing, the idea of cloudlet as a trusted, resource-rich computer which is well-connected to the Internet and available for use by nearby mobile devices was introduced in [58]. The notion of the cloudlet or a "data center in a box" has been further developed by a research team at Carnegie Mellon University by introducing and developing various mechanisms [22, 34, 50, 59, 60]. In parallel with the development of fog computing and the cloudlet concept, the so called Mobile Edge Computing (MEC) idea has being standardized by an Industry Specification Group (ISG) lunched by the European Telecommunications Standards Institute (ETSI) [36]. MEC recognized as one of the key emerging technologies for 5G networks aims at providing computing capabilities in proximity of Mobile Users (MUs) and within the Radio Access Network (RAN), thereby reducing the latency and improving the Quality of Service (QoS) [36]. Moreover, MEC is becoming an important enabler of consumer-centric IoT with potential applications such as smart mobility, smart cities, and location-based services [23]. Therefore, in such user-centric IoT concept in which the users participate in sensing and computing tasks, computation-intensive tasks still need to be offloaded to either the cloud or the computing resources at the edge.

A large and cohesive body of work investigated the major limitations of Mobile Cloud Computing (MCC), e.g., the radio access associated energy consumption of mobile devices and the latency experienced over Wide Area Network (WAN), and the researchers came up with a variety of policies and algorithms. For instances, the computation offloading problem via joint optimization of the communication and computation resources is explored in [8] and a message-passing approach for the same problem is proposed in [40]. A cloudlet network planning approach for mobile access networks is introduced in [17] which optimally places the cloudlet facilities among a given set of available sites and then assigns a set of access points to the cloudlets by taking into consideration of the user mobility. Chiang *et al.* [21] summarized the opportunities and challenges of edge computing in the networking context of IoT and indicated that the fog concept can fill the technology gaps in IoT. Gonzalez *et al.* [32] also explored the state of the art of edge computing and its applications in IoT. Moreover, adaptive edge computing solutions for IoT networking are presented in [39], which aims to optimize traffic flows and network resources.

1.1 Contributions

We have made the following major contributions.

1.1.1 Green Versus Brown

Green and price diversities are considered as an opportunity to design a green and low cost GLB approach that not only can maximize the utilization of green energy but also minimize the cost of electricity. However, due to the different costs and different environmental impacts of the renewable energy and brown energy, such a GLB approach should tap on the merits of the separation of green energy utilization maximization and brown energy cost minimization problems. To this end, in this thesis, we propose the concept of decomposing the workload into the workloads

served by green and brown energy. In other words, the notion of green workload and green service rate, versus brown workload and brown service rate, respectively, to facilitate the separation of green energy utilization maximization and brown energy cost minimization problems.

The idea is to distinguish the servers at each data center based on the energy which is utilized to power them. In fact, some servers are turned on and powered by the available green energy (green servers) and the others if needed by purchasing brown energy (brown servers). Therefore, the distinction between green and brown workloads is made mainly based on the server which is utilized to serve the workload. In specific, the workload served by a green server is defined as the green workload and similarly the workload served by a brown server is defined as the brown workload. Moreover, using this idea, we can tackle the shortcoming in some studies, which propose an integrated optimization framework but under the assumption that local renewable generation is always less than the local power consumption. In fact, using the green versus brown concept, each data center utilizes green energy as much as possible, and purchases brown energy only when the green energy generation is not adequate to serve all incoming workloads.

Nevertheless, most of the existing studies in the field of geographically dispersed data centers either neglect an accurate queueing analysis or assume Poisson workload arrivals. Thus, in this thesis, we formulate an optimization framework for profit maximization which relies on the accuracy of the G/D/1 queue [30, 46] in capturing the workload distribution. In particular, we propose a new workload distribution strategy for geographically dispersed green data centers in which our strategy aims at maximizing the revenue and minimizing the energy expenditures. In a G/D/1 queueing model, the arrival rate of the requests can be modeled by a random process with an arbitrary and general probability distribution function like Gaussian processes that have received significant attention as accurate models for the arrival process. In

addition, assuming a fixed service rate for each time interval allows us to model the SLA-deadline as a finite-size queue. Our optimization-based workload distribution strategy taps on the merits of workload decomposition into green and brown workloads served by green and brown energy resources, respectively.

1.1.2 Fundamental Energy Trade-offs

Most of the proposed GLB strategies aim at reducing the energy cost or brown energy consumption via distributing the requests to the locations with cheaper price of electricity or higher renewable energy generation. However, such strategies may increase the total power consumption due to the fact that different data centers have different servers with different service capabilities, and also a request sent to different data centers experiences different network delays. In fact, consuming the same or even more amount of energy at one data center may handle less number of requests than another data center. In other words, the idea of sending a request to another data center with higher network delay or less service capability only in order to benefit from cheaper electricity or utilize more renewable energy may lead to a significant increase in the total power consumption.

The extra green energy generation at a data center can be injected into the power grid, and the data center can receive compensation for the injected power. In the case of electricity, the cheap electricity at a data center can be stored at energy storage devices to be utilized later when the electricity becomes more expensive. Therefore, the more green energy utilization or the cheaper electricity at the expense of increasing the total energy consumption is not necessarily the best option. To find the achievable tradeoffs between total power consumption and green energy utilization, we propose to model geo-dispersed data centers with an information flow graph. Note that this idea may be adopted to capture the achievable tradeoffs between total power consumption and the cost of electricity.

1.1.3 Hierarchical Mobile Edge Computing

In a MEC environment, a mobile subscriber/user can be considered as a person/entity with one or more IoT devices that can utilize the computing and storage capabilities at the edge. However, it is still desirable to investigate an efficient strategy that can be used to offer the computing and storage facilities, and accordingly the required communications bandwidth to a mobile subscriber. Such strategy not only has to allow the users to adapt their computing and communications capacities according to their requirements but also has to change its economics by allowing the users to pay only for the resources that they utilize. In this regard, the main challenge is the resource poverty at the edge where we are dealing with resource-poor computing facilities not big data centers. To this end, the current study aims to address the aforementioned issue by proposing an auction-based profit maximization approach in Chapter 4. While there are some studies that investigate auction models for the resource allocation in a could computing system, only a small body of work has studied auction mechanisms for the resource allocation in MEC. For example, Zhang *et al.* [68] modelled the resource allocation process of a mobile cloud computing system as an auction mechanism by taking into consideration of premium and discount factors and derived the optimal solutions of the resource allocation in their proposed auction mechanism. In addition, a concurrent Virtual Machine (VM) pricing and the distribution of VM instances across Physical Machines (PMs) in a data center are presented in [48]. Zheng *et al.* [70] developed an optimization model for the spot pricing system and answered the question of how users should bid for cloud resources. The auction model in this dissertation is inspired by the equilibrium pricing models, such as the model presented in [48] tailored for a could computing system, i.e., a data center. However, we face the issues of user mobility and the resource poverty at the edge when we apply such pricing models to an MEC environment, and thus, we propose a hierarchical network architecture as well as a two time scale resource

allocation approach to address these issues. Moreover, we formulate our auction model as a profit maximization problem in which the gained profit is established by considering not only the revenue of serving the VM demands and the electricity cost of running the computing and network facilities, but also the revenue lost due to network delay.

1.1.4 Capacity Provisioning

To shed some light on the idea of capacity provisioning, let's consider distributed CCTV video cameras as a potential application of edge computing. For example, more than 400 CCTV video cameras are distributed over the state of New Jersey and they are generating a huge amount of video data each day. These data have to be processed and stored for different applications such as traffic congestion mitigation strategies. However, sending all of these data to a backend system such as Traffic Management Centers (TMC), which is equipped with computational and storage capabilities, is not practical due to two main reasons: 1) The opportunity to process video data and act on the processed data might be gone after the time it takes to send data all the way to TMC over the backhaul network. 2) Continuously capturing video on the cameras poses a permanent stress on the network paths to the centralized controller. One simple solution to mitigate the congestion on the backhaul network may offer buffering data at the intermediate network nodes for later transmission. This solution is not useful because cameras are capturing videos 24/7 and there will never be a future time when the backhaul network is not overwhelmed. Another solution towards this problem can be a distributed edge computing network architecture by leveraging the concept of the cloudlets. In such a distributed network architecture, each camera itself as well as the aggregation nodes in the network such as the network hubs and routers are all the potential sites to install the cloudlets. Therefore, two important questions must be answered about such a distributed edge computing architecture: 1) Should

we consider a flat or hierarchical design? 2) What is the size of each cloudlet, i.e., how much capacity should be provisioned at each cloudlet location? To this end, the current study aims to address the aforementioned issue by proposing a hierarchical capacity provisioning scheme. In fact, the idea here is to efficiently provision a total capacity budget at the edge while the distribution of the computation workload at different locations is given.

1.1.5 Optimal Code Partitioning

Computation offloading requires code partitioning to decide which tasks should be executed locally and which tasks should be offloaded to the mobile edge depending on different parameters such as energy and delay. Existing computation offloading problems in the literature such as [8, 26] propose joint optimization framework for the code partitioning problem and the radio resource optimization. Such joint optimization frameworks lead to Mixed Integer Nonlinear Programming (MINLP) models in which finding the optimal solution requires an exhaustive search over all the useful call graph partitions, i.e, all the configurations that satisfy the feasibility conditions. Accordingly, these schemes propose to find sub-optimal solutions for code partitioning and then optimize the radio resources for a given partitioning. A message-passing approach for the same problem is proposed in [40] which reduces the complexity of the computation offloading problem. However, the proposed model in [40] considers the code partitioning problem between a mobile device and only one remote location.

In summary, in Chapter 2:

- We develop a new model to maximize the profit of running geographically dispersed data centers. In our model, it is assumed that each data center is offering multiple classes of services and we take into account of individual SLA-deadline for each type of service. Also, we assume that each data center either has a renewable power source or is powered by a nearby wind or solar farm thereby taking into account of green energy. However, as the green energy resources may not be adequate to meet the QoS requirements for all incoming

workloads, each data center is also provisioned by on-grid energy. We further elaborate our model by taking into consideration of geographical electricity price diversity due to different electricity markets and time zones of the dispersed data centers.

- Based on the developed model, we design an optimal workload distribution strategy in terms of the gained profit by the data centers. The profit is defined as $revenue - cost$ by considering the deadline, service income, penalty for the service requests of each class, and also both green and brown energy costs. Our strategy relies on the accuracy of the G/D/1 queueing model in capturing the workload distribution. Furthermore, we prove the convexity of our optimization and therefore its appropriateness for practical purposes. In the optimization frameworks such as [30] which are proposed for a single data center, the service rate is the only decision variable. As our model is an extension for a group of data centers, our objective function and the constraints are functions of both allocated workloads to the data centers and the service rate at each data center. In other words, we maximize the profit by not only optimizing the service rates at data centers but also allocating optimized workload to each data center. To prove the convexity of our problem, we introduce the average number of dropped requests at each data center as an extended SLA constraint and based on that we can prove the convexity of the whole problem by using the convexity of the perspective of a function.

- Our optimization model relies on the potential merit of the decomposition of the workload to the green and brown workloads thereby taking into account of different costs and different environmental impacts of green and brown energy. In this way, we can allocate the green workload to the data centers based on the availability and cost variation of the green energy at different locations. However, for the brown workload, our strategy takes into account of electricity price diversity and hence distinguishes the data centers by the price of electricity. In fact, we take into consideration of not only the cost of brown energy but also one time capital and maintenance expenses of renewable energy. Therefore, unlike some of the existing works in the literature, our optimal profit is not under the assumption that local renewable generation is always less than the local power consumption.

In Chapter 3:

- We define a new service efficiency parameter for geo-dispersed data centers based on an M/GI/1 Processor Sharing (PS) queue analysis by taking into consideration of the network delay.

- We develop a new information flow graph based model for geo-dispersed data centers to capture the tradeoff between the total and brown power consumption.

- Based on the developed model, we characterize the achievable tradeoff between total and brown power consumption.

In Chapter 4:

- We propose a HIerachichal Mobile Edge Computing (HI-MEC) architecture in accordance with the principles of LTE-Advanced backhaul network and introduce the notion of *field*, *shallow* and *deep* cloudlets.

- We propose a two time scale mechanism to allocate the computing and communications resources to the MUs. The importance of the proposed two time scale is due to the fact that the economics of computing resources cannot change as quickly as the traffic loads of the MUs. In particular, the decision about the price and distribution of the computing resources are made in longer time frames, while the bandwidth allocations are updated in shorter time slots. To this end, we formulate a Binary Linear Programming (BLP) aimed at maximizing the profit of the service provider and a convex optimization problem for bandwidth allocation. We also design heuristic algorithms to solve the BLP problem and a centralized solution is proposed for the bandwidth allocation problem.

- We evaluate the performance of the heuristic algorithms via extensive simulations.

In Chapter 5:

- We propose a hierarchical capacity provisioning scheme by considering a 2-tier edge computing network architecture consisting of shallow and deep cloudlets.

- We investigate two different network scenarios based on accurate queueing analysis. In particular, we study the case that the network delay between the shallow cloudlets and the deep cloudlet is negligible as well as the case in which the deep cloudlet is located somewhere deeper in the network, and thus the network delay between the shallow cloudlets and the deep cloudlet matters. We also formulate optimization problems for each case and investigate the solution to each problem by using stochastic ordering and optimization algorithms.

In Chapter 6:

- Inspired by distributed processing systems [11], we propose to use the shortest tree algorithm to optimally schedule tasks in mobile edge networks. More importantly, we extend the code partitioning problem to scheduling problem over time and a hierarchical mobile edge.

- We investigate two different optimization scenarios. In particular. the first scenario aims at finding an optimal task scheduling for given radio parameters. In the second scenario, we investigate joint optimization of task scheduling and the mobile device's transmission power, and show that by using the proposed scheduling scheme, the transmission power optimization problem becomes a disjoint problem from the task scheduling problem.

CHAPTER 2

PROFIT MAXIMIZATION FOR GEOGRAPHICAL DISPERSED GREEN DATA CENTERS

This chapter aims at maximizing the profit associated with running geographically dispersed green data centers, which offer multiple classes of service. To this end, we formulate an optimization framework which relies on the accuracy of the G/D/1 queue in characterizing the workload distribution, and taps on the

2.1 System Model

Figure 2.1 shows the proposed system model in which we consider a group of $|N|$ data centers dispersed at different regions. Each data center is equipped with a collection of M_i homogeneous servers.

The data centers are supplied by multiple types of power. The major power supply of each data center is on-grid or brown energy. The data center has to pay brown energy prices according to its contract with the power company. The electricity pricing contract for each data center depends on the electricity markets at the data center's location. If the market is regulated, the electricity price has a flat rate during the day. On the other hand, if the region is following a deregulated market, the price of electricity is varying. In most cases, the data center pays less during off-peak hours and more during on-peak period. Therefore, we note the price variability among data centers located at different locations and time zones.

To reduce the cost of electricity and to capitalize on the environmental and sustainability advantages of green energy, we assume that each data center either is equipped with a renewable power source or has access to a nearby renewable energy source such as solar panels or a wind farm. It is worth mentioning that we assume the available renewable energy at each data center can only be used to supply power locally.

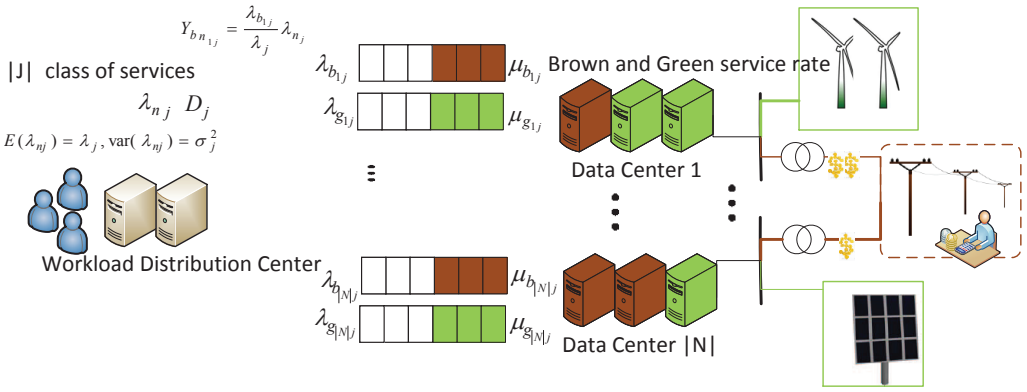


Figure 2.1 System model.

Each data center is offering $|J|$ multiple classes of service like web services, video streaming, etc. Each type of service has its specific deadline according to the SLA.

The service requests are initiated by users and arrive at the workload distribution center. One or a group of servers can serve as the workload distribution center [29]. These servers can be treated as the front-end devices that exist in multi-data center Internet services like Google and Itunes [49]. The distribution center facilitates workload flexibility at the demand side. In other words, this center inspects the arriving requests from all users and manages the distribution of the incoming workload to the geographically dispersed data centers based on the availability of green energy and the price of electricity. In our formulation, the total power consumption at each data center takes into account of the Base Load and Proportional Load [30],

$$\textit{Total Power Consumption at data center } i=$$
$$m_i[P_{idle} + (E_{usage} - 1)P_{peak}] + m_i[(P_{peak} - P_{idle})U_i], \tag{2.1}$$

where the base load, $m_i[P_{idle}+(E_{usage}-1)P_{peak}]$, indicates the power consumption even when all of the turned on servers are idle. The proportional load, $m_i[(P_{peak}-P_{idle})U_i]$, is the extra power consumption which is proportional to the CPU utilization of the

servers, U_i, and accordingly to the workload. It is worth mentioning that both base and proportional loads are computed based on the number of switched on servers, m_i, idle power, P_{idle}, and average peak power of a single server, P_{peak}. Moreover, due to different energy efficiencies at different data centers, the definition of the total power consumption incorporates the Power Usage Effectiveness (PUE) ratio, E_{usage}, thereby amalgamating the power consumption at facility for cooling, lighting, and other overhead [15].

2.2 Problem Formulation

We divide the running time of the data centers into a sequence of time slots at equal length, T, e.g., a few minutes. Our goal is to maximize the data centers' total profit during the interval T. To this end, we propose an optimization problem to be solved at the beginning of each time slot in which we update the number of turned on servers as well as the allocated workload to each data center. Note that for the analysis, we consider a single time slot, e.g., Δ as the time slot of interest, and omit the explicit time dependence in the notations.

At the beginning of each time slot, we allocate the workload (total number of service requests) to the data centers based on the availability of green energy and the price of electricity. As the renewable energy and brown energy incur different costs and different environmental impacts, we decompose the total workload into the green and brown workloads. In fact, we distinguish the servers at each data center based on the energy which is utilized to power them. Some of the servers are turned on and powered by the available green energy (green servers), and the others, if needed, by purchasing brown energy (brown servers). Therefore, the distinction between green and brown workloads is made mainly based on the server which is utilized to serve the workload. Specifically, the requests served by a green server are defined as the green workload and similarly those by a brown server the brown workload.

The data center's profit is modeled as $Revenue - Cost$, where the data center's revenue is calculated based on the QoS requirements satisfaction and the cost indicates the energy cost. Owing to the limited computational resources at the data centers, the allocated requests to a data center are first placed in a queue before they can be processed by any available server. Accordingly, to satisfy the QoS requirements, the queueing delay for each service request should be limited by a deadline. If the data center can handle the service requests by the deadline, it receives the service income. Otherwise, it has to pay penalty to its customers. These three parameters, i.e., the deadline, service income, and penalty, depend on the type of service and are determined by the SLA [30, 47]. Thus, we assume that the waiting requests of different classes of service at each data center are placed in different queues. Denote D_j, δ_j, and γ_j as the deadline, service income, and penalty for the service requests of class j, respectively. The service requests that are not handled by the deadlines are discarded [65]. In our problem formulation which is based on the workload decomposition, we distinguish the profit gained by serving green workload from the brown workload as the green and brown profit, respectively. To this end, we assume the green and brown requests of each class are placed in two different queues at a data center. In the next two subsections, we will formulate the green and brown profits.

2.2.1 Green Profit Formulation

We assume that the request rate of each class of service at the workload distribution center is a random process with an arbitrary and general probability distribution function, and λ_{n_j} denotes the service request rate of class j at time n. Let λ_j be the average rate of receiving service requests of class j at the workload distribution center within time slot Δ of length T. Also, σ_j^2 denotes the variance of the class j service request rate's probability distribution function. Request interarrival times are

assumed to be much shorter than a time slot duration, so that the request allocation can be based on the average arrival rate during the time slot.

We allocate $\frac{\lambda_{g_{ij}}}{\lambda_j}$ fraction of the service requests to the data center i's green servers. These requests are first placed in a particular queue on green servers. The input process to this queue, i.e., $\lambda_{g_{n_{ij}}} = \frac{\lambda_{g_{ij}}}{\lambda_j}\lambda_{n_j}$, has the same general probability distribution function as the request rate of class j. Thus, $\lambda_{g_{ij}} \neq 0$ and $\sigma^2_{g_{ij}} = (\frac{\lambda_{g_{ij}}}{\lambda_j})^2\sigma^2_j$ are the mean and variance of the input process to the corresponding queue, respectively.

Based on the aforementioned QoS model, the green revenue earned by the data center i for serving the green requests of different classes of service within a time slot can be calculated as, $R_i(\lambda_{g_{ij}}, \mu_{g_{ij}}) = \sum_{j=1}^{|J|}([1 - P_L(\lambda_{g_{ij}}, \mu_{g_{ij}})]\delta_j\lambda_{g_{ij}}T - P_L(\lambda_{g_{ij}}, \mu_{g_{ij}})\gamma_j\lambda_{g_{ij}}T)$, where $P_L(\lambda_{g_{ij}}, \mu_{g_{ij}})$ denotes the probability that the waiting time for a service request of class j exceeds its SLA-deadline. Note that $\mu_{g_{ij}}$ denotes the green service rate, i.e., the rate that the requests of class j are removed (i.e., served) from the corresponding queue by the data center i's green servers.

To obtain $P_L(\lambda_{g_{ij}}, \mu_{g_{ij}})$, the SLA-deadline is translated into the loss probability of a G/D/1 queue. In a nutshell, it is assumed the service rate that the service requests are removed from the queue, i.e., $\mu_{g_{ij}}$, is fixed over the time slot. Thus, for instance, if there are Q_{ij} number of requests waiting in the queue upon the arrival of a new service request, it takes $\frac{Q_{ij}}{\mu_{g_{ij}}}$ seconds until the new request can be handled by any available server. If $\frac{Q_{ij}}{\mu_{g_{ij}}} \leq D_j$, then the new request can be handled before the deadline. Therefore, the SLA-deadline can be modeled by a finite-size queue with length $\mu_{g_{ij}}D_j$. In other words, in order to handle a new request by the SLA-deadline, it has to enter a queue with length $\mu_{g_{ij}}D_j$ [30]. According to queueing analysis [46], the loss probability of the finite-size queue can be accurately estimated from the tail of the queue length distribution for any general probability distribution. However, it is known that the estimation yields the highest level of accuracy when the service

request rate is characterized by a Gaussian process [46]. Therefore, through out the rest of this thesis, the request rate of each class of service, accordingly the input process to the queues is assumed to be a Gaussian process, and the loss probability can be obtained as,

$$P_L(\lambda_{g_{ij}}, \mu_{g_{ij}}) = \alpha(\lambda_{g_{ij}}, \mu_{g_{ij}}) e^{-\frac{1}{2}\min_{n\geq 1} M_n(\lambda_{g_{ij}}, \mu_{g_{ij}})}, \tag{2.2}$$

where

$$\alpha(\lambda_{g_{ij}}, \mu_{g_{ij}}) = \frac{1}{\lambda_{g_{ij}}\sqrt{2\pi}\sigma_{g_{ij}}} e^{\frac{(\mu_{g_{ij}} - \lambda_{g_{ij}})^2}{2\sigma^2_{g_{ij}}}} \int_{\mu_{g_{ij}}}^{\infty} (r - \mu_{g_{ij}}) e^{\frac{-(r-\lambda_{g_{ij}})^2}{2\sigma^2_{g_{ij}}}} dr, \tag{2.3}$$

and for each $n \geq 1$,

$$M_n(\lambda_{g_{ij}}, \mu_{g_{ij}}) = \frac{((D_j - d_i)\mu_{g_{ij}} + n(\mu_{g_{ij}} - \lambda_{g_{ij}}))^2}{nC_{\lambda_{g_{ij}}}(0) + 2\sum_{l=1}^{n-1}(n-l)C_{\lambda_{g_{ij}}}(l)}, \tag{2.4}$$

where $C_{\lambda_{g_{ij}}}(l)$ is the autocovariance of the class j service request rate's probability function at data center i, and we have $\sigma^2_{g_{ij}} = C_{\lambda_{g_{ij}}}(0)$. Also, d_i is the network delay experienced by a request from the workload distribution center to data center i.

The green power consumption at each data center depends on the number of switched on green servers as well as the CPU utilization of each green server. The total number of switched on green servers at data center i can be expressed based on the total green service rate as $m_{g_i} = \sum_{j=1}^{|J|} \frac{\mu_{g_{ij}}}{k_j}$, where each server can handle k_j service requests of class j per second. Also, within the interval of T, each switched on green server handles $\frac{T(1-P_L(\lambda_{g_{ij}}, \mu_{g_{ij}}))\lambda_{g_{ij}}}{m_{g_i}}$ requests of class j [30]. Thus, the total CPU busy time of each server can be obtained as $\sum_{j=1}^{|J|} \frac{T(1-P_L(\lambda_{g_{ij}}, \mu_{g_{ij}}))\lambda_{g_{ij}}}{m_{g_i}k_j}$. By dividing the total server busy time by T, we have the CPU utilization $U_{g_i} = \sum_{j=1}^{|J|} \frac{(1-P_L(\lambda_{g_{ij}}, \mu_{g_{ij}}))\lambda_{g_{ij}}}{m_{g_i}k_j}$.

Therefore, referring to the definition of power consumption in (6.1), the total green power consumption in data center i at the time of interest can be expressed as,

$$E_i(\lambda_{g_{ij}}, \mu_{g_{ij}}) = (P_{idle} + (E_{usage} - 1)P_{peak}) \sum_{j=1}^{|J|} \frac{\mu_{g_{ij}}}{k_j} + (P_{peak} - P_{idle}) \sum_{j=1}^{|J|} \frac{(1 - P_L(\lambda_{g_{ij}}, \mu_{g_{ij}}))\lambda_{g_{ij}}}{k_j}. \quad (2.5)$$

Note that the total number of the green servers at each data center, and accordingly the green service rate is limited by the available green energy at the time slot of interest. Let W_i be the available green energy at data center i within the time slot. W_i is predicted at the beginning of the time slot, and depends, for example, on wind speed and solar irradiance. Similar to some other published thesiss in the literature such as [29, 30] it is assumed that the time slot is small enough (e.g., every few minutes). Therefore, while the amount of renewable energy is changing at different time of a day, it is reasonable that solar irradiance and wind speed are relatively stable within a slot. We assume C_{g_i} is the cost of renewable energy at data center i. The cost of green energy generation includes one time capital and maintenance expenses. The average unit cost of renewable energy can be obtained by averaging over the total amount of energy generated during the whole operation period. Therefore, the total green profit gained by all the data centers during the time slot of interest can be calculated as $Profit_g = \sum_{i=1}^{|N|}(R_i(\lambda_{g_{ij}}, \mu_{g_{ij}}) - C_{g_i} TE_i(\lambda_{g_{ij}}, \mu_{g_{ij}}))$.

2.2.2 Brown Profit Formulation

If green energy generation is not adequate to serve all incoming workload, brown energy is purchased. Brown energy is considered as an additional resource to power on additional servers referred to as the brown servers. We allocate $\lambda_{b_{n_{ij}}} = \frac{\lambda_{b_{ij}}}{\lambda_j}\lambda_{n_j}$ service requests, as the brown requests, to the data center i's brown servers. These requests are first placed in their particular queue on brown servers, and $\lambda_{b_{ij}} \neq 0$ and $\sigma^2_{b_{ij}} = (\frac{\lambda_{b_{ij}}}{\lambda_j})^2\sigma_j^2$ are the mean and variance of the input process to the queue, respectively.

When using brown energy, we note the different deregulated electricity markets of data centers located at different regions. Denote C_{b_i} as the price of electricity at data center i within the time slot of interest. In order to benefit from the electricity price diversity, the distribution center can employ the day-ahead electricity price forecasting methods [7, 66]. Therefore, the total brown profit gained by all the data centers during the time slot of interest can be calculated as, $Profit_b = \sum_{i=1}^{|N|}(R_{b_i}(\lambda_{b_{ij}}, \mu_{b_{ij}}) - C_{b_i}TE_{b_i}(\lambda_{b_{ij}}, \mu_{b_{ij}}))$. In the next section, we propose an optimization framework for the service request distribution. The objective of our framework is to maximize the total profit earned by the data centers within each time slot. Our optimization framework uses the results of renewable energy and electricity price forecasting methods.

2.3 Optimization Framework

In order to maximize the total profit earned by the data centers, we update the allocated workload and the service rates for each data center. In fact, we seek to maximize the total profit by optimizing the allocated green and brown requests (i.e., $\lambda_{g_{ij}}$ and $\lambda_{b_{ij}}$) as well as the green and brown service rates (i.e., $\mu_{g_{ij}}$ and $\mu_{b_{ij}}$) within each time slot. To this end, the following optimization problem is proposed to be solved at the beginning of the time slot of interest,

$$\underset{\lambda_{g_{ij}}, \mu_{g_{ij}}, \lambda_{b_{ij}}, \mu_{b_{ij}}}{\text{maximize}} \quad (Profit_g + Profit_b) \tag{2.6}$$

subject to

$$0 < \lambda_{g_{ij}} \leq \mu_{g_{ij}}, \ \forall i \in N, \ \forall j \in J, \tag{2.7}$$

$$0 < \lambda_{b_{ij}} \leq \mu_{b_{ij}},\ \forall i \in N,\ \forall j \in J, \tag{2.8}$$

$$\sum_{j=1}^{|J|} \frac{\mu_{g_{ij}}}{k_j} \leq \lfloor \frac{W_i(t)}{P_{peak} E_{usage}} \rfloor,\ \forall i \in N, \tag{2.9}$$

$$\sum_{i=1}^{|N|} (\lambda_{g_{ij}} + \lambda_{b_{ij}}) = \lambda_j,\ \forall j \in J, \tag{2.10}$$

$$\lambda_{g_{ij}} P_L(\lambda_{g_{ij}}, \mu_{g_{ij}}) \leq\ TH_j,\ \forall i \in N,\ \forall j \in J, \tag{2.11}$$

$$\lambda_{b_{ij}} P_L(\lambda_{b_{ij}}, \mu_{b_{ij}}) \leq\ TH_j,\ \forall i \in N,\ \forall j \in J, \tag{2.12}$$

where the inequality constraints (6.9), (6.8) are to lower bound the service rate of each queue by the average of the input process to that queue and are necessary for stabilizing the service request queue. In addition, the inequality constraint (2.9) is used to limit the green service rates by the available renewable energy in which we make full CPU utilization assumption. Also, we use equality constraint (2.10) to allot all the requests of each class to the data centers based on the average rate of receiving service requests. Moreover, by inequality constraints (2.11), (2.12), we add an extended SLA requirement in which the average number of dropped requests at each queue is upper bounded by a constant TH_j.

The proposed optimization problem is a convex optimization problem, as proven in the following theorem, and consequently can be solved by efficient optimization techniques, such as the interior point method (IPM).

Theorem 2.3.1. *The constrained optimization problem (6.7) is a convex optimization problem if data centers are profitable for each class of service and*

$$\mu_{g_{ij}} \geq 1 \quad and \quad \mu_{b_{ij}} \geq 1, \forall\ i, j \tag{2.13}$$

Proof. To show the convexity of the proposed optimization problem, we require to prove [14]:

- The objective function, i.e., $Profit_g + Profit_b$, is concave.
- The inequality constraint functions are convex.
- The equality constraint functions, i.e., $\sum_{i=1}^{|N|}(\lambda_{g_{ij}} + \lambda_{b_{ij}}) - \lambda_j$, are affine.

Since the corresponding functions of the constraints (6.9), (6.8), (2.9) and (2.10) are all linear, we start by proving the convexity of the following function,

$$f(\lambda_{g_{ij}}, \mu_{g_{ij}}) \triangleq \lambda_{g_{ij}} P_L(\lambda_{g_{ij}}, \mu_{g_{ij}}) - TH_j,\ \forall i \in |N|,\ \forall j \in |J|. \tag{2.14}$$

From (5.3), as e^{-x} is non-increasing, we have

$$P_L(\lambda_{g_{ij}}, \mu_{g_{ij}}) = \max_{n \geq 1} \alpha(\lambda_{g_{ij}}, \mu_{g_{ij}}) e^{-\frac{1}{2} M_n(\lambda_{g_{ij}}, \mu_{g_{ij}})}. \tag{2.15}$$

Since max preserves convexity [14] and TH_j is constant, the function $f(\lambda_{g_{ij}}, \mu_{g_{ij}})$ is proven to be convex if we can prove the following function,

$$f_n(\lambda_{g_{ij}}, \mu_{g_{ij}}) = \lambda_{g_{ij}} \alpha(\lambda_{g_{ij}}, \mu_{g_{ij}}) e^{-\frac{1}{2} M_n(\lambda_{g_{ij}}, \mu_{g_{ij}})}, \tag{2.16}$$

is convex for each $n \geq 1$.

After reordering the terms in (5.4), we can show that,

$$\begin{aligned} &\alpha(\lambda_{g_{ij}}, \mu_{g_{ij}}) = \\ &\frac{\sigma_{g_{ij}}}{\lambda_{g_{ij}}\sqrt{2\pi}}[1 - \frac{(\mu_{g_{ij}} - \lambda_{g_{ij}})}{\sigma_{g_{ij}}} e^{\frac{(\mu_{g_{ij}} - \lambda_{g_{ij}})^2}{2\sigma^2_{g_{ij}}}} \int_{\frac{(\mu_{g_{ij}} - \lambda_{g_{ij}})}{\sigma_{g_{ij}}}}^{\infty} e^{\frac{-u^2}{2}} du] \end{aligned} \tag{2.17}$$

By substituting $\sigma_{g_{ij}} = (\frac{\lambda_{g_{ij}}}{\lambda_j})\sigma_j$ in (2.17) and (5.5) respectively, and after simple algebraic manipulation we have,

$$\alpha(\lambda_{g_{ij}}, \mu_{g_{ij}}) = \\ \frac{C_{v_j}}{\sqrt{2\pi}}[1 - \frac{1}{C_{v_j}}(\frac{\mu_{g_{ij}}}{\lambda_{g_{ij}}} - 1)e^{\frac{1}{2C_{v_j}^2}(\frac{\mu_{g_{ij}}}{\lambda_{g_{ij}}}-1)^2} \int_{\frac{1}{C_{v_j}}(\frac{\mu_{g_{ij}}}{\lambda_{g_{ij}}}-1)}^{\infty} e^{\frac{-u^2}{2}} du] \tag{2.18}$$

and

$$M_n(\lambda_{g_{ij}}, \mu_{g_{ij}}) = \frac{((D_j - d_i + n)(\frac{\mu_{g_{ij}}}{\lambda_{g_{ij}}} - 1) + (D_j - d_i))^2}{\rho_{n_j}}, \tag{2.19}$$

where $C_{v_j} = \frac{\sigma_j}{\lambda_j}$ is the coefficient of variation of the class j's service request rate. Also,

$$\rho_{n_j} \triangleq\ nC_{v_j}^2 + 2\sum_{l=1}^{n-1}(n-l)\frac{C_{\lambda_j}(l)}{\lambda_j^2} \tag{2.20}$$

Equations (2.18) and (2.19) indicate that $f_n(\lambda_{g_{ij}}, \mu_{g_{ij}})$ is the perspective of the following function,

$$g_n(\mu_{g_{ij}}) = \alpha(\mu_{g_{ij}})e^{-\frac{1}{2}M_n(\mu_{g_{ij}})}, \tag{2.21}$$

where

$$\alpha(\mu_{g_{ij}}) = \\ \frac{C_{v_j}}{\sqrt{2\pi}}[1 - \frac{1}{C_{v_j}}(\mu_{g_{ij}} - 1)e^{\frac{1}{2C_{v_j}^2}(\mu_{g_{ij}}-1)^2} \int_{\frac{1}{C_{v_j}}(\mu_{g_{ij}}-1)}^{\infty} e^{\frac{-u^2}{2}} du] \tag{2.22}$$

and

$$M_n(\mu_{g_{ij}}) = \frac{((D_j - d_i + n)(\mu_{g_{ij}} - 1) + (D_j - d_i))^2}{\rho_{n_j}}. \tag{2.23}$$

If $g_n(\mu_{g_{ij}})$ is convex, so is its perspective function $f_n(\lambda_{g_{ij}}, \mu_{g_{ij}})$ [14]. Therefore, we continue our proof by proving the convexity of $g_n(\mu_{g_{ij}})$. Let's define

$$t \triangleq \frac{(\mu_{g_{ij}} - 1)}{C_{v_j}} \tag{2.24}$$

Then, we have

$$g_n(t) = \alpha(t)e^{-\frac{1}{2}M_n(t)} \tag{2.25}$$

$$\alpha(t) = \frac{C_{v_j}}{\sqrt{2\pi}}[1 - te^{\frac{t^2}{2}} \int_t^\infty e^{\frac{-u^2}{2}} du] \tag{2.26}$$

and

$$M_n(t) = \frac{((D_j - d_i + n)C_{v_j}t + (D_j - d_i))^2}{\rho_{n_j}}. \tag{2.27}$$

Then, the function $g_n(\mu_{g_{ij}})$ is proven to be convex if we can show for each $n \geq 1$,

$$\begin{aligned} g_n''(t) = e^{-\frac{1}{2}M_n(t)}(\alpha''(t) + \alpha(t)\frac{M_n'^2(t)}{4} \\ -\alpha'(t)M_n'(t) - \alpha(t)\frac{M_n''(t)}{2}) \geq 0 \end{aligned} \tag{2.28}$$

By simple algebra, we can show that,

$$\alpha'(t) = (\frac{t^2+1}{t})\alpha(t) - \frac{C_{v_j}}{\sqrt{2\pi}t} \tag{2.29}$$

and

$$\alpha''(t) = (t^2+3)\alpha(t) - \frac{C_{v_j}}{\sqrt{2\pi}} \tag{2.30}$$

By substituting (2.29) and (2.30) in $g_n''(t)$, we have

$$\begin{aligned} g_n''(t) = \frac{\alpha(t)e^{-\frac{1}{2}M_n(t)}}{t}[t^3 - t^2M_n'(t) \\ +(3 + \frac{M_n'^2(t)}{4} - \frac{M_n''(t)}{2})t - M_n'(t) + \frac{C_{v_j}}{\sqrt{2\pi}\alpha(t)}(M_n'(t) - t)] \end{aligned} \tag{2.31}$$

Now, we show (2.28) for all $t \geq 0$.

First, since $nC_{v_j}^2 \leq \rho_{n_j} \leq n^2C_{v_j}^2$, we can show that,

$$\frac{M_n'^2(t)}{4} - \frac{M_n''(t)}{2} \geq t^2 - 1 \tag{2.32}$$

Then, from the following upper and lower bounds [6]

$$\frac{2}{t + \sqrt{t^2+4}} \leq e^{\frac{t^2}{2}} \int_t^\infty e^{\frac{-u^2}{2}} du \leq \frac{2}{t + \sqrt{t^2 + \frac{8}{\pi}}}. \tag{2.33}$$

we have,

$$\frac{C_{v_j}}{\sqrt{2\pi}}[1 - \frac{2t}{t + \sqrt{t^2 + \frac{8}{\pi}}}] \leq \alpha(t) \leq \frac{C_{v_j}}{\sqrt{2\pi}}[1 - \frac{2t}{t + \sqrt{t^2+4}}] \tag{2.34}$$

which indicates $\alpha(t) \geq 0$ and we can show that

$$\frac{C_{v_j}}{\sqrt{2\pi}\alpha(t)} \geq (\frac{t+\sqrt{t^2+4}}{2})^2 \geq \ t^2+1 \tag{2.35}$$

From (2.35) and (2.32), the following inequality holds,

$$\begin{aligned} g_n''(t) &\geq \frac{\alpha(t)e^{-\frac{1}{2}M_n(t)}}{t}[t^3 - t^2 M_n'(t) \\ &+(3+t^2-1)t - M_n'(t) + (t^2+1)(M_n'(t)-t)] \\ &= \alpha(t)e^{-\frac{1}{2}M_n(t)}(t^2+1) \geq 0 \end{aligned} \tag{2.36}$$

Therefore, for all $t \geq 0$, i.e., $\mu_{g_{ij}} \geq 1$, $g_n(\mu_{g_{ij}})$ and consequently $f(\lambda_{g_{ij}}, \mu_{g_{ij}})$ is convex. The convexity of the following function:

$$f(\lambda_{b_{ij}}, \mu_{b_{ij}}) \triangleq \lambda_{b_{ij}} P_L(\lambda_{b_{ij}}, \mu_{b_{ij}}) - TH_j, \ \forall i \in N, \ \forall j \in J, \tag{2.37}$$

can be similarly be proven and we conclude the convexity of inequality constraints (2.11), (2.12).

Now, we prove the concavity of the objective function. Note that the nonnegative weighted sum of concave functions is concave [14]. Also, the functions $-\lambda_{g_{ij}} P_L(\lambda_{g_{ij}}, \mu_{g_{ij}})$ and $-\lambda_{b_{ij}} P_L(\lambda_{b_{ij}}, \mu_{b_{ij}})$ are concave. Therefore, by rewriting the objective functions based on $-\lambda_{g_{ij}} P_L(\lambda_{g_{ij}}, \mu_{g_{ij}})$ and $-\lambda_{b_{ij}} P_L(\lambda_{b_{ij}}, \mu_{b_{ij}})$, we can show that if the data centers are profitable for each class of service, i.e.,

$$\begin{aligned} &\delta_j + \gamma_j - \frac{P_{peak} - P_{idle}}{k_j} \max(C_{b_i}, C_{g_i}) \\ \geq \ &\delta_j - \frac{P_{peak} - P_{idle}}{k_j} \max(C_{b_i}, C_{g_i}) > 0, \forall \ i \end{aligned} \tag{2.38}$$

the objective function is concave and the proof is complete. □

It is worth mentioning that the G/D/1 model in [46] is valid only for the range of service rates, $\mu_{g_{ij}} \geq \lambda_{g_{ij}}$ and $\mu_{b_{ij}} \geq \lambda_{g_{ij}}$, which we have already considered in our constraints. Therefore, even if we do not allocate any workload to a data center, the service rate has to be set greater than one for the problem to be convex.

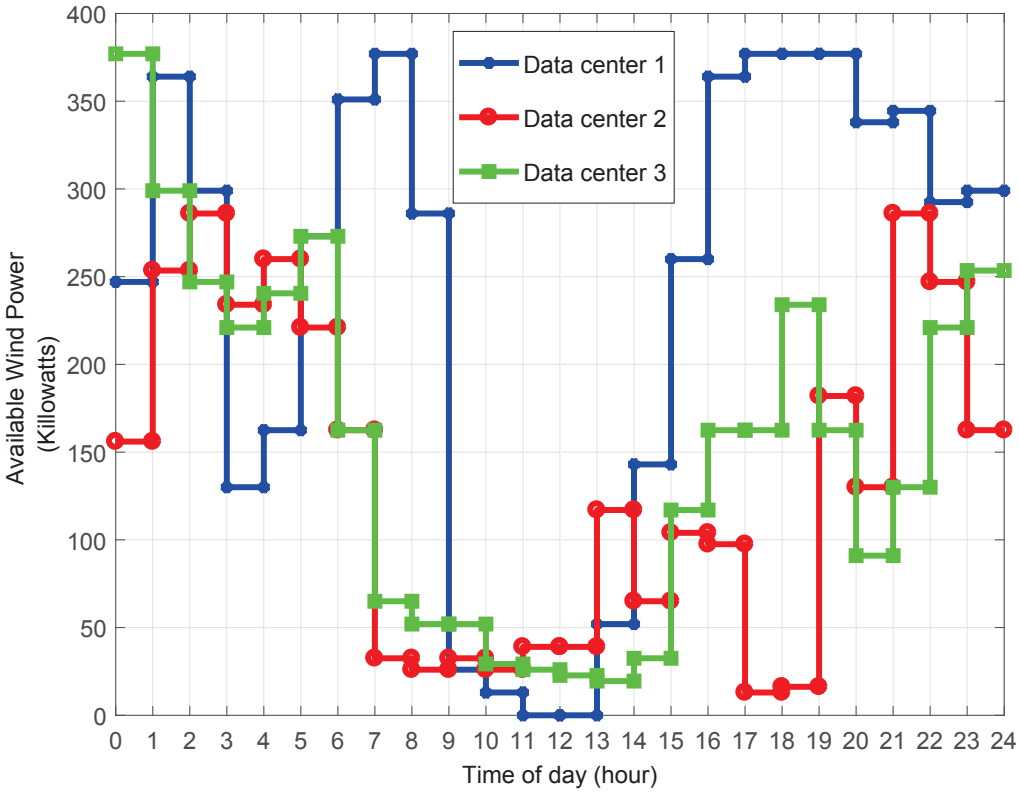


Figure 2.2 Wind power generation.

2.4 Simulation Results

We consider $|N| = 3$ data centers offering $|J| = 2$ different classes of service. Each data center is integrated with a wind farm as a renewable power source. It is assumed that the data centers are located at three different regions with deregulated electricity market. Our simulation data are based on the trends of wind power and electricity price shown in Figures 2.2 and 2.3, respectively, which are updated every hour. We simulated the total workload of two classes of service using two sample days of the requests made to the 1998 World Cup web site [4] shown in Figure 2.4. Also, for each turned on server, we have assumed $P_{peak} = 0.2$ kw, $P_{idle} = 0.1$ kw, and $E_{usage} = 1.2$.

Figure 2.5 compares the normalized profit gained by running three data centers. As shown in this figure, the curves represent the normalized profit of our proposed

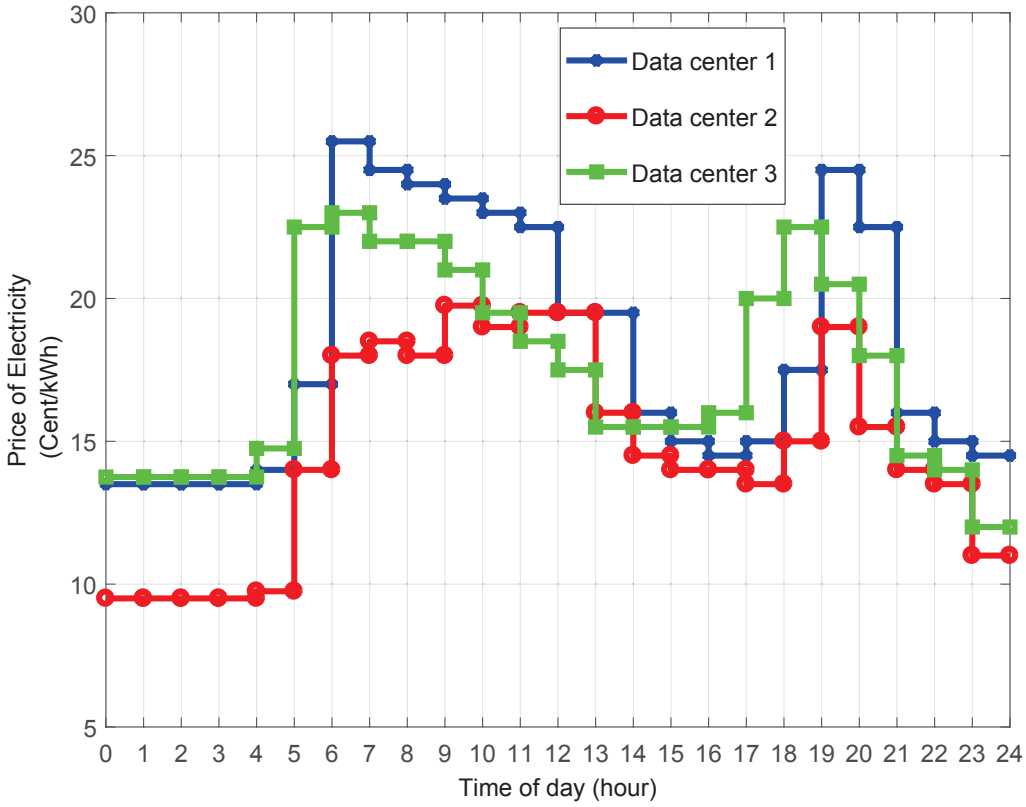


Figure 2.3 Price of electricity.

optimization problem and the design which is based on M/M/1 queueing [55]. The normalized profit gain is calculated as $(Profit - Profit_{Base})/(Profit_{Max} - Profit_{Base})$ where $Profit_{Base}$ is the profit obtained when $\mu = \lambda$ and $Profit_{Max}$ is the maximum of the profit curve obtained by simulation [30]. We can see that the proposed design outperforms the normalized profit gain of M/M/1 queueing because the G/D/1 queueing model can capture the workload distribution more accurately than M/M/1. In other words, this figure demonstrates that the gained profit of the G/D/1 queuing model is closer to the maximum profit obtained via simulations as compared to the M/M/1 queuing model.

Figure 2.6 demonstrates the better performance of our proposed design than the design in [30] adapted for the case of multiple data centers. While Figure 2.6(a) compares the gained profits of 24 hours operation of the data centers, Figure 2.6(b)

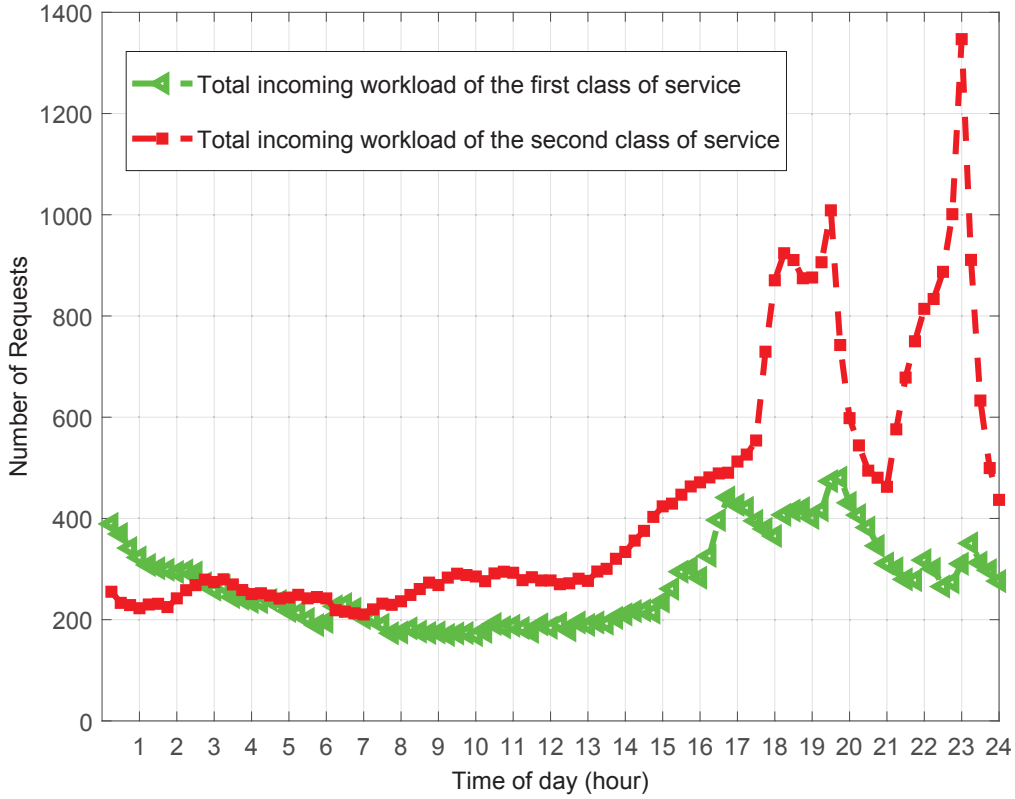


Figure 2.4 Total incoming workload.

shows the gained profit of a sample time slot versus the relative increase in green energy. As demonstrated in Figure 3.6(a), our design yields higher profit as compared to the design in [30] adapted for the case of multiple data centers that cannot fully utilize the green resources. To understand this reason, we note the result in Figure 2.6(b). As we can see in this figure, while our proposed design has a better performance for the initial available wind power, we can improve the gained profit of both designs by increasing the wind power. After a 30% increase in the wind power, our proposed design achieves its maximum profit since at this point the utilized wind power is higher than the total required power to serve all the incoming requests. Meanwhile, the design in [30] adapted for the case of multiple data centers achieves its maximum profit after a 80% increase in the wind power.

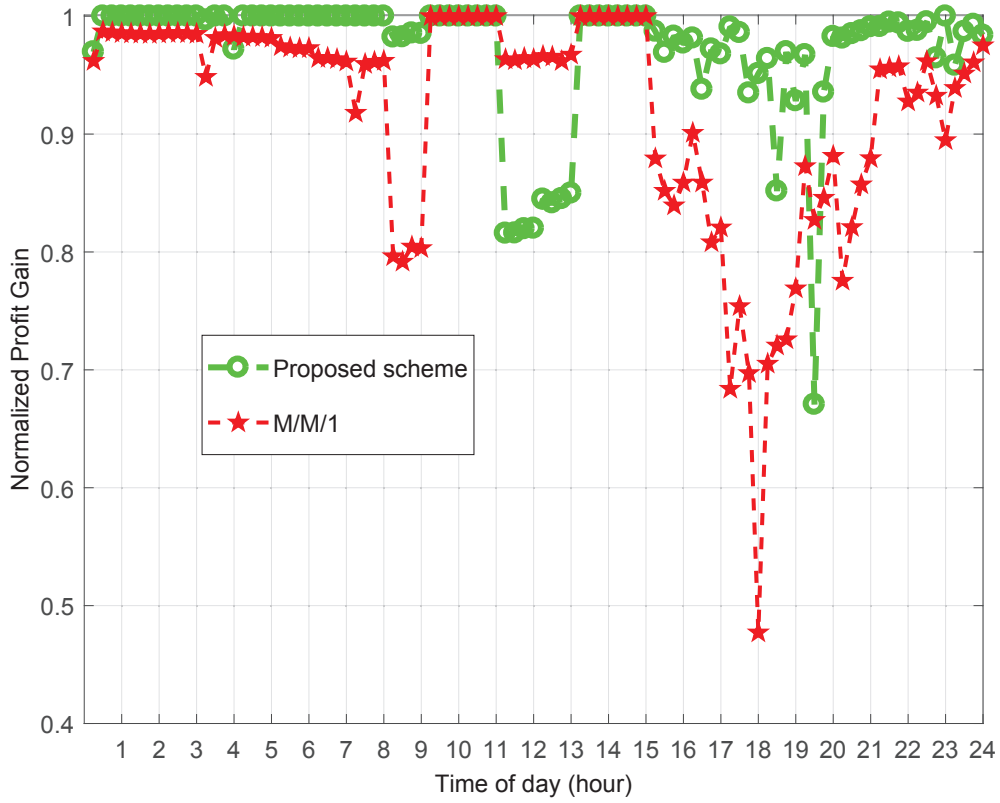


Figure 2.5 Normalized profit gain.

Figure 2.7(a) and (b) demonstrate the allocated green workloads of the first and second class of service to each data center, respectively. For example, the trend of wind power indicates that after hour 15 most of the green workload is assigned to data center 1 where the highest wind power is available. However, from hours 10 to 13, the available wind power at data center 1 is lower than the other data centers, and thus less of the green workload is allocated to this data center. Finally, Figure 2.8 shows the allocated brown workloads of the first and second class of service to each data center. For example, as shown in the Figure 2.8(a), from hours 8 to 11, all of the left over of the requests of both classes (the requests that are not served by green energy) are allocated to data center 2 where the price of electricity is the lowest. Moreover, from hours 4 to 8, the available wind power is adequate to serve all the requests of both classes of service, and the brown workload is thus not allocated to

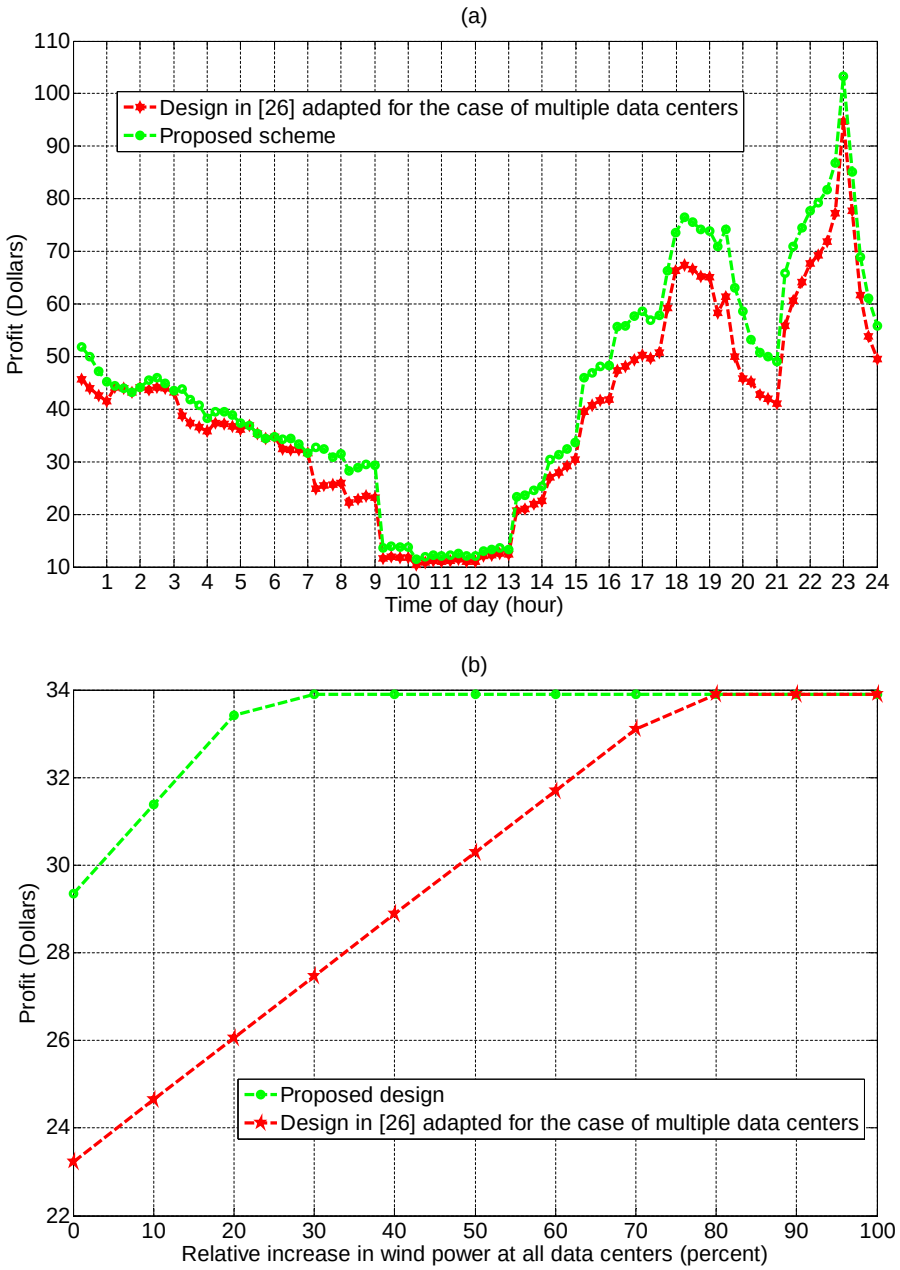


Figure 2.6 Performance comparison between the profit gain of the proposed design and design in [30] adopted for the case of multiple data centers. (a) 24 hours operation. (b) One time slot.

the data centers. In other words, from hours 4 to 8, the available wind power is the key decision factor to allocate workloads among the data centers.

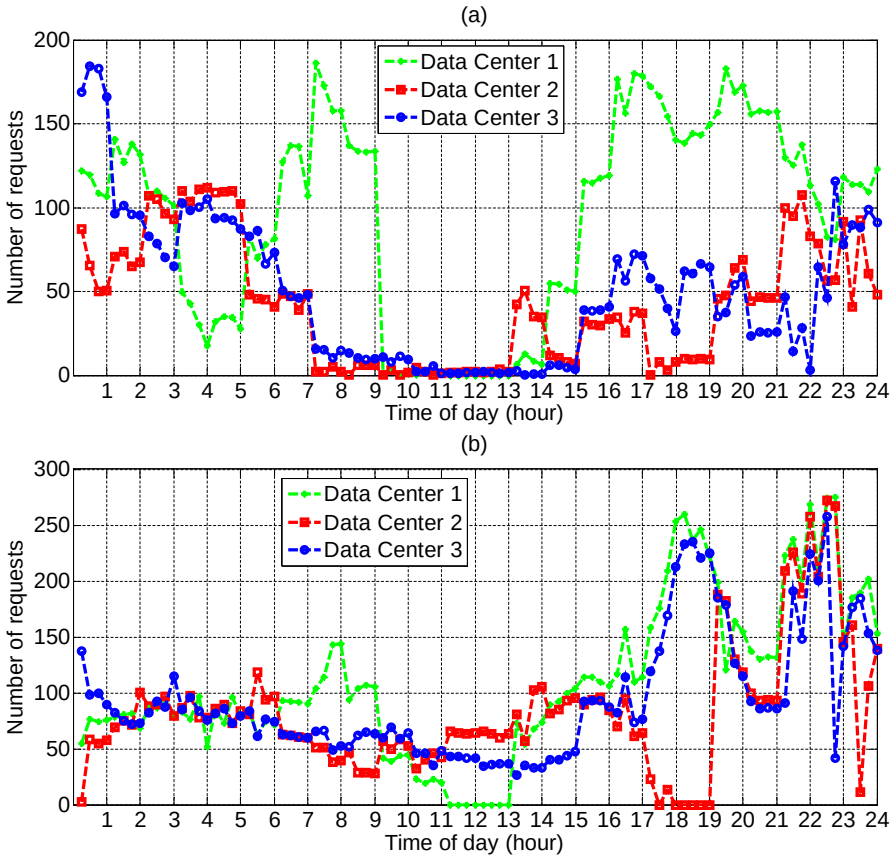


Figure 2.7 Allocated green workload to the data centers. (a) First class of service. (b) Second class of service.

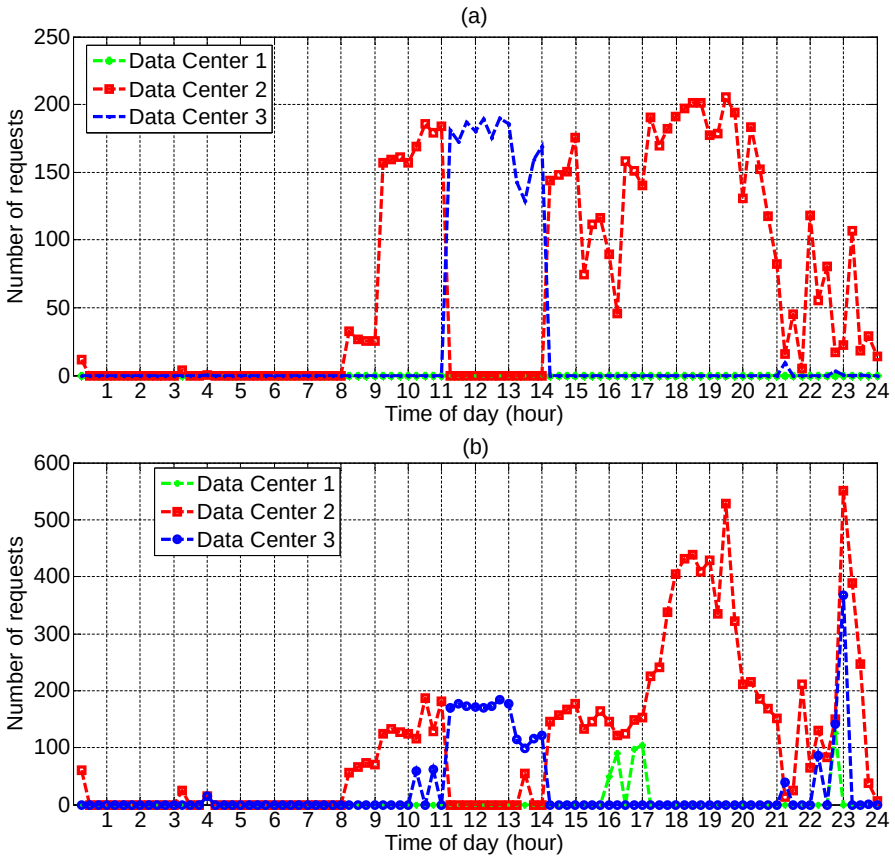


Figure 2.8 Allocated brown workload to the data centers. (a) First class of service. (b) Second class of service.

CHAPTER 3

A FUNDAMENTAL TRADEOFF BETWEEN TOTAL AND BROWN POWER CONSUMPTION IN GEOGRAPHICALLY DISPERSED DATA CENTERS

This chapter aims at deriving a fundamental tradeoff between the total and brown power consumption associated with geographical dispersed data centers, where utilizing more green energy mostly happens at the cost of increasing the total power consumption. To this end, we define a new service efficiency parameter for data centers in satisfying the QoS requirements based on the queueing analysis. More importantly, we propose the idea of modeling geo-dispersed data centers with an information flow graph to capture a total-brown power consumption tradeoff region. Accordingly, we characterize the achievable tradeoff between total and brown power consumption.

3.1 System Model and Problem Formulation

Figure 3.1 shows the proposed system model in which we consider a group of N data centers dispersed at different regions. The service requests are initiated by users and arrive at a Workload Distribution Center (WDC). One or a group of servers can serve as the workload distribution center [29]. These servers can be treated as the front-end devices that exist in multi-data center Internet services like Google and Itunes [49]. The distribution center facilitates workload flexibility at the demand side. In other words, this center inspects the arriving requests from all users and manages the distribution of the incoming workload to the geo-dispersed data centers. We divide the runtime of the data centers into a sequence of time slots at equal length, T, e.g., a few minutes. Our goal is to capture a fundamental tradeoff between the total and brown power consumption. To this end, we propose an optimization problem to be solved at the beginning of each time slot in which we update the number of the allocated requests to each data center. Note that for the analysis, we consider a single

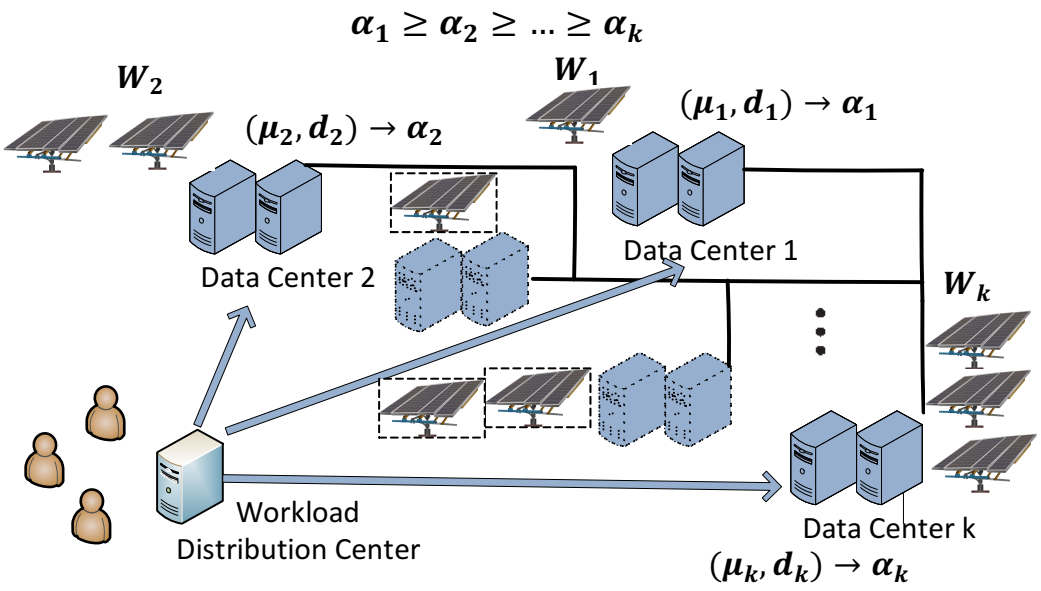


Figure 3.1 System model.

time slot, e.g., Δ as the time slot of interest, and omit the explicit time dependence in the notations.

The data centers are supplied by both on-grid and renewable types of power. The main power supply of each data center is on-grid or brown energy. To capitalize on the environmental and sustainability advantages of green energy, we also assume that each data center either is equipped with a renewable power source or has access to a nearby renewable energy source such as solar panels or a wind farm. Let W_i be the total available renewable power at data center i at the beginning of the time slot.

The allocated requests to a data center are first placed in a queue before they can be processed by any available server. We model each queue as an M/GI/1 PS queue which has been commonly adopted in modeling the waiting time of the requests at a data center in many studies like [52]. Therefore, the queuing delay at data center i can be computed as $\frac{1}{\mu_i - \frac{\lambda_i}{m_i}}$, where λ_i and μ_i are the allocated requests to data center i and the service rate of a single server at data center i, respectively. Also, m_i represents the total number of servers at data center i. The total number of servers that are turned on and run at full utilization can be computed as $m_i = \frac{P_i}{P_{peak} E_{usage}}$, where P_i is

the power consumption of data center i. P_{peak} also indicates the average peak power of a turned on server in handling a service request. Moreover, E_{usage} is the Power Usage Effectiveness (PUE) of a data center and is defined as the ratio of the data center's total power consumption to the power consumption of the servers [31, 42].

To satisfy the QoS requirements, the queueing delay for each service request should be limited by a given deadline determined by the Service Level Agreement (SLA) between the data centers and clients. Let D be the SLA deadline. Therefore, according to our queuing delay, the allocated rate to each data center is upper bounded by

$$\lambda_i \leq \frac{P_i}{P_{peak} E_{usage}} (\mu_i - \frac{1}{D - d_i}), \tag{3.1}$$

where d_i denotes the network delay from the workload distribution center to data center i. The workload distribution center sorts N data centers based on $\alpha_i \triangleq \mu_i - \frac{1}{D-d_i}$ such that $\alpha_{i-1} \geq \alpha_i$.

Denote λ_T as the total number of requests arrived at the workload distribution center at the beginning of the time slot. To capture the tradeoff between the total and brown power consumption, we model geo-dispersed data centers with an information flow graph. The information flow graph is a directed acyclic graph which includes three types of nodes: (i) a single source node (S), (ii) some intermediate nodes, and (iii) data collector nodes [27, 41]. As depicted in Figure 3.2, the workload distribution center can be thought as the source node which is the source of original requests (WDC node). Also, the intermediate nodes are data centers, and data collector node can correspond to the users that receive processed requests. The information flow graph, which models the geo-dispersed data centers, varies across time. At any given time, each node in the graph is either active or inactive. At the initial time of each time slot, the WDC node as the only active node contacts all N data center nodes and sorts them based on the service efficiency parameter, i.e., α_i. Then, it connects to a

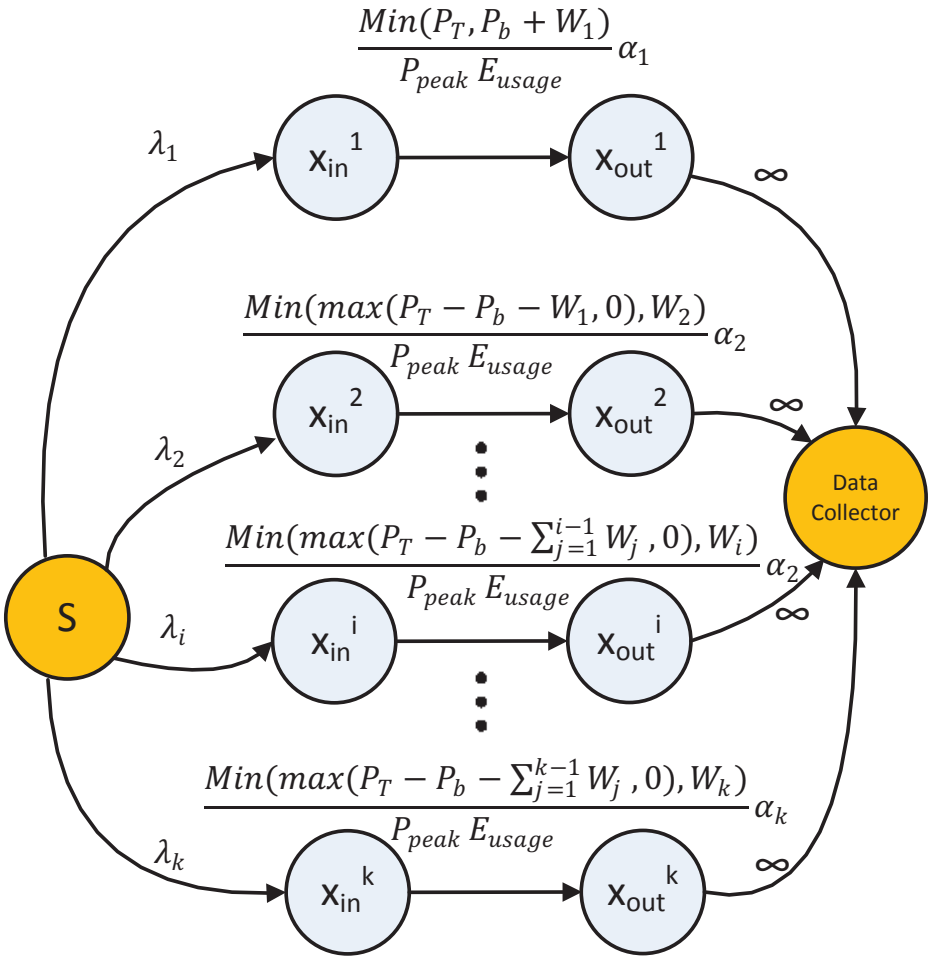


Figure 3.2 Information flow graph.

set of the first k data center nodes, i.e., $i = 1, ..., k$, with capacities of the edges equal to the allocated workloads to these nodes. It is assumed the total service provided by all the available renewable energy at these k data centers is not more than the required service to serve all the arriving requests. In fact, brown energy consumption is also required to serve all the requests and satisfy the QoS requirements. As the first data center has the highest service efficiency parameter and is assumed to have enough resources to satisfy the QoS requirements, it is more efficient to consume the brown energy only at this data center. Therefore, we have $P_1 = min(P_T, P_b + W_1)$

and $P_i = min(max(P_T - P_b - \sum_{j=1}^{i-1} W_j, 0), W_i)$ for $i = 2, ..., k$, where P_b is the brown power consumption, and P_T is assumed to be the total power consumption of all k data centers. Note that the brown power consumption depends on the number of connected data center nodes to the WDC node, i.e., k. In other words, in our model, connecting to different number of data centers will result in different amount of brown power consumption, and accordingly total power consumption. From this point onwards, WDC becomes and remains inactive, and selected data center nodes become active. Note that each data center node is represented by a pair of incoming and outgoing nodes connected by a directional edge whose capacity is the maximum number of requests that the data center can handle by the deadline. Finally, when the deadline comes, the data collector node becomes active and connects to the data center nodes to receive the processed requests. The edges that connect from the data center nodes to the data collector node are assumed to have infinite capacity, i.e., users have access to all the processed requests. In the next section, we will show how this model can capture the whole trade-off region between the total and brown power consumption.

3.2 Total-Brown Power Consumption Trade-off

In this section, we will characterize the optimal total-brown power consumption tradeoff region. As mentioned earlier, our workload allocation strategy needs k active data center nodes to connect to, and has to be designed such that the WDC node allocates $\lambda_1 = \frac{min(P_T, P_b+W_1)\alpha_1}{P_{peak}E_{usage}}$ requests to the first node and $\lambda_i = \frac{min(max(P_T-P_b-\sum_{j=1}^{i-1} W_i, 0), W_i)\alpha_i}{P_{peak}E_{usage}}$ requests to nodes $i = 2, ..., k$.

Theorem 3.2.1. *For some given (k, P_T), there exists $P_b^*(k, P_T)$ such that if $P_b \geq P_b^*(k, P_T)$, the points (k, P_b, P_T) are feasible, i.e., $P_b - P_T$ tradeoff is achievable. If $P_b \leq P_b^*(k, P_T)$, it is information theoretically impossible to serve all the arriving*

requests by the deadline. The threshold function $P_b^*(k, P_T)$ *is,*

$$\overset{\star}{P}_b(k, P_T) = \begin{cases} \frac{\lambda_T P_{peak} E_{usage} - \sum_{j=1}^{k} W_j \alpha_j}{\alpha_1}, & P_T \in \big[f(k), \infty\big) \\ \\ \frac{\lambda_T P_{peak} E_{usage} - \sum_{j=1}^{i-1} W_j(\alpha_j - \alpha_i) - P_T \alpha_i}{\alpha_1 - \alpha_i}, & P_T \in \big[f(i-1), f(i)\big), \end{cases} \tag{3.2}$$

where

$$f(i) \triangleq \frac{\lambda_T P_{peak} E_{usage} - \sum_{j=1}^{i} W_j(\alpha_j - \alpha_1)}{\alpha_1}, \tag{3.3}$$

and $i = 2, ..., k$.

Note that the tradeoff region which is verified in (5.6) has two extremal points corresponding to the minimum P_T and the minimum P_b, respectively. The point that minimizes P_T is always achieved when we send all the requests to the first data center. In (2), this point can be verified by letting $i = 2$, i.e., $(P_b, P_T) = (\frac{\lambda_T P_{peak} E_{usage} - \sum_{j=1}^{2-1} W_j(\alpha_j - \alpha_2) - P_T \alpha_2}{\alpha_1 - \alpha_2}, f(1))$. On the other hand, the point that minimizes P_b is achieved when $P_T = f(k)$, i.e., when we send the requests to all available data centers.

Proof. Consider a given information flow graph. The minimum cut is a cut between the source node (WDC node) and the data collector node in which its total sum of the edge capacities is the smallest. According to Figure 3.2, the capacity of the WDC-data collector minimum cut can be computed as

$$C = min(P_T, P_b + W_1)\frac{\alpha_1}{P_{peak} E_{usage}} + \sum_{i=2}^{k} min(max(P_T - P_b - \sum_{j=1}^{i-1} W_i, 0), W_i)\frac{\alpha_i}{P_{peak} E_{usage}}. \tag{3.4}$$

If C is larger than or equal to the total number of requests (λ_T), the data collector node can receive all the processed requests by the deadline, and so the workload distribution strategy can meet the SLA requirements. To derive the optimal tradeoff between P_b and P_T, one can fix P_T and k (to some integer values) and then find the minimum value of P_b that satisfies $C \geq \lambda_T$. To this end, we define $\overset{\star}{P}_b(k, P_T)$ as follows:

$$\begin{aligned} \overset{\star}{P}_b(k, P_T) &\triangleq \min\ P_b \\ \text{subject to}: C\ &\geq\ \lambda_T. \end{aligned} \tag{3.5}$$

Note that C is a function of P_b. Therefore, $C(P_b)$ can be computed by considering the possible intervals of P_b.

$$C(P_b)P_{peak}E_{usage} = \begin{cases} P_b\alpha_1 + \sum_{j=1}^{k} W_j\alpha_j, & P_b \in \left(0, P_T - \sum_{j=1}^{k} W_j\right] \\ P_b(\alpha_1 - \alpha_k) + P_T\alpha_k + \sum_{j=1}^{k-1} W_j(\alpha_j - \alpha_k), & P_b \in \left(P_T - \sum_{j=1}^{k} W_j, P_T - \sum_{j=1}^{k-1} W_j\right] \\ \vdots & \\ P_b(\alpha_1 - \alpha_i) + P_T\alpha_i + \sum_{j=1}^{i-1} W_j(\alpha_j - \alpha_i), & P_b \in \left(P_T - \sum_{j=1}^{i} W_j, P_T - \sum_{j=1}^{i-1} W_j\right] \\ \vdots & \\ P_b(\alpha_1 - \alpha_2) + P_T\alpha_2 + W_1(\alpha_1 - \alpha_2), & P_b \in \left(P_T - \sum_{j=1}^{2} W_j, P_T - W_1\right]. \end{cases}$$

As a result by noting $C \geq \lambda_T$ and letting $\overset{\star}{P}_b(k, P_T) = C^{-1}(\lambda_T)$, we have

$$\overset{\star}{P}_b(k, P_T) = \begin{cases} \frac{\lambda_T P_{peak}E_{usage} - \sum_{j=1}^{k} W_j\alpha_j}{\alpha_1}, & \lambda_T P_{peak}E_{usage} \in A \\ \frac{\lambda_T P_{peak}E_{usage} - \sum_{j=1}^{i-1} W_j(\alpha_j - \alpha_i) - P_T\alpha_i}{\alpha_1 - \alpha_i}, & \lambda_T P_{peak}E_{usage} \in B, \end{cases}$$

where $A \triangleq (\sum_{j=1}^{k} W_j\alpha_j, P_T\alpha_1 + \sum_{j=1}^{k} W_j(\alpha_j - \alpha_1)]$ and $B \triangleq (P_T\alpha_1 + \sum_{j=1}^{i} W_j(\alpha_j - \alpha_1), P_T\alpha_1 + \sum_{j=1}^{i-1} W_j(\alpha_j - \alpha_1)]$. By changing the conditions in the above expression from $\lambda_T P_{peak}E_{usage}$ to P_T, our tradeoff region, i.e., (5.4), is derived. □

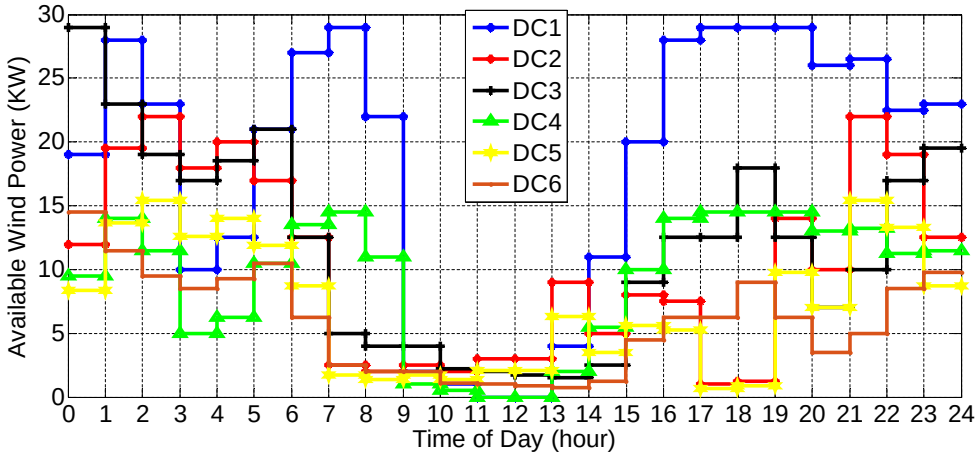


Figure 3.3 Wind power generation.

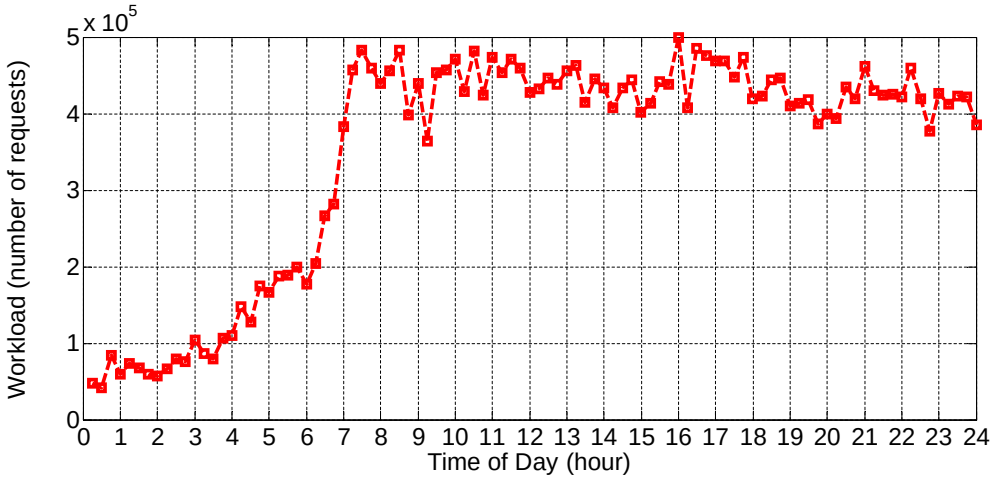


Figure 3.4 Total incoming workload.

3.3 Numerical Results

We consider $k = 6$ data centers, each integrated with a wind farm as a renewable power source. Our simulation data are based on the trends of wind power and the total workload shown in Figures. 3.3 and 3.4, respectively. Figure 3.5 shows the tradeoff curves between the total and brown power consumption for different values of D, which is the deadline to serve the requests. The tradeoff curves in this figure confirm that we can decrease brown power consumption by increasing the total power consumption. Also, the green power utilization-total power consumption tradeoff

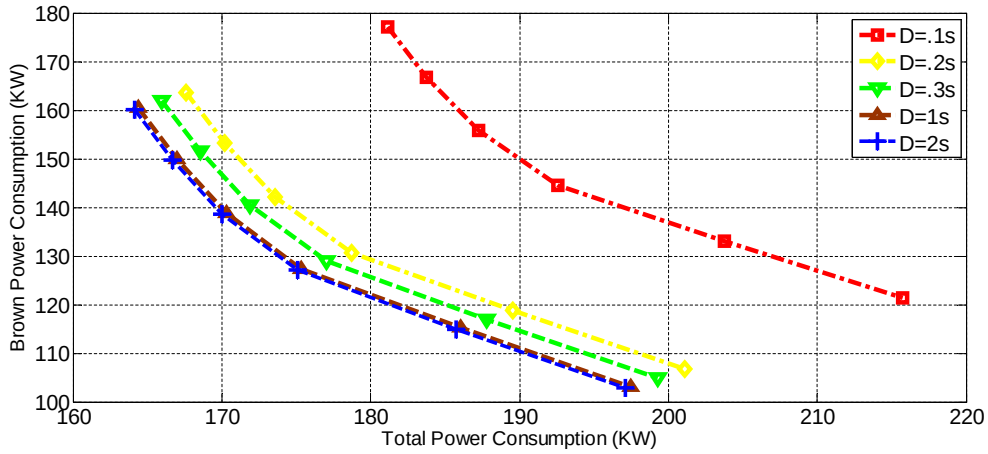


Figure 3.5 Total-brown power consumption tradeoff curves for different values of D.

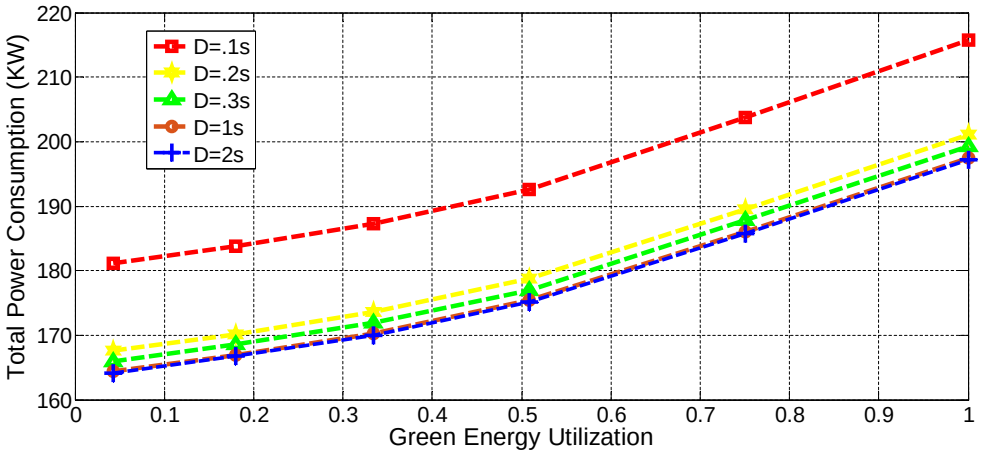


Figure 3.6 Green power utilization-total power consumption tradeoff curves for different values of D.

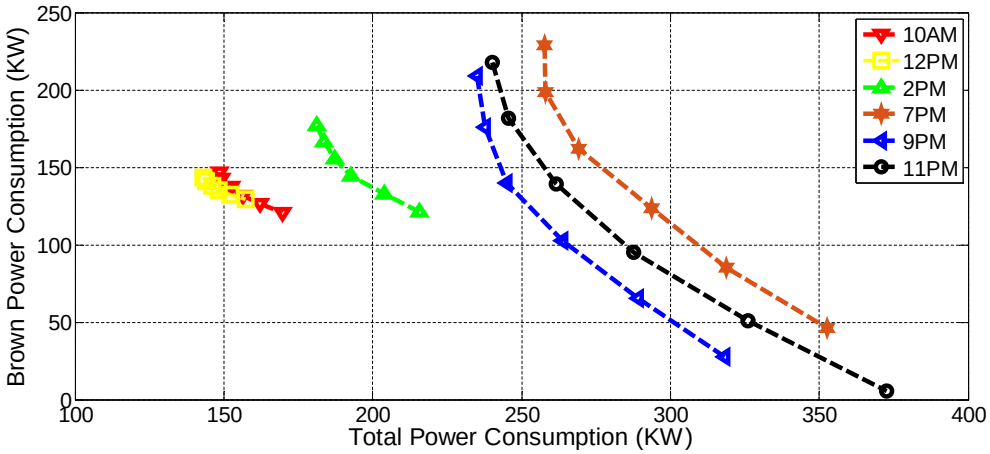


Figure 3.7 Total-brown power consumption tradeoff curves at different hours of day.

curves for different values of D are shown in Figure 3.6. The green energy utilization is defined as the consumed wind power divided by the total available wind power at 6 data centers. Figures. 3.5 and 3.6 demonstrate that the curve corresponding to the highest deadline outperforms that of the curves with lower deadline values. Finally, Figure 3.7 provides the total-brown power consumption tradeoff at some sample hours of the day when $D = .1$. As shown in Figure 3.7, for example, the tradeoff curve at hour 12PM outperforms those of the other curves due to the less number of arrival requests.

CHAPTER 4

TOWARD HIERARCHICAL EDGE COMPUTING

The multi-tiered concept of Internet of Things (IoT) devices, cloudlets and clouds is facilitating a user-centric IoT. However, in such three tier network, it is still desirable to investigate efficient strategies to offer the computing, storage and communications resources to the users. To this end, this Chapter proposes a new hierarchical model by introducing the concept of *field*, *shallow*, and *deep* cloudlets where the cloudlet tier itself is designed in three hierarchical levels based on the principle of LTE-Advanced backhaul network. Accordingly, we explore a two time scale approach in which the computing resources are offered in an auction-based profit maximization manner and then the communications resources are allocated to satisfy the users' QoS.

4.1 System Model

Figure 4.1 shows our proposed HI-MEC architecture designed for provisioning mobile edge computing services by an edge-computing service provider (a service provider in short). Based on the principles of LTE-Advanced backhaul network [57], we introduce the notion of *field*, *shallow* and *deep* cloudlets. In particular, in a HI-MEC environment, we have several field cloudlets as the resource-poor facilities co-located with Small Cell enhanced Node Bs (SCeNBs). The shallow cloudlets as the resource-middle class facilities are also hosted at the first level of aggregation nodes, i.e., at Point of Presences (PoPs). Moreover, in order to leverage the resource-rich facilities, we consider one deep cloudlet for each HI-MEC enviroment located at mobile backhual. In the proposed hierarchical model, each SCeNB is assumed to be connected to one PoP using a dedicated last mile link. Moreover, there is a dedicated aggregation link between each PoP and the aggregation node. In other words, each field cloudlet has access to only one shallow cloudlet connected via a dedicated last mile link, and

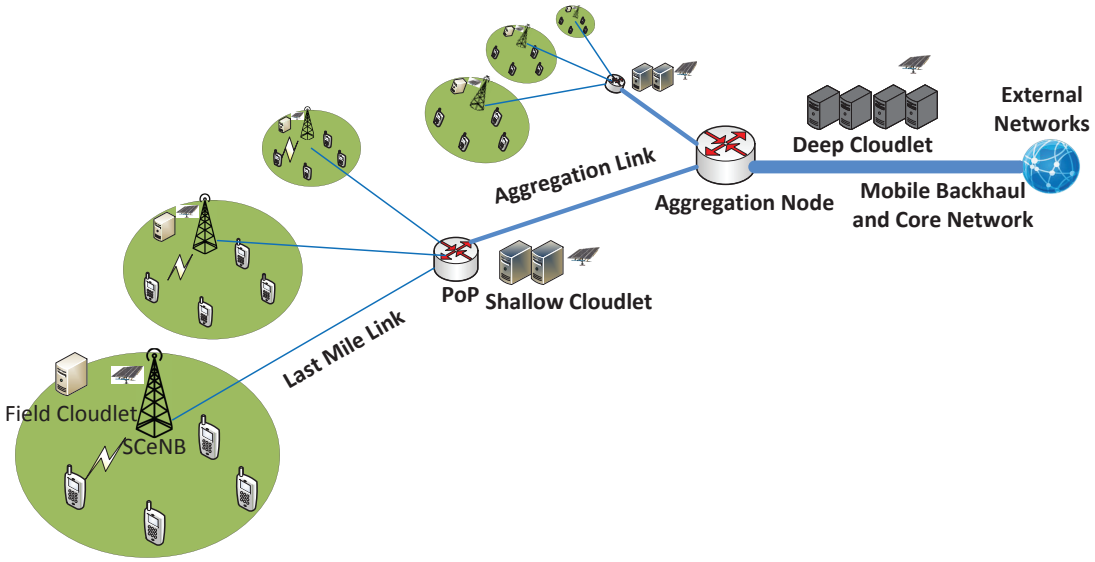


Figure 4.1 System model.

Table 4.1 Description of Symbols

Symbols	Description
	Provider Side
$\mathcal{A} \subseteq \mathbb{N}$	Set of provisioned SCeNBs as the APs
$\mathcal{C} \subseteq \mathbb{N}$	Set of all cloudlets
$\mathcal{C}_f \subseteq \mathcal{C}$	Set of field cloudlets
$\mathcal{C}_s \subseteq \mathcal{C}$	Set of shallow cloudlets
$c_d \in \mathcal{C}$	Deep cloudlet.
$\mathcal{A}_{c_s} \subseteq \mathcal{A}$	Set of APs connected to shallow cloudlet $c_s \in \mathcal{C}_s$
$\mathcal{C}_a \subseteq \mathcal{C}$	Set of cloudlet locations connected to AP $a \in \mathcal{A}$
$\mathcal{V} \subseteq \mathbb{N}$	Set of offered VMs
$\mathcal{P} \subseteq \mathbb{N}$	Set of available types of PMs
$\mathcal{P}_c \subseteq \mathcal{P}$	Set of available types of PMs at cloudlet location $c \in \mathcal{C}$
$M_c^p \subseteq \mathbb{N}$	Available number of PMs of type $p \in \mathcal{P}$ at cloudlet $c \in \mathcal{C}$
$\mathcal{R} \subseteq \mathbb{N}$	Set of resource types such as memory
D^v	Maximum allowed data transfer to/from VM type $v \in \mathcal{V}$ within a time frame
r_{min}^v	Base bandwidth of VM type $v \in \mathcal{V}$
RD_r^v	Resource demand of VM type $v \in \mathcal{V}$ for resource type $r \in \mathcal{R}$
RS_r^p	Resource supply of PM type $p \in \mathcal{P}$ for resource type $r \in \mathcal{R}$
R_a	Capacity of the last mile link between AP $a \in \mathcal{A}$ and its connected shallow cloudlet
R_{c_s}	Capacity of the aggregation link between shallow cloudlet $c_s \in \mathcal{C}_s$ and the aggregation node
R_{c_d}	Capacity of the backhual link which connects the aggregation node to the deep cloudlet
	Demand Side
$\mathcal{B}$	Set of bids submitted for all types of VMs
$\mathcal{B}_c \subseteq \mathcal{B}$	Set of $b \in \mathcal{B}$ that can be served at $c \in \mathcal{C}$
$\mathcal{B}^v \subseteq \mathcal{B}$	Set of bids submitted for VM type $v \in \mathcal{V}$
$\mathcal{B}_a \subseteq \mathcal{B}$	Set of bids submitted at AP location $a \in \mathcal{A}$
$\mathcal{B}_a^v \subseteq \mathcal{B}$	Set of bids submitted for VM type $v \in \mathcal{V}$ at AP location $a \in \mathcal{A}$
$(1, ..., \lvert\mathcal{B}_a^v\rvert)$	Sequence of bids $b \in \mathcal{B}_a^v$ in a decreasing order of the corresponding prices
a_b	AP location of $b \in \mathcal{B}$
T_b	Desired VM type of $b \in \mathcal{B}$
k_b	Rank of $b \in \mathcal{B}$ in the corresponding sequence $(1, ..., \lvert\mathcal{B}_{a_b}^{T_b}\rvert)$
$e_{k,a}^v$	Corresponding willingness price of the kth bid in $(1, ..., \lvert\mathcal{B}_a^v\rvert)$
	Profit
$x_{k,a}^v \in \{0, 1\}$	Binary decision variable that indicates whether the kth bid in sequence $(1, ..., \lvert\mathcal{B}_a^v\rvert)$ is served or not. $x_{k,a}^v = 1$ if the kth bid is served, and $x_{k,a}^v = 0$ otherwise
$y_{m,c}^p \in \{0, 1\}$	Binary decision variable that indicates whether the mth PM of type $p \in \mathcal{P}$ at cloudlet $c \in \mathcal{C}$ is on or not. $y_{m,c}^p = 1$ if the mth PM is on, and $y_{m,c}^p = 0$ otherwise
$z_{b,m,c}^p \in \{0, 1\}$	Binary decision variable that indicates the assignments of bid $b \in \mathcal{B}$ to the mth PM of type $p \in \mathcal{P}$ at cloudlet $c \in \mathcal{C}$. $z_{b,m,c}^p = 1$ if bid $b \in \mathcal{B}$ is assigned to mth PM of type $p \in \mathcal{P}$ at cloudlet $c \in \mathcal{C}$, and $z_{b,m,c}^p = 0$ otherwise
q_c	Cost of electricity at cloudlet location $c \in \mathcal{C}$
P_{idle}^p	Idle power consumption of PM $p \in \mathcal{P}$
P_{peak}^v	Average peak power consumption of a VM type $v \in \mathcal{V}$
E_{usage}	Total power consumption (including that of network facilities) divided by the power consumption at the cloudlets

all shallow cloudlets are connected to the deep cloudlet via aggregation links and mobile backhual. The main advantage of the HI-MEC architecture is to efficiently manage the fluctuations in user demands while taking into consideration of the limits in available resources at the edge. The HI-MEC network can efficiently handle the peak loads at an AP location. In other words, when the computing capacity of a field cloudlet is not enough to handle the loads from its corresponding MUs, the loads are handled by utilizing the shallow and deep computing facilities at higher levels. We assume that the network has been optimally designed in terms of the connections of the SCeNBs to the PoPs by taking into consideration of different parameters like link lengths and capacities. A list of the most symbols is summarized in Table 4.1. However, in order to ease the reading, the symbols used in Section 4.4 are not included in this table and are explained in the corresponding sections.

We consider a two time scale model in which the running time of the HI-MEC environment is divided into a sequence of time frames at equal length, T, e.g., five minutes. Each time frame itself is also divided into a sequence of time slots at equal length, τ, e.g., a few seconds. Our goal is to maximize the service provider total profit during the time frame T and minimize the total delay experienced by the users during the time slot τ. Note that for the analysis, we consider a single time frame, e.g., Δ as the time frame of interest (or a single time slot, e.g., δ as the time slot of interest) and omit the explicit time dependence in the notations through the paper.

4.1.1 Provider Side

The service provider provides the MUs (users in short) by a set of computing and communications facilities as an augmentation to their mobile device capacities. The computing facilities are provisioned as different types of Virtual Machines (VMs) running on Physical Mashines (PMs) located at different cloudlet sites. To manage the fluctuations in user demands while taking into consideration of the limitations of

available resources at the edge, the service provider should consider a flexible pricing methods in which the resources are priced according to the demands. To this end, we consider an auction-based pricing model such as Amazon's Elastic Compute Cloud (EC2) spot pricing [1, 48, 70]. In such strategy, the service provider updates the prices for each type of VM at the beginning of each time frame that depend on the available resources and demands. The minimum granularity in offering the computing resource is assumed to be one VM instance in one time frame. The service provider also renders the required communications bandwidth between the users and the VMs, i.e., the SCeNBs as the Access Points (APs) as well as the network connection between the APs and the cloudlet locations.

4.1.2 Demand Side

The service provider tenders the communications and edge-computing facilities as a service to the MUs. The MUs can benefit from the provided service, e.g., by offloading their mobile applications, and hereby prolong their device battery life-time. However, the users must submit their demand bids for the offered service stating their maximum willingness price for their desired VM type. The maximum willingness price can be decided using the spot price history. We assume that the users can submit their bids at any time but the service provider runs the auction at the beginning of each time frame in which the bids above the spot price are served, and those below the spot price are rejected. In fact, it is assumed that the demand bids are submitted based on the required VM type but the service provider will guarantee communications bandwidth for the served bids. Without loss of generality, if a user demands more than one instance of a specific type of VM type, we treat the requested instances as different bids but with the same maximum willingness price.

4.2 Problem Formulation

The service provider not only has to decide the final price, which depends on the number of served bids for each type of VM, but also has to determine the assignments of the VMs among the cloudlet locations such that the communications requirements are also guaranteed. To this end, we propose an auction-based profit maximization problem to be solved by the service provider. The profit gained by running the proposed HI-MEC environment is assumed to be given by the revenue of serving the VM demands minus the electricity cost of running the computing and network facilities, and the revenue lost due to network delay.

4.2.1 Revenue

The revenue of the service provider in a time frame depends on its decision about the spot price for each type of VM. We consider a local pricing approach in which the price for a specific type of VM varies from one AP location to another AP depending on the demand and supply but all the served bids in one AP location pay an identical price, i.e., equilibrium price per instance of a VM type. On the other hand, at each AP location, for a given type of VM, only those bids whose respective prices are greater than or equal to the equilibrium price can be served with their desired VM instances. We thus establish the revenue of the service provider in one time frame as,

$$R = \sum_{a \in \mathcal{A}} \sum_{v \in \mathcal{V}} \sum_{k=1}^{|\mathcal{B}_a^v|} x_{k,a}^v (k * e_{k,a}^v - (k-1) * e_{k-1,a}^v) \tag{4.1}$$

where we assume that the binary variables $x_{k,a}^v$ are decided such that $x_{k,a}^v \leq x_{k-1,a}^v$. In the presented definition for revenue, for example, at AP location a, the final local price for one instance of VM type v, is set to the maximum willingness price of the last served bid in sequence $(1, ..., |\mathcal{B}_a^v|)$. In other words, all the bids with willingness prices above this bid are served, and on the other hand, all the bids with willingness

prices below this bid are rejected. The total revenue is thus calculated by summing over all the bids in sequence $(1, ..., |\mathcal{B}_a^v|)$ with consideration of their willingness prices $(e_{k,a}^v)$. Going from the $(k-1)$th bid to the kth bid, if $(x_{k,a}^v = 1)$, the new revenue, $k * e_{k,a}^v$, is added to the summation and the previous revenue, $(k-1) * e_{k-1,a}^v$, is deducted from the summation.

4.2.2 Electricity Cost

The electricity cost of the service provider depends on different variables like the number of turned on PMs at each cloudlet and the distribution of the VMs among the PMs. Following the power consumption model adapted for data centers [15,30,43,45], the total electricity cost (EC) in one time frame can be computed as,

$$EC = TE_{usage}(\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}_{a_b}}\sum_{p\in\mathcal{P}_c}\sum_{m=1}^{min(|\mathcal{B}|,M_c^p)} q_c z_{b,m,c}^p P_{peak}^{T_b} + \sum_{c\in\mathcal{C}}\sum_{p\in\mathcal{P}_c}\sum_{m=1}^{min(|\mathcal{B}|,M_c^p)} q_c y_{m,c}^p P_{idle}^p) \tag{4.2}$$

where the first term corresponds to the electricity cost of VMs' power consumption and the second term is to consider the related cost of PMs' idle power consumption. In fact, we take into consideration of both a fixed electricity cost which is due to the idle power consumption of a PM and a variable electricity cost which is attributed to the extra power consumption of the VMs running on that PM. Moreover, we incorporate the Power Usage Effectiveness (PUE) ratio, Eusage, to amalgamate the power consumption at the network facilities.

4.2.3 Lost Revenue

The proposed architecture is a MEC architecture where the users expect to experience a low latency connecting to their VMs. Therefore, for QoS satisfaction, we incorporate a lost revenue into our profit maximization problem due to the network delay

experienced by the users. The idea is to first serve the bids as close as possible to the edge, and then allocate bandwidth to those bids that have to be served at a shallow/deep cloudlet due to high demands at their corresponding AP locations. In other words, field cloudlets have to be the first priority to serve a bid while shallow and deep cloudlet facilities have the second and third priorities, respectively.

Let r_b be the bandwidth allocated to bid b on all the links that it has to go through. For example, if bid b is served at the deep cloudlet, r_b is allocated to bid b on all corresponding last mile, aggregation and mobile bakchual links. In other words, there is a dedicated link of capacity r_b between the corresponding AP of bid b and its assigned cloudlet location. Since the users are interested in their QoS, rather than their allocated bandwidth, we translate the allocated bandwidth to our lost revenue.

In a nutshell, at any time $t \in T$, we denote the traffic load of a given bid b on its dedicated link, i.e., r_b, by $A_b(t)$. Therefore, within interval T , bid b makes its dedicated link busy for $\frac{\int_0^T A_b(t)dt}{r_b}$ seconds. Thus, the link utilization for bid b is $\frac{\int_0^T A_b(t)dt}{Tr_b}$. Here, the network delay is related to the link utilization such that the less time is the link busy, the less network delay is experienced. The total traffic load of a bid within a time slot is upper bounded by its maximum data transfer to/from the VM, i.e., $\int_0^T A_b(t)dt \leq D^{T_b}$. Moreover, we assume that the allocated bandwidth of each bid is lower bounded by the base bandwidth of its VM type, i.e., $r_b \geq r_{min}^{T_b}$. Therefore, the link utilization of a bid is upper bounded with its maximum data transfer as well as the base bandwidth as follows,

$$\frac{\int_0^T A_b(t)dt}{Tr_b} \leq \frac{D^{T_b}}{Tr_{min}^{T_b}} \tag{4.3}$$

The idea is to incorporate this upper bound into our profit maximization which is solved every time frame and then update the bandwidth allocated to the bids every

time slot based on the traffic loads. We thus define our lost revenue as,

$$LR = \sum_{a \in \mathcal{A}} \sum_{b \in \mathcal{B}_a} \sum_{c \in \mathcal{C}_a \setminus \mathcal{C}_f} \sum_{p \in \mathcal{P}} \sum_{m \in M_c^p} \xi_{a,c} \frac{z_{b,m,c}^p D^{T_b}}{Tr_{min}^{T_b}} \tag{4.4}$$

where $\xi_{a,c}$ are the coefficients set by the service provider based on the importance of QoS compared to the profit and by taking into consideration of the link lengths between APs and their connected cloudlets. Moreover, the reason behind using the upper bound is to derive a QoS satisfaction which is VM type oriented.

4.3 Profit Maximization

Note that users can submit or cancel their bids or change their willingness prices. The AP location of a user changes when she moves to other location. Therefore, the service provider must update its decision on serving the bids periodically. To this end, we propose to maximize the auction-based profit at the beginning of each time frame.

4.3.1 Binary Linear Programming

The proposed optimization problem is formulated as,

$$\underset{x_{k,a}^v,\ y_{m,c}^p,\ z_{b,m,c}^p}{\text{maximize}} \quad (R - EC - LR)$$

$$\text{C1} : \sum_{c \in \mathcal{C}_{a_b}} \sum_{p \in \mathcal{P}_c} \sum_{m=1}^{min(|\mathcal{B}|, M_c^p)} z_{b,m,c}^p = x_{k_b,a_b}^{T_b} \ \forall b \in \mathcal{B}$$

$$\text{C2} : \sum_{b \in \mathcal{B}} z_{b,m,c}^p RD_r^{T_b} \leq RS_r^p y_{m,c}^p \ \forall p, m, c, r$$

$$\text{C3} : \sum_{b \in \mathcal{B}_a} \sum_{c \in \mathcal{C}_a \setminus \mathcal{C}_f} \sum_{p \in \mathcal{P}_c} \sum_{m=1}^{min(|\mathcal{B}_a|, M_c^p)} z_{b,m,c}^p r_{min}^{T_b} \leq R_a \ \forall a$$

$$\text{C4}: \sum_{a \in \mathcal{A}_{c_s}} \sum_{b \in \mathcal{B}_a} \sum_{p \in \mathcal{P}_{c_d}} \sum_{m=1}^{min(|\mathcal{B}_a|, M_{c_d}^p)} z_{b,m,c_d}^p r_{min}^{T_b} \leq R_{c_s} \ \forall c_s$$

$$\text{C5}: \sum_{b \in \mathcal{B}} \sum_{p \in \mathcal{P}_{c_d}} \sum_{m=1}^{min(|\mathcal{B}|, M_{c_d}^p)} z_{b,m,c_d} r_{min}^{T_b} \leq \ R_{c_d}$$

$$\text{C6}: x_{k,a}^v \leq x_{k-1,a}^v \ \forall v, a, 2 \leq k \leq |\mathcal{B}_a^v|$$

$$\text{C7}: y_{m,c}^p \leq y_{m-1,c}^p \ \forall c, p, 2 \leq m \leq M_c^p$$

$$\text{C8}: x_{k,a}^v \in \{0,1\} \ \forall v, a, 1 \leq k \leq |\mathcal{B}_a^v|$$

$$\text{C9}: z_{b,m,c}^p \in \{0,1\} \ \forall b, m, c, p$$

$$\text{C10}: y_{m,c}^p \in \{0,1\} \ \forall m, c, p \tag{4.5}$$

where the objective is to maximize the profit defined as the $Revenue - ElectricityCost - LostRevenue$. The equality constraint C1 in (5.6) is to ensure that the served bids are assigned to a PM at a cloudlet location connected to their AP locations. Inequality constraint C2 is also to lower bound the total resource demands of all the bids assigned to a PM by the resource supply of that machine. In addition, we use inequality constraints C3, C4 and C5 to bound the total minimum bandwidth of the bids traversing a link by the bandwidth capacity of that link. Note that C3, C4 and C5 are formulated for the last mile, aggregation and backhual links, respectively. Moreover, by inequality constraints C6, we enforce the requirement of our defined

revenue function. Constraint C7 is designed to give priority to the PMs with lower running index at one cloudlet location over those with higher index at the same location. Finally, constraints C8, C9 and C10 are to restrict our variables to the binary choices. The computational complexity of the proposed BLP is exponential and corresponds to $\mathcal{O}(2^{|\mathcal{B}|^2*|\mathcal{P}|*|\mathcal{C}|})$.

4.3.2 Heuristics

While the proposed BLP optimization model offers flexibility, finding an optimal solution presents computational complexity. The complexity grows fast with the number of bids and PMs. In order to obtain high quality solutions in a reasonable time, we propose two heuristic algorithms that employ VM pricing and VM distribution techniques [48]. The pseudo codes for VM pricing and VM distribution algorithms are shown in Algorithms 1 and 2, respectively. In fact, we follow a two phases approach.

In the first phase (Algorithm 1), for each type of VM at each AP location, we first estimate the serving cost of one VM instance, i.e., φ_a^v, by taking a weighted average over all suitable type of PMs across all connected field, shallow, and deep cloudlets to that location. In our cost estimation, we consider both electricity cost and the lost revenue (lines 2-19). We then identify the favorable number of the bids to be served, i.e., $\hat{k}_a^v$ and the final local price, i.e., ω_a^v such that the estimated profit is maximized (lines 21-26). Finally, for each AP a and VM type v, we store all those bids with a rank less than or equal to $\hat{k}_a^v$ in the set of served bids, i.e., S (line 27).

In the VM distribution phase (Algorithm 2), we first initialize an instance count m_p^c for each type of PM at each AP location (lines 1-5). We then search the set of all the available PMs and the cloudlet locations to find a favorite PM, i.e., $\hat{p}$, at a favorite cloudlet, i.e., $\hat{c}$. For a given instance of a PM type at a given cloudlet, we scan the set of all the served bids and create a packing list for that machine, i.e., L_c^p. The

packing list for a PM is created based on its resource constraints and the possibility of serving a bid at that PM. We subsequently compute the utility function for each PM at each cloudlet location, i.e., u_c^p. Accordingly, both the favorite PM type and cloudlet location are identified by comparing all the utility functions (lines 8-25), and all the bids in the corresponding packing list, i.e, $L_{\hat{c}}^{\hat{p}}$, are assigned to one instance of $\hat{p}$ at $\hat{c}$ (lines 26-32). Finally, the assigned bids are removed from the set of served bids and this process is repeated until all the served bids are assigned or no suitable PM and cloudlet location is found for the VM assignment (lines 33-34). The complexity of the VM distribution presented in Algorithm 2 corresponds to $\mathcal{O}(|\mathcal{B}|^2 * |\mathcal{P}| * |\mathcal{C}|)$.

Algorithm 1 VM pricing

1: $S \leftarrow \emptyset$
2: **for all** $v \in \mathcal{V}$ **do**
3: **for all** $a \in \mathcal{A}$ **do**
4: $g_a^v \leftarrow 0, EC_a^v \leftarrow 0$
5: **for all** $c \in \mathcal{C}_a$ **do**
6: **for all** $p \in \mathcal{P}_c$ **do**
7: **if** $p.canHost(v) = true$ **then**
8: $g_{a,c}^p \leftarrow min(M_c^p, |\mathcal{B}_a^v|)$
9: **if** $c \in C_a \setminus C_f$ **then**
10: $g_{a,c}^p \leftarrow min(g_{a,c}^p, \frac{R_a}{r_{min}^v})$
11: **end if**
12: $g_a^v \leftarrow g_a^v + g_{a,c}^p$
13: $f_{a,c}^p \leftarrow Tq_c(p_{peak}^v + \frac{p_{idle}^p}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \frac{RD_r^v}{RS_r^p}) + \xi_{a,c} \frac{D^v}{r_{min}^v}$
14: $\varphi_a^v = \varphi_a^v + g_{a,c}^p * f_{a,c}^p$
15: **end if**
16: **end for**
17: **end for**
18: **if** $g_a^v > 0$ **then**
19: $\varphi_a^v \leftarrow \frac{\varphi_a^v}{g_a^v}$
20: $\hat{\rho}_a^v \leftarrow 0, \hat{k}_a^v = 0, \omega_a^v \leftarrow 0$
21: **for** $k = 1 \rightarrow |B_a^v|$ **do**
22: $\rho_a^v \leftarrow k * (e_{k,a}^v - \varphi_a^v)$
23: **if** $\rho_a^v \geq \hat{\rho}_a^v$ **then**
24: $\hat{\rho}_a^v \leftarrow \rho_a^v, \omega_a^v \leftarrow e_{k,a}^v, \hat{k}_a^v = k$
25: **end if**
26: **end for**
27: $S \leftarrow S \cup \{b \in \mathcal{B}_a^v \mid k_b <= \hat{k}_a^v\}$
28: **end if**
29: **end for**
30: **end for**

Algorithm 2 VM distribution

for all $c \in \mathcal{C}$ **do**
  for all $p \in \mathcal{P}_c$ **do**
    $m_c^p \leftarrow 0$
  end for
end for
repeat
  $\hat{p} \leftarrow \emptyset, \hat{c} \leftarrow \emptyset, \hat{u} \leftarrow 0$
  for all $p \in \mathcal{P}$ **do**
    for all $c \in \mathcal{C}$ **do**
      if $p \in \mathcal{P}_c$ **then**
        if $m_c^p < M_c^p$ **then**
          $L_c^p \leftarrow \emptyset$
          for all $b \in S \cap \mathcal{B}_c$ **do**
            if $c.canHost(a_b) \vee m_c^p.canHost(L_c^p \cup b) = true$ **then**
              $L_c^p \leftarrow L_c^p \cup b$
            end if
          end for
          $u_c^p \leftarrow \frac{\sum_{b \in L_c^p} \omega_b^{T_b}}{q_c(p_{idle}^p + \sum_{b \in L_c^p} p_{peak}^{T_b}) + \sum_{b \in L_c^p} \frac{\xi_{a_b,c} D^{T_b}}{Tr_{min}^{T_b}}}$
          if $u_c^p > \hat{u}$ **then**
            $\hat{u} \leftarrow u_c^p, \hat{p} \leftarrow p, \hat{c} \leftarrow c$
          end if
        end if
      end if
    end for
  end for
  if $\hat{p} \neq \emptyset$ **then**
    $y^{\hat{p}}_{m^{\hat{p}}_{\hat{c}}, \hat{c}} \leftarrow 1$
    $m^{\hat{p}}_{\hat{c}} \leftarrow m^{\hat{p}}_{\hat{c}} + 1$
    for all $b \in L^{\hat{p}}_{\hat{c}}$ **do**
      $z^{\hat{p}}_{b, m^{\hat{p}}_{\hat{c}}, \hat{c}} \leftarrow 1$ and *update the capacity of all links between* a_b *and* $\hat{c}$ *according to* $r_{min}^{T_b}$
    end for
  end if
  $S \leftarrow S \setminus L^{\hat{p}}_{\hat{c}}$
until $S = 0 \vee \hat{p} = \emptyset$

4.4 Bandwidth Allocation

Based on the VM assignment in the previous section, we now define an optimization problem to find the optimal bandwidth allocated to each served bid, i.e., r_b. Our goal is to minimize the total network delay experienced by the served users on the link between their corresponding APs and cloudlets. Note that the delay between a user and AP which is related to the radio resource allocation is not the focus of this paper since it has already been addressed in other studies such as [8]. Let $\{1, ..., N\}$ be the set of all bids served at a shallow or deep cloudlet. a_b and c_b are also the corresponding AP and cloudlet locations of bid b, respectively. Moreover, we define $\{1, ..., M\}$ as the set of all the links in our HI-MEC environment including all the last mile, aggregation links and the mobile backhual link. Let v_{mb} be a binary variable such that $v_{mb} = 1$ if the traffic load of bid b traverses link m.

4.4.1 Convex Optimization

We propose to solve the bandwidth allocation problem shown in (5.12) at the beginning of the time slot of interest. The objective of this optimization problem is to minimize the total delay experienced by the users who have been served at shallow cloudlets or the deep cloudlet location, by taking into consideration of the traffic load of each user at the beginning of the time slot of interest, i.e., λ_b. Constraints C1 and C2 are to bound the bandwidth allocated to bid b by the the lower and upper boundary values l_b and u_b, respectively. Note that these values are positive and decided by the service provider for example based on the VM types and the traffic loads. The lower bound l_b is also lower bounded by the base bandwidth considered during the auction, i.e., $l_b \geq r_{min}^{T_b}$. Moreover, constraint C3 is to bound the bandwidth allocation by the physical bandwidth capacity of the links. In fact, the total bandwidth allocated to the bids traversing link m is upper bounded by its capacity, i.e., R_m.

$$\underset{r_b}{\text{minimize}} \sum_{b=1}^{N} \xi_{a_b,c_b} \frac{\lambda_b}{r_b}$$

$$\text{C1} : r_b \geq l_b \ \forall b \in 1, ..., N$$

$$\text{C2} : r_b \leq u_b \ \forall b \in 1, ..., N$$

$$\text{C3} : \sum_{b=1}^{N} v_{mb} r_b \leq R_m \ \forall m \in 1, ..., M \tag{4.6}$$

4.4.2 Centralized Optimal Solution

The proposed bandwidth allocation problem is a convex optimization with $2N + M$ constraints. The complexity of this problem may increase as the numbers of the served bids and the links increase. However, a HI-MEC network is assumed to be limited by the number of the links and the computing capacity to serve as few as several thousand bids. Therefore, it is desirable to derive a centralized optimal solution for this problem. To this end, we define the matrix $V = (v_{mb})_{M \times N}$ to show the traverse of the bids on each link based on our already defined binary variable v_{mb}. Let $R = (R_1, ..., R_M)$ and $r = (r_1, ..., r_N)$ also be the vectors of the capacity of the links and the bandwidth allocated to the bids, respectively. To derive the optimal solution, we apply the method of Lagrange multipliers since the constraints of Problem (5.12) are linear, and the Kuhn-Tucher conditions are necessary and sufficient for an existing optimal solution [14, 33].

Theorem 4.4.1. *There exists* $\gamma_m \geq 0$ *(*$m \in 1, ..., M$*) such that* $\forall b \in 1, ..., N$*:*

$$\hat{r_b} = \sqrt{\frac{\xi_{a_b,c_b} \lambda_b}{\sum_{m=1}^{M} \gamma_m v_{mb}}} \tag{4.7}$$

$$l_b \leq r_b \leq u_b,$$

and $\forall m \in 1, ..., M$

$$\gamma_m((V.r)_m - R_m) = 0 \tag{4.8}$$

where $\hat{r_b}$ *is the optimal solution for Problem (5.12).*

Proof. Our proof is based on the assumption that the bandwidth allocation space of Problem (5.12) is a nonempty, convex and compact set and thus our objective function is strictly convex with respect to r_b. Then, we define $\alpha_b \geq 0$ and $\beta_b \geq 0$ $\forall b \in 1, ..., N$ as well as $\gamma_m \geq 0$ $\forall m \in 1, ..., M$ as the Lagrange multipliers for constraints C1, C2 and C3 in problem (5.12), respectively. Therefore, the Lagrangian becomes,

$$\mathcal{L}(r, \alpha, \beta, \gamma) = \sum_{b=1}^{N} \xi_{a_b, c_b} \frac{\lambda_b}{r_b} + \sum_{b=1}^{N} \alpha_b(l_b - r_b)$$
$$+ \sum_{b=1}^{N} \beta_b(r_b - u_b) + \sum_{m=1}^{M} \gamma_m((V.r)_m - (R)_m). \tag{4.9}$$

To optimize the objective by applying the necessary and sufficient conditions, we have

$$\Delta\mathcal{L}(\hat{r}, \alpha, \beta, \gamma) = 0 \Leftrightarrow$$

$$-\xi_{a_b, c_b} \frac{\lambda_b}{{r_b}^2} - \alpha_b + \beta_b + \sum_{m=1}^{M} \gamma_m v_{mb} = 0 \ \forall b \in 1, ..., N \tag{4.10}$$

and

$$\alpha_b(l_b - \hat{r_b}) = 0 \ \forall b \in 1, ..., N,$$
$$\beta_b(\hat{r_b} - u_b) = 0 \ \forall b \in 1, ..., N,$$
$$\gamma_m((V.r)_m - (R)_m) = 0 \ \forall m \in 1, ..., M, \tag{4.11}$$

where $\hat{r} = (\hat{r_1}, ..., \hat{r_N})$ is the optimal solution to Problem (5.12). Noting the values of the Lagrange multipliers in (6.2) and focusing on the general case when $l_b < r_b < u_b$,

Table 4.2 Computation Times Comparison Between Heuristic and Optimal.

	50 (bids)	100 (bids)	1000 (bids)	2000 (bids)
Heuristic case 1	0.052 (s)	0.79 (s)	1.89 (s)	5.55 (s)
Optimal case 1	2.17 (s)	76.32 (s)	107.65 (s)	458.86 (s)
Heuristic case 2	0.31 (s)	0.94 (s)	2.716(s)	5.75 (s)
Optimal case 2	2.53 (s)	31.99 (s)	97.34 (s)	570.53 (s)

one can conclude $\alpha_b = 0$ and $\beta_b = 0$. In fact, we are not interested in special cases when r_b is equal to the boundary values. Therefore, by solving (5.21) for $\alpha_b = 0$ and $\beta_b = 0$, $\hat{r_b}$ is derived and the proof is complete. □

The result of Theorem 4.4.1 indicates that the optimal bandwidth for each bid can be achieved by the optimal multipliers of its associated links. For example, when a bid is served at the deep cloudlet, its optimal bandwidth can be solved by the optimal multipliers of its associated last mile and aggregation links as well as the mobile backhual link. Therefore, solving this problem in a distributed manner for the case that the numbers of bids and the links scale up can be investigated in a future work.

4.5 Simulation Results

In this section, we compare the results of the heuristic VM pricing and VM distribution algorithms with the optimal results in solving the proposed profit maximization problem (BLP). We consider a HI-MEC environment consisting of five AP locations, each co-located with a field cloudlet, and two PoPs, each equipped with a shallow cloudlet in which APs 1, 2, and 3 are connected to the first PoP, and APs 4 and 5 to the second PoP. The network model is also assumed to have a deep cloudlet. We fix the bandwidth capacity of all the links to 1Gbps. Moreover, we consider three types of VMs (m3 large, c3 xlarge, and r3 2xlarge) and three types of resources (CPU, memory, and storage) [1]. The cloudlets are assumed to be equipped with the same type of PM but different numbers of PMs are available at different hierarchical levels.

The power consumption of a PM is set to 0.7kWh and the power consumption of each type of VM is estimated accordingly based on its resource demands and the resource supply of the PM. The price of electricity is fixed to 2 cent/kWh. The price of the bids are generated randomly using a triangle distribution [48] assuming that the submitted price for each type of VM will not exceed its on-demand price available at [1].

CVX [2] combined with Gurobi [3] and MATLAB are used to simulate the BLP and the two phases heuristic approach. For performance evaluations, we study two cases, each with four different scenarios, i.e., 50, 500, 1000 and 2000 bids. In the first case study, we fix the ratio of bids submitted for three types of VMs as m3:c3:r3=2.5:1.5:1, corresponding to the case that the users are more interested in a smaller type of VM, i.e., m3. On the other hand, for the second case study, we change the ratio to m3:c3:r3=1:1.5:2:5 assuming that the users are more interested in a larger type of VM, i.e., r3. The AP locations for the bids are generated randomly in each case.

The computation time of the optimal approach (BLP) and the heuristic algorithm for different scenarios are compared in Table 4.2. While the heuristic algorithm provides the suboptimal solution within a few seconds, the computation time of the optimal approach grows fast with the number of bids. The reason is in accordance with our qualitative discussion of the complexities of the BLP and VM distribution algorithm in which the former grows exponentially with the number of bids and the latter is polynomial.

Figures 4.2 and 4.3 show the profits gained in one time frame for case 1 and case 2, respectively. The corresponding approximate ratios of the heuristic algorithm in Figure 4.2 are 0.989, 0.991, 0.987 and 0.982 for 50, 500, 1000 and 2000 bids, respectively. The ratios in Figure 4.3 also equal to 0.995, 0.995, 0.965 and 0.961 for 50, 500, 1000 and 2000 bids, respectively. As we can see in these figures, the heuristic

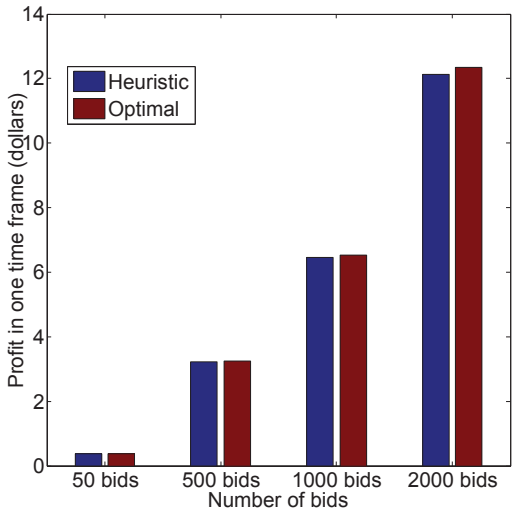


Figure 4.2 Profit comparison between heuristic and optimal approaches for case 1.

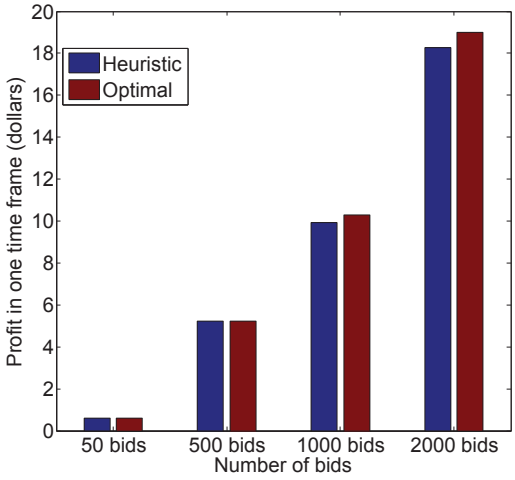


Figure 4.3 Profit comparison between heuristic and optimal approaches for case 2.

algorithm results in a profit quite close to the profit of the optimal approach. To understand the reason of this observation, we should analyze the performance of the heuristic approach in terms of the number of the served bids as well as the VM pricing. To this end, we compare the performance of the heuristic and optimal approaches by providing the ratio of the served bids in Figures 4.4 and 4.5 for case 1 and 2, respectively. Here, the ratio of the served bids is defined as the total number of served bids divided by the total number of submitted bids. As demonstrated in these

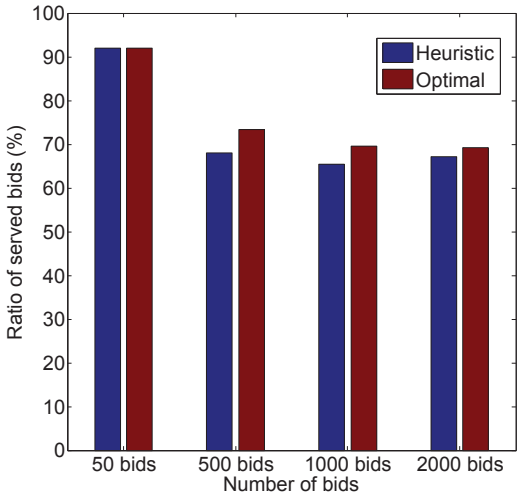


Figure 4.4 Ratios between the served bids and the total bids for case 1.

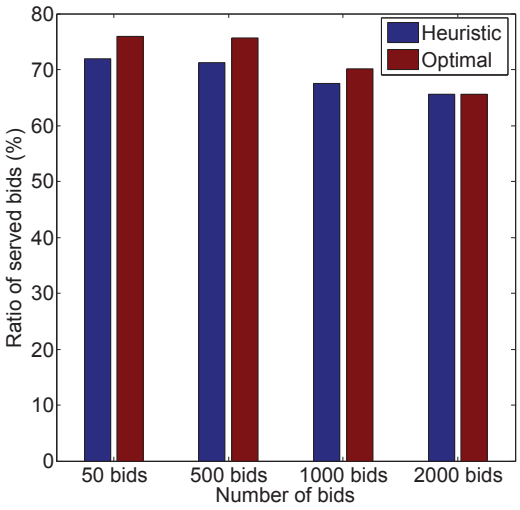


Figure 4.5 Ratios between the served bids and the total bids for case 2.

figures, the heuristic approach serves nearly the same number of bids as the optimal approach. We validate the performance of the VM pricing algorithm in Figure 4.6. Owing to similarity, we only compare two prices as examples, and we choose m3 for case 1 and r3 for case 2 since m3 and r3 are the most demanded VMs in case 1 and 2, respectively. As demonstrated in the figure, the estimated price of the heuristic VM pricing for most scenarios is slightly higher than the optimal price. This result is due to the reason that the heuristic VM pricing algorithm serves fewer bids than the

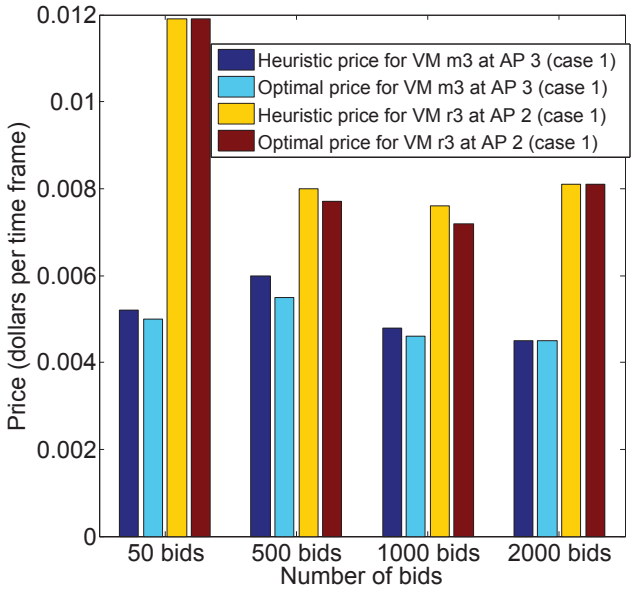


Figure 4.6 Local prices comparison between heuristic and optimal approaches.

optimal one, as also confirmed by the results shown in Figures 4.4 and 4.5. Finally, we compare the average delay per bid of the heuristic algorithm with that of the optimal algorithm in Figure 4.7 for case 1. To obtain the average delay per bid, we solve the bandwidth allocation problem based on both the results of the heuristic VM pricing and distribution algorithms as well as the optimal approach. As we can see in this figure, the delay per bid achieved by the heuristic algorithm is slightly higher than that of the optimal approach.

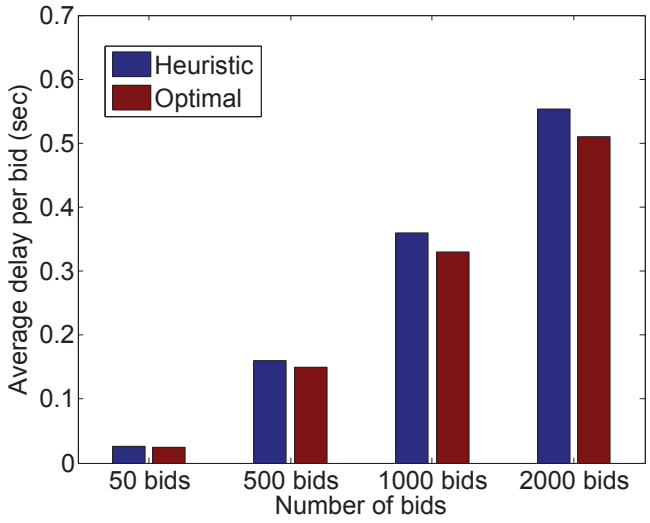


Figure 4.7 Average delay per bid comparison between heuristic and optimal.

CHAPTER 5

HIERARCHICAL CAPACITY PROVISIONING

The concept of fog computing is centered around providing computation resources at the edge of network, thereby reducing the latency and improving the quality of service. However, it is still desirable to investigate how and where at the edge of the network the computation capacity should be provisioned. To this end, we propose a hierarchical capacity provisioning scheme. In particular, we consider a two-tier network architecture consisting of shallow and deep cloudlets and explore the benefits of hierarchical capacity based on queueing analysis. Moreover, we explore two different network scenarios in which the network delay between the two tiers is negligible as well as the case that the deep cloudlet is located somewhere deeper in the network and thus the delay is significant. More importantly, we model the first network delay scenario with bufferless shallow cloudlets as well as the second scenario with finite-size buffer shallow cloudlets, and formulate an optimization problem for each model. We also use stochastic ordering to solve the optimization problem formulated for the first model and an upper bound based technique is proposed for the second model. The performance of the proposed scheme is evaluated via simulations in which we show the accuracy of the proposed upper bound technique as well as the queue length estimation approach for both randomly generated input and real trace data.

5.1 System Model and Problem Formulation

We consider a fog computing network consisting of M shallow cloudlets as the first tier of a two-tier hierarchical fog computing architecture. Accordingly, the second tier of fog computing nodes called the deep cloudlet is connected to all the shallow cloudlets. Therefore, we assume that each shallow cloudlet can cooperatively manage

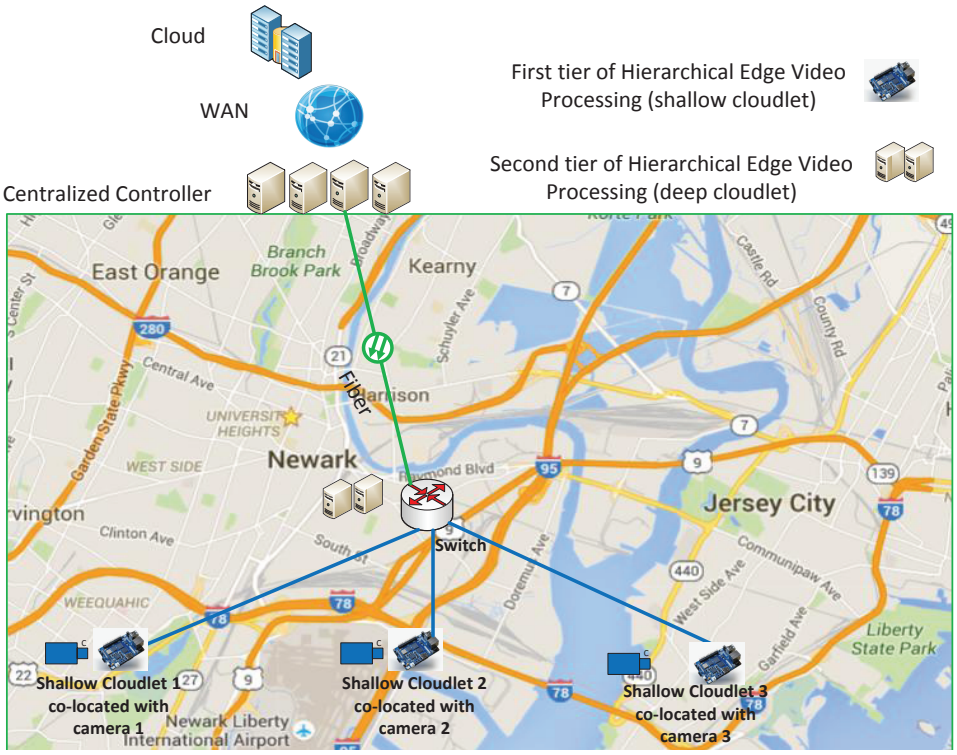


Figure 5.1 System model.

its incoming workload with the deep cloudlet. That is, the peak computing load at a shallow cloudlet can be forwarded to the deep cloudlet. As a practical case, we consider a distributed edge video processing environment shown in Figure 5.1. However, the proposed hierarchical capacity provisioning framework in this paper is not limited to only this example and it is applicable to all similar edge computing architectures. As depicted in this example, the shallow cloudlets are co-located with CCTV cameras and the deep cloudlet is installed at an aggregation switch. Moreover, in order to leverage the resource-rich facilities, the deep cloudlet is connected to the cloud via fibers. Our focus in this paper is on the capacity provisioning at the edge, i.e., the shallow and deep cloudlets.

We assume that the amount of edge computing workload at each shallow cloudlet at a given time follows a general distribution. We also assume that C is the total capacity budget to be provisioned at the edge where a portion α of the

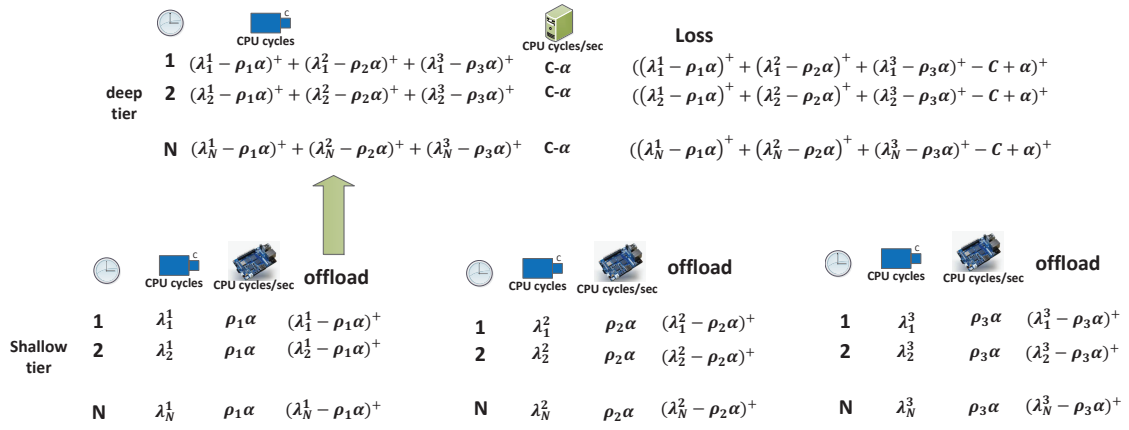


Figure 5.2 System model for bufferless shallow cloudlets.

capacity is provisioned at the shallow cloudlets and $C - \alpha$ at the deep cloudlet. Both the workload and the capacity are measured in CPU cycles. We use CPU cycles to measure the workload since it has been widely used in the literature to measure the computation requirements of the computing tasks [24]. Accordingly, to be consistent with the workload unit, we use CPU cycles per second as the unit of the computing capacity. Moreover, we consider a finite size queuing system at each cloudlet location where all the queuing systems are modeled as a discrete-time fluid system. In particular, at each time n, the queuing system at shallow cloudlet i consists of a server with constant rate $\rho_i\alpha$ and a fluid input λ_n^i which is assumed to be ergodic and stationary. We assume that λ_n^i's are independent but have a common distribution and $E(\lambda_n^i) = \overline{\lambda}_i$. The normalized coefficient ρ_i is also defined as $\rho_i = \frac{\overline{\lambda}_i}{\sum_{i=1}^{M} \overline{\lambda}_i}$. The system is assumed to be stable, i.e., $\sum_{i=1}^{M} \overline{\lambda}_i \leq C$.

5.2 Capacity Provisioning

We investigate two different network scenarios for the proposed system model. In particular, we first investigate the case that the network delay between the shallow cloudlets and the deep cloudlet is negligible. In the second scenario, we consider the

case in which the deep cloudlet is located somewhere deeper in the network, and thus the network delay between the shallow cloudlets and the deep cloudlet is significant.

5.2.1 Bufferless Shallow Cloudlets

We first investigate a network model in which the network delay between shallow cloudlets and the deep cloudlet is negligible. As shown in Figure 5.2, for such a network, we consider a buffer of size zero at each shallow cloudlet. Note that going from a flat architecture consisting of only shallow cloudlets to a hierarchical architecture with both the shallow cloudlets and the deep cloudlet, we take a portion of the capacity of the shallow cloudlets and allocate it to the deep cloudlet. Such a hierarchical capacity provisioning model is fair only if one unit of the capacity at a shallow cloudlet results in the same delay as compared to that at the deep cloudlet. Therefore, when the network delay is negligible, this fairness requirement is satisfied with bufferless shallow cloudlets since the deep cloudlet is assumed to be bufferless too. In other words, considering buffers at the shallow cloudlets while the deep cloudlet is bufferless is not a fair assumption from the perspective of the proposed capacity provisioning model. At each time n, the amount of the computing workload forwarded to the deep cloudlet is equal to $\sum_{i=1}^{M}(\lambda_n^i - \rho_i\alpha)^+$ where $(x)^+ = max(x, 0)$. Accordingly, the queuing system of the deep cloudlet can be modeled as a discrete-time fluid system consisting of a single server of constant rate $C - \alpha$ and a fluid input $\sum_{i=1}^{M}(\lambda_n^i - \rho_i\alpha)^+$. At time n, the total amount of fluid loss in the system can be established as $(\sum_{i=1}^{M}(\lambda_n^i - \rho_i\alpha)^+ - (C - \alpha))^+$. The average fluid loss in the system is calculated as

$$\begin{aligned} \overline{L}_{bl}(\alpha) &= \lim_{N\to\infty} \frac{\sum_{n=1}^{N}(\sum_{i=1}^{M}(\lambda_n^i - \rho_i\alpha)^+ - (C-\alpha))^+}{N} \\ &= E((\sum_{i=1}^{M}(\lambda_n^i - \rho_i\alpha)^+ - (C-\alpha))^+) \end{aligned} \tag{5.1}$$

where the second equality is due to the ergodicity assumption. Note that the focus of this paper is on proposing a network capacity planning framework rather

than a workload placement algorithm. Therefore, to achieve an optimum capacity provisioning, we propose to solve the following optimization problem

$$\underset{\alpha}{\text{minimize}}\ \overline{L}_{bl}(\alpha) \tag{5.2}$$

$$s.t.\ C_1 : \sum_{i=1}^{M} E(\lambda_n^i - \rho_i \alpha)^+ \leq C - \alpha$$

$$C_2 : 0 \leq \alpha \leq C$$

where the objective is to minimize the average fluid loss and constraint C_1 is necessary for stabilizing the queue at the deep cloudlet. The following theorem provides an optimal solution to problem (5.2).

Theorem 5.2.1. *The optimal solution to optimization problem (5.2) is achieved when $\alpha = 0$, i.e., when all the computing capacity is provisioned at the deep cloudlet.*

Proof. To prove Theorem 5.2.1, we need to show that $\overline{L}_{bl}(\alpha)$ is an strictly increasing function with respect to α. After some simple algebraic manipulation on $\overline{L}_{bl}(\alpha)$, we have,

$$\overline{L}_{bl}(\alpha) = E((\sum_{i=1}^{M} \max(\lambda_n^i, \rho_i \alpha) - C)^+) \tag{5.3}$$

Function $\overline{L}_{bl}(\alpha)$ is proven to be strictly increasing if we can show that $\overline{L}_{bl}(\alpha_h) < \overline{L}_{bl}(\alpha_k)$ for all $\alpha_h < \alpha_k$, where $0 \leq \alpha_h, \alpha_k \leq C$. Consider two random variables $X_n = \sum_{i=1}^{M} \max(\lambda_n^i, \rho_i \alpha_h)$ and $Y_n = \sum_{i=1}^{M} \max(\lambda_n^i, \rho_i \alpha_k)$. If X_n and Y_n satisfy the stop-loss order, written as $X_n <_{sl} Y_n$, then $\overline{L}_{bl}(\alpha_h) < \overline{L}_{bl}(\alpha_k)$ for all C. In addition, the stop-loss order is maintained under the summation of independent random variables. Therefore, if random variable $\max(\lambda_n^i, \rho_i \alpha_h)$ precedes random variable $\max(\lambda_n^i, \rho_i \alpha_k)$ in stop-loss order, so X_n precedes Y_n. Moreover, the dangerous order relation is known

to be a sufficient condition for the stop-loss order [20, 37, 38]. Therefore, we continue our proof by showing the satisfaction of the two known conditions for dangerous order relation. In terms of the first condition, we observe that random variables $\max(\lambda_n^i, \rho_i\alpha_h)$ and $\max(\lambda_n^i, \rho_i\alpha_k)$ satisfy the once-crossing condition for crossing point $\rho_i\alpha_h$. Regarding the second condition, it is simple to show that,

$$E(\max(\lambda_n^i, \rho_i\alpha_h)) \leq E(\max(\lambda_n^i, \rho_i\alpha_k)) \tag{5.4}$$

Therefore, $\max(\lambda_n^i, \rho_i\alpha_h)$ precedes $\max(\lambda_n^i, \rho_i\alpha_k)$ in a dangerous order, and accordingly X_n and Y_n have the stop-order relation and the proof is complete. □

5.2.2 Finite-Size Buffer Shallow Cloudlets

In this section, we investigate the case when the network delay between the shallow cloudlets and the deep cloudlet is not negligible. Therefore, $\alpha = 0$ is not the optimal solution since the reduction in the average loss is achieved at the expense of a higher delay. Let D be the average network delay per unit of workload (one CPU cycle) if it is served at the deep cloudlet and let's define each unit of workload as a job. For this scenario, we enforce a deadline equal to D seconds at each shallow cloudlet's buffer. In fact, a job is forwarded to the deep cloudlet only if it cannot be handled by deadline D. That is, sizes of the buffers at the shallow cloudlets are calculated based on D such that the maximum waiting time in each shallow cloudlet's buffer is D seconds. In other words, if one unit of capacity at a shallow cloudlet can handle a job within D seconds, it is not fair/justfiable to consider the allocation of that capacity to the deep cloudlet since the network delay is D seconds. Therefore, if Q^i is the number of waiting jobs in the corresponding buffer of shallow cloudlet i right before the arrival of a new job, the new job can be handled after $\frac{Q^i}{\rho_i\alpha}$ seconds. If $\frac{Q^i}{\rho_i\alpha} \leq D$, then the job can be handled before the deadline D. Otherwise, the job is not handled before

the deadline and it is forwarded to the deep cloudlet. Therefore, we can model the deadline by a finite-size queue with length $\rho_i \alpha D$. Accordingly, the average fluid loss is calculated as

$$\overline{L}_{fb}(\alpha) \quad = \quad E((\sum_{i=1}^{M}(\lambda_n^i \ + \ Q_{n-1}^i(\alpha) \ - \ \rho_i\alpha \ - \ \rho_i\alpha D)^+ \ - \ (C \ - \ \alpha))^+) \quad (5.5)$$

where Q_{n-1}^i is the queue length at shallow cloudlet i at time $n-1$. Therefore, we propose to solve the following optimization problem,

$$\underset{\alpha}{\text{minimize}}\ \overline{L}_{fb}(\alpha) \tag{5.6}$$

$$s.t.\ C_1 : \sum_{i=1}^{M} E(\lambda_n^i + Q_{n-1}^i(\alpha) - \rho_i\alpha - \rho_i\alpha D)^+ \leq C - \alpha$$

$$C_2 : 0 \leq \alpha \leq C$$

where the objective is to minimize the average loss via optimizing α and constraint C_1 is required for stabilizing the queue at the deep cloudlet.

Note that the optimization problem (5.6) can be compared to an stop-loss reinsurance model where the objective of the problem is the stop-loss pure premium $E(X-d)^+$ with retention equal to $d = C - \alpha$ [16, 62, 63]. Here, the retention $d = 0$, i.e., a flat design with only shallow cloudlets, can be considered as the special case where the insurer transfers all loss to the reinsurer, i.e., full reinsurance. On the other hand, case $d = C$, i.e., a flat design with only a deep cloudlet, denotes the special case where the insurer retains all loss, i.e., the case that implies no reinsurance. In terms of finding the optimal solution for the reinsurance models, most of the existing studies assume that the distribution function of X is known and satisfies some properties. However, here the distribution function of X, i.e., $\sum_{i=1}^{M}(\lambda_n^i + Q_{n-1}^i(\alpha) - \rho_i\alpha(1+D))^+$, is not known for two reasons. First, the distribution of $Q_{n-1}^i(\alpha)$ is not known.

Second, even if we have the knowledge of the distribution function for $Q^i_{n-1}(\alpha)$, it is cumbersome to calculate the M-fold convolution of M pdfs. Moreover, in practice, we usually know the average of λ^i_n's rather than their distribution function. There are a few studies such as [35, 56], that consider the case when incomplete information of X is available. However, those solutions are not applicable here because they either have to know at least the average and variance of X or they are interested in finding the optimal retention d or estimating the minimal stop-loss rather than the optimum value of X. Note that here we only know the average of X, i.e., $\sum_{i=1}^{M} E(\lambda^i_n + Q^i_{n-1}(\alpha) - \rho_i\alpha(1+D))^+$ based on the loss probability of the G/D/1 queue. Therefore, we propose two different strategies to find the optimal value of α. Both strategies are developed based on the Markov's inequality. That is, instead of minimizing the original objective, we minimize an upper bound calculated based on the Markov's inequality in the following theorem.

Theorem 5.2.2. *The objective function of optimization problem (5.6) is upper bounded as follows,*

$$\overline{L}_{fb} \leq \int_C^{\tau} \frac{\sum_{i=1}^{M} E(\lambda^i_n + Q^i_{n-1}(\alpha) - \rho_i\alpha(1+D))^+}{C-\alpha} dx \tag{5.7}$$

Proof.

$$\begin{aligned}
&E(\sum_{i=1}^{M}(\lambda^i_n + Q^i_{n-1}(\alpha) - \rho_i\alpha - \rho_i\alpha D)^+ - (C-\alpha))^+ \\
&= \int_C^{\infty} (x-C) dP(\sum_{i=1}^{M}(\lambda^i_n + Q^i_{n-1}(\alpha) - \rho_i\alpha - \rho_i\alpha D)^+ + \alpha \leq x) \\
&= -\int_C^{\infty} (x-C) dP(\sum_{i=1}^{M}(\lambda^i_n + Q^i_{n-1}(\alpha) - \rho_i\alpha - \rho_i\alpha D)^+ + \alpha \geq x) \\
&= \int_C^{\infty} P(\sum_{i=1}^{M}(\lambda^i_n + Q^i_{n-1}(\alpha) - \rho_i\alpha - \rho_i\alpha D)^+ + \alpha > x) dx \\
&\approx \int_C^{\tau} P(\sum_{i=1}^{M}(\lambda^i_n + Q^i_{n-1}(\alpha) - \rho_i\alpha - \rho_i\alpha D)^+ + \alpha > x) dx
\end{aligned}$$

where τ in the approximation can be decided based on the tail of the distribution of λ_n^i such that $P(\sum_{i=1}^{M}(\lambda_n^i + Q_{n-1}^i(\alpha) - \rho_i\alpha - \rho_i\alpha D)^+ + \alpha > \tau) \leq \epsilon$, i.e.,

$$\int_{\tau}^{\infty} P(\sum_{i=1}^{M}(\lambda_n^i + Q_{n-1}^i(\alpha) - \rho_i\alpha - \rho_i\alpha D)^+ + \alpha > x)dx$$
$$<< \int_{C}^{\tau} P(\sum_{i=1}^{M}(\lambda_n^i + Q_{n-1}^i(\alpha) - \rho_i\alpha - \rho_i\alpha D)^+ + \alpha > x)dx$$

Then, we have

$$\int_{C}^{\tau} P(\sum_{i=1}^{M}(\lambda_n^i + Q_{n-1}^i(\alpha) - \rho_i\alpha - \rho_i\alpha D)^+ + \alpha > x)dx$$
$$\leq \int_{C}^{\tau} P(\sum_{i=1}^{M}(\lambda_n^i + Q_{n-1}^i(\alpha) - \rho_i\alpha - \rho_i\alpha D)^+ > C - \alpha)dx$$
$$\leq \int_{C}^{\tau} \frac{\sum_{i=1}^{M} E(\lambda_n^i + Q_{n-1}^i(\alpha) - \rho_i\alpha(1+D))^+}{C-\alpha}dx$$

where the last inequality is in accordance with the Markov's inequality. The proof is complete. □

G/D/1 Loss Probability Approach In the first approach, we rely on the loss probability of the G/D/1 queue. According to queueing analysis [46], we have,

$$E(\lambda_n^i + Q_{n-1}^i(\alpha) - \rho_i\alpha(1+D))^+ = P_i(\alpha)\overline{\lambda}_i \tag{5.8}$$

where $P_i(\alpha)$ is the loss probability of the finite-size queue and can be accurately estimated from the tail probability (overflow probability) of an infinite buffer system as follows [46],

$$P_i(\alpha) = \gamma_i(\alpha)e^{-\frac{1}{2}\min_{n\geq 1} M_n^i(\alpha)}, \tag{5.9}$$

where

$$\gamma_i(\alpha) = \frac{1}{\overline{\lambda}_i\sqrt{2\pi}\sigma_i} e^{\frac{(\rho_i\alpha - \overline{\lambda}_i)^2}{2\sigma_i^2}} \int_{\rho_i\alpha}^{\infty} (r - \rho_i\alpha) e^{\frac{-(r-\overline{\lambda}_i)^2}{2\sigma_i^2}} dr, \tag{5.10}$$

and for each $n \geq 1$,

$$M_n^i(\alpha) = \frac{(\rho_i \alpha D + n(\rho_i \alpha - \overline{\lambda}_i))^2}{nC_{\lambda_n^i}(0) + 2\sum_{l=1}^{n-1}(n-l)C_{\lambda_n^i}(l)}, \tag{5.11}$$

and $C_{\lambda_n^i}(l)$ is the autocovariance of λ_n^i probability function and we have $\sigma_i^2 = C_{\lambda_n^i}(0)$. Note that function (5.9) is valid when $\rho_i \alpha \geq \overline{\lambda}_i$, i.e., when $\alpha \geq \sum_{i=1}^{M} \overline{\lambda}_i$. In addition, it is known that the estimation yields the highest level of accuracy when λ_n^i is characterized by a Gaussian process. Therefore, in this approach, we focus on the case that the input process to each queue, i.e., λ_n^i, follows a Gaussian process and propose to solve the following optimization problem,

$$\underset{\alpha}{\text{minimize}} \ \frac{\sum_{i=1}^{M} P_i(\alpha)\overline{\lambda}_i}{C - \alpha} \tag{5.12}$$

$$s.t.\ C_1 : \sum_{i=1}^{M} P_i(\alpha)\overline{\lambda}_i \leq C - \alpha$$

$$C_2 : \sum_{i=1}^{M} \overline{\lambda}_i \leq \alpha < C$$

Algorithm 3

find a feasible stepsize $\epsilon \geq 0$
$r \leftarrow 1 + \epsilon$
$\hat{\alpha} \leftarrow C$
repeat
  solve problem (5.13) for α in range (5.14) and find $\overset{\star}{\alpha}$
  if $\overset{\star}{\alpha} \neq \emptyset$ **then**
    $\hat{\alpha} \leftarrow \overset{\star}{\alpha}$
    $r \leftarrow \frac{C - \hat{\alpha}}{\sum_{i=1}^{M} P_i(\hat{\alpha})\overline{\lambda}_i} + \epsilon$
  end if
until $\overset{\star}{\alpha} = \emptyset$

To solve optimization problem (5.12), we propose a centralized heuristic algorithm. Our algorithm is motivated by two observations. First, $p_i(\alpha)$ is a

non-increasing function with respect to α when $\alpha \geq \sum_{i=1}^{M} \overline{\lambda}_i$ [30, 45]. Second, alternative optimization problem (5.13) is a convex optimization problem if α is limited to some specific range and can be solved efficiently by interior point methods. In other words, problem (5.12) is generally nonconvex. Therefore, we introduce a new variable r such that $r = \frac{C-\alpha}{\sum_{i=1}^{M} P_i(\alpha)\overline{\lambda}_i}$. Accordingly, inspired by coordinate descent techniques [10], we solve successively alternate minimizations (5.13) in α while holding r fixed. As shown in Algorithm 3, we first choose a feasible value for stepsize ϵ. Note that Algorithm 3 converges to the optimal solution provided that the stepsize is selected small enough. We also set initial ratio $r = 1+\epsilon$ and C is chosen as the initial solution. Then, we solve the following optimization problem for the given value of r,

$$\underset{\alpha}{\text{minimize}} \sum_{i=1}^{M} P_i(\alpha)\overline{\lambda}_i \tag{5.13}$$

$$s.t.\ C_1 : \sum_{i=1}^{M} P_i(\alpha)\overline{\lambda}_i - \frac{C-\alpha}{r} \leq 0$$

$$C_2 : \sum_{i=1}^{M} \overline{\lambda}_i \leq \alpha < C$$

Finally, we update the ratio r and optimal solution $\hat{\alpha}$ as shown in Algorithm 3. We repeat this procedure until there is no optimal solution for problem (5.13). The convexity of problem (5.13) is proven in the following theorem.

Theorem 5.2.3. *The constrained optimization problem (5.13) is a convex optimization problem if α is limited to,*

$$\alpha \in \sum_{i=1}^{M} \overline{\lambda}_i + [\max_i\ .07071\frac{\sigma_i}{\rho_i}, \min_i\ 1.4477\frac{\sigma_i}{\rho_i}] \tag{5.14}$$

Proof. To show the convexity of the proposed optimization problem, we are required to prove [14]:

- The objective function, i.e., $\sum_{i=1}^{M} P_i(\alpha)\overline{\lambda}_i$, is convex.
- The inequality constraint C_1 is convex.

We start by proving the convexity of $P_i(\alpha)$, i.e., loss probability function. It is known that the loss probability is a convex function when the service rate $\rho_i\alpha$ [30, 45] is limited to,

$$\rho_i\alpha \in [\overline{\lambda}_i + .07071\sigma_i, \overline{\lambda}_i + 1.4477\sigma_i] \tag{5.15}$$

Accordingly, $P_i(\alpha)$ is a convex function for all i if,

$$\alpha \in \sum_{i=1}^{M} \overline{\lambda}_i + [\max_i .07071\frac{\sigma_i}{\rho_i}, \min_i 1.4477\frac{\sigma_i}{\rho_i}] \tag{5.16}$$

Then, the inequality constraint function of C_1 and the objective function are both proven to be convex since they are summations of convex functions, and the proof is complete. □

An interesting extension for the optimization problem (5.12) is the case when the loss probability at each shallow cloudlet i is upper bounded by a constant TH_i. In other words, this extension limits the number of jobs that can be forwarded to the deep cloudlet from the shallow cloudlets. Therefore, we incorporate this requirement into our optimization problem by adding the inequality constraints $P_i(\alpha) < TH_i$ as follows,

$$\underset{\alpha}{\text{minimize}} \ \frac{\sum_{i=1}^{M} P_i(\alpha)\overline{\lambda}_i}{C - \alpha} \tag{5.17}$$

$$s.t.\ C_1 : \sum_{i=1}^{M} P_i(\alpha)\overline{\lambda}_i \leq C - \alpha$$

$$C_2 : P_i(\alpha) \leq TH_i \ \forall i = 1, ..., M$$

$$C_3 : \sum_{i=1}^{M} \overline{\lambda}_i \leq \alpha < C$$

Note that the new inequality constraints C_2 form a convex set under the same requirement as Theorem 5.2.3. Therefore, Algorithm 3 can still be used to solve problem (5.17).

Queue Length Estimation Approach In the previous approach, we rely on the accuracy of loss probability of a G/D/1 queue and replace loss $E(\lambda_n^i + Q_{n-1}^i(\alpha) - \rho_i\alpha(1+D))^+$ with $P_i(\alpha)\overline{\lambda}_i$. However, as mentioned earlier, function $P_i(\alpha)$ is accurate when the input process λ_n^i is characterized by a Gaussian distribution, and more importantly, it is derived based on the assumption that $\rho_i\alpha \geq \overline{\lambda}_i$. Therefore, in this section, we propose another approach which can be accurate for other distributions such as the uniform distribution and is valid for all values of α. The idea is to replace the queue length Q_{n-1}^i in $E(\lambda_n^i + Q_{n-1}^i(\alpha) - \rho_i\alpha(1+D))^+$ with a linear estimation of the Average Queue Length (AQL). We propose the following linear estimation,

$$e_{AQL_i} = \begin{cases} 0, & \overline{\lambda}_i \leq \rho_i\alpha \\ a\alpha + b, & \rho_i\alpha < \overline{\lambda}_i \leq \rho_i\alpha(1+D) \\ \rho_i\alpha D, & \overline{\lambda}_i > \rho_i\alpha(1+D) \end{cases} \tag{5.18}$$

where constants a and b can be calculated by solving two equations $a(\frac{\sum_{i=1}^{M}\overline{\lambda}_i}{1+D}) + b = \rho_i D(\frac{\sum_{i=1}^{M}\overline{\lambda}_i}{1+D})$ and $a(\sum_{i=1}^{M}\overline{\lambda}_i) + b = 0$. After reordering, we have

$$e_{AQL_i} = \begin{cases} 0, & \alpha \geq \sum_{i=1}^{M}\overline{\lambda}_i \\ -\rho_i\alpha + \rho_i\sum_{i=1}^{M}\overline{\lambda}_i, & \frac{\sum_{i=1}^{M}\overline{\lambda}_i}{1+D} \leq \alpha < \sum_{i=1}^{M}\overline{\lambda}_i \\ \rho_i\alpha D, & \alpha < \frac{\sum_{i=1}^{M}\overline{\lambda}_i}{1+D} \end{cases} \tag{5.19}$$

Note that estimation (5.19) yields a higher accuracy for a smaller variance of λ_n^i. In case that the variance is not small, we can adjust the estimation as follows

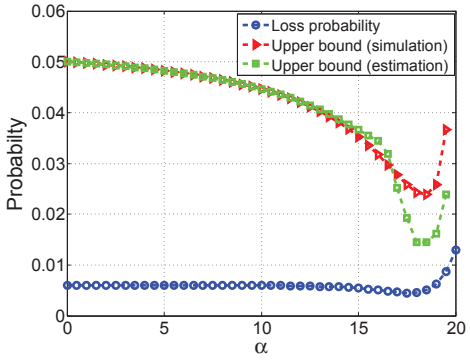
$$e_{AQL_i} = \begin{cases} 0, & \alpha \geq \sum_{i=1}^{M}\overline{\lambda}_i + \kappa_i \\ -\rho_i(\alpha - \kappa_i) + \rho_i\sum_{i=1}^{M}\overline{\lambda}_i, & \frac{\sum_{i=1}^{M}\overline{\lambda}_i}{1+D} + \kappa_i \leq \alpha < \sum_{i=1}^{M}\overline{\lambda}_i + \kappa_i \\ \rho_i(\alpha - \kappa_i)D, & \alpha < \frac{\sum_{i=1}^{M}\overline{\lambda}_i}{1+D} + \kappa_i \end{cases} \tag{5.20}$$

where constant κ_i is calculated heuristically and according to the variance of λ_n^i. Therefore, in order to find an approximate solution, we can replace the optimization problem (5.12) with the following problem,

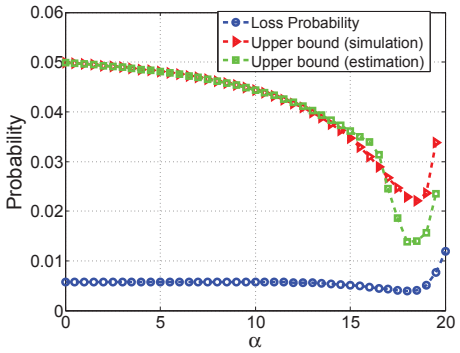
$$\underset{\alpha}{\text{minimize}} \ \frac{\sum_{i=1}^{M} E(\lambda_n^i + e_{AQL_i} - \rho_i\alpha(1+D))^+}{C - \alpha} \tag{5.21}$$

$$s.t.\ C_1 : \sum_{i=1}^{M} E(\lambda_n^i + e_{AQL_i} - \rho_i\alpha(1+D))^+ \leq C - \alpha$$

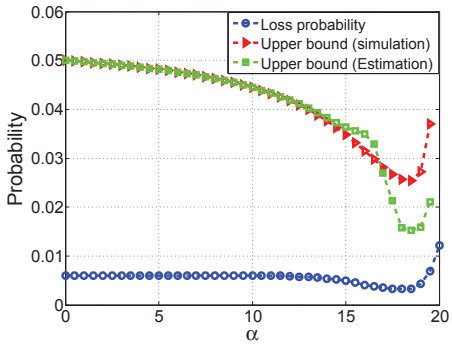
$$C_2 : 0 \leq \alpha \leq C$$



(a) When the input is a Gaussian AR process.



(b) When the input is a Gaussian process.



(c) When the input is a uniform process.

Figure 5.3 The comparison between the shape of the loss probability with the shape of the proposed upper bound versus α for $D = 0.1$.

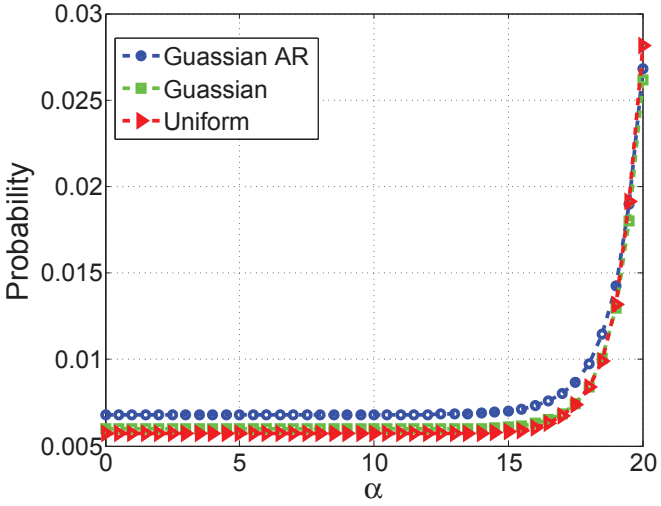


Figure 5.4 Loss probability versus α for different input processes and when $D = 0$.

The same procedure as Algorithm 3 is still valid to solve problem (5.21) for two reasons. That is, function $E(\lambda_n^i + e_{AQL_i} - \rho_i\alpha(1+D))^+$ is a non-increasing and convex function with respect to α as proved in the following theorem.

Theorem 5.2.4. *Function $g_i(\alpha) := E(\lambda_n^i + e_{AQL_i} - \rho_i\alpha(1+D))^+$ is a non-increasing and convex function with respect to α.*

Proof.

$$g_i(\alpha) = \int_{\rho_i\alpha(1+D)-e_{AQL_i}}^{\infty} (x - \rho_i\alpha(1+D) + e_{AQL_i}) f_{\lambda_n^i}(x)dx \tag{5.22}$$

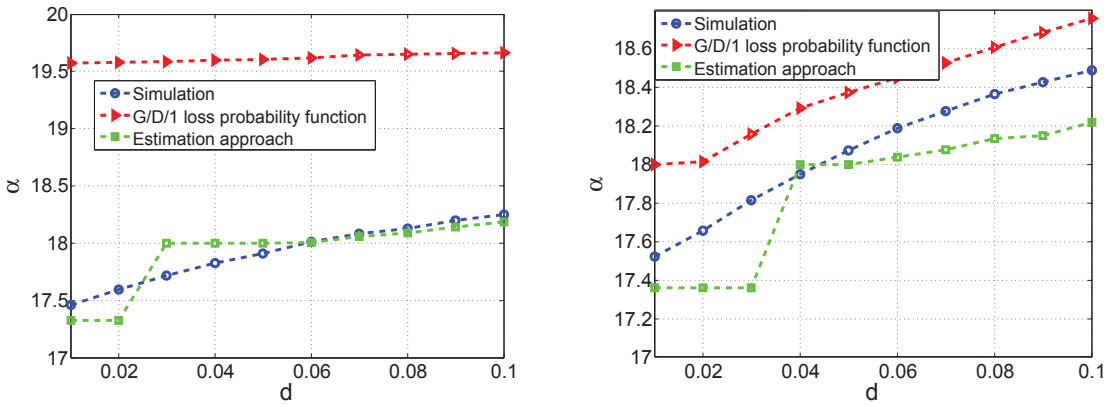
Then, according to Leibniz integral rule, we have

$$g_i'(\alpha) = \int_{\rho_i\alpha(1+D)-e_{AQL_i}}^{\infty} (-\rho_i(1+D) + e'_{AQL_i}) f_{\lambda_n^i}(x)dx \tag{5.23}$$

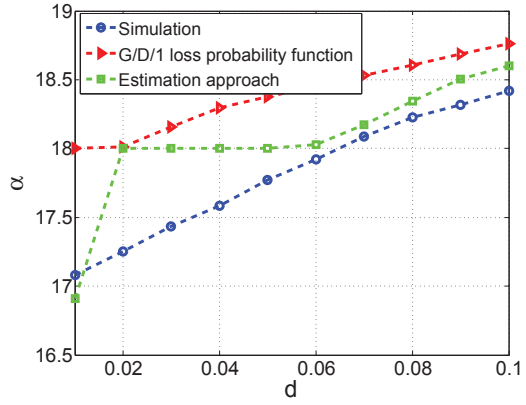
where $-\rho_i(1+D)+e'_{AQL_i} \leq -\rho_i$ and thus $g_i'(\alpha) \leq 0$. Therefore, function $g_i(\alpha)$ is proven to be non-increasing. Moreover, by taking the second derivative with respect to α, we have

$$g_i''(\alpha) = (\rho_i(1+D) - e'_{AQL_i})^2 f_{\lambda_n^i}(\rho_i\alpha(1+D) - e_{AQL_i}) \geq 0 \tag{5.24}$$

Therefore, $g_i(\alpha)$ is convex and the proof is complete. □



(a) When the input is a Gaussian AR process. **(b)** When the input is a Gaussian process.



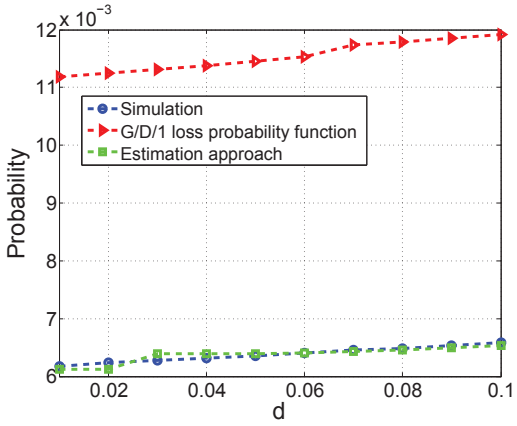
(c) When the input is a uniform process.

Figure 5.5 Optimal α versus D.

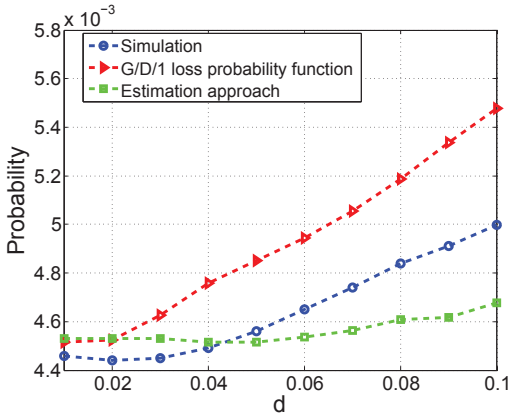
5.3 Simulation Results

In this section, we evaluate the performance of the proposed upper bound for the average loss based on both randomly generated input and real trace data. In both cases, we consider a fog computing network consisting of three shallow cloudlets connected to a deep cloudlet, i.e., a network architecture similar to Figure 5.1.

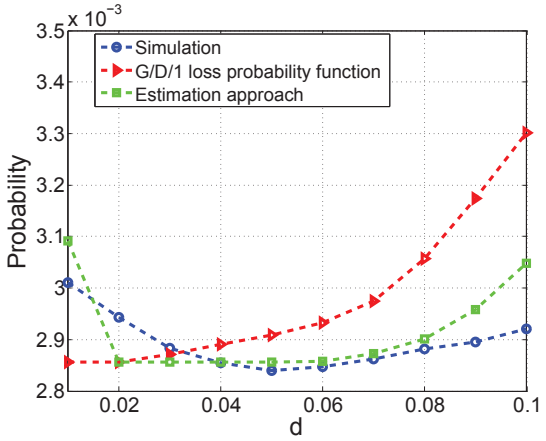
We assume a total capacity budget of 20 Gigacycles per second. It is also assumed that the average computation workload at shallow cloudlets 1, 2, and 3 is equal to 4, 8, and 6 Gigacycles, respectively. The variance of the input process is also set to one. Moreover, when the input process is modeled by a Gaussian autoregressive (AR) process, the autocovariance is set to $\frac{(0.3)^n}{1-(0.3)^2}$. For the simulation curves in the



(a) When the input is a Gaussian AR process.

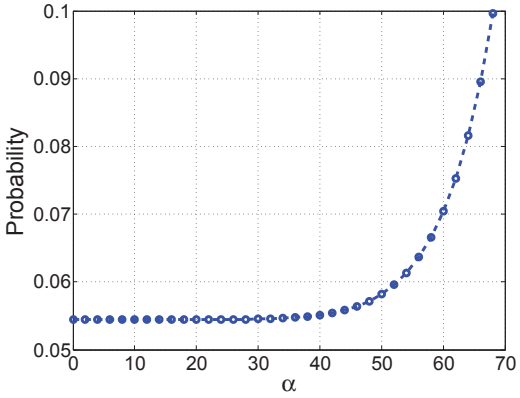


(b) When the input is a Gaussian process.

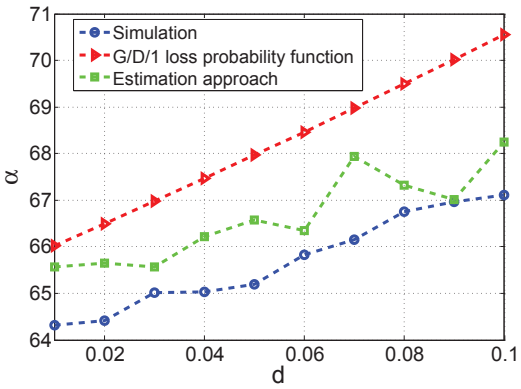


(c) When the input is a uniform process.

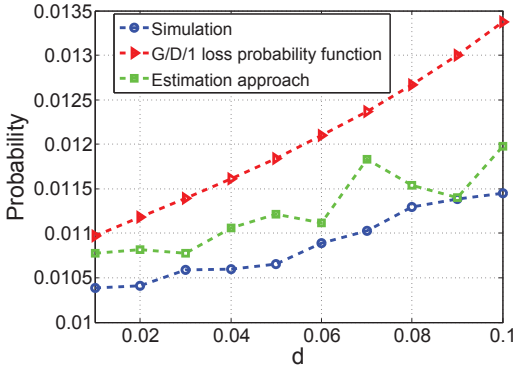
Figure 5.6 Optimum loss probability versus D.



(a) Loss probability versus α when $D = 0$.



(b) Optimal α versus D.



(c) Optimum loss probability versus D.

Figure 5.7 Real data trace based simulations.

figures, the corresponding loss probabilities are calculated by simulations. That is, we neither use the loss probability function formula nor our estimation technique.

Figure 5.3 compares the shape of the loss probability with the that of the proposed upper bound versus α when $D = 0.1$ sec. In particular, Figures 5.3 (a), (b) and (c) show the results for Gaussian AR, Gaussian, and uniform processes, respectively. Note that the loss probability is defined as the average loss divided by $\sum_{i=1}^{M} \overline{\lambda}_i$. To be comparable with loss probability, the upper bound is also divided by $\sum_{i=1}^{M} \overline{\lambda}_i$ in all the corresponding figures. As depicted in this figure, the upper bound is minimized almost for the same value of α as the loss probability which confirms the accuracy of the proposed upper bound in terms of optimizing α. This figure also evaluates the accuracy of the queue length estimation approach by comparing the upper bound based on this approach with the upper bound based on the simulation. We do not include the upper bound based on G/D/1 loss probability function (5.9) since this function is valid only for $\alpha \geq \sum_{i=1}^{M} \overline{\lambda}_i$. Moreover, Figure 5.4 shows the loss probability versus α for Gaussian, Gaussian AR, and uniform input processes when $D = 0$. As shown in Figure 5.4, in the case of $D = 0$, no matter what distribution, the loss probability exhibits a non-decreasing shape versus α, which confirms the result of Theorem 1.

Figures. 5.5 and 5.6 provide the optimization results for different values of D and different input processes. Specifically, Figure 5.5 compares the optimum α of the simulation result with both the G/D/1 loss probability function approach and the queue length estimation approach. Note that the optimum α is increased by increasing D because the queue length at the shallow cloudlets in increased by increasing D and thus, it is more efficient to provide higher capacity at the shallow cloudlets.

Figure 5.6 also compares the same approaches but in terms of the optimum loss probability which is equivalent to the optimum average loss since the loss probability is the average loss divided by constant $\sum_{i=1}^{M} \overline{\lambda}_i$. As depicted in Figures. 5.5 and 5.6,

while both approaches have high accuracy, the estimation approach yields higher accuracy because the loss probability function is limited to a short range of values of α. In other words, the optimum α in the case of G/D/1 loss probability approach is lower bounded by $\sum_{i=1}^{M} \overline{\lambda}_i$. In addition, the better performance of G/D/1 loss probability approach when the input is Gaussian is due to the higher accuracy of function (5.9) for Gaussian input. Nevertheless, while the estimation approach provides an accurate solution quite close to the simulation, the loss probability of the estimation approach is sometimes lower than that of the simulation. This observation is attributed to the fact that the estimation approach can underestimate the average queue lengths. For example, the queue length estimation method estimates the average queue length as zero ($e_{AQL_i} = 0$) for $\alpha \geq \sum_{i=1}^{M} \overline{\lambda}_i$ while the average queue length based on the simulation is not necessarily zero. We also simulate the total incoming tasks at shallow cloudlets by the requests made to the 1998 World Cup web site [4] in which we use one hour trend of a sample day for each shallow cloudlet. We also assume that each task requires on average 1 Gigacycles. Figure 5.7 (a) depicts the shape of the loss probability versus α when $D = 0$. The loss probability versus α when $D = 0$ is a non-decreasing function which confirms the result of Theorem 1 for real trace data as well. Moreover, Figures 5.7 (b) and (c) compare the optimization results, i.e., the optimum α and optimum loss probability, of two proposed approaches with the simulation result. As depicted in these figures, the queue length estimation approach outperforms the G/D/1 approach for the real trace data as well.

CHAPTER 6

OPTIMAL CODE PARTITIONING OVER TIME AND HIERARCHICAL CLOUDLETS

This letter proposes a task scheduling scheme designed for code partitioning over time and the hierarchical cloudlets in a mobile edge network. To this end, we define the so called energy-time cost parameters to optimally schedule tasks over time and hierarchical cloudlet locations. Accordingly, we investigate two different optimization scenarios. In particular, the first scenario aims at finding the optimal task scheduling for given radio parameters. In the second scenario, we carry out the optimization of both the task scheduling and the mobile device's transmission power. More importantly, we show that by adopting the proposed code partitioning scheme in this letter, the transmission power optimization problem becomes a disjoint problem from the task scheduling problem.

6.1 System Model and Problem Formulation

We consider a Hierarchical Mobile Edge Computing (HI-MEC) architecture shown in Figure 6.1. The HI-MEC architecture consists of field, shallow and deep cloudlets. In particular, in a HI-MEC environment, the field cloudlets as the resource-poor facilities are co-located with Small Cell enhanced Node Bs (SCeNBs). The shallow cloudlets as the resource-modest facilities are also hosted at the first level of aggregation nodes, i.e., at Point of Presences (PoPs). Moreover, a resource-rich facility called the deep cloudlet is located at the mobile backhaul. We consider a two-time scale model in which the running time of the HI-MEC environment is divided into a sequence of time frames at equal length, T, e.g., five minutes. Each time frame itself is also divided into a sequence of time slots at equal length, τ, e.g., one minute. We assume one time frame consists of N time slots and denote $t_0, ..., t_{N-1}$ as the set of time slots in a time frame. At the beginning of each time frame, each SCeNB

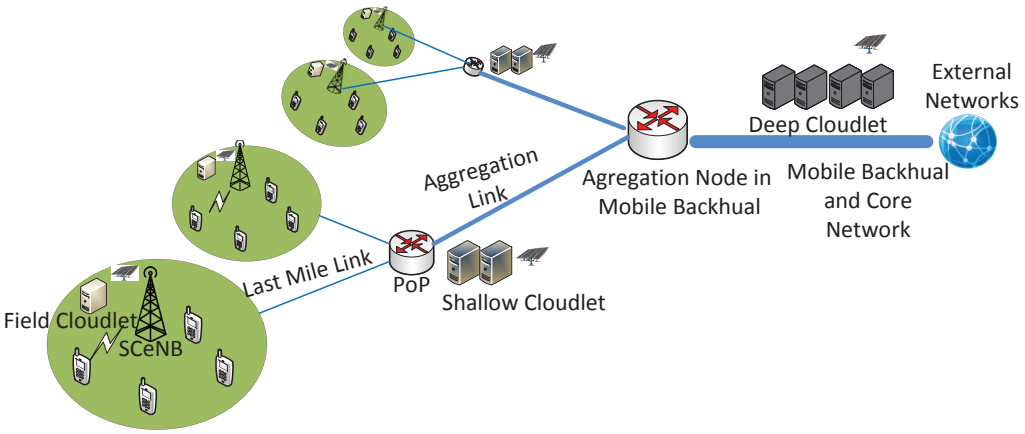


Figure 6.1 System model.

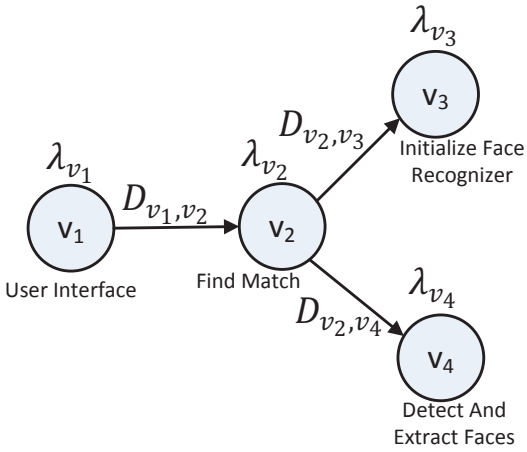


Figure 6.2 Call tree.

broadcasts the available computational capacities at the field, shallow and deep level to their MUs. In fact, a centralized controller at the deep cloudlet equipped with a data management model as well as a global view of the network predicts the workloads during the next few time slots and accordingly allocates the resources to the MUs. The centralized controller informs the SCeNBs about the allocated resources to each MU. The allocated resources within a time slot are assumed to be fixed but changing from a time slot to the next time slot.

A MU's application is described by a call graph, i.e., a directed acyclic graph as $G = (\mathcal{V}; \mathcal{E})$. The call graph represents the relation among the tasks in which the

MU's application can be partitioned. For example, the call graph of a face recognition application [24] is shown in Figure 6.2. Each vertex represents a task v_i in the call stack and each edge $e = (v_i; v_j)$ shows an invocation of task v_j from task v_i. Each task node v_i is characterized by its workload, λ_{v_i}, i.e., the number of CPU cycles required to complete the execution of the task. Each edge $(v_i; v_j) \in \mathcal{E}$ is also characterized by the number of bits (D_{v_i,v_j}) that must be transferred from the parent task v_i to child task v_j. In the rest of the letter, we consider a given MU of interest in defining the corresponding notations. The MU can decide to execute a task locally at the mobile device or remotely at the available cloudlet locations. The MU's decision depends on two factors, energy and delay. The MU's energy consumption is the energy required to execute a task locally or to transmit the required bits to the remote cloudlet (when the parent task is executed locally and the child remotely), or to receive the required bits from the cloudlet (when the parent task is executed remotely and the child locally). On the other hand, the delay is the time required to execute the task locally or to transmit the required bits to the remote location. Therefore, not only the local parameters but also the remote parameters are contributing to the corresponding cost of each task, i.e, the energy and time costs. Here, the local parameters include transmit power P_{up}, reception power P_{rx}, local computational capacity μ_{loc} (in CPU cycles per second), and local processing power P_{loc}. Unlike the remote parameters, we assume that the local parameters are not changing time slot by time slot.

In terms of the wireless access parameters, we define C_{dl} and $C_{up}(P_{up})$ as the capacities of the downlink and the uplink channels between the MU and its associated SCeNB, respectively. The remote parameters are the available cloudlet locations for the MU, the computational capacities at the cloudlets and the data rates on the corresponding links. Let's assume the MU is associated with SCeNB s and $\mathcal{AC}$ is the set of all available remote locations, which provision a field, a shallow and the deep cloudlet. Let $\mu_x^{t_n}$ also be the remote computational capacity that can be assigned

to the MU at cloudlet $x \in \mathcal{AC}$ during time slot t_n. Moreover, we assume $C_{s,x}^{t_n}$ is the maximum data rate that can be allocated to the MU between SCeNB s and the cloudlet x during time slot t_n. Similarly, $C_{x,y}^{t_n}$ is the maximum data rate that can be allocated to the MU between two cloudlets $x \in \mathcal{AC}$ and $y \in \mathcal{AC}$. It is assumed that a task is allowed to start execution only at the beginning of a time slot. However, the data from a parent task to a child task is assumed to be transferred as soon as the parent task execution is completed. We also assume that once a task starts executing during a time slot (once a data from a parent task to a child task starts transferring over the network), it is allowed to execute to completion (to transfer to completion) even if the time slot ends during execution (transfer) but with the same allocated computational capacity (data rate). Based on the defined local and remote parameters, we can translate the computation and communication requirements of the tasks and the edges on the MU's call graph to an energy-time cost parameter as follows,

$$ETC = \zeta_1(energy\ cost) + \zeta_2(time\ cost) \tag{6.1}$$

where ζ_1 and ζ_2 are two coefficients as the weights of the energy and time, respectively. The MU can flexibly choose the coefficients that favors more their demands. For example, a user with a low battery level may like to put more weight on the energy [18]. According to the proposed model, the MU not only has the option to execute a task at $|\mathcal{AC}|+1$ different local and remote computing locations (including the mobile device) but also in N time slots. Thus, the task offloading decision problem can be modeled as an assignment problem in a distributed processors system with $(|\mathcal{AC}|+1) \times N$ processors. In terms of the local ETC of a task, let's define $ETC_{loc}^{t_n}(v_i)$ as the ETC of task v_i when executed locally at the mobile device in time slot t_n. $ETC_{x}^{t_n}(v_i)$ is also defined as the ETC of task v_i when executed at location x during time slot t_n. Moreover, $ETC_{loc,x}^{t_n,t_m}(D_{v_i,v_j})$ is assumed to be the ETC between two tasks v_i executed

locally at the mobile device during time slot t_n and v_j executed remotely at cloudlet x during time slot t_m for $m > n$ where D_{v_i,v_j} is the number of bits that must be transferred from task node v_i to v_j. $ETC^{t_n,t_m}_{x,loc}(D_{v_i,v_j})$ indicates the same ETC but v_i executed remotely at cloudlet x and v_j locally at the mobile device. Similarly, let $ETC^{t_n,t_m}_{x,y}(D_{v_i,v_j})$ be the ETC between two tasks v_i executed remotely at cloudlet x during time slot t_n and v_j executed at cloudlet y during time slot t_m. Then, we can calculate the following ETCs,

$$ETC^{t_n}_{loc}(v_i) = \zeta_1 P_{loc}(\frac{\lambda_{v_i}}{\mu_{loc}}) + \zeta_2(n\tau + \frac{\lambda_{v_i}}{\mu_{loc}}) \tag{6.2}$$

$$ETC^{t_n}_{x}(v_i) = \zeta_2(n\tau + \frac{\lambda_{v_i}}{\mu^{t_n}_{x}}) \tag{6.3}$$

$$ETC^{t_n,t_m}_{loc,x}(D_{v_i,v_j}) = \begin{cases} \zeta_1 P_{up}(\frac{D_{v_i,v_j}}{C_{up(P_{up})}}) + \zeta_2(D_{v_i,v_j}(\frac{1}{C_{up(P_{up})}} + \frac{1}{C^{t_l}_{s,x}})),\ m \geq k+1 \\ \infty,\ m < k+1 \end{cases} \tag{6.4}$$

where $l = \lfloor n + \frac{\lambda_{v_i}}{\tau\mu_{loc}} \rfloor$ and $k = \lfloor n + \frac{\lambda_{v_i}}{\tau\mu_{loc}} + \frac{D_{v_i,v_j}}{\tau}(\frac{1}{C_{up(P_{up})}} + \frac{1}{C^{t_l}_{s,x}}) \rfloor$.

$$ETC^{t_n,t_m}_{x,loc}(D_{v_i,v_j}) = \begin{cases} \zeta_1 P_{rx}(\frac{D_{v_i,v_j}}{C_{dl}}) + \zeta_2(D_{v_i,v_j}(\frac{1}{C_{dl}} + \frac{1}{C^{t_{l'}}_{s,x}})),\ m \geq k'+1 \\ \infty,\ m < k'+1 \end{cases} \tag{6.5}$$

$$ETC^{t_n,t_m}_{x,y}(D_{v_i,v_j}) = \begin{cases} \zeta_2(\frac{D_{v_i,v_j}}{C^{t_{l'}}_{x,y}}),\ m \geq k''+1 \\ \infty,\ m < k''+1 \end{cases} \tag{6.6}$$

where $l' = \lfloor n + \frac{\lambda_{v_i}}{\tau \mu_x^{t_n}} \rfloor$, $k' = \lfloor n + \frac{\lambda_{v_i}}{\tau \mu_x^{t_n}} + \frac{D_{v_i,v_j}}{\tau}(\frac{1}{C_{dl}} + \frac{1}{C_{s,x}^{t_{l'}}}) \rfloor$ and $k'' = \lfloor n + \frac{\lambda_{v_i}}{\tau \mu_x^{t_n}} + \frac{D_{v_i,v_j}}{\tau C_{x,y}^{t_{l'}}} \rfloor$.

$$ETC_{local,local}^{t_n,t_{m\geq l+1}}(D_{v_i,v_j}) = ETC_{x,x}^{t_n,t_{m\geq l'+1}}(D_{v_i,v_j}) = 0 \tag{6.7}$$

Moreover, some of the tasks in a call graph are required to be executed locally. For example, the user interface task in Figure 6.2 which initiates the application must be executed locally at the mobile device. Therefore, the ETC of executing such tasks remotely is set to infinity. Based on the defined ETC parameters, the code partitioning problem over time and hierarchical cloudlets can be formulated as the following MINLP

$$\begin{aligned} &\underset{0\leq P_{up}\leq P_{max},\ I_{v_i}^{x,t_n}\in\{0,1\}}{\text{minimize}} \sum_{v_i\in\mathcal{V}} \sum_{x\in\mathcal{AC}'} \sum_{n=0}^{N-1} I_{v_i}^{x,t_n} ETC_x^{t_n}(v_i) \\ &+ \sum_{(v_i;v_j)\in\mathcal{E}} \sum_{x\in\mathcal{AC}'} \sum_{y\in\mathcal{AC}'} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} I_{v_i}^{x,t_n} I_{v_j}^{y,t_m} ETC_{x,y}^{t_n,t_m}(D_{v_i,v_j}) \end{aligned}$$

$$s.t. \sum_{x\in\mathcal{AC}'} \sum_{n=0}^{N-1} I_{v_i}^{x,t_n} = 1 \ \forall v_i \in \mathcal{V}$$

$$\sum_{v_i\in\mathcal{V}} I_{v_i}^{x,t_n} = 1 \ \forall x \in \mathcal{AC}', n = 1, ..., N-1 \tag{6.8}$$

where $I_{v_i}^{x,t_n} = 1$ if task v_i is executed at cloudlet x during time slot t_n, and $I_{v_i}^{x,t_n} = 0$ otherwise. Set $\mathcal{AC}'$ is also the set of all cloudlets plus the mobile device.

6.2 Optimal Hierarchical Task Scheduling

Note that (6.8) defines a mixed integer program which involves binary and real variables. Finding an optimal solution to this problem requires an exhaustive search over all the useful code partitions and entails a complexity that is exponential in the number of tasks. Therefore, we investigate an optimal scheduling scheme to solve problem (6.8) for two optimization scenarios. In the first scenario, we are interested

in finding an optimal task scheduling for given radio parameters, i.e., the case that variable P_{up} in the optimization problem (6.8) is fixed. In the second scenario, beside finding the optimal scheduling, we also optimize the transmission power at the mobile device, i.e., P_{up}. Note that in the scheduling scheme to be presented in this section, it is assumed that the MU's call graph is a directed tree.

6.2.1 Optimal Scheduling for Given Radio Parameters

Figure 3.3 shows an scheduling graph for a time frame consisting of four time slots and one of its corresponding assignment trees. Each node of the scheduling graph corresponds to the execution of a task in a given time slot and at a given cloudlet location. As shown in Figure 3.3, in time slot t_1, the local, shallow and deep cloudlets are all available to execute task v_1. However, as task v_1 initiates the application, it is required to be executed locally. Therefore, task v_1 is scheduled only at the local location. In time slot t_2, while the local and the deep locations are available to execute task v_2, the field and shallow locations are unavailable due to for example peak load at the corresponding SCeNBs. Accordingly, task v_2 is scheduled to be executed either locally or at the deep cloudlet. Moreover, we assume that the execution of task v_2 takes more than the duration of one time slot. Therefore, no matter which locations are available during time slot t_3, child tasks v_3 and v_4 have to wait until the execution of parent task v_2 is completed, i.e., time slot t_4. Then, tasks v_3 and v_4 can be scheduled at the local, field and deep locations. We assume that two tasks cannot be scheduled at the same cloudlet location in one time slot. In fact, if task v_3 is scheduled to be executed locally, task v_4 has to be executed either at the field cloudlet or the deep cloudlet. An assignment graph also has some distinguished nodes including one source node and several terminal nodes. In particular, there is one terminal node for each leaf node of the call tree.

Note that each scheduling of the tasks to different cloudlet locations and different time slots corresponds to a subgraph of the scheduling graph. The subgraph plus the source and the terminal nodes is called an assignment tree, and it connects the source node to all the terminal nodes. The weight of an edge on the assignment tree connecting parent task v_i, executed at cloudlet x during time slot t_n, to child task v_j, executed at cloudlet y during time slot t_m, is equal to $ETC_y^{t_m}(v_j) + ETC_{x,y}^{t_n,t_m}(D_{v_i,v_j})$. The ETC of the source and the terminal nodes as well as the weight of the edges that connect the leaf tasks to the terminal nodes are assumed to be zero (see the assignment tree in Figure 6.3). Moreover, the weight of each assignment tree which indicates the ETC of that assignment is established by the sum of the weights of all edges in it. Therefore, the optimal assignment corresponds to the assignment tree which has the minimum weight. The minimum weight assignment tree of an application, which involves M tasks, N time slots, and $|\mathcal{AC}| + 1$ cloudlet locations, can be found by dynamic programming with complexity $\mathcal{O}(M \times N^2 \times (|\mathcal{AC}|+1)^2)$ [11].

6.2.2 Optimal Scheduling While Optimizing Transmission Power

In this section, we are interested in both finding the optimal scheduling and optimizing the transmission power at the mobile device, i.e., P_{up}. We show in the following theorem that the optimal scheduling and the optimization of the transmission power are disjoint optimization problems that can be solved independently.

Theorem 6.2.1. *The scheduling optimization problem and the transmission power optimization are disjoint optimization problems.*

Proof. We first assume that the transmission power is given. Then, following the optimal scheduling scheme for given radio parameters, the optimal scheduling corresponds to the assignment tree that has the minimum weight. On the other hand, according to the defined ETCs, factors $\frac{\zeta_1 P_{up}+\zeta_2}{C_{up}(P_{up})}$ appear on the weight of an assignment

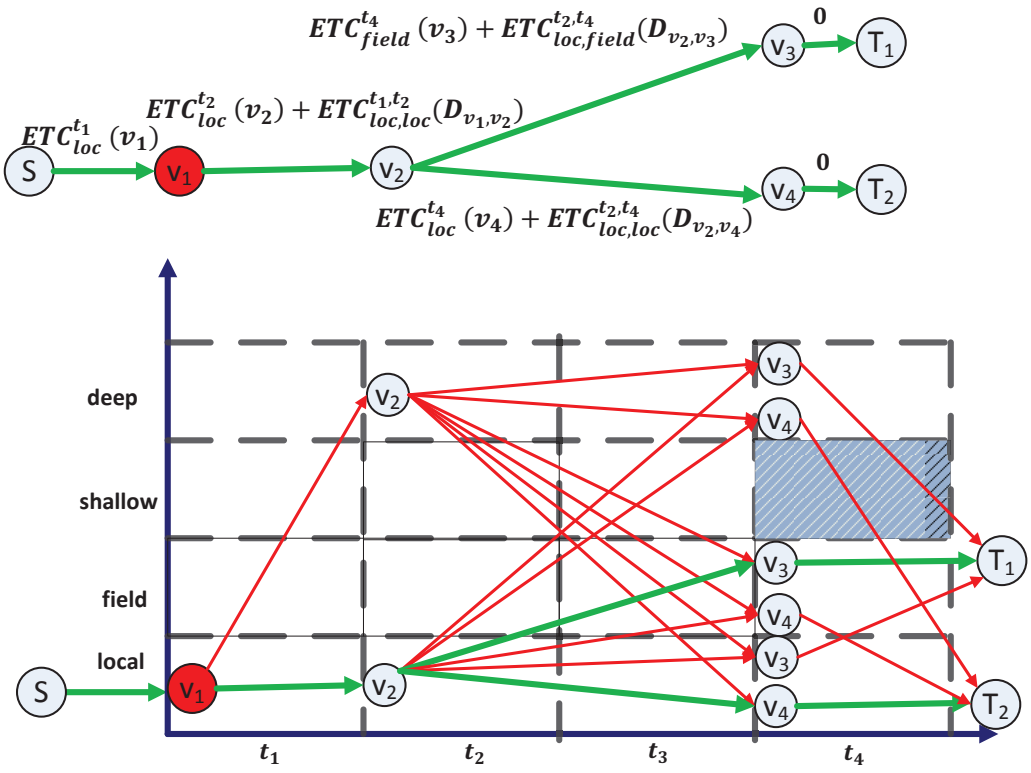


Figure 6.3 Scheduling graph and one of the corresponding assignment trees.

tree. Therefore, one can first minimize $\frac{\zeta_1 P_{up}+\zeta_2}{C_{up}(P_{up})}$ by optimizing P_{up} and then find the optimal scheduling for the given optimal P_{up}. The proof is complete. □

Therefore, we propose a disjoint optimization framework in which we first solve the following optimization problem to find the optimal transmission power,

$$\underset{0\le P_{up}\le P_{max}}{\text{minimize}} \quad \frac{\zeta_1 P_{up} + \zeta_2}{C_{up}(P_{up})} \tag{6.9}$$

Then, we follow the optimal scheduling scheme in the previous section for the given optimal transmission power. The optimization problem in (6.9) becomes strictly convex with the change of variables $Z = C_{up}(P_{up})$ [26] and thus can be solved by efficient convex optimization techniques.

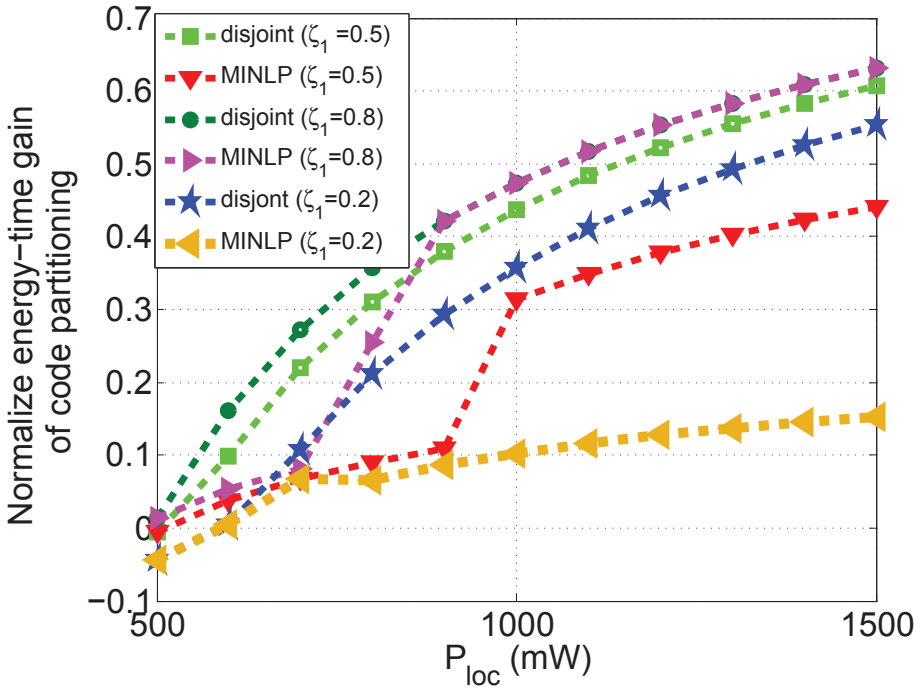


Figure 6.4 Normalized energy-time gain of code partitioning versus local processing power.

6.3 Simulation Results

In this section, we evaluate the results of the proposed optimal code partitioning scheme. To this end, we consider the call tree of Figure 6.2 and assume $\lambda_{v_1} = 2$ M cycles. We also set $\lambda_{v_2} = 18.1$ M cycles, $\lambda_{v_3} = 92.6$ M cycles, $\lambda_{v_4} = 256.1$ M cycles, $D_{v_1,v_2} = 182$ kB, $D_{v_2,v_3} = 4675$ kB and $D_{v_2,v_4} = 13860$ kB [24]. Moreover, we consider an scheduling graph consisting of four time slots and we assume the computational capacity of a cloudlet location during a time slot is fixed and between 10 to 14 G cycles per second if it is available, and is equal to zero otherwise. The local computational capacity is also set to 100 M cycles per second. In terms of the uplink channel, we set the channel bandwidth to 5 MHz, the transmission power budget constraint to 100 mW, and the background noise to -100 dBm [18]. For performance evaluations, we define the normalized energy-time gain as $\frac{ETC_{all\ \ locall} - ETC_{code\ \ partitioning}}{ETC_{all\ \ locall}}$ where $ETC_{all\ \ locall}$ is the ETC incurred if all the tasks are executed locally. Figure 6.4 compares the normalized energy-time gain of the proposed scheme with MINLP

model. The OPTI toolbox combined with NOMAD, which is an excellent derivative free MINLP solver, is used to solve the MINLP problem. As demonstrated in Figure 3.4, the proposed scheme performs better than the MINLP model. This result is attributed to the fact that in the proposed scheme we first optimize P_{up} and then carry out the scheduling optimization for the optimal value of P_{up}.

CHAPTER 7

CONCLUSION

In this dissertation, first, we have developed a new model to maximize the profit of running geographically dispersed data centers. Our model considers multiple classes of service and takes into account of individual SLA-deadline for each type of service. The proposed model is elaborated by taking into consideration of geographical electricity price diversity due to different electricity markets at each data center's location and the availability of renewable energy. Based on the developed model, we have designed an optimization-based workload distribution scheme that relies on the accuracy of G/D/1 queue in characterizing the workload distribution and the workload decomposition to the green and brown workloads. We have also proven the convexity of the formulated optimization problem and evaluated the performance of our workload distribution scheme via extensive simulations.

Second, we have developed a new information flow graph based model for geo-dispersed data centers. Based on the developed model, we have derived a fundamental tradeoff between the total and brown power consumption. Furthermore, we have characterized the achievable points on this tradeoff in which one can know how much green energy is possibly utilized for a given amount of total power consumption budget.

Third, we have proposed a new hierarchical architecture in the context of mobile edge computing called HI-MEC. Specifically, we have introduced the concept of field, shallow and deep cloudlets deployed in three hierarchical levels in accordance with the principle of LTE-advanced mobile backhaul network. Based on the proposed model, a two time scale optimization approach for resource allocation is introduced. In particular, a BLP is formulated to maximize an auction-based profit for concurrent

VM pricing and VM distribution, and accordingly heuristic algorithms are designed to solve this problem in a reasonable time. A convex optimization problem for bandwidth allocation is formulated and a centralized solution to this problem is derived. The proposed hierarchical model and the two time scale optimization platform have been demonstrated to effectively facilitate the resource allocation to the subscribers of an MEC network.

Fourth, we have proposed a new hierarchical capacity provisioning scheme based on accurate queueing analysis. Specifically, we have considered a 2-tier edge computing network architecture consisting of shallow and deep cloudlets, and explored both the case that the network delay between the shallow cloudlets and the deep cloudlet is negligible as well as the case in which the deep cloudlet is located somewhere deeper in network. We have formulated optimization problems for each case and investigated the solution to each problem by using stochastic ordering and optimization algorithms. We have also validated the performance of our capacity provisioning scheme via extensive simulations.

Fifth, we have proposed a task scheduling scheme for offloading computation over time and the hierarchical mobile edge. To this end, we have studied two different optimization scenarios. In particular, in the first scenario, we have found an optimal task scheduling for given radio parameters. In the second scenario, we have investigated the joint optimization of task scheduling and the mobile device's transmission power in which we have showed that by using the scheduling task in this letter, the problem of optimizing the transmission power becomes a disjoint problem from the scheduling problem.

Finally, in line with this dissertation's research, we recently focused on Non Orthogonal Multiple Access (NOMA) and MEC, two of the emerging technologies of 5G, and proposed a novel MEC aware NOMA technique for 5G networks [44]. Our proposed scheme is motivated by the fact that the joint allocation of communication

and computing resources greatly improves the performance of the system. In other words, it may happen that one type of resources is wasted due to congestion of other type of resources. While several works have investigated the joint allocation of computing and communication resources, none of the existing works consider a joint optimization technique in the context of NOMA with consideration of intra-cell interferences. To this end, we aimed to address the aforementioned issue by proposing a joint optimization technique to allocate the computing and communication resources based on the requirements of both MEC and NOMA. We proposed a novel NOMA augmented edge computing model that captures the gains of uplink NOMA in MEC users' energy consumption. Specifically, we designed a NOMA based optimization framework that minimizes the energy consumption of MEC users via optimizing the user clustering, computing and communication resource allocation, and transmit powers. Similar to frequency Resource Blocks (RBs), we defined the notion of computing RBs and investigated the joint allocation of the frequency and computing RBs. More importantly, we designed an efficient heuristic algorithm for user clustering and RBs allocation, and formulated a convex optimization problem for the transmission power control to be solved independently per NOMA cluster.

BIBLIOGRAPHY

[1] Amazon ec2. `https://aws.amazon.com/ec2`. Accessed: 11/23/2016.

[2] Cvx: Matlab software for disciplined convex programming, version 2.0. `http://cvxr.com/cvx`. Accessed: 11/23/2016.

[3] Gurobi optimization. `http://www.gurobi.com`. Accessed: 11/23/2016.

[4] Requests made to the 1998 World Cup Web site. `http://ita.ee.lbl.gov/html/contrib/WorldCup.html`. Accessed: 8/13/2013.

[5] Expanding renewable energy options for companies through utility-offered. *Google White Paper*, https://www.google.com/green/pdf/renewable-energy-options.pdf, April 19, 2013.

[6] M. Abramowitz and I. Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Number 55. Courier Corporation, 1965.

[7] P. Areekul, T. Senjyu, H. Toyama, and A. Yona. A hybrid arima and neural network model for short-term price forecasting in deregulated market. *IEEE Transactions on Power Systems*, 25(1):524–530, 2010.

[8] S. Barbarossa, S. Sardellitti, and P. Di L. Communicating while computing: Distributed mobile cloud computing over 5g heterogeneous networks. *IEEE Signal Processing Magazine*, 31(6):45–55, 2014.

[9] C. Belady, D. Azevedo, M. Patterson, J. Pouchet, and R. Tipley. Carbon usage effectiveness (CUE): A green grid data center sustainability metric. *The Green Grid White Paper*, 2010.

[10] J. Bezdek, R. Hathaway, R. Howard, C. Wilson, and M. Windham. Local convergence analysis of a grouped variable version of coordinate descent. *Springer Journal of Optimization Theory and Applications*, 54(3):471–477, 1987.

[11] S. H. Bokhari. A shortest tree algorithm for optimal assignments across space and time in a distributed processor system. *IEEE Transaction on Software Engineering*, (6):583–589, 1981.

[12] F. Bonomi. Connected vehicles, the internet of things, and fog computing. In *ACM International Workshop on Vehicular Inter-Networking (VANET), Las Vegas, USA*, pages 13–15, 2011.

[13] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the internet of things. In *ACM Workshop on Mobile Cloud Computing (MCC)*, pages 13–16, 2012.

[14] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2009.

[15] R. E. Brown, R. Brown, E. Masanet, B. Nordman, B. Tschudi, A. Shehabi, J. Stanley, J. Koomey, D. Sartor, P. Chan, et al. Report to congress on server and data center energy efficiency: Public law 109-431. Technical report, Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA (US), 2007.

[16] J. Cai and K. S. Tan. Optimal retention for a stop-loss reinsurance under the VaR and CTE risk measures. *Astin Bulletin*, 37(01):93–112, 2007.

[17] A. Ceselli, M. Premoli, and S. Secci. Cloudlet network design optimization. In *IEEE IFIP Networking*, pages 1–9, 2015.

[18] X. Chen. Decentralized computation offloading game for mobile cloud computing. *IEEE Transaction Parallel Distributed Systems*, 26(4):974–983, 2015.

[19] Y. Chen, S. Jain, V. K. Adhikari, and Z. Zhang. Characterizing roles of front-end servers in end-to-end performance of dynamic content distribution. In *ACM SIGCOMM Conference on Internet Measurement*, pages 559–568, 2011.

[20] Y. Cheng and J.S. Pai. The maintenance properties of nth stop-loss order. In *International ASTIN Colloquium*, volume 95, page 118, 1999.

[21] M. Chiang and T. Zhang. Fog and IoT: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6):854–864, 2016.

[22] S. Clinch, J. Harkes, A. Friday, N. Davies, and M. Satyanarayanan. How close is close enough? understanding the role of cloudlets in supporting display appropriation by mobile users. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 122–127, 2012.

[23] P. Corcoran and S. K. Datta. Mobile-edge computing and the internet of things for consumers: Extending cloud computing and services to the edge of the network. *IEEE Consumer Electronics Magazine*, 5(4):73–74, 2016.

[24] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. Maui: making smartphones last longer with code offload. In *ACM Conference on Mobile Systems, Applications, and Services*, pages 49–62, 2010.

[25] P. Delforge. America's data centers are wasting huge amounts of energy. *National Resources Defense Council*, Issue Brief 14-08-A, August 2014.

[26] P. Di Lorenzo, S. Barbarossa, and S. Sardellitti. Joint optimization of radio resources and code partitioning in mobile edge computing. *arXiv preprint arXiv:1307.3835v3*, 2016.

[27] A.G. Dimakis, P. Godfrey, Y. Wu, M.J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Transactions on Information Theory*, 56(9):4539–4551, 2010.

[28] B. Diniz, D. Guedes, Jr. W. Meria, and R. Bianchini. Limiting the power consumption of main memory. In *34th annual Symposium on Computer Architecture*, pages 290–301, 2007.

[29] M. Ghamkhari and H. Mohsenian-Rad. Optimal integration of renewable energy resources in data centers with behind-the-meter renewable generator. In *IEEE International Conference on Communications (ICC)*, pages 3340–3344, 2012.

[30] M. Ghamkhari and H. Mohsenian-Rad. Energy and performance management of green data centers: A profit maximization approach. *IEEE Transactions on Smart Grid*, 4(2):1017–1025, 2013.

[31] M. Ghamkhari, H. Mohsenian-Rad, and A. Wierman. Optimal risk-aware power procurement for data centers in day-ahead and real-time electricity markets. In *IEEE Conference on Computer Communications (INFOCOM WKSHPS)*, page 2014.

[32] N. M. Gonzalez, W. A. Goya, R. Fatima P., K. Langona, E. A. Silva, Tereza C. M. Brito C., C. C. Miers, J. Mångs, and A. Sefidcon. Fog computing: Data analytics and cloud distributed processing on the network edges. In *IEEE International Conference of the Chilean*, pages 1–9, 2016.

[33] J. Guo, F. Liu, D. Zeng, J. CS Lui, and H. Jin. A cooperative game based allocation for sharing data center networks. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 2139–2147, 2013.

[34] K. Ha, Y. Abe, Z. Chen, W. Hu, B. Amos, P. Pillai, and M. Satyanarayanan. Adaptive vm handoff across cloudlets. *Technical Report CMU-CS-15–113, CMU School of Computer Science*, 2015.

[35] X. Hu, H. Yang, and L. Zhang. Optimal retention for a stop-loss reinsurance with incomplete information. *Insurance: Mathematics and Economics*, 65:15–21, 2015.

[36] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young. Mobile edge computing-A key technology towards 5G. *ETSI white paper*, 11(11):1–16, 2015.

[37] W. Hürlimann. Extremal moment methods and stochastic orders. *Application in Actuarial Science. Monograph manuscript (available from the author)*, 1998.

[38] W. Hürlimann. Higher degree stop-loss transforms and stochastic orders(i) theory. *Springer Blätter der DGVFM*, 24(3):449–463, 2000.

[39] M. Jutila. An adaptive edge router enabling internet of things. *IEEE Internet of Things Journal*, 3(6):1061–1069, 2016.

[40] S. Khalili and O. Simeone. Inter-layer per-mobile optimization of cloud mobile computing: a message-passing approach. *Transactions on Emerging Telecommunications Technologies*, 27(6):814–827, 2016.

[41] A. Kiani and S. Akhlaghi. Selective regenerating codes. *IEEE Communications Letters*, 15(8):854–856, 2011.

[42] A. Kiani and N. Ansari. Towards low-cost workload distribution for integrated green data centers. *IEEE Communications Letters*, 19(1):26–29, 2015.

[43] A. Kiani and N. Ansari. A fundamental tradeoff between total and brown power consumption in geographically dispersed data centers. *IEEE Communications Letters*, 20(10):1955, 2016.

[44] A. Kiani and N. Ansari. Edge computing aware NOMA for 5G networks. *IEEE Internet of Things Journal*, 5(2), 2018.

[45] A. Kiani and N. Ansari. Profit maximization for geographical dispersed green data centers. *IEEE Transactions on Smart Grids*, 9(2), 2018.

[46] H. S. Kim and N. B. Shroff. Loss probability calculations and asymptotic analysis for finite buffer multiplexers. *IEEE/ACM Trans. on Networking (TON)*, 9(6):755–768, 2001.

[47] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang. Power and performance management of virtualized computing environments via lookahead control. *Cluster computing*, 12(1):1–15, 2009.

[48] U. Lampe, M. Siebenhaar, A. Papageorgiou, D. Schuller, and R. Steinmetz. Maximizing cloud provider profit from equilibrium price auctions. In *IEEE International Conference on Cloud Computing*, pages 83–90, 2012.

[49] K. Le, R. Bianchini, M. Martonosi, and T. D. Nguyen. Cost-and energy-aware load distribution across data centers. In *Workshop on Power Aware Computing and Systems*, 2009.

[50] G. A. Lewis, S. Echeverría, S. Simanta, B. Bradshaw, and J. Root. Cloudlet-based cyber-foraging for mobile systems in resource-constrained edge environments. In *ACM International Conference on Software Engineering*, pages 412–415, 2014.

[51] J. Li, Z. Li, K. Ren, and X. Liu. Towards optimal electric demand management for internet data centers. *IEEE Transactions on Smart Grid*, 3(1):183–192, 2012.

[52] Z. Liu, M. Lin, A. Wierman, S. Low, and L.L. Andrew. Greening geographical load balancing. *IEEE/ACM Transactions on Networking (TON)*, 23(2):657–671, 2015.

[53] T. Pering, T. Burd, and R. Brodersen. Dynamic voltage scaling and the design of a low-power microprocessor system. In *Power Driven Microarchitecture Workshop*, 1998.

[54] L. Rao, X. Liu, L. Xie, and W. Liu. Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 1–9, 2010.

[55] L. Rao, X. Liu, L. Xie, and W. Liu. Coordinated energy cost management of distributed internet data centers in smart grid. *IEEE Transactions on Smart Grid*, 3(1):50–58, 2012.

[56] R. Reijnen, W. Albers, and W. Kallenberg. Approximations for stop-loss reinsurance premiums. *Elsevier Insurance: Mathematics and Economics*, 36(3):237–250, 2005.

[57] J. Robson. Small cell backhaul requirements. *NGMN White Paper*, pages 1–40, 2012.

[58] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for VM-based cloudlets in mobile computing. *IEEE pervasive Computing*, 8(4):14–23, 2009.

[59] M. Satyanarayanan, G. Lewis, E. Morris, S. Simanta, J. Boleng, and K. Ha. The role of cloudlets in hostile environments. *IEEE pervasive Computing*, 12(4):40–49, 2013.

[60] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, and B. Amos. Edge analytics in the internet of things. *IEEE Pervasive Computing*, 14(2):24–31, 2015.

[61] S. S. Soman, H. Zareipour, O. Malik, and P. Mandal. A review of wind power and wind speed forecasting methods with different time horizons. In *IEEE North American Power Symposium (NAPS)*, pages 1–8, 2010.

[62] K. S. Tan, C. Weng, and Y. Zhang. VaR and CTE criteria for optimal quota-share and stop-loss reinsurance. *Taylor & Francis North American Actuarial Journal*, 13(4):459–482, 2009.

[63] K. S. Tan, C. Weng, and Y. Zhang. Optimality of general reinsurance contracts under CTE risk measure. *Elsevier Insurance: Mathematics and Economics*, 49(2):175–187, 2011.

[64] A. Wierman, Z. Liu, I. Liu, and H. Mohsenian-Rad. Opportunities and challenges for data center demand response. In *IEEE Green Computing Conference (IGCC)*, pages 1–10, 2014.

[65] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowtron. Better never than late: Meeting deadlines in datacenter networks. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 50–61, 2011.

[66] L. Wu and M. Shahidehpour. A hybrid model for day-ahead price forecasting. *IEEE Transactions on Power Systems*, 25(3):1519–1530, 2010.

[67] Y. Zhang and N. Ansari. On architecture design, congestion notification, TCP incast and power consumption in data centers. *IEEE Communications Surveys and Tutorials*, 15(1):39–64, 2013.

[68] Y. Zhang, D. Niyato, and P. Wang. An auction mechanism for resource allocation in mobile cloud computing systems. In *Springer International Conference on Wireless Algorithms, Systems, and Applications*, pages 76–87, 2013.

[69] J. Zhao, H. Li, C. Wu, Z. Li, Z. Zhang, and F. Lau. Dynamic pricing and profit maximization for the cloud with geo-distributed data centers. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 118–126, 2014.

[70] L. Zheng, C. Joe-Wong, C. W. Tan, M. Chiang, and X. Wang. How to bid the cloud. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 71–84, 2015.