Leveraging Workload Relocation and Resource Pruning for Electricity Cost Minimization in Service Provider Networks

PhD Thesis

Muhammad Saqib Ilyas

2005-06-0024

Advisor: Dr. Zartash Afzal Uzmi

Department of Computer Science

School of Science and Engineering

Lahore University of Management Sciences

Dedicated to dedication

Lahore University of Management Sciences

School of Science and Engineering

CERTIFICATE

I hereby recommend that the thesis prepared under my supervision by ***Muhammad Saqib Ilyas*** titled ***Leveraging Workload Relocation and Resource Pruning for Electricity Cost Minimization in Service Provider Networks*** be accepted in partial fulfillment of the requirements for the degree of doctor of philosophy in computer science.

Dr. Zartash Afzal Uzmi (Advisor)

Recommendation of Examiners' Committee:

Name	**Signature**
Dr. Zartash Afzal Uzmi	————————
Dr. X	————————
Dr. Y	————————
Dr. Z	————————

Acknowledgements

Contents

1 Introduction **1**

1.1 Networks and systems pervade 1

1.2 Electricity costs in networks and systems 2

1.3 Energy inefficiency characterizes today's networks 3

1.4 Prevalent electricity cost reduction techniques 5

1.4.1 Reducing the amount of energy consumed 6

1.4.2 Using cheaper electricity - Workload Relocation (WR) 7

1.5 Our thesis 8

1.6 Contributions 8

1.7 Organization 9

2 Background - Different Types of Networks and Their Similarities **10**

2.1 Geo-Diverse Data Centers 10

2.1.1 Structure 11

2.1.2 Request routing 14

2.1.3 Power consumption model 17

2.2 Cellular Networks 18

2.2.1 Structure 19

2.2.2 Call placement 21

2.2.3 Power consumption model 22
2.3 Similarities between different types of networks 23
2.4 Differences between different types of networks 23

3 A generalized framework for electricity cost optimization 25
3.1 Problem Model 26
3.1.1 Illustrative Example 26
3.1.2 Problem complexity 28
3.2 Optimization problem formulation 31
3.2.1 The objective function 31
3.2.2 The constraints 32
3.2.3 Comments on the problem formulation 33

4 Case Study I: Geo-diverse Data Centers 35
4.1 Prelude 35
4.2 Related work 37
4.3 Sources of transition costs in the data center scenario 40
4.4 Instantiating the generalized optimization formulation 41
4.5 Experimental setup 50
4.5.1 Application workload 50
4.5.2 Electricity prices 51
4.5.3 Algorithms for Workload Distribution/Relocation 51
4.6 Results 55
4.6.1 Sensitivity of electricity cost savings to extent of overprovisioning . . 56
4.6.2 Sensitivity of electricity cost savings to magnitude of transition costs 57
4.6.3 Sensitivity of electricity cost savings to resource pruning granularity . 58
4.6.4 Sliding window re-optimization 60

4.7 Sensitivity of electricity cost savings to the server idle-peak power ratio . . . 66
4.8 Performance of the heuristic algorithm 66
4.9 Discussion 67

5 Case Study II: Cellular Networks **69**

5.1 Instantiating the generalized optimization formulation 69
5.2 Experimental setup 69
5.3 Results 69
5.3.1 Sensitivity of electricity cost savings to the duration of an optimization interval 69
5.3.2 Sensitivity of electricity cost savings to the resource pruning granularity 70
5.3.3 Sensitivity of electricity cost savings to the margin of state-change damping 70
5.4 Discussion 70

6 Conclusions and Future Work **71**

6.1 Contributions 71
6.2 Limitations 71
6.3 Future work 71

List of Figures

1.1 Networks lack energy proportionality 4
1.2 Call traffic for an oeprational cellular site over two days 5

2.1 Google Data Center Locations - Source: royal.pingdom.com 12
2.2 The modern data center's architecture 13
2.3 Resolving the IP address for a server hosted in a data center 16

3.1 An example of mapping variable workload to capacity-limited network resources with geo-temporal diversity in electricity prices. Three consecutive intervals t_1, t_2 and t_3 are considered. Workload and electricity prices may only change between two consecutive intervals. (a) Workload considered in this example. (b) Electricity prices for the locations at which the two network resources are situated. (c) A uniform mapping of workload to network resources does not exploit electricity price diversity. (d) Mapping workload to network resources in order of their current electricity price. Due to lack of energy proportionality, only slight savings in electricity cost are possible. (e) Deactivating idle resources alongwith the resource mapping strategy of (d) may result in significant electricity cost savings. 29

4.1 Normalized workload 50
4.2 Workload intensity histogram 50

4.3 Percentage savings with over-provisioning 55
4.4 Total cost vs transition overhead 58
4.5 Cost estimation error due to workload estimation error 58
4.6 Flow for Sliding Window Optimization Experiments 63
4.7 Mean absolute workload prediction error vs sliding window size 63
4.8 Distribution of workload prediction error for sliding window size of 12 hours 63
4.9 Local trajectory correction technique for three consecutive intervals 63
4.10 Percentage error of sliding window forecasts compared to global optimal with error-free workload 65
4.11 Cost saving vs (de)activation granularity 65
4.12 For bs = 0.01 66
4.13 For bs = 0.65 67
4.14 The minimum, maximum and average percentage difference between the cost of our heuristic and RED-BL 68

List of Tables

4.1 Data Center Network Model Parameters 44
4.2 Sources of electricity prices used in our work 51
4.3 Algorithms compared in our work 52
4.4 A comparison of the algorithms studied in this paper 52

Abstract

Abstraction

Chapter 1

Introduction

1.1 Networks and systems pervade

Different types of networks play a critical role in our every day lives. We use Public Switched Telephone Networks (PSTN) and cellular networks to communicate with people by making voice calls (and sending text messages in case of cellular networks). PSTN and cellular networks also server as access media for connecting to the Internet, which offers several key services. We communicate and collaborate using email, voice/video calls over Internet Protocol (IP) and social networks. We also use the Internet to access teaching/learning material, course registration systems on campus and even pathological examination reports.

The Internet itself is an interconnection of several different types of networks. First, there are the packet-switched networks operated by Internet Service Providers (ISPs) that provide us access to Internet resources worldwide by carry information between hosts on the Internet. A second type of networks that the Internet is comprised of are the geo-diverse data centers operated by companies like Facebook, Amazon, Microsoft and Google. Servers in these data center networks run applications like Google Search, Gmail, Youtube, Twitter, Bing and Facebook. A third type of network which are also part of the Internet are the

Content Distribution Network (CDN), that place mutlple copies of Internet resources such as web pages across the globe. The role of CDNs is to keep the latency from a user to an Internet resource small (compared to having the resource located at a fixed single location). For instance, if Google's home page were only located at a server in San Jose, CA, the latency (the time it takes for a web browser to send a packet to the server) for users in Pakistan would be hundreds of milliseconds. Placing a replica of the Google home page close to Pakistan lowers the packet latency significantly, thereby allowing the web browser to display the page much faster.

The deployment of these different types of networks involves huge expenses. For instance, Google announced building a data center in Iowa at a cost of $400 Million [1]. Furthermore, according to [2], the capital cost of a typical cellular network site is $550,000[1].

The recurring operational cost of these networks is also quite high. For instance, in 2009, Facebook spent $50 Million on leasing the data center space, alone [3]. In the context of geo-diverse data centers, other contributors to operational expenses include staff salaries, maintenance related costs, the cost of inter-data center network connectivity and electricity bill. Similar trends may also be observed in other types of networks. Optimizing operational costs is critical for network operators in order to offer cost-effective services to consumers and maximize their profit.

1.2 Electricity costs in networks and systems

For many tyeps of networks, electricity costs contribute a significant fraction of operational costs. For instance, electricity costs may be as much as 15% of operational costs in data centers [4]. Similarly, for an operator with 7000 cellular sites in a country as small as Pak-

[1]This does not include spectrum licensing costs. Furthermore, an operator needs to deploy many sites. A site at about every 800 meters is common in urban settings

istan, the annual electricity cost can be roughly estimated at $9.19 Million[2]. Telecom Italia reported a consumption of 1.793 GWh in 2012 [6], which is significantly higher compared to our estimates for the Pakistani network operator and hence the electricity costs are also expected to be much higher.

1.3 Energy inefficiency characterizes today's networks

For most networks, the power consumption is well-approximated as an affine function of workload [7, 8]. Furthermore, these networks are not energy proportional. In Figure 1.1, the green line shows the ideal energy proportional behaviour where the network consumes no power when there is no workload. No real network exhibits this ideal behaviour for one of the following reasons.

1. The network activity under no workload conditions is not significantly less than that under peak workload. For instance, a cellular network's radio components must continue operating and drawing power to offer uninterrupted connectivity to prospective allers, even when no call is in progress. In packet switching networks, many data link layer technologies continuously transmit frames irrespective of traffic activity.

2. The components of the network may not be energy proportional. For instance, in data centers, server power consumption is a large fraction of the total power consumption and the server idle power consumption is a large fraction of their peak power consumption.

A network that is not energy proportional is energy inefficient (i.e., consumes a lot more energy than it should) in the low-workload regimes. It has been observed for many networks

[2]Using a 1.5 kW draw for a single cellular site [5], Rs. 10 per kWh and Rs. 100 per US$. Note that the Rs. 10 per kWh is a gross under-estimation, given that it is the approximate current price of grid power, which is note reliable. In the absence of grid power, diesel generators power a cellular site and the resulting cost per kWh is much higher.

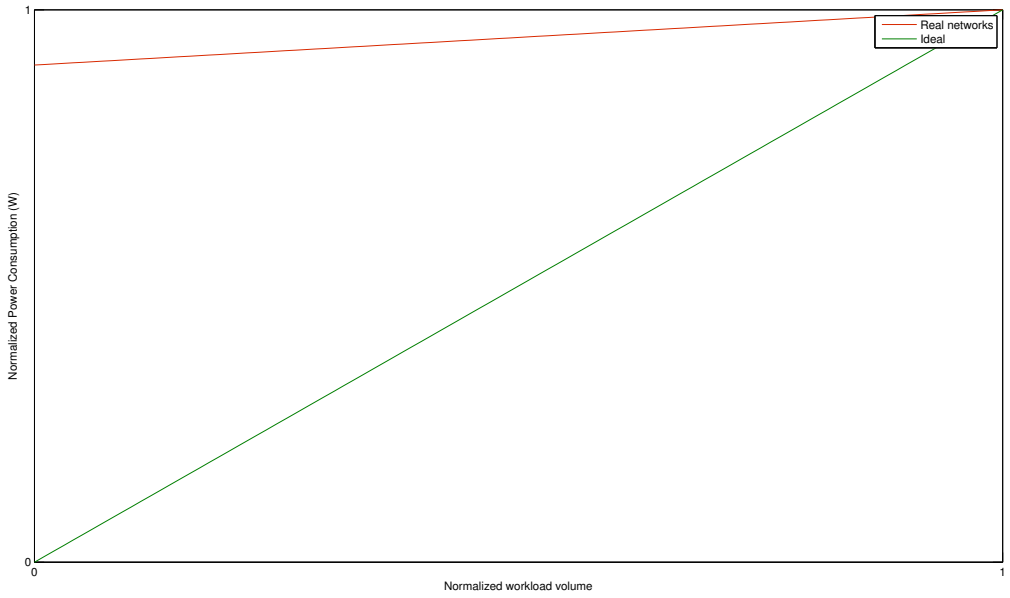

Figure 1.1: Networks lack energy proportionality

that workload is variable and periodic. Figure 1.2 shows the workload for call traffic at a cellular site in a large operational GSM network in Pakistan. It shows that call traffic has diurnal cycles and that traffic peaks for only a short period of time during a day. Furthermore, the workload peak is quite high compared to the trough. ISP [9] and data center [10] traffic also show similar trends. In order to meet peak expected workload amicably, networks are dimensioned according to the peak workload. Since the workload is far from the peak most of the time and networks are not energy proportional, most networks are energy inefficient. Recent years have witnessed significant research effort aimed at improving network energy efficiency in packet networks, cellular networks and geo-diverse data centers. Effectively, such research aims to lower the y-intercept of the red line in Figure 1.1.

We have observed earlier that network electricity costs are quite high. An energy efficient network would only incur high electricity costs if it handled a lot of workload. On the other hand, for energy inefficient networks, such as those prevalent today, high electricity costs

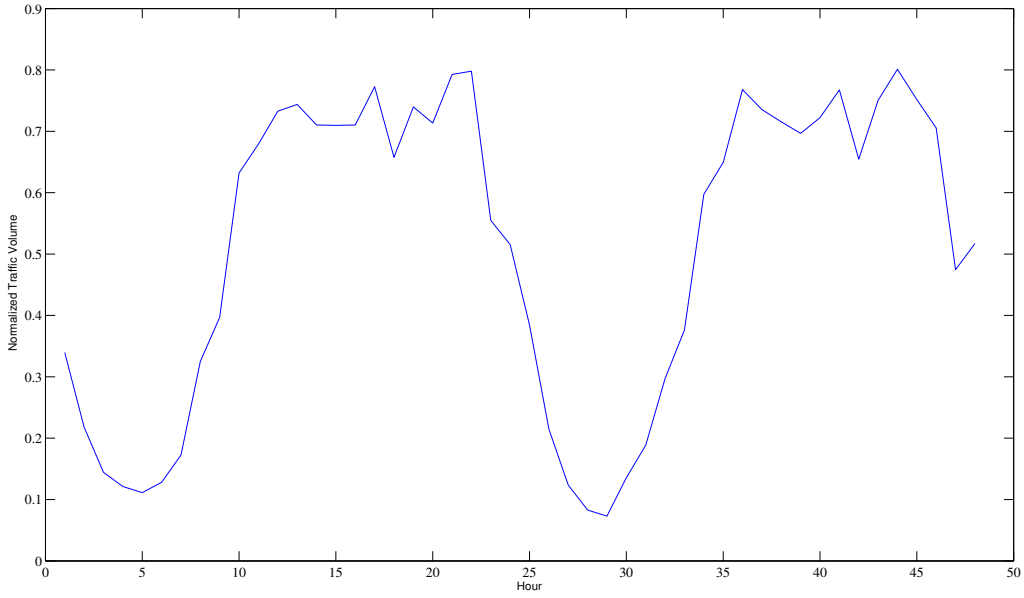

Figure 1.2: Call traffic for an oeprational cellular site over two days

are not justifiable by high workload because the workload is quite variable. In other words, today's networks have very poor performance per Watt characteristics. Therefore, reducing the electricity costs in today's networks is critically important.

1.4 Prevalent electricity cost reduction techniques

The electricity cost for a network during a unit duration of time is given by:

$$\text{Electricity cost} = \text{amount of energy consumed} \times \text{unit price of electricity} \tag{1.1}$$

Conseuqently, the electricity cost for a network may be reduced by minimizing either or both of the terms on the right handside of the above equation. From prior research work and current operational practices in different types of networks, we observe the following techniques to reduce electricity cost in networks by reducing one or both of the two quantities

in equation 1.1.

1.4.1 Reducing the amount of energy consumed

1. **Hardware upgrades:** Due to ecological challenges, improved energy efficiency is generally a key requirement when developing new technologies and devices. For a given workload demand, an improvement in device energy efficiency lowers the amount of energy consumed, thereby reducing electricity cost. Therefore, hardware upgrades are a way to reduce electricity costs. An operator would, however, opt for hardware upgrades in their network only after they have obtained the Return on Ivestment (ROI) of the initial deployment. The initial investment not only involves capital cost of equipment but other factors such as spectrum licensing as well. In the cut-throat competition prevalent in most of today's networks, the ROI is slow to achieve. This means that existing energy efficient networks would stay that way for a considerable tiem into the future.

2. **Hardware virtualization:** With the advent of ever faster CPUs, it was observed that servers tend to operate at relatively low CPU utilization most of the time. This was seen as an opportunity to statistically multiplex multiple servers onto a single physical machine by slicing the latter into multiple virtual servers. In this way, virtualization cuts capital costs for procurement of hardware. Since the virtual servers share the same resources (power supply, CPU, network interface, disks), if two servers are multiplexed onto a single physical server, the electricity consumption may be cut by as much as 50%. A more aggressive server consolidation may cut electricity costs by upto 80% [11].

3. **Resource Pruning (RP):**Since network resources must be deployed according to peak demand while the workload peaks only for a short period of time, the excess resource may be deactivated (shutdown or put in power-saving mode depending on

what is supported by the equipment) when workload is low [12, 13, 14, 15, 7]. When evaluating the reduction in electricity costs through resource pruning, it is imperative to consider any costs associated with activation and deactivation of network resources.

1.4.2 Using cheaper electricity - Workload Relocation (WR)

Electricity prices exhibit geographic diversity [16], i.e., the price of electricity varies from one location to another. The variation in electricity price is generally noticeable only at large distances. For instance, the electricity price anywhere within a city is generally the same[3]. Most networks span large enough distances for geographic diversity in electricity prices to be apparent. If the network workload is quite flexible in terms of where it is handled, then the workload originating at a location with high electricity price may be relocated to a different location that has lower electricity price, thereby cutting electricity cost. We call this technique Workload Relocation (WR). We observe that different networks have different levels of geo-flexibility in workload. In geo-diverse data centers, for instance, the workload is highly geo-flexible, i.e., a client's request may be handled close by or even hundreds of miles away. On the other hand, the workload in cellular networks has very low geo-flexibility, i.e., a call mus tbe handled at a cellular base station within a few hundred meters from the caller.

Electricity prices also exhibit temporal diversity [16], i.e., the relative order of electricity prices at different locations keeps changing. If a city in Kansas presently has chepaer electricity than one in Oklahoma, an hour later, the reverse may be true. This means that mapping of workload to locations must be periodically updated. The granularity of these updates depends on how frequently electricity prices change. Electricity markets exhibit price changes at two different time scales (15 minutes for real-time electricity prices and an hour for day-ahead prices).

[3]With the exception of factors such as different tarriffs for domestic, commercial and industrial consumers

1.5 Our thesis

Based on the similarity in workload characteristics and the dependence of power consumption on workload, we opine that a generalized power optimization framework may be formulated that is applicable to many different types of networks. Our generalized electricity cost optimization framework would use workload relocation and resource pruning in tandem to reduce electricity costs[4].

1.6 Contributions

This thesis makes the following contributions:

- We present a generalized model for electricity cost optimization applicable to different types of networks that jointly uses workload relocation and resource pruning. We show that this problem is NP-Hard.
- We present a framework called Relocate Energy Demand to Better Locations (RED-BL), pronounced Red Bull, that solves this problem. We apply RED-BL to geo-diverse data centers as well as cellular networks using real data traces.
- We exactly solve some reasonably-sized instances of this problem using real data. We also propose some heuristics that would be useful for larger instances of the problem.
- We evaluate RED-BL on two different types of networks, namely, geo-diverse data centers and cellular networks.
- Prior efforts in this area had mostly ignored the costs associated with activation and deactivation of network resources. To the best of our knowledge, we are the first to incorporate these in our optimization framework.

[4]Hardware virtualization is complimentary to our framework

- We evaluate the benefits of geographical diversity exhibited by electricity prices and network deployments.

- A network with significant overprovisioning may handle most of the workload at cheaper locations while the more expensive ones may be pruned from the network. In other words, geographic diversity in electricity prices incentivises over-provisioning. We study the benefits of increased over-provisioning and find diminishing returns when increasing over-provisioning.

1.7 Organization

The rest of the document is structured as follows. In Chapter 2, we compare two different types of networks and describe how similar they are in terms of workload handling and power consumption. In chapter 3, we derive a generalized power consumption model, applicable to different types of networks and formulate RED-BL, a generalized electricity cost optimization problem. We present an evaluation of RED-BL on geo-diverse data centers and cellular networks in chapters 4 and 5, respectively. In chapter 6, we draw the conslusions about our thesis and provide some future directions.

Chapter 2

Background - Different Types of Networks and Their Similarities

In this thesis, we claim that many different types of networks are quite similar in terms of power consumption. In this chapter, we take an essentials-only look at two different types of networks with a view to establishing the similarity between them. This similarity motivates the formulation of a generalized electricity cost optimization framework.

2.1 Geo-Diverse Data Centers

Organizations like Microsoft, Facebook, Amazon and Google run a plethora of applications. Some of these applications are accessible by the general public. Google Docs is one such application. Such organizations also run private applications for the consumption of authorized internal users only. These applications run on servers that are hosted at sites called data centers.

A data center is a site that has equipment such as servers, storage and networking equipment, in addition to some allied equipment such as airconditioning and power supplies.

A given data center may host only public applications, only private applications or even both. Furthermore, some public data center operators allow a client to host their own applications, whereas some only offer a fixed set of internally developed applications. For instance, one may run a custom application on a server leased on Amazon's data centers, but on the other hand, Google's search cluster only hosts the Google search application.

Operators typically deploy multiple data centers at different geographic locations. This is done for two reasons. First, having data centers at different locations provides fault tolerance. If one site goes down for some reason, the other site may take over as a backup. Also, multiple remote sites are less likely to affected simultaneously by a natural disaster. A second reason to have multiple data centers is to have low latency to clients at different locations. For instance, Amazon has multiple data centers in different continents, thereby ascertaining that no matter where a client may be, there is an Amazon data center relatively close by compared to the case if Amazon only had one data center in the US. Figure 2.1 shows the locations of Google's data centers across the globe (according to royal.pingdom.com as of April 2008).

2.1.1 Structure

Before delving into the internal structure and composition of a data center, let us consider a data center as a single resource. This view helps provide only the high-level details of an operator's network. At this level, each one of an operator's data centers are inter-connected by means of high-speed inter data center network links. These links serve to carry various types of traffic, some of which are given below:

- **Consistency traffic:** To maintain consistency amongst replicas of an application's servers hosted in different data centers, some overhead in terms of network traffic must be incurred. For instance, a customer's website may be hosted at two different data

Figure 2.1: Google Data Center Locations - Source: royal.pingdom.com

centers and whenever a change is made to one copy of the website, the same changes must be reflected at the replica as well.

- **Traffic due to load-balancing:** Some traffic on the inter-data center links may be a result of the effort to achieve load-balancing amongst the data centers. For instance, the data centers may be oeprated by a web-based email service provider and the user inboxes may be partitioned over the data centers. In this case, an operator might desire that a roughly balanced amount of storage be used at each of the data centers. To this end, the operator might want to spread the inboxes over the data centers such that the cumulative size of the inboxes at each data center is roughly the same. Over time, due to changes in user behaviour and activity, the operator would need to re-adjust the inboxes assigned to each data center, thus requiring migration of inboxes between data centers.

- **Background traffic:** Yet another source of inter data center traffic is background

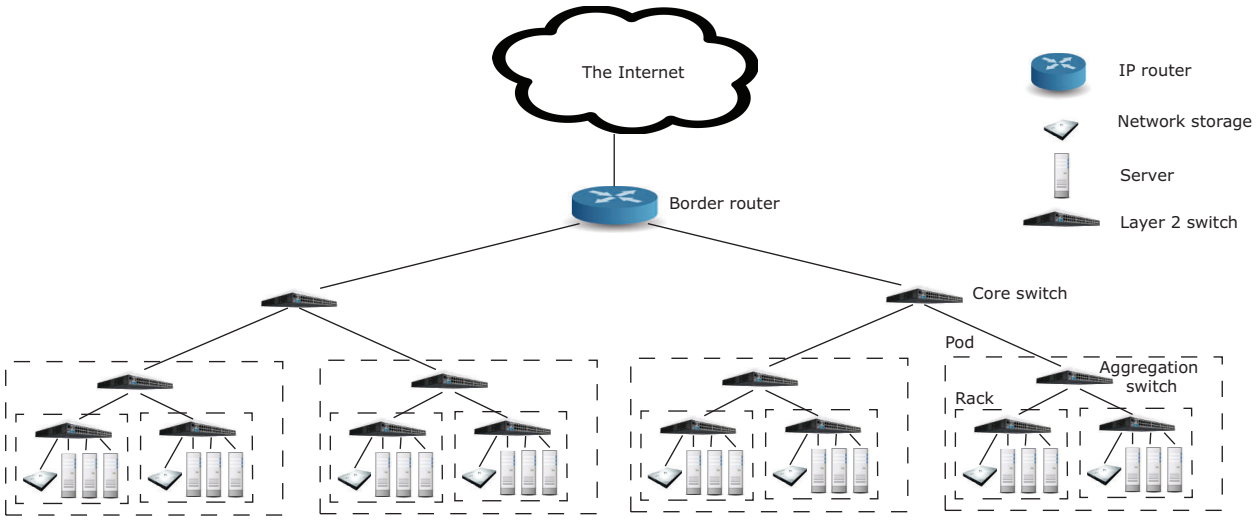

Figure 2.2: The modern data center's architecture

traffic. For isntance, different data centers belonging to an Internet search engine operator may collaboratively compute search results. In this example, the search indices may also be updated periodically in the background.

Having taken a high-level view of a geo-diverse data center operator's network, now let us delve into the internal structure and composition of a data center. Today's data center architecture is heirarchical [17] as shown in Figure 2.2. A typical data center hosts tens of thousands of servers [18]. The servers are installed in vertical racks. Apart from servers, the racks host other equipment as well. In addition to built-in hard drives in the servers, some dedicated storage nodes are also installed in the racks. A high speed Ethernet switch provides interconnection between the devices installed in the rack and connectivity to the rest of the data center and beyond. Power supply and distribution units for the equipment are also installed in the rack.

A group of racks, called a pod (or a cluster), are interconnected by means of aggregattion switches. An aggregation switch allows servers in different racks to communicate with each other. All the pods within a data center are interconnected by core switches. This allows servers in different pods to communicate with each other. The core switches are intercon-

nected through one or more border routers. These border routers are the avenues for traffic coming in and going out of the data center.

All of the equipment is quite tightly packed within a pretty small space in a data center. The equipment generates a lot of heat and to prevent thermal damage to it, cooling must be provided. This is generally done by air-cooling, i.e., heat is transported away from the equipment by circulating cool air around it.

2.1.2 Request routing

As noted in chapter 1, electricity cost depends not only on how much workload is handled, but also where it is handled. In order to develop a model for electricity cost in a geo-diverse data center, we need to first understand how workload from all over the globe is distributed amongst the data centers. In this section, we will use as an example a client request for viewing a web page hosted by a geo-diverse data center operator.

To access a web resource, the user types a uniform resource locator (URL) in the web browser's address bar. The URL typically contains the DNS name corresponding to the web server that hosts the requested resource[1]. Since a single server would hardly be sufficient to handle all traffic for a typical web site, several servers must be mapped to the same DNS name. However, the web browser must connect to exactly one of these servers during a browsing session. Figure 2.3 briefly describes how this web server's IP address is picked. For details on DNS resolution process, see [19, 20].

When the user enters a URL in her web browser, the browser invokes the local Domain Name System (DNS) resolver on the client machine which attempts to determine the IP address corresponding to the DNS name of the remote host specified in the URL. The local DNS server communicates with the DNS server for the client's ISP[2]. The DNS query

[1]It is also possible to specify the IP address of the web server directly in the URL. However, remembering IP addresses for all web sites of interest is not humanly possible

[2]Some people configure other DNS servers, such as Google's Open DNS Servers on their machines. In

eventually reaches the authoritative server for the remote host's domain. In our example, this would be operated by the data center operator. The DNS server for the data center operator resolves the DNS name by returning an IP address corresponding to the DNS name specified by the client. The data center operator's DNS server performs an attempt at load-balancing so that roughly the same amount of workload is sent to each server hosting the requested web site. Notice in Figure 2.3 that caches are available at various DNS resolvers in order to improve the latency of DNS resolution. These caches will keep the IP address corresponding to recently queried DNS names until the timeout specified by the authoritative DNS server expires.

The data center operator has a large pool of IP addresses, also known as IP address space, for their layer 3 devices. This IP address space is segmented over the geo-distributed data centers. The IP address resolved by the operator's DNS server belongs to one of the data centers and the client must now send it's Hyper Text Transfer Prototcol (HTTP) [21] request to the appropriate server at the corresponding data center. The client's web browser now establishes a Transport Control Protocol (TCP) connection with the server. To this end, the client sends packets to the web server's IP address that was just resolved. The packets leave the client's network interface and go to the ISP's gateway. Once in the ISP's network, the routers determine a path to the destination IP address and forward the packets hop by hop until the packets reach the data center where the required web server is hosted.

Having determined the IP address of the web server, the client's web browser establishes an HTTP session with that IP address over a TCP connection. The IP packets belonging to this connection destined to the web server arrive at the border router in the corresponding data center and are forwarded to the server, traversing the core, aggregation and top of rack switches. The response packets are forwarded from the web server to the border router which routes it back to the client machine.

such cases, the local DNS server would communicate with the Google Open DNS Server

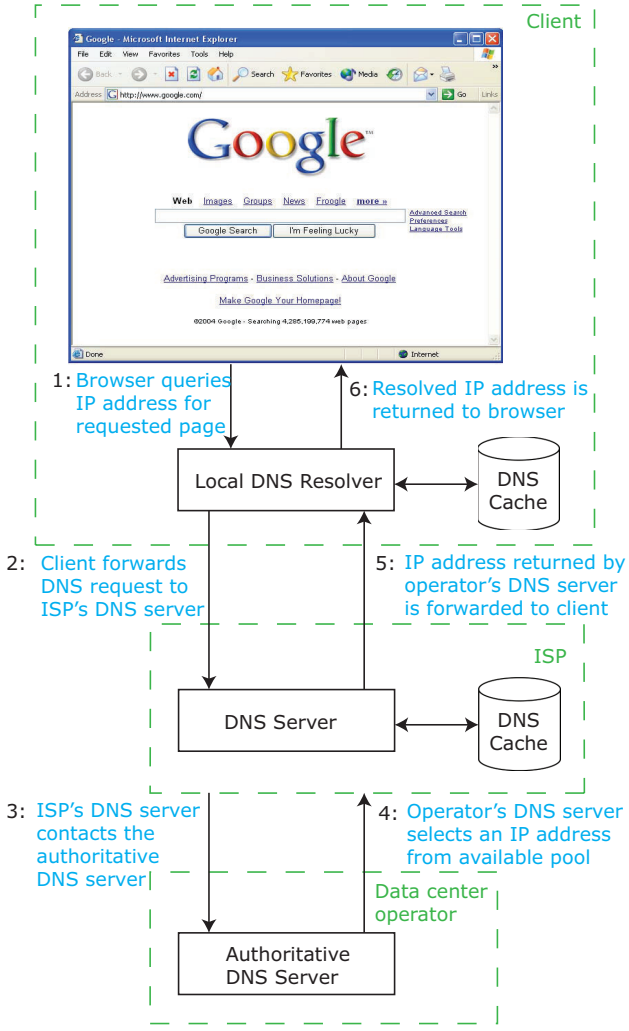

Figure 2.3: Resolving the IP address for a server hosted in a data center

2.1.3 Power consumption model

Fan et. al. used the results of a measurement study to show that the power consumption in a data center can be well-modeled as a linear function of the average CPU utilization [8]. As more and more client traffic arrives at servers in a data center, the average CPU utilization increases. If we consider homogenous client requests, the CPU utilization can be modeled as a linear function of workload. In case of heterogenous requests, one can approximate all request types as consisting of an integer number of micro-requests. Using the micro-request as our workload unit, we can still model CPU utilization as a linear function of workload. Since CPU utilization can be modeled as a linear function of workload, server power consumption can be modeled as a linear function of it's workload. The total power consumed by servers in a data center can, therefore, be represented as a linear function of the cumulative workload handled by the servers at the data center.

In our thesis, we wish to minimize the total electricity cost in a data center and servers are not the only power consuming equipment in a data center. Nonetheless, server power consumption is related to total data center power consumption by a measure called Power usage effectiveness (PUE). PUE is a measure of the efficiency with which a data center handles its power. It is defined as the ratio of total facility power to the IT equipment power. IT equipment power consumption includes power consumption by servers, storage and networking equipment. We assume that the power consumption in storage is related to that in servers, i.e., a unit workload consumes a fixed amount of power in storage devices. Networking equipment's power consumption is almost invariable with workload [22, 23, 24]. Therefore, we can consider total IT power as being a constant multiple of server power consumption plus a constant amount (which is not important in our thesis since it plays no role in an electricity cost minimization problem). Therefore, PUE being a constant (depending on how efficiently data center is architected), data center power consumption is proportional to workload handled by the data center.

2.2 Cellular Networks

Being the older sibling of the Internet, telephony services are a more integral part of our every day lives than the Internet. Mobile telephone systems have enabled not only untethered access to traditional telephony services but also new types of services. We make phone calls, send text messages and can even connect to the Internet using our mobile phones. Just as Internet connectivity services are provided by ISPs and Internet applications are powered by data center operators, mobile phone services are provided by mobile network operators (MNOs).

Over the years, mobile networks have been deployed based on different technologies. Literature often categorizes mobile network technologies in terms of *generations*. First generation cellular networks (1G) were based on Advanced Mobile Phone System (AMPS). AMPS networks were deployed starting in 1978. The AMPS system also evolved into Digital-AMPS (D-AMPS) networks. Two technologies were part of the second generation (2G) cellular networks, namely Global System for Mobile communication (GSM) and Code Division Multiple Access (CDMA). Today, 90% of the world's top 20 cellular networks use GSM technology [25]. Anticipating the increased demand for mobile access to data services such as Internet access, vendors introduced General Packet Radio Service (GPRS) as an add-on to GSM networks. GPRS offers data rates between 56 kbps and 114 kbps. 2G networks with GPRS are sometiems referred to as 2.5G. GPRS bit rates are insufficient for many high bandwidth applications such as video calls, video streaming and video conferencing. To enable such services, broadband mobile services were introduced in third generation (3G) networks networks such as High Speed Downlink Packet Access (HSDPA) and Universal Mobile Telephone System (UMTS). The increasing trends in the use of high-bandwidth applications in mobile networks has spawned the fourth generation (4G) cellular networks such as WiMAX and Long Term Evolution (LTE).

2.2.1 Structure

In this thesis, as far as cellular networks are concerned, we focus specifically on GSM networks. Mobile phone networks are also referred to as cellular networks. The term cellular network stems from the fact that the area covered by the operator is logically divided into several small areas called cells. A cell in an urban setting (a macrocell) is typically upto a few hundred meters in radius, whereas in suburban or rural settings, the cell radius may be upto tens of kilometers. A *cell site*, typically situated in the middle of a cell, enables subscribers in that cell to connect to the mobile network. A cell site is also often referred to as a Base Transceiver Station (BTS)[3] or simply a base station. A cell site hosts a number of transceivers (TRXs), radio antennas, power amplifiers and other allied equipment.

Typically a government regulator such as Pakistan Telecommunication Authority (PTA) allocates a frequency band to each of the operators providing cellular service in the host country. The allocation is such that each operator gets a different frequency band. The spectrum allocated to a cellular operator is an integer multiple of the bandwidth of a single GSM channel (200 kHz). A cellular operator distributes their allocated frequencies to cells in their network. The channels allocated to an operator are much fewer than the number of cells in the network. Therefore, a given channel must be reused in an operator's network. Frequency reuse is done in such a way that any two cells that share the same frequency channel are sufficiently far apart so that the radio signal from any one of the cells does not noticably intefere with that in the other. In fact, each cell is typically divided into three sectors (resembling 120 degree pie-slices), therefore, the frequency allocation is done on a per-sector basis. Nonetheless, for a high-level view, the set of frequencies allotted to all sectors in a cell can be considered as allotted to the cell itself. Each TRX at a cell site operates at a distinct frequency.

[3]A single cell site sometimes hosts multiple BTSs, for instance, when multiple network operators share a single site

Given two communicating parties at fixed locations, if the transmitted signal power is kept constant, the received radio signal strength would differ depending on the frequency used. Also, this frequency selective behaviour of the radio communication medium keeps changing with time, i.e., if frequency A receives better propagation compared to frequency B at time t_1, the same will not necessarily be true at time $t_1 + \epsilon$. This means that we can't statically pick the best frequencies to use for a particular cell by considering, for instance, the type of terrain. In order to make decent communication conditions available to all callers, on average, GSM networks also use frequency hopping, whereby the frequency allocation to cells are changed periodically.

To improve GSM's spectrum utilization, each frequency is also time-divided. For each frequency, a 120 ms duration transmission unit is called a GSM multiframe. It is named so because it consists of 26 frames of duration approximately 4.6ms each. Each frame is also divided into 8 bursts of duration approximately 0.577 ms each. The recurrence of a particular burst is what may be called a channel in GSM. In other words, a particular frequency and position within every frame defines a channel.

A MS often receives radio signals from multiple BTSs nearby. The MS picks the BTS from which it receives the strongest signal as it's serving BTS. A MS will do all communication such as call reception and placement through the serving BTS. When a subcriber moves around, the signal from the serving BTS might weaken. In such an event, the MS requests the network to allow it to change it's serving BTS to the one from which it currently receives the strongest signal. This is called a call handoff and is coordinated by a Base Station Controller (BDC).

Whereas a BTS is the access-side of a cellular, having no intelligence and performing only radio transmission and reception, the operations requiring intelligence such as frequency allocation, handoff coordination and frequency-hopping are controlled by a BSC. A BSC is responsible for several BTSs, which are connected to the BSC by means of some backhaul,

such as E-1 or microwave links. A cellular network will typically have multiple BSCs, with a BSC being responsible for several BTSs in a vicinity. All BSCs are also interconnected by the cellular network's backbone, so that actions that require global coordination such as frequency assignment, frequency hopping and call handoff can be done smoothly.

Another key component of a GSM network is the Mobile Switching Center (MSC). The MSC is responsible for call routing both within the GSM network and beyond (to a landline phone, for instance). Since the focus of our thesis is power consumption in the network and 50% [26] - 80% [27] of a cellular netowrk's electricity consumption is due to the BTSs, we will not dwell on the MSC and other components of the cellular network any more than necessary.

2.2.2 Call placement

For intelligible wireless communication, only one transmitter may transmit on a given frequency. Therefore, a call to/from a MS requires the allocation of two GSM frequencies, one for uplink (voice traffic from the MS to BTS) and the other for downllink (from BTS to MS). For coordinated acquisition of these frequencies, certain frequencies are reserved in each cell to serve as control channels. In fact, a caller does not get complete access to a particular pair of frequencies. Each of the 8 bursts in a GSM frame for a particular frequency may be used by different callers. Therefore, a particular frequency may be shared between multiple callers at a given time. In GSM terminology, they would all be using different channels, however, because a channel is characterized not only by the frequency but also the position within a GSM frame. Hence, the voice traffic for a call in GSM operates over two channels.

It appears that if n frequencies are assigned to a particular sector, then it can support up to $8n$ simultaneous calls because that is the number of GSM channels available. However, this is not true for two reasons.

- Some channels are reserved for control purposes. The exact number of such channels varies from operator to operator.

- GSM supports two different types of codecs, namely the full-rate codec and the half-rate codec. The full-rate codec corresponds to a caller using a burst in every GSM frame during a call, whereas the half-rate codec corresponds to a caller using a burst in every alternate GSM frame. By default, the full-rate codec is used for every call. However, when traffic congestion rises above an operator-configured threshold, the network attempts to admit every new call using a half-rate codec, if the corresponding MS supports it. If the traffic rises further and crosses a second threshold as configured by the operator, the network also re-assigns current calls to use a half-rate codec depending on the corresponding MS support. This enables a BTS to support more than $8n$ simultaneous calls during times of congestion.

2.2.3 Power consumption model

BTSs account for most of the power consumed in a cellular network. [26] claims that BTSs contribute 50% of overall network power consumption, whereas [27] puts this number at 80%. For this reason, most of the prior work related to power consumption in cellular networks focuses on BTSs.

Lorincz et. al. performed a measurement study of BTS power consumption under real-traffic conditions and concluded that the power consumption may be approximated as a linear function of call traffic [28]. Thus, as traffic varies during a given day, instantaneous power consumption would follow a similar curve as the traffic.

2.3 Similarities between different types of networks

From our discussion of two different types of networks, we can see that they are both essentially a collection of interconnected sites (data centers and BTSs) which are a collection of resources (data centers and TRXs). The workload in both types of networks exhibits diurnal patterns [10, 7]. The network in both cases is provisioned according to peak workload demand. Since the network resources are not energy proportional, this means that in low-workload regimes, the network is heavily over-provisioned. The resulting energy inefficiency can be dealt with by deactivating some resources when the traffic is low.

In terms of power consumption also, the two networks considered in this chapter, namely geo-diverse data centers and cellular network are quite similar. Both have a linear mapping from workload to power consumption. This motivates the possibility of establishing a common framework that smartly schedules the network resources in response to workload variations so as to minimize electricity costs.

2.4 Differences between different types of networks

All characteristics of different network types are not essentially similar. Several attributes are different as well. The following list summarizes some differences between geo-diverse data centers and cellular networks

- **Workload granularity:** The workload capacity of a data center is very large, potentially in millions of client requests per second, whereas the workload capacity of a TRX is less than eight simultaneous calls. For this reason, instead of determining the exact integer number of requests to handle at each data cneter, the fraction of workload to be handled at each data center can be determined as a real-number (which is a much simpler problem to solve), and the resulting number of requests will most likely be an

integer or may be rounded off with little change in electricity cost. Meanwhile, in case of cellular networks, the cumulative workload for a cell is a small number of calls and each call must be handled at exactly one of a few candidate cells. Hence, the call to cell mapping must be binary in nature and fractional mapping algorithm will not work.

- **Geo-diversity in electricity prices:** In a geo-diverse data center scenario, the network resources, i.e., data centers are quite far from each other and hence electricity price differential due to geo-diversity in electricity prices is quite likely. However, in case of cellular networks, the cell sites are within a few hundred meters of each other (in an urban setting) and an electricity price differential is highly improbable.

Chapter 3

A generalized framework for electricity cost optimization

In Chapter 1, we have seen that many different types of networks are plagued by high electricity costs. If the networks were energy efficient, this would be justifiable due to high workload. However, many types of networks today are characterized by lack of energy proportionality and significant daily variations in workload which result in energy inefficiency. Hence, high electricity costs in networks is a major concern and minimizing electricity costs is an important research problem.

Since electricity costs depend on two things: i) the amount of electricity consumed, ii) electricity prices. We developed insights into the former by investigating network operation as it relates to power consumption. In doing so, we observed that there are several similarities and a few minor differences between different network types in terms of power consumption. In this chapter, we will leverage the similarities in different types of networks to formulate an optimization problem that minimizes electricity costs for different types of networks. We will also comments on how the subtle differences between different network types will be incorporated in this optimization problem.

3.1 Problem Model

In order to develop a generalized model for network electricity cost and to formulate an optimization problem for minimizing electricity cost, we use an illustrative example. We will then comment on the complexity of the problem before developing an optimization problem formulation.

3.1.1 Illustrative Example

Let us illustrate network operation from the standpoint of electricity consumption and cost using an example shown in Figure 3.1. The example uses a test tube to represent a network resource and marbles to represent a unit workload. The network resource would be a data center in the context of geo-diverse data center operator, whereas it would be a transceiver in the case of a cellular operator. Similarly, the workload unit would be a client request in the data center context, whereas it would be a call in a cellular network setting. The operator's goal is to assign workload to network resources and, if needed, periodically update this mapping in response to variations in workload.

We consider the largest possible quantum of time for which the workload (and electricity price) remains fixed and term each such quantum as an *interval*. We assume that workload for several consecutive intervals is known and term this sequence of intervals as a planning window. The example demonstrates three different ways of mapping this workload to two network resources situated at different locations. For simplicity we assume in this example that the workload is geographically split such that half of it originates near each of the two resource. For this example, we consider temporal variatioin in workload as shown in Figure 3.1 (a). Meanwhile, Figure 3.1 (b) shows the geo-temporal variation in electricity prices for the two network resources.

One possible operational strategy is to map each workload unit to the nearest available

resource as shown in Figure 3.1 (c). In a sense, this is the default strategy in cellular networks, whereby a call is handled by the BTS from which the mobile station (MS) receives the strongest radio signal[1]. In geo-diverse data center settings, this sort of mapping is also often the default strategy because it minimizes the access latency for all clients[2].

The above workload-resource mapping strategy pays no attention to geo-diversity in electricity prices. We can exploit geo-diversity in electricity prices to reduce the electricity cost over the planning window by mapping more workload to resources at cheaper locations. To this end, we must change the way workload is mapped to resources as the electricity prices at various locations changes. We term such changes in workload-resource mapping as Workload Relocation (WR). Figure 3.1 (d) shows a mapping strategy that uses WR to map as much workload as possible to resources at cheaper locations. In interval t_1, since the cumulative workload equals the total network capacity, both network resources will be operating at capacity. Accordingly, there is not opportunity to reduce electricity costs using either WR or RP. In interval t_2, on the other hand, as shown in Figure 3.1 (d), we may use WR to move all workload to network resource B, which is situated at the location with the cheapest electricity price, thereby reducing electricity cost for that interval as compared to the default workload-mapping strategy shown in Figure 3.1 (c). While doing this WR, care must be taken not to violate the workload capacity of a network resource. In interval t_3, again, as shown in Figure 3.1 (d) WR may be used to shift workload to network resource A to reduce electricity costs compared to the default strategy shown in at the location with cheapest electricity price during that interval.

Due to lack of energy proportionality in networks, the power consumption of idle resources

[1]Signal from the physically nearest BTS may be weakened considerably due to natural or man-made obstructions. In such cases, the nearest BTS may not be the one from which the strongest signal is received. Hence, we take "nearest" to mean the BTS from which the MS receives the strongest signal

[2]Network latency has been shown to have a strong correlation with the physical shortest path distance between two locations on the globe [29]. So, the commonly understood physical measure of "shortest" applies in this case.

is a large fraction of their peak power consumption. Hence, consolidation of workload to cheaper locations offers a limited benefit in terms of reducing electricity cost compared to the default workload-mapping strategy. To avail considerable savings in electricity cost, one must use resource pruning (RP), i.e., deactivate idle resources. Notice that in the deafult workload-resource mapping strategy of Figure 3.1 (c), there is no opportunity to deactivate idle resources in any of the three intervals. However, the purely-WR strategy of Figure 3.1 (d) may be augmented with RP, as shown in Figure 3.1 (e), to achieve maximal savings in electricity cost. The strategy in Figure 3.1 (e) not only shifts workload to the cheapest possible resources, but also deactivates as many resources as possible.

In claiming that Figure 3.1 (e) shows the maximal savings in electricity cost, we have assumed that activation and deactivation of network resources is free of cost. However, such costs may exist in practice and in some network types may even be significant compared to the total electricity cost of network operation. In such cases, care must be taken when defining the optimal strategy for network operation. With this in mind, we draw parallels with similar problems in other domains with known results on optimal solutions and hence draw conclusions on the computational complexity of the optimal electricity cost network operation.

3.1.2 Problem complexity

During each interval, the network operation problem maps to the multiple knapsacks problem [30, 31], whereby a subset of a given set of items, each with a certain weight, must be selected and placed into several weight-limited knapsacks such that the total profit from the selected items is maximized. Since the single-interval instancce of our problem is analogous to the multiple knapsacks problem, which is known to be NP-Hard [30, 31], each single-interval instance of our problem is NP-Hard as well. Hence, the multi-interval planning problem must also be NP-Hard. This applies to cellular networks, for instance, where every call may

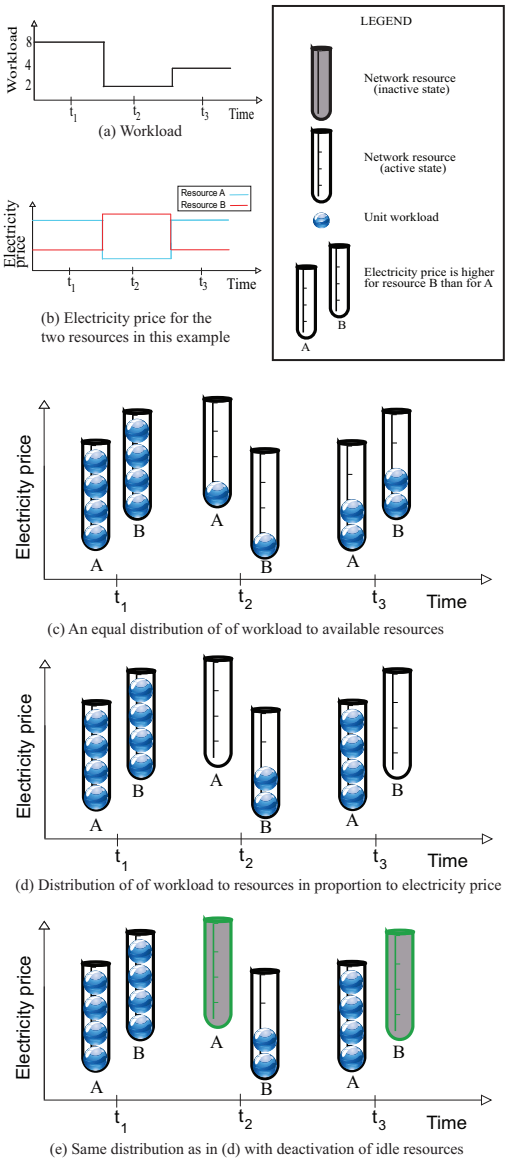

Figure 3.1: An example of mapping variable workload to capacity-limited network resources with geo-temporal diversity in electricity prices. Three consecutive intervals t_1, t_2 and t_3 are considered. Workload and electricity prices may only change between two consecutive intervals. (a) Workload considered in this example. (b) Electricity prices for the locations at which the two network resources are situated. (c) A uniform mapping of workload to network resources does not exploit electricity price diversity. (d) Mapping workload to network resources in order of their current electricity price. Due to lack of energy proportionality, only slight savings in electricity cost are possible. (e) Deactivating idle resources alongwith the resource mapping strategy of (d) may result in significant electricity cost savings.

be associated to exactly one BTS.

In certain type of networks, workload may be fractionally distributed, i.e., the cumulative workload during interval j, denoted x^j, may be distributed amongst the network resources such that each network resource gets a fraction of the workload dentoed x_i^j. In such networks, the sum of x_i^j over all network resources must equal x^j. As discussed in section 2.4, workload mapping in geo-diverse data centers may be approximated using such a scheme[3]. One would expect that the resulting problem would be simpler to solve and shouldn't be NP-Hard, because the single interval instance now resembles the fractional knapsack problem which may be solved optimally using a greedy strategy. But as we shall see, when RP is applied in a multi-interval setting, even this version of the problem is NP-Complete.

We compare this second version of our problem to the unit commitment problem [32] in distributed electricity generation and distribution scenario, which is known to be NP-Complete. The unit commit problem determines the generation levels of several power generating resources, given time-varying demand for electricity that may be derived from any of the active generating resources in any fraction. The generating resources may be turned off when electricity demand is lower than the cumulative capacity of running generating resources while incurring a ramp-down cost. Similarly, a generating resource may be turned on when demand exceeds the capacity of resources that are currently operating while incurring a ramp-up cost. Ramp-up and ramp-down costs as well as costs of generating a unit of electricity at each generating resource depends on the fuel prices at the location where the corresponding generating resource is situated. Furthermore, a generator is allowed to run on no-load as a spinning reserve while incurring idle fuel costs. If we represent the time-varying electricity demand as the network operator's workload and replace the generating resources by data centers, we have a one-to-one mapping of the unit commit problem to

[3]The number of client requests per interval is so large that the fractional distribution (to a reasonable precision) of workload amongst resources results in a solution that will most likely have an integer number of requests mapped to each data center

our geo-diverse data center scenario. Since the unit commitment problem is NP-Complete, it follows that so is the fractional workload-mapping version of our problem.

3.2 Optimization problem formulation

On a high-level, routine network operation (in the context of our thesis) involves distributing workload to network resources and periodically updating the fraction of workload mapped to each network resource. For simplicity of modeling and analysis, we assume that the mapping of workload to network resources, henceforth referred to as *workload mapping* or simply *mapping*, is updated at the beginning of intervals of fixed duration. We define resource i's state during interval j, denoted s_i^j, as the corresponding resource's status (on or off) and the amount of workload mapped to it. The aggregated state of all network resources during interval j may be termed as the *network state* during the corresponding interval, denoted by S^j. The routine network operation can thus be modeled as determing a sequence of states for a time horizon, called a *planning window*, consisting of a set of consecutive intervals of equal duration. In the context of our thesis, the objective is to determine the state trajectory that is optimal in the sense that it minimizes the electricity cost over the planning window.

3.2.1 The objective function

The optimal state trajectory problem attempts to determine a sequence of states such that the sum of state costs and the cost of transitions between states in consecutive intervals is minimized. Mathematically, we may present the objective function as:

$$\sum_{j=1}^{n} C(S^j) + T(S^j, S^{j-1}) \tag{3.1}$$

Here $C(S^j)$ represents a function that evaluates the cost of being in state S^j and $T(S^j, S^{j-1})$ represents the cost of transitioning from state S^{j-1} to state S^j. In the context of our thesis, the function $C(S^j)$ should evaluate the electricity cost of network operation given that the network is in state S^j. Similarly, $T(S^j, S^{j-1})$ should compute the cost of changing network state between two consecutive intervals that may arise due to factors such as turning network resources on or off.

3.2.2 The constraints

The state trajectory problem must be subject to a number of problem-specific constraints. Some constraints are common to all types of networks.

- **Resource capacity must be respected:** During all intervals, we must ensure that the workload mapped to each network resource does not exceed it's capacity.
- **All workload must be handled:** During all intervals, the sum of workload mapped to all network resources must equal the offered workload for that interval.
- **Network resource status is binary:** The status of a network must be represented as a binary variable which takes on the value 1 if the said network resource is on during a given interval, and 0 otherwise. We can not simply represent the resource state using a real-valued variable, representing the fraction of the current cumulative workload mapped to the corresponding resource, because it is not possible to determine instances of activation/deactivation of resources using such real-valued variables.

Several network-specific constraints must also be formulated. These constraints arise from subtle differences between different types of network. For instance, while any client request can be handled at any data center, a given call may be handled by only a few BTSs that are in the immediate vicinity of a caller.

3.2.3 Comments on the problem formulation

The decision variables in our problem formulation are the state of network resources for each interval in the planning window. A resource's state has two parts: the amount of workload it handles and it's status (on or off). The resource status needs to be a discrete (binary) variable. The amount of workload may also be a whole number in some network types. For instance, in a cellular network context, the workload mapped to a resource represents the number of active calls being handled by a BTS. In some other cases, on the other hand, such as the geo-diverse data center scenario, the amount of workload mapped to a resource may be represented using a real-valued variable, representing the fraction of total workload being handled by a data center during a given interval. In the first scenario, such as cellular networks, the resource state is purely discrete, whereas in the second scenario, such as geo-diverse data centers, the resource state is composed of a discrete as well as real-valued parts. In the former case, our optimization problem is an integer program (IP), whereas in the latter, the problem is a mixed integer program (MIP). Both IP and MIP are NP-Hard, however, and must be solved using techniques such as branch and bound [33] or other heuristics. If functions $C(.)$ and $T(.,.)$ as well as all constraints are linear and convex, the formulation is termed as an integer linear program or mixed integer linear program. Since the branch and bound technique repeatedly solves constrained and integer-relaxed versions of the IP (or MIP), having linear objective functions results in lower computational complexity. Fortunately, the nature of energy consumption in networks is such that the power consumption function $C(.)$ is linear and convex. For this reason, in our thesis, we strive to make the transition cost function $T(.,.)$ as well as all problem constraints as linear.

In this chapter, we have presented only an abstract formulation of the general optimization problem for minimizing electricty cost in service provider networks. We find it appropriate to comment here on the more concrete mathematical formulation of the problem in specific network types as well as the techniques used to solve those concrete instances

of the problem. Two concrete instances of the optimization problem are discussed in the next two chapters. In order to solve both of those concrete instances, we used the CPLEX solver, which uses the branch and bound heuristic, available with the ILOG CPLEX Studio which is available free of cost for academic use. Our primary focus in this thesis is to investigate the potential that the use of RP and WR offers for electricity cost optimization. Therefore, we focus on solving both concrete instances of the optimization problem *exactly* rather than proposing heuristics for approximate solutions to the problem[4]. As we shall see in the next two chapters, we were able to solve problems of reasonable size using simple desktop PCs within a reasonable amount of time.

[4]We do, however, propose a heuristic for the optimization problem in a cellular network setting.

Chapter 4

Case Study I: Geo-diverse Data Centers

4.1 Prelude

To enable robust and low-latency cloud services, companies such as Amazon, Google and Microsoft deploy large-scale infrastructure in the form of distributed data centers. The cost of electricity needed to run these data centers accounts for a significant portion (15%) of the total capital and operational expenditure [4]. Furthermore, due to increasing service demand as well as electricity prices, the fraction representing the cost of electricity is on the rise [34, 35], making it important for the cloud providers and data center operators to cut down on their electric bills.

In the previous chapter we have presented an abstract and generalized formulation of an electricity cost optimization framework for various types of networks. In this chapter, we will derive a specialized instance of the general problem, discuss it's solution (exact as well as heuristic) and perform a sensitivity analysis. We will also discuss our formulation's benefits and limitations.

Our problem formulation talks about cutting electricity cost by shutting down resources, data centers in this case, during low-workload regimes to improve energy efficiency of the overall network. However, each data center is a very large distributed system. Several possible ramifications of shutting down entire data centers raises some concerns. First, when a data center is booted up, some equipment requires ramp-up time before the data center can start regular operation. If the ramp-up time is of the order of the time at which electricity prices change, our solution will not be able to fully exploit the geo-temporal variation in electricity prices. An example of such equipment is air-conditioning. Another concern is that a restart of networking equipment such as border routers often results in convergence delays before the forwarding information across the network is consistent. During this time, the forwarding information may have undesirable states such as forwarding loops. For this reason, it may not be wise to shutdown networking equipment even when the data center is not expected to handle any workload. Third, it may be important to keep application data stores (such as search indices) consistent, hence the database servers on all data centers need to be up and running irrespective of presence or absence of workload. Fourth, some clients might keep cached location information about server's addresses and may continue to send traffic to a data center that an operator has shutdown. Dropping such *unexpected traffic* does not make good business sense and some servers may need to be kept active to handle it. It appears from this discussion that a data center may only be partially shutdown, if needed, so that performance and revenues are not impacted.

The electricity load at a data center may be partitioned into two sets: one that is inelastic and must be on all the time and the other elastic portion that may be shutdown if there is no workload for the data center to handle. Let the first set comprise a fraction r of the total data center power draw. The second part accordingly contributes a fraction $(1 - r)$ of the total power draw. According to [36], the breakup of power draw in a typical data center is servers: 56%, cooling: 30%, power conditioning: 8%, network: 5% and lighting: 1%. Since

we put only the servers in the elastic pool of devices, $r = 0.44$.

The inelastic electricity load in a data center results in a fixed power consumption which can not be minimized by a workload relocation scheme. Hence, while formulating the electricity cost minimization problem, we must focus solely on the elastic electricity load in the data center, i.e., servers. To model server power consumption, we observe that it is known to be an affine function of CPU utilization [37]. Another way to arrive at this model is to note that data center power consumption is an affine function of average CPU utilization [8] and server power consumption is a known fraction of a data center's total power consumption.

4.2 Related work

While much prior work has considered scaling the capacity within a data center by scaling the number of servers that are on [38, 39, 40, 41, 42], to the best of our knowledge the cost of shutting down entire data center(s) has largely been ignored [43]. In the present work, we take a macro-scale view of an entire data center, whereby we use decision variables representing a whole data center's state. Micro-scale approaches that attempt to manage per-server state across all distributed data centers tend to suffer from scalability problems and thus rely mostly on approximation algorithms. Our formulation is more scalable because of the macro-scale view. [1].

Li et. al. determined the electricity cost optimal mapping of workload to geo-diverse data centers by controlling the state of the individual servers within each data center [38]. The state of the servers and their electricity consumption was controlled using Dynamic Voltage and Frequency Scaling (DVFS) and Dynamic Cluster Server Configuration (DCSC). Since their optimization problem formulation used Mixed Integer Programming (MIP) with decision variables per server, their approach is effective for small-scale problems such as an

[1]The trade-off is that this macro-level focus might produce somewhat sub-optimal results. Investigation of the magnitude of the sub-optimality is left as future work.

individual data center or a fraction thereof. This limitation is evident from the small number of servers used in the simulation-based evaluation in [38]. One way to scale their approach to large distributed data centers is to use coarse granularity in their problem formulation. For instance, instead of controlling the state of each server independently, all servers in a single rack could be configured in the same state at a given time. They used the Soccer World Cup 1998 webserver workload traces and electricity prices at four different locations to evaluate their proposal.

Some researchers have also proposed algorithms for the data center electricity cost optimization problem. In the context of a web-search query processing system hosted on a geo-diverse data center network, Kayaaslan et. al. presented a bin-packing type of algorithm for shifting search query workload between data centers in [44]. Buchbinder et. al. proposed online algorithms for relocating MapReduce jobs between geo-diverse data centers to reduce the electricity bill while considering the cost of inter-data center bandwidth [45]. Their proposed algorithms consider the uncertainty in electricity prices and workload estimates while mapping the jobs to data centers. They evaluated their algorithms using electricity prices from 30 locations across the US and workload data from a 10000 node MapReduce cluster. Bhaskar et. al. proposed online algorithms for mixed packing and covering, a problem which may be applied to optimally map workload to geo-diverse data centers [46]. For configuring servers in a single data center with a view of minimizing electricity costs, Lin et. al. presented offline as well as online algorithms for dynamic scaling of server computational capacity [39]. Urgaonkar et. al. proposed an online optimization algorithm while proposing to disconnect data center devices from mains and running on UPS when the electricity prices are high [47]. Their proposed scheme recharges the UPS units when the electricity prices are lower.

Investigation of strategies of infrastructure scaling to conserve power in a single data center is reported in [40, 48, 49, 12]. Chen et. al. proposed three different solutions that

either shut down or frequency-scale servers in a web hosting data center with the objective of minimizing electricity and maintenance cost while ensuring SLA compliance [40]. The first two of the proposed algorithms were based on a queuing theoretic and control theoretic analysis, respectively, while the third one was a hybrid scheme. While scaling the deployed capacity, their proposed scheme considers the cost of turning the servers on and off in terms of the resulting wear and tear. Mazzucco et. al. present similar strategies in [48]. Oh et. al. considered a virtualized environment and proposed solutions for optimally placing VMs on servers and map workload to the VMs such that electricity costs are minimized [49]. In [12], Chase et. al. present policies for resource allocation in a hosting center alongwith a switching infrastructure for routing requests to servers.

Most of the prior work in this area considers applications with short request-response type jobs. In [13], however, Chen et. al. considered connection-intensive applications such as video streaming, Internet gaming and instant messaging in the context of energy cost aware load dispatch.

Rao et. al. consider data center operation in a futures electricity market and the possibility of hedging against uncertain electricity prices under a smart grid environment in [50]. The authors used workload data from Google search cluster and evaluated a scenario of an operator with data centers at two different locations.

When one talks about cutting data center electricity costs without cutting the total electricity consumption, questions of the sustainability also arise. Efforts towards *green* mapping of workload to data centers have been reported in [51, 52, 53, 54, 55, 56, 57, 58]. For minimizing total electricity consumption on participant hosts in a peer-to-peer file download system, Sucevic et. al. studied various approaches such that peers could be turned off when they are not needed [59].

All of the above work deals with problems that can be categorized broadly as optimal scheduling problems. Such problems arise in many different domains and prior work in

such domains is relevant. For instance, System on Chip (SOC) [60], electric power systems and smart grid [61, 62, 63, 64], WiFi access points [65], wide area networks [66], cellular networks [7] and high performance computing [67, 68, 69, 70, 71, 72].

4.3 Sources of transition costs in the data center scenario

Our present work uses the overhead of data center activation/deactivation as the transition costs. However, in practice, there can be other forms of transition costs as well, which would vary from one deployment to the other. Examples of other sources of transition costs include, but may not be limited to, the following:

- An operator might change the way user traffic is routed to data centers by modifying entries in the Domain Name System (DNS). In such cases, it has been reported that many client-side DNS caches violate the DNS entry Time-To-Live (TTL) by continuing to cache expired DNS entries [73]. This means that user traffic may continue to arrive at a data center that the operator does not intend to handle workload at. If this overwhelms the active data center capacity, it may result in lost revenue due to excessive response times.

- The operator might change the way user traffic is routed to data centers by making changes to the Border Gateway Protocol (BGP) routing table. Due to the complex nature of inter-service provider connectivity and BGP dynamics, BGP routing table changes have been shown to take an unpredictable amount of time to reflect globally [74]. This means that we run into a similar situation as for the DNS-based method described above.

- In order for any request to be handled anywhere, the data store for the applications

must be replicated and consistency must be maintained. The cost of inter-data center traffic is quite high, hence this form of transition costs would be quite significant. The magnitude is not easy to predict because replication schemes are operator and application dependent. To the best of our knowledge, the current body of knowledge lacks a generic model for such traffic. Therefore, similar to [43], in the present work, we assume that content is perfectly replicated.

It is clear from the above discussion that the factors contributing towards transition costs depend not only on how the data center network is deployed and operated but also on the applications being hosted. Since this information is confidential, the utility of modeling a specific deployment is limited. The question that is significant, however, is the impact on the possible electricity cost savings (resulting from geo-temporal diversity in electricity prices) of variation in the magnitude of transition costs relative to the electricity cost in a given interval. Therefore, we have used a normalized and parametrized model for transition costs in our problem formulation. Our aim in this paper is to present electricity cost optimization solutions which operators can use, along with the parameter data (idling costs, transition costs, number of data centers and their locations) from their own data center network.

4.4 Instantiating the generalized optimization formulation

Consider a geo-distributed data center infrastructure comprising m interconnected data centers. The workload consists of client requests for hosted applications. Every client request is routed to one of the data centers in the network. The fraction of requests mapped to each data center may be revised periodically. For ease of modeling, we assume that these changes may be done at the start of discrete intervals of duration λ. Let the workload handled at

data center i during interval j be x_i^j. We consider a normalized measure of workload, so that $\sum_{i=1}^{m} x_i^j \leq 1$.

Let c_i be the workload capacity for data center i. Also, let P^{max} and P^{min} be the sum of the peak and idle power consumption, respectively, of the elastic portion of electric load over all data centers. In the present work, we consider homogenous data centers, so an individual data center's peak and idle server power consumption are proportional to it's workload capacity, i.e., $P_i^{max} = c_i P^{max}$ and $P_i^{min} = c_i P^{min}$. Since the average utilization of data center i during interval j is x_i^j/c_i^j, it's power consumption during that interval may be given by:

$$P_i^j = c_i(P^{min} + \frac{x_i^j(P^{max} - P^{min})}{c_i}) \tag{4.1}$$

Let the ratio of P_{min} to P_{max} be f. The case when $f = 0$ corresponds to complete energy-proportionality in the elastic load, which is purely ideal and does not occur in reality, whereas $f = 1$ means that the elastic load's power consumption is completely independent of workload. Substituting $P^{min} = fP^{max}$ in the above expression gives:

$$P_i^j = c_i P^{max}(f + \frac{x_i^j(1-f)}{c_i}) \tag{4.2}$$

Since P^{max} is a constant parameter, it plays no role in electricity cost minimization. Hence, we may normalize P_i^j over P^{max} to get:

$$\hat{P_i^j} = fc_i + x_i^j(1-f) \tag{4.3}$$

If we set $x_i^j = 0$, i.e., data center i is not computing any workload during interval j, then the second term in equation 4.3 goes to zero and the power consumption reduces to the first term in equation 4.3 only, which we refer to as *idle power consumption*. The second term in

equation 4.3 indicates the workload-dependent *computational power consumption*, which is independent of the data center capacity.

Electricity cost incurred at data center i during interval j is a product of it's power consumption (P_i^j), duration of the interval (λ) and the unit price of electricity (e_i^j). Hence, the sum of costs for idling and computational energy consumption incurred at data center i during interval j may be given as:

$$e_i^j \lambda (c_i f + (1 - f) x_i^j) \tag{4.4}$$

Now, let us consider the activation/deactivation costs for the elastic portion of the data center workload. Let σ and δ be the average power consumption, over a single interval, required to activate or deactivate, respectively, the elastic load at a unit capacity data center. Then, the bootup power consumption for the elastic load at data center i is σc_i and the shutdown power consumption is δc_i.

The electricity cost for booting up data center i's elastic load at the start of interval j is given by $c_i e_i^j \sigma$. Multiplication with the duration of an interval, i.e., λ is not necessary, because σ is defined as the per interval cost. We are assuming that the elastic load at a data center may be booted up within a single interval. The value of λ that we used in our experiments is equal to one hour, which seems to be a sufficiently large interval for this boot up. Similarly, the electricity cost for shutting down data center i's elastic load at the start of interval j is given by $c_i e_i^j \delta$.

The total cost of electricity for data center i during interval j is the sum of computational, idling, activation and deactivation electricity costs incurred during that interval. Therefore, the RED-BL optimization problem formulation, a summation of electricity costs over all data centers and over all intervals in a planning window may be given as:

Parameter	Description
m	Number of data centers
r	Fraction of data center load that is inelastic, i.e., may never be shutdown
n	Number of intervals in a planning window
f	The ratio between a data center's peak and idle power consumption
c_i	Normalized workload capacity of data center i
σ	Penalty for activating the elastic load at a unit capacity data center as a fraction of its energy consumption at full load in one interval
δ	Penalty for deactivating the elastic load at a unit capacity data center as a fraction of its energy consumption at full load in one interval
e_i^j	Unit cost of electricity at data center i during interval j
λ	Duration of an interval in hours
w^j	Operator's workload during interval j
x_i^j	Workload mapped to data center i during interval j
p_i^j	1 if data center i is active (either computing workload or idling) during interval j, 0 otherwise
b_i^j	1 if data center i's elastic load is activated at interval j, 0 otherwise
s_i^j	1 if data center i's elastic load is deactivated at interval j, 0 otherwise

Table 4.1: Data Center Network Model Parameters

$$\text{minimize} \sum_{j=1}^{n}\sum_{i=1}^{m} c_i e_i^j (p_i^j \lambda (f + (1-f)\frac{x_i^j}{c_i}) + b_i^j \sigma + s_i^j \delta) \tag{4.5}$$

subject to:

$$x_i^j \leq c_i \ \ \forall i, \forall j \tag{4.6}$$

$$\sum_{i=1}^{m} x_i^j = w^j \ \ \forall j \tag{4.7}$$

$$p_i^j, b_i^j, s_i^j \in \{0,1\} \ \ \forall i, \forall j \tag{4.8}$$

$$p_i^j \geq x_i^j \ \ \forall i, \forall j \tag{4.9}$$

$$b_i^j \geq p_i^j - p_i^{j-1} \ \ \forall i, 2 \leq j \leq n \tag{4.10}$$

$$s_i^j \geq p_i^{j-1} - p_i^j \ \ \forall i, 2 \leq j \leq n \tag{4.11}$$

$$b_i^0 = p_i^0, s_i^0 = 0 \ \ \forall i \tag{4.12}$$

In addition to the parameters that we've already described, the RED-BL optimization problem includes three auxiliary indicator decision variables: p_i^j, b_i^j and s_i^j (a summarized description of all parameters and variables is given in Table 4.1). p_i^j is 1 if the elastic load in data center i is active during interval j, or 0 otherwise. Similarly, b_i^j (and s_i^j) is 1 if the elastic load in data center i is booted up (shut down) at the start of interval j. In equation 4.5, multiplication of the first two terms by p_i^j ensures that computation and idling costs are accounted for in interval j, only if the elastic load in data center i is active during that interval. Similarly, multiplication of the last two terms in equation 4.5 by b_i^j and s_i^j, respectively, ensures that bootup and shutdown costs contribute to the summation only when the elastic load in a data center is booted up or shutdown.

The workload capacity constraint is given in (4.6). Eq. (4.7) ensures that all incident workload is handled, while (4.8) represents the binary-value constraint. Inequality (4.9) ensures that the elastic load in a data center is active whenever there is any workload mapped to it. If the elastic load in data center i is inactive in interval $j-1$ and active in interval j, then there is an activation at the beginning of interval j. The constraint in Eq. (4.10) ensures that b_i^j is 1 in such cases and 0 otherwise. Similarly, the constraint in Eq. (4.11) ensures that s_i^j takes on the correct value depending on the deactivation of elastic load in the data centers. We assume that the elastic load in all data centers is initially shutdown, therefore, an activation may be necessary at the first interval whereas deactivation in the first interval is not necessary. These conditions are ensured by the constraints in Eq. (4.12). It is easy to customize this constraint such that all data centers are assumed to be initially active.

As discussed in Section 3.1.2, the optimal workload mapping and relocation problem

is NP-Complete. While we are able to solve reasonably sized instances of the problem to global optimality over 24-hour planning windows, it is appropriate to also propose a heuristic algorithm for the problem.

Assume that the workload vector for the planning window starts at a trough, then rises in a non-decreasing manner to the peak before falling off in a non-increasing mannner to another trough. Under this assumption, the two ends of the workload vector define the intervals for which the minimum sufficient number f data centers is the smallest over the entire planning window. Without loss of generality, let's assume that one data center is sufficient for the workload at the two ends of the planning window. The single data center to be picked to run continuously from the first interval to the last, then, would be the one that has the lowest average electricity price. Based on this idea, we have defined a heuristic that places two pointers at the intervals that define the first and last intervals for which a certain number of data centers is sufficient, then picks the ones that have the least average electricity price in those intervals before sliding these pointers to their new appropriate position. The pseudo-code of our heuristic algorithm for RED-BL is given in Algorithm 1. It is designed based on the observation that the cumulative hourly workload increases or decreases slowly in somewhat fixed patterns. Since the bootup/shutdown costs are expected to be significant, our heuristic is designed to select a small number of data centers to operate in long continuous stretches during a given day.

On line 1, the algorithm starts by placing two pointers, g_1 and g_2, at the extremes of the workload vector. Our algorithm would work best if the beginning and end of the workload vector conincides with the two troughs of the workload in the planning window. Also, on line 1, our algorithm makes a local copy l of the workload vector w, so that l may be used to keep track of the algorithm's progress without modifying the input vector. Line 1 also includes the initialization of a list of available data center indices, a, and the initialization of the number of data centers currently in use, n_c, to zero. These steps would run in O(m+n).

The algorithm will store it's solution in 2-D arrays y and z. Here, $y[i][j]$ is 1 if data center i is to be on during interval j and 0 otherwise. Furthermore, $z[i][j]$ holds the amount of workload to be mapped to data center i during interval j. This initialization takes O(mn).

The algorithm starts by computing the minimum number of data centers required to handle the workload during intervals g_1 and g_2 as the values d_1 and d_2, respectively. Since we have considered homogenous data centers, all c_i are equal, therefore, this computation uses c_1 as the capacity of a data center. The smaller of d_1 and d_2 is picked as n_d. If n_d, the minimum data center demand during $[g_1, g_2]$ is greater than the current number of active data centers n_c, the algorithm enters the if-block on line 6. On line 7, the algorithm first computes the average electricity price for the available data centers between intervals g_1 and g_2 in O(mn) and then sorts the available data center indices in ascending order of average electricity price from g_1 to g_2 in O(m lg m). Between lines 8 and 15, the algorithm marks $n_d - n_c$ data centers to be on from g_1 to g_2 (line 10), updates the current number of data centers that have been used by the algorithm, n_c (line 10), assigns workload to each of these data centers (line 11), updates the local copy of the workload vector, l, by subctracting the amount of workload that the algorithm has handled so far for the relevant intervals and removes the data centers that have been assigned workload from the list of available data center a (line 13). Since line 13 involves removal of some contiguous entries from the beginning of an array, it runs in O(m).

Lines 17-22 update the two pointers until they either demark a section of the workload vector that requires a greater number of data centers than n_c or g_1 exceeds g_2 (which means that we are done). If the workload during intervals g_1 and g_2 is such that they both require the same number of data centers, both of the repeat until loops run and pointers g_1 and g_2 are moved until they reach a point where the workload for each pointer requires a greater number of data centers. Otherwise, only one of the repeat-until loops runs and the pointer corresponding to the interval requiring the smaller number of data centers is moved towards

the other pointer until it reaches an interval where the workload requires a greater number of data centers.

The if-block (line 6-16) will dominate the overall execution time. Within this if-statement, on line 7, the average electricity price is computed and then the data center indices are sorted in ascending order. In the worst case, the if-block will be entered in every iteration of the outer repeat-until loop and exactly one data center index will be removed from a (line 13) in each iteration of the outer repeat-until loop. In this case, the computation of average electricity prices will require mn + (m-1)(n-1) + (m-2)(n-2) + (m-3)(n-3) + ... + (m-n+1) primitive operations. Sorting the electricity prices will require O(m lg m) + O((m-1) lg (m-1)) + ... + O((m-n+1) lg (m-n+1)) running time, overall. Within the nested for-loops, line 13 is most complex which will require, in the worst case, (m-1) + (m-2) + (m-3) + ... + (m-n-1) primitive operations. This is smaller compared to the running time of the average electricity price computation, so in Big-Oh notation, we can ignore it. The overall worst case running time for the electricity price averaging can be computed as follows. We first consider that the electricity price averaging is done exactly i times and will later replace i by it's actual value of n.

$$
\begin{aligned}
mn + (m-1)(n-1) + (m-2)(n-2) + ... + (m-i+1)(n-i+1) \\
imn - n - 2n - ... - (i-1)n - m - 2m - ... - (i-1)m + 1 + 4 + ... + (i-1)^2 \\
imn - (n+m)(1 + 2 + ... + (i-1)) + 1 + 4 + ... + (i-1)^2 \\
imn - (n+m)\frac{i(i+1)}{2} + \frac{n(n-1)(2n-1)}{6}
\end{aligned}
$$

Substituting i by n, we get the overall worst case running time for the average electricity price calculation step as O(mn^2 + n^3). The overall worst case running time of the entire algorithm (including the average electriciry price sorting step) is O(mn^2 + n^3 + m lg m).

Algorithm 1 Heuristic for the RED-BL problem

Require: $w[1..n]$: Cumulative data center workload for the planning window,
$e[1..m][1..n]$: Electricity prices for all data centers over the planning window

Ensure: $z[1..m][1..n]$: workload assigned to each data center over all intervals in the planning window
$y[1..m][1..n]$: Array of data center status (1=on/0=off) over the planning window

1: $g_1 = 0; \quad g_2 = n - 1; \quad l = w; \quad a = 1..m; \quad n_c = 0;$
2: $\quad y[i][j] = 0; \quad z[i][j] = 0; \quad (\forall i, \quad \forall j)$
3: **repeat**
4: $\quad d_1 = \lceil w[g_1]/c_1 \rceil; \quad d_2 = \lceil w[g_2]/c_1 \rceil;$
5: $\quad n_d = \min(d_1, d_2)$
6: $\quad$ **if** $n_d > n_c$ **then**
7: $\quad\quad$ Sort a in ascending order of average electricity price in $[g_1, g_2]$
8: $\quad\quad$ **for all** $i \in a$ **do**
9: $\quad\quad\quad$ **for all** $j \in [g_1, g_2]$ **do**
10: $\quad\quad\quad\quad y[i][j] = 1; \quad n_c{+}{+}$
11: $\quad\quad\quad\quad z[i][j] = \min(l[j], c_i)$
12: $\quad\quad\quad\quad l[j] = l[j] - z[i][j]$
13: $\quad\quad\quad\quad$ Remove i from a
14: $\quad\quad\quad$ **end for**
15: $\quad\quad$ **end for**
16: $\quad$ **end if**
17: $\quad$ **repeat**
18: $\quad\quad g_1{+}{+}$
19: $\quad$ **until** $(\lceil w[g_1]/c_1 \rceil > n_c)$**or**$(g_1 > g_2)$**or**$(\lceil w[g_1]/c_1 \rceil > \lceil w[g_2]/c_1 \rceil)$
20: $\quad$ **repeat**
21: $\quad\quad g_2{-}{-}$
22: $\quad$ **until** $(\lceil w[g_2]/c_1 \rceil > n_c)$**or**$(g_1 > g_2)$**or**$(\lceil w[g_1]/c_1 \rceil < \lceil w[g_2]/c_1 \rceil)$
23: **until** $g_1 > g_2$

4.5 Experimental setup

In this section, we describe the experimental setup to perform a comparative study of different workload placement algorithms under various scenarios.

4.5.1 Application workload

We used an year-long trace of hourly workload for 3 social networking applications, with a subscription base of over 8 million users [75]. In order to make the dataset representative of a large data center network operator, we normalized these traces into a week long trace as follows. We sliced the trace into week-long segments and considered each slice as workload for a different application, for the same week. We, then, normalized the sum of these trace vectors so that the peak cumulative workload corresponds to a value of 0.9. The normalized workload intensity is plotted in Figure 4.1. The statistical characteristics of our workload, as plotted in Figure 4.2 are quite similar to those reported by Google for "thousands of servers during a six-month interval at a Google data center" [10].

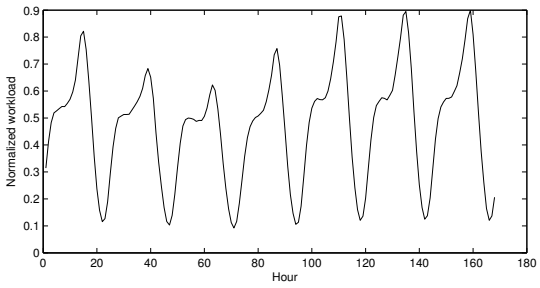

Figure 4.1: Normalized workload

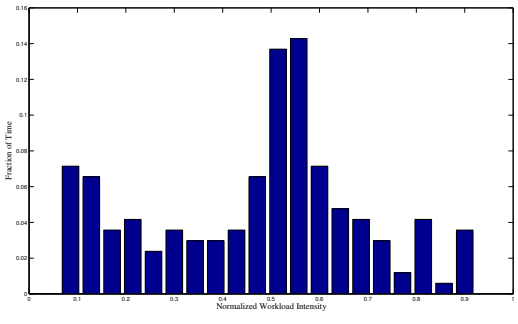

Figure 4.2: Workload intensity histogram

S. No.	Source	S. No.	Source	S. No.	Source	S. No.	Source	S. No.	Source	S. No.	Source
1	CAPITL	7	LONGIL	13	OH	19	MISO	25	WCMass	31	SEMass
2	CENTRL	8	MHKVL	14	PJM1	20	Illinois	26	Boston	32	Vermont
3	DUNWOOD	9	MILLWD	15	WEST	21	Cinergy	27	CT	33	PJM2
4	GENESE	10	NYC	16	PGAE	22	Michigan	28	Maine		
5	HQ	11	NORTH	17	SCE	23	Minnesota	29	NH		
6	HUDVL	12	NPX	18	SDGE	24	First Energy	30	Rhode		

Table 4.2: Sources of electricity prices used in our work

4.5.2 Electricity prices

We selected 33 different regions in the USA for which hourly electricity prices are available online. To ease reproducibility of our results, the names for these locations as used in the online datasets are listed in Table 4.2. We used the day-ahead prices for these locations, i.e., the electricity price negotiated for the same hour on the following day. We formulated the following deployment scenarios using this dataset:

1. **Single site operator:** An operator that only has one data center at a fixed location. We considered 33 different cases in this type of deployment, corresponding to each location for which we had electricity price data.

2. **Multi-site operator:** For a multi-site operator, we considered six different deployment scenarios. Five of these scenarios represent operators with data centers at the first 10, 15, 20, 25 and 30 locations that we selected, while the sixth scenario represents an operator with a data center at all 33 locations.

4.5.3 Algorithms for Workload Distribution/Relocation

The workload relocation problem has the following dimensions based on which different algorithms may be formulated. These are:

- For a given interval, the strategy for distribution of workload amongst data centers.

Algorithm	Remarks
LI	Local optimal with idling
LD	Local optimal with deactivation
LS	Local optimal with selection
LO	Local optimal without transition costs
RED-BL	The global optimal solution for the planning window
UNIFORM	Distribute workload equally over all data centers in all intervals
STATIC_MIN	A single site data center operator

Table 4.3: Algorithms compared in our work

Workload mapping strategy	
LI	Greedy
LD	Greedy
LS	Greedy
LO	Greedy
RED-BL	Based on global optimal solution
UNIFORM	Workload equally divided amongst all data centers
STATIC_MIN	There is only one data center that gets all workload
State of a data center in an interval when it has no workload	
LI	Active and idling
LD	Inactive
LS	Either inactive or idling, whichever is cheaper
LO	Inactive
RED-BL	Based on global optimal solution
UNIFORM	Active
STATIC_MIN	Active
Is transition cost reported in the total electricity cost reported?	
LI	N/A
LD	Yes
LS	Yes
LO	No
RED-BL	Yes
UNIFORM	N/A
STATIC_MIN	N/A

Table 4.4: A comparison of the algorithms studied in this paper

- For a given interval, the strategy for the elastic load at a data center which has not been assigned any workload. For such a data center, an algorithm may either keep the elastic load active, thereby incur idling electricity costs, or it may deactivate the elastic load.

- Over the planning window, does the algorithm report transition costs in the total electricity cost?

In this paper, we report comparative results for seven workload placement algorithms, listed in Table 4.3. The following list describes and differentiates these algorithms. the same comparison is also presented in tabular form in Table 4.4.

- **RED-BL:** This is our proposed algorithm that determines the global optimal cost of electricity over a planning window while considering and reporting the transition costs. The choice of workload distribution is governed by the optimal solution as determined by the CPLEX solver. Similarly, RED-BL decides to idle or deactivate unneeded elastic load based on the optimal solution returned by the CPLEX solver.

- **Heuristic:** This is the heuristic algorithm that we proposed in Section 4.4.

- **UNIFORM:** This algorithm equally distributes workload amongst all data centers. It has the sometimes desirable property of even utilization across all data centers. This algorithm does not deactivate elastic loads and hence does not incur transition costs.

- **STATIC_MIN:** This algorithm considers only one data center with unit workload capacity that is situated at the location that has the lowest average electricity price over the planning window. While this runs counter to the latency-minimizing philosophy of distributed data centers, it offers a good baseline for evaluation of benefits of geographic diversity. Since there is only one data center, the workload mapping strategy is trivial, there is no workload relocation and hence no transition costs.

- **Greedy algorithms:** The originally proposed algorithm in [43] distributes workload to data centers such that it picks the fewest data centers needed from the list of data centers sorted in order of increasing electricity prices. It then assigns workload to data centers starting with the cheapest one, until all workload is distributed. Furthermore, this original algorithm keeps all data centers active in all intervals, incurring significant idling costs and hence is naturally disadvantaged against RED-BL. To have a fair comparison with the greedy workload distribution strategy, we use several variants of the original algorithm as well.
  - **Local optimal with Idling (LI):** This is the originally proposed algorithm from [43]. It does not deactivate elastic load.
  - **Local optimal withOut transition costs (LO):** This is the variation of LI that was also proposed in [43]. However, this algorithm ignores transition costs and does not report them. The utility of this algorithm in our work is that it defines the lower bound on electricity cost that any algorithm can ever achieve.
  - **Local optimal with Deactivation (LD):** This algorithm always deactivates unneeded elastic load at data centers and reports the resulting transition costs in the total cost as well. This would be an improvement over LI making it somewhat competitive with RED-BL.
  - **Local optimal with Selection (LS):** In cases where transition costs are high compared to idling costs it would be better to keep the elastic resources at a data center active and incur idling costs if the elastic load is not needed at a particular data center for a small number of consecutive intervals. LS is a variant of LD that is empowered with the ability to *select* whether to deactivate unneeded elastic load at a data center or keep it idling.

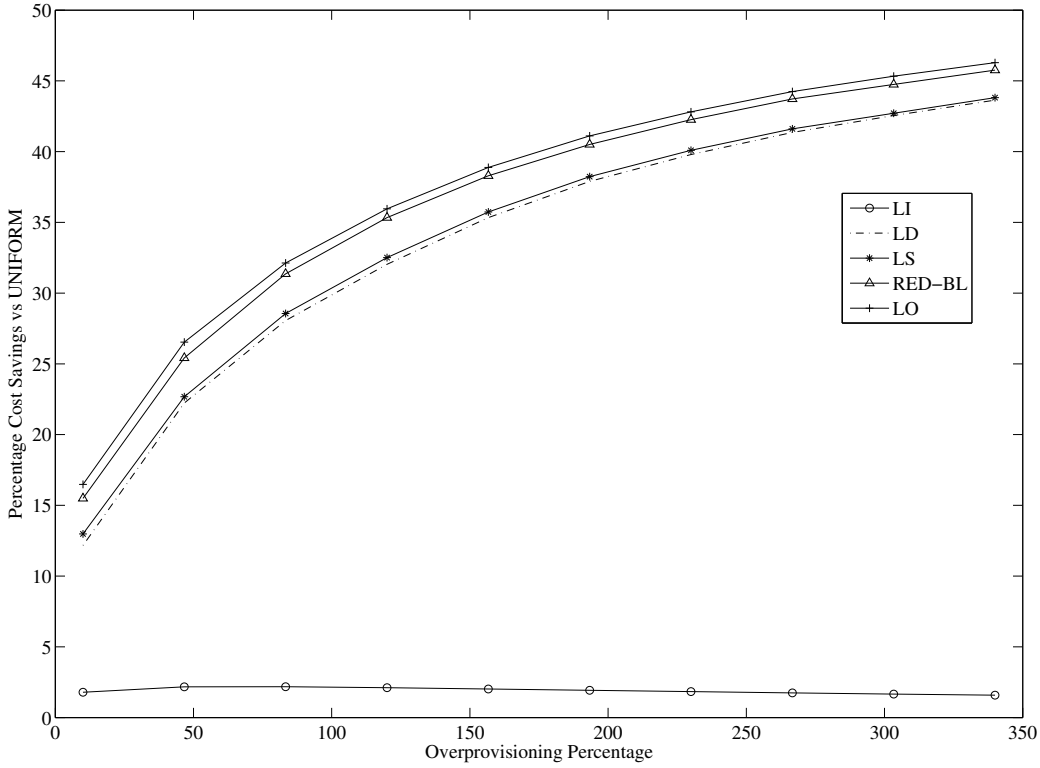

Figure 4.3: Percentage savings with over-provisioning

4.6 Results

To evaluate the utility of workload relocation for electricity cost minimization, we formulated five different scenarios. For each scenario, we ran 7 experiments (one for each day's workload in our dataset) and report the average of the total electricity cost for each algorithm. Each experiment determines an operational plan for a planning window consisting of 24 consecutive intervals, each with a duration of one-hour.

4.6.1 Sensitivity of electricity cost savings to extent of overprovisioning

In this scenario, we investigate the relationship of the amount of data center network capacity over-provisioning with the electricity cost savings. As we increase the amount of over-provisioning, each individual data center's capacity would increase enabling more and more workload to be mapped to data centers at locations with cheaper electricity price.

With data centers at all 33 locations in our dataset, we varied c_i between 0.03 and 0.12 (in increments of 0.01). This covers a variety of operators whose workload capacity ranges from just over expected peak workload to almost 300% over-provisioning.

We computed the total electricity cost for all algorithms, except STATIC_MIN[2]. In this scenario, we set $f = \sigma = \delta = 0.65$. The percentage savings in total electricity cost by various algorithms compared to UNIFORM are plotted against the data center capacity over-provisioning in Figure 4.3. An interesting observation is that for the wide range of capacity over-provisioning that we considered, LI is able to do only slightly better than UNIFORM. This is because LI incurs significant idling costs because it does not deactivate the elastic load at data centers that are not computing any workload.

The cost of the greedy strategy can be significantly improved, compared to that of LI, by deactivating the elastic load at data centers when they are not needed. Prior work has reported this (shown as LO) while assuming that there are no transition costs. Such costs, however, exist and the curves for LD and LS indicate that the total cost (including transitions) is indeed higher. For example, LS offers 10.35% less cost savings than LO, on average. The good news is that the cost of RED-BL solution is quite close the ideal lower bound (LO).

The reason for greater savings with RED-BL compared to the greedy solutions (LS and

[2]STATIC_MIN uses a single data center which is big enough to handle peak workload, so scaling it's capacity would not bring any extra savings.

LD) is that, the transition costs being significant, the former does fewer state transitions in the latter. In several intervals, RED-BL choses data ceWQhj34e43nters with relatively higher electricity price than the greedy solutions, but makes up for the higher computational cost by a reduction in the transition costs incurred.

4.6.2 Sensitivity of electricity cost savings to magnitude of transition costs

As the magnitude of transition costs relative to the state cost for an interval grows beyond a certain point, the benefits of deactivating elastic load at data centers would diminish. Accordingly, the electricity cost savings achievable by the workload relocation schemes would drop with increase in transition costs. In this scenario, we determine the percentage savings in total electricity cost for each algorithm, except STATIC_MIN[3], compared to UNIFORM, while varying the activation/deactivation overhead between 0 and 1, in increments of 0.1. The lower bound on σ (and δ) implies the ideal condition of no transition overheads. We set the upper bound to 1 so that the transition costs equal the cost of operating a data center at full load for an interval. A transition cost higher than this does not make sense as a workload relocation scheme would be better off keeping the elastic load at unloaded data centers idle. In this scenario, we kept $f = 0.65$.

LI does not (de)activate unneeded elastic load and thus it's electricity cost is independent of the magnitude of transition costs. We observed taht it offered a saving of merely 1.74% compared to UNIFORM. Figure 4.4 shows the electricity cost savings for the other algorithms compared to UNIFORM. The LS and LD adaptations of the LI algorithm offer savings that scale almost linearly to the magnitude of transition costs. Both LS and LD also bring an average reduction in the elastic load's electricity cost sby a factor of 4, compared to that of LI. RED-BL not only scales better than LS and LD but also achieves electricity cost saving

[3]STATIC_MIN uses a static workload mapping so it is insensitive to variations in transition costs

that is fairly close (only 3% higher, on average) than the ideal lower bound as reported by LO.

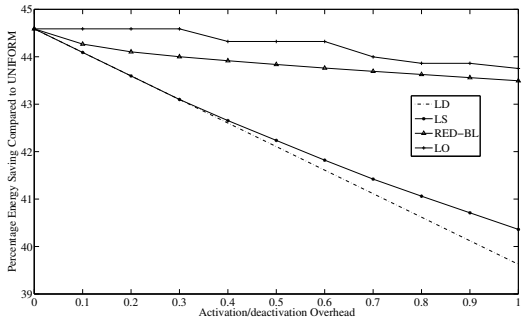

Figure 4.4: Total cost vs transition overhead

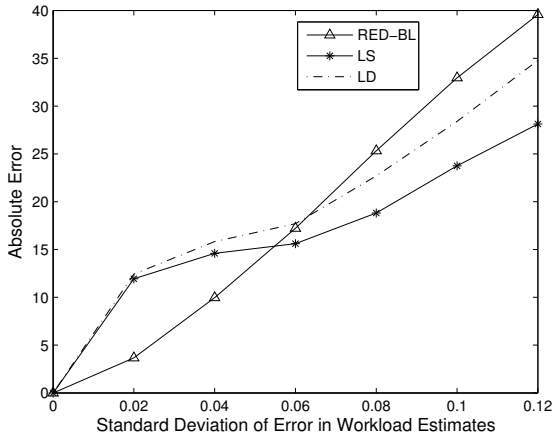

Figure 4.5: Cost estimation error due to workload estimation error

4.6.3 Sensitivity of electricity cost savings to resource pruning granularity

In this scenario, we investigate the potential benefits of deactivating the elastic load in a data centers in equal sized chunks instead of an all or nothing approach. The size of the portion of the elastic load in a data center that may be independently (de)activated may be deployment-dependent or operator-dependent. Possible choices of granularity may be a rack, a pod or one half of the elastic load etc. This granular deactivation can also be seen as throttling the equipment in a data center using facilities like processor frequency scaling.

Granular (de)activation is expected to bring additional power savings. For instance, if the elastic load in a data center is operating at 10% of it's capacity, then the remaining 90% of the load is still consuming significant idling power. If we had the ability to power down half of the elastic load at a data center, we could cut idling energy cost significantly.

The optimization problem formulation with l granular (de)activation levels is given by:

$$\text{minimize } \sum_{j=1}^{n}\sum_{i=1}^{m} c_i e_i^j (p_i^j \lambda (\frac{f}{l} + (1-f)\frac{x_i^j}{c_i}) + \frac{b_i^j \sigma}{l} + \frac{s_i^j \delta}{l})$$

subject to:

$$x_i^j \leq c_i \ \ \forall i, \forall j \tag{4.13}$$

$$\sum_{i=1}^{m} x_i^j = w^j \ \ \forall j \tag{4.14}$$

$$p_i^j, b_i^j, s_i^j \in \{0, 1, ..., l\} \ \ \forall i, \forall j \tag{4.15}$$

$$p_i^j \geq x_i^j * l/c_i \ \ \forall i, \forall j \tag{4.16}$$

$$b_i^j \geq p_i^j - p_i^{j-1} \ \ \forall i, 2 \leq j \leq n \tag{4.17}$$

$$s_i^j \geq p_i^{j-1} - p_i^j \ \ \forall i, 2 \leq j \leq n \tag{4.18}$$

$$b_i^0 = p_i^0, s_i^0 = 0 \ \ \forall i \tag{4.19}$$

There are three primary differences from the vanilla RED-BL formulation. The first difference is in the objective function, where the idling, bootup and shutdown costs depend on the number of granular units involved in the idling, bootup or shutdown process respectively. Since the computational cost component of power consumption only depends on the workload and is independent of the data center capacity, it is independent of the number of granular units being used at a data center during a given interval. The second difference is in the domain of p_i^j, b_i^j and s_i^j (see constraint 4.15). The third difference is in the constraint 4.16, which ensures that p_i^j takes on an appropriate value from $0, 1, ..., l$.

In the current evaluation scenario, we explore how the RED-BL electricity cost savings scale with change in size of the unit of independent (de)activation. In Figure 4.11, we have

plotted the percentage savings in electricity cost vs the granularity of data center's elastic load (de)activation. The savings are computed against the scenario where only the entire elastic load in the data center may be (de)activated as a whole. This baseline corresponds to a value of l equal to one. Accordingly, in Figure 4.11 we see no savings for that value of l. We also see that the ability to independently (de)activate half of a data center's elastic load provides around 2.5% additional such savings on top of what the vanilla RED-BL can achieve. The electricity cost savings grow almost linearly when going to more granular size of independent (de)activation.

4.6.4 Sliding window re-optimization

With the exception of scenario 3, all of our simulation scenarios have been driven by error-free workload traces. The underpinning assumption to the corresponding results, therefore, is the availability of accurate workload estimates. We opine that this is not such a bad assumption given that the cumulative workload on the granularity of an hour changes slowly from one hour to the next and from one hour on a day to the same hour the next day. However, workload forecasting will have some error, however small it may be.

In order to accommodate workload mis-estimation, we propose a sliding window-based algorithm that is somewhat similar to receding horizon control [76]. The algorithm is so called because it performs workload forecasts for a planning window and later slides the window by a constant offset before making another workload forecast, this time for intervals currently in the planning window.

Our approach compensates for workload estimation errors on two (often) different timescales. At an (often) longer timescale, the workload for the next n intervals is forecast and a RED-BL plan is generated. This forecasting is repeated every γ intervals, the *window slide interval* parameter. The motivation is that at interval number 1, the workload forecast for interval number $gamma + k$ is expected to contain a greater amount of error than if the fore-

cast for the same interval is done at interval γ, due to availability of more historical workload data at the later interval. This step of repeated forecasting and subsequent generation of a RED-BL plan is called *global trajectory correction*. The basic idea behind this approach is that lowering of workload estimation error should bring the planned state trajectory closer to the optimal state trajectory (the one that results from perfect workload estimates).

On a relatively shorter timescale, in each interval, our algorithm locally corrects for forecasting errors. The planned state for an interval may be *infeasible* in the sense that it may be based on an under-estimation of workload and we might not have sufficient active data center capacity for the actual workload. Also, in case of over-estimation of workload, the originally planned state may be *locally sub-optimal* as some data center resources would unnecessarily consume idling costs. This step that corrects for locally infeasible or sub-optimal states, considers only a single interval and, hence, is called *local trajectory correction*. Upon entering an infeasible or locally sub-optimal planned state, we perform a local correction by finding a new state for the current interval. As shown in Figure 4.9, we start at the initial state S_0 and are scheduled to transition to state $\hat{S}_1$ at interval 1. However, at interval 1, we know the actual workload received and might discover that the planned state is locally infeasible or sub-optimal. To accommodate this, we transition to a locally better state S_1. At the end of interval 1, we are scheduled to transition to state $\hat{S}_2$, and the process repeats. Since we only have accurate information about the workload for the present interval, the revision of the next state is always deferred to the next local trajectory correction step.

The local trajectory correction for interval j is an optimization problem that attempts to minimize the electricity cost of the corrected state S_j and the cost of transition between the planned state $\hat{S}_j$ and the corrected state. The mixed integer linear programming formulation for the local trajectory correction step for interval j is as follows:

$$\text{minimize} \sum_{i=1}^{m} c_i e_i^j (p_i^j \lambda (f + (1-f)\frac{x_i^j}{c_i}) + (b_i^j + \hat{b}_i^j)\sigma + (s_i^j + \hat{s}_i^j)\delta) \tag{4.20}$$

subject to:

$$x_i^j \leq c_i \ \forall i \tag{4.21}$$

$$\sum_{i=1}^{m} x_i^j = w^j \tag{4.22}$$

$$p_i^j, b_i^j, s_i^j \in \{0, 1\} \ \forall i \tag{4.23}$$

$$p_i^j \geq x_i^j \ \forall i \tag{4.24}$$

$$b_i^j \geq p_i^j - \hat{p}_i^j \ \forall i \tag{4.25}$$

$$s_i^j \geq \hat{p}_i^j - p_i^j \ \forall i \tag{4.26}$$

$$\hat{b}_i^j \geq \hat{p}_i^j - p_i^{j-1} \ \forall i \tag{4.27}$$

$$\hat{s}_i^j \geq p_i^{j-1} - \hat{p}_i^j \ \forall i \tag{4.28}$$

Given that the planning window size is n intervals, the possible values for γ are $1, 2, .., \gamma$. We experimented with all possible values for γ. Figure 4.6 shows the flow of our experiments. We first pick a value for the window sliding interval and estimate the workload for the next n-intervals and then invoke RED-BL. As an example, consider $\gamma = 2$. We start by forecasting the workload for the first n-intervals, denoted by $\hat{W}_1^2 = [\hat{w}_{1,1}^2, \hat{w}_{2,1}^2, ..., \hat{w}_{n,1}^2]$. Here, $\hat{w}_{j,1}^2$, for instance, represents the workload forecast for interval j during the first forecasting operation while the value of γ is 2. For workload forecasting, we trained an ARMA(4, 4) [77] model on a day's workload. Using $\hat{W}_1^2$ as the expected workload vector, we propose a RED-BL deployment plan for the first n-intervals. After the lapse of γ intervals, i.e., at

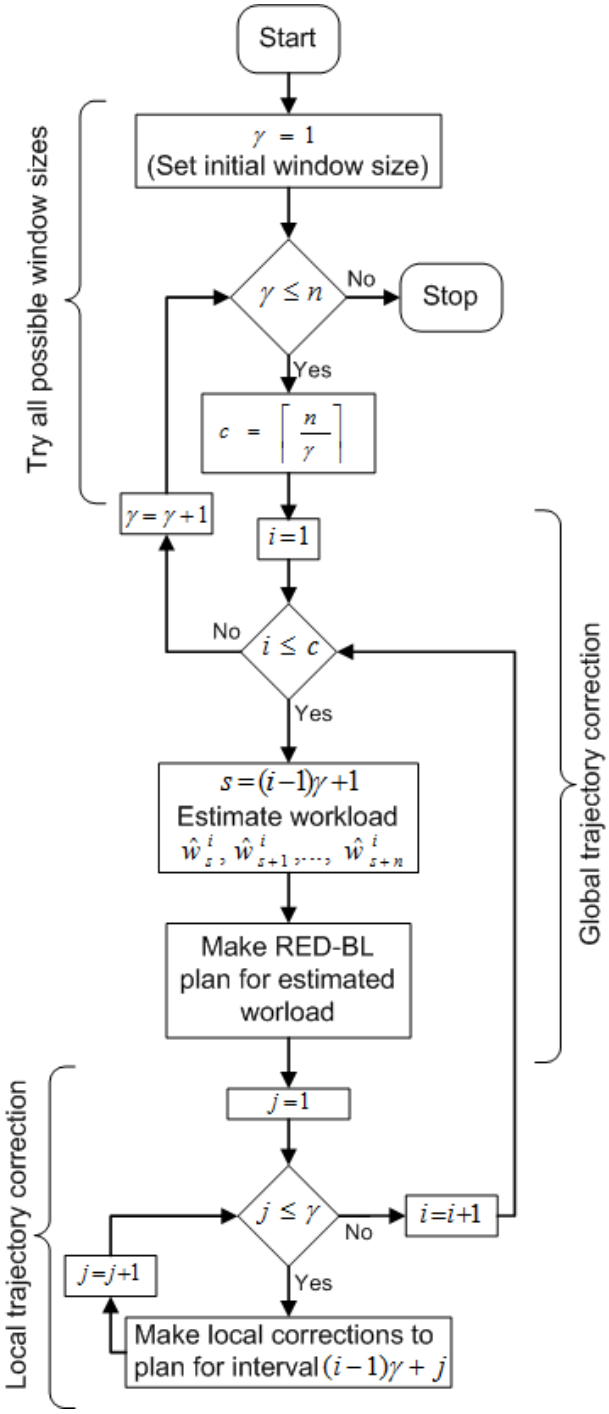

Figure 4.6: Flow for Sliding Window Optimization Experiments

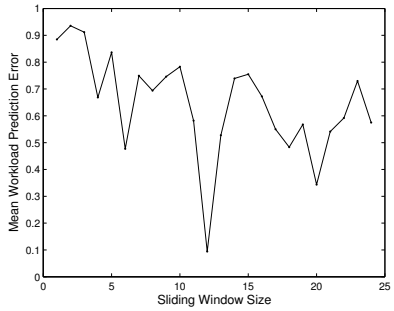

Figure 4.7: Mean absolute workload prediction error vs sliding window size

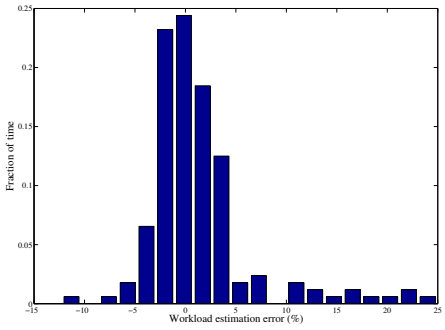

Figure 4.8: Distribution of workload prediction error for sliding window size of 12 hours

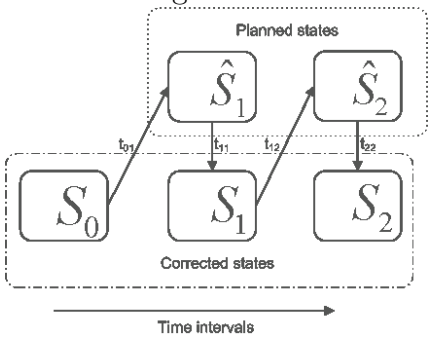

Figure 4.9: Local trajectory correction technique for three consecutive intervals

the start of the third interval (for $\gamma = 2$), we forecast the workload for the next n intervals, leveraging the additional information about the actual workload for the first two intervals which was not available in the first forecast step at $t = 0$. This forecast is denoted by $\hat{W}_2^2 = [\hat{w}_{3,2}^2, \hat{w}_{4,2}^2, ..., \hat{w}_{n+2,2}^2]$. Then, we compute the RED-BL deployment plan for intervals $3, 4, ..., n + 2$ as the global trajectory correction step. Since the window sliding interval size is γ and the number of intervals in our experiments is n, the number of times the window must slide, for a given value of γ is $\lceil n/\gamma \rceil$. For $\gamma = 1$, our scheme reduces to something resembling receding horizon control.

Having trained the model on the first day's data, we ran experiments for the last six days' workload in our dataset. We computed the average error of the daily electricity cost reported by these experiments compared to the total daily electricity cost for the same period with perfect workload estimates. The size of the planning window was set to 24 hours.

The first set of results in this scenario is the percentage workload estimation error for various sliding window sizes. We see in Figure 4.7 that the mean absolute percentage prediction error is less than 1%. The minimum mean error is for a sliding window size of 12 hours. For this sliding window size, the distribution of percentage workload estimation error is plotted in Figure 4.8. Most of the workload estimates are quite close to error-free, while a few estimates are as much as 24% off. This low average error for $\gamma = 12$ is expected due the nature of daily variations in the cumulative workload.

The difference of the electricity cost resulting from the use of the sliding window trajectory correction approach compared to the optimal solution with perfect workload knowledge is plotted in Figure 4.10. We see that the electricity cost achievable with RED-BL in a sliding window fashion is within 5-7% of the optimal cost achievable with perfect workload estimates.

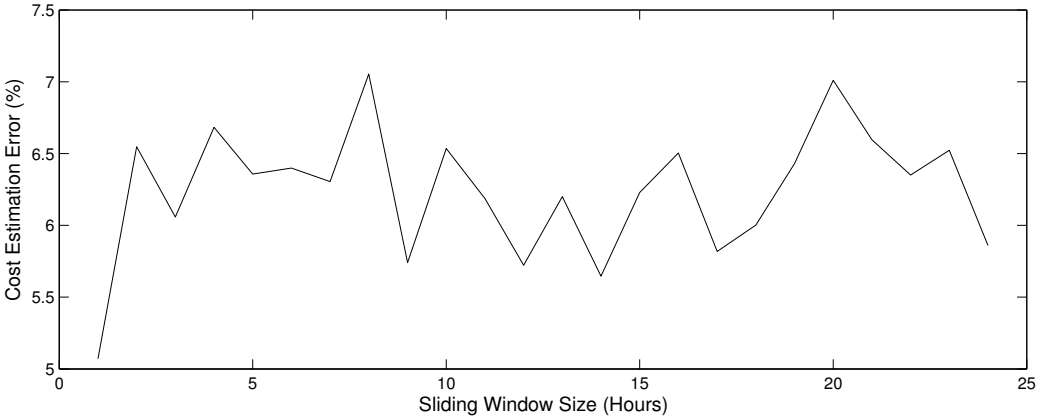

Figure 4.10: Percentage error of sliding window forecasts compared to global optimal with error-free workload

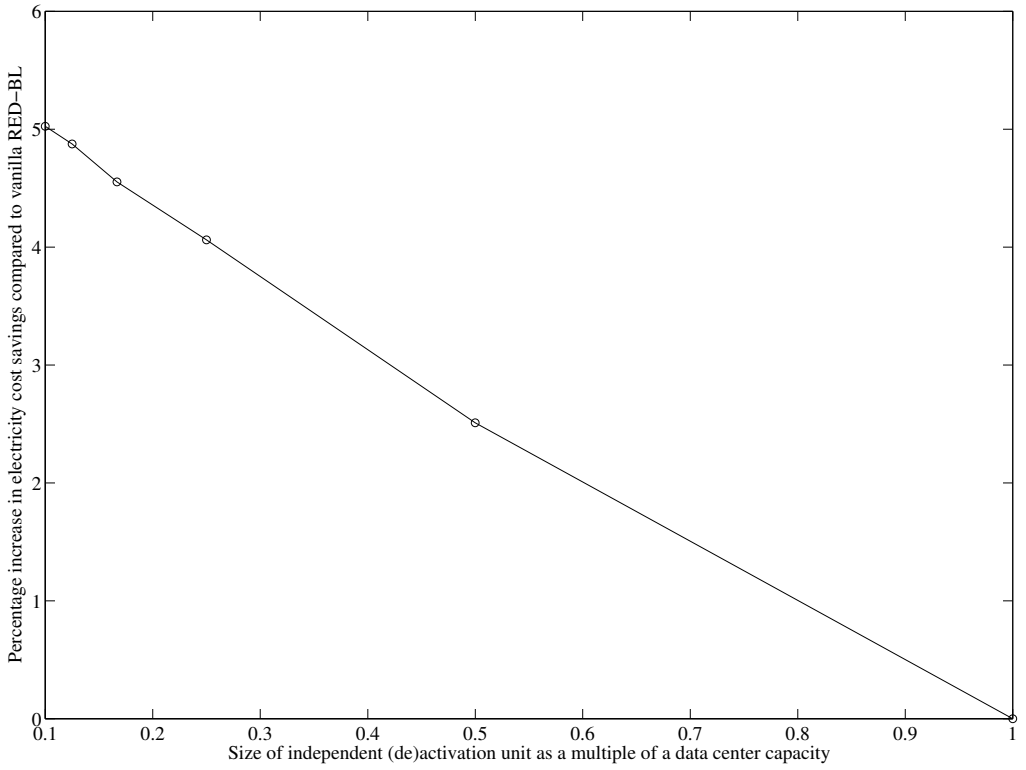

Figure 4.11: Cost saving vs (de)activation granularity

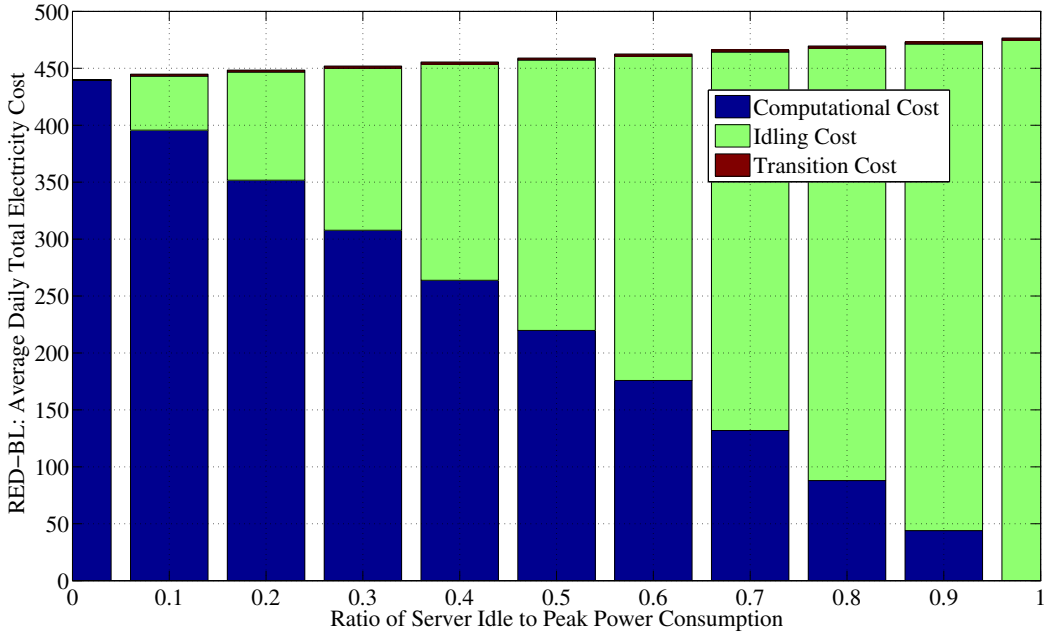

Figure 4.12: For bs = 0.01

4.7 Sensitivity of electricity cost savings to the server idle-peak power ratio

4.8 Performance of the heuristic algorithm

Figure 4.14 shows the performance of our heuristic algorithm compared to the optimal solution of the problem for various values of the (de)activation overhead parameters. For each value of the b/s parameters, we have plotted the average error over the seven days in our workload dataset (the curve) as well as the minimum and maximum error for any given day (the vertical bars). The performance of the heuristic is the worst for $b/s = 0$, because the heuristic avoids bootup/shutdown which has zero cost for $b/s = 0$. For other small values of b/s also, the bootup/shutdown overhead is not significant and by avoiding it, our heuristic fares relatively poorly compared to the optimal solution. As the value of b/s increases, our

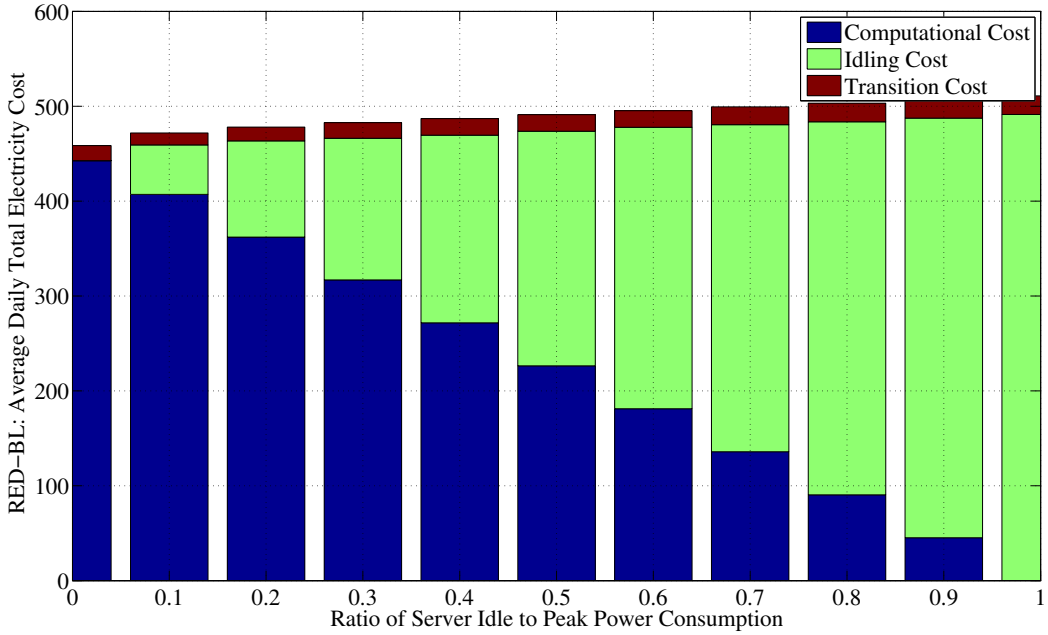

Figure 4.13: For bs = 0.65

heuristic's error compared to the optimal solution drops until it starts a slight rise. The rising trend in the heuristic's performance for relatively high values of b/s is because in this regime, it may often be better to allow idling of some data centers instead of a bootup at the intervals defined by p_1 and shutdown at those defined by p_2. We observed similar trends for other values of f as well, when b/s is varied from 0 to 1.

4.9 Discussion

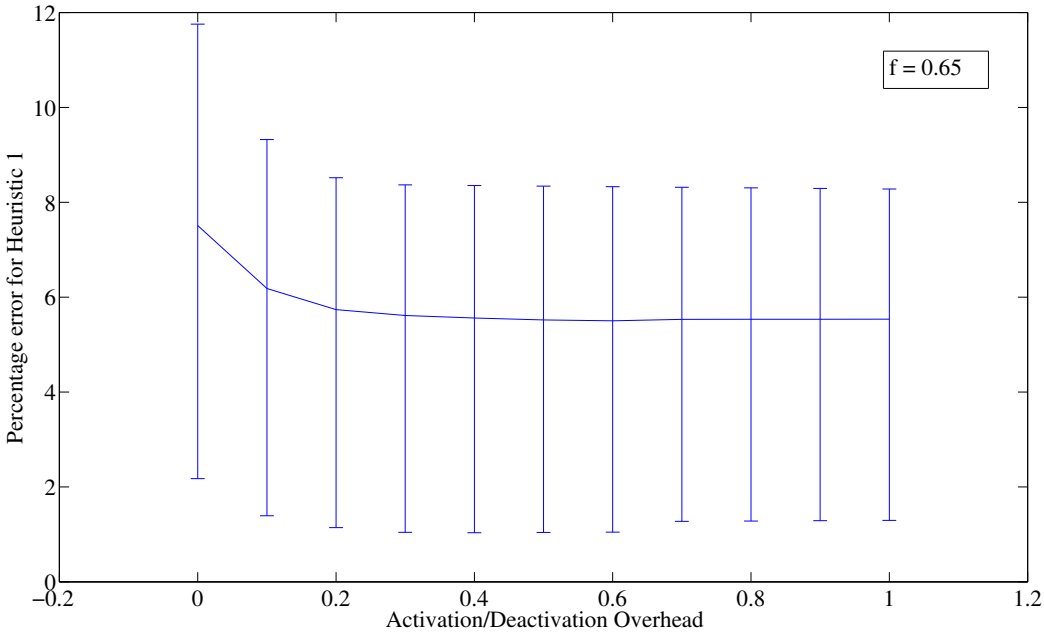

Figure 4.14: The minimum, maximum and average percentage difference between the cost of our heuristic and RED-BL

Chapter 5

Case Study II: Cellular Networks

5.1 Instantiating the generalized optimization formulation

Derive the objective function and constraints. Clearly outline the assumptions that we've made about the geo-diverse data centers.

5.2 Experimental setup

5.3 Results

5.3.1 Sensitivity of electricity cost savings to the duration of an optimization interval

We may optimize at different frequencies, such as once an hour or twice an hour. In this section, we study the sensitivity of electricity cost savings to the frequency of re-optimization

5.3.2 Sensitivity of electricity cost savings to the resource pruning granularity

We may have two states for a BTS: (i) 6+6+6, (ii) 3+3+3. Or, we may have three states: (i) 6+6+6, (ii) 4+4+4, and (iii) 2+2+2. How do the two-state and three-state resource pruning granularity settings comapre in terms of electricity cost savings?

5.3.3 Sensitivity of electricity cost savings to the margin of state-change damping

Suppose that we are using a two-state resource pruning model. If t_{max} is the call capacity of a 6+6+6 site, then the call capacity of the half-pruned site is $t_{max}/2$. If we deactivate TRXs immediately when the instantaneous call volume reaches $t_{max}/2$, we are likely to have many transitions due to short-term variations in call volume. We, therefore, wait until the instantaneous call volume is $t_{max}/2 - \epsilon$ before we switch to a $3 + 3 + 3$ configuration. The value of ϵ is a configurable parameter which can take a value from 0 (very aggressive, lots of transients, perhaps more savings) to $t_{max}/2$ (very conservative, no transients, no savings either). How do the electricity cost savings vary with the value of ϵ.

5.4 Discussion

Chapter 6

Conclusions and Future Work

6.1 Contributions

Describe the contributions made by this thesis

6.2 Limitations

Discuss the limitations of our work

6.3 Future work

Future directions

Bibliography

[1] J. Verge, "Google Pumps $400 Million More into Iowa, Investment Now Tops $1.5 Billion," April 2013. [Online; accessed 23-May-2013].

[2] "Mobile Broadband: The Benefits of Additional Spectrum," tech. rep., Federal Communications Commission, October 2010. White Paper.

[3] R. Miller, "Facebook: $50 Million A Year on Data Centers," September 2010. [Online; accessed 24-May-2013].

[4] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," in *Computer Communications Review*, vol. 39, January 2009.

[5] S. Mbakwe, M. T. Iqbal, and A. Hsaio, "Design of a 1.5kw hybrid wind/photovoltiac power system for a telecoms base station in remote location of benin city nigeria," in *IEEE NECEC*, November 2011.

[6] T. Italia, "Telecom Italia Annual Report 2012," January 2013. [Online; accessed 24-May-2013].

[7] C. Peng, S.-B. Lee, S. Lu, H. Luo, and H. Li, "Traffic-driven power saving in operational 3g cellular networks," in *Proceedings of the 17th annual international conference on*

Mobile computing and networking, MobiCom '11, (New York, NY, USA), pp. 121–132, ACM, 2011.

[8] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture*, (New York, NY, USA), pp. 13–23, ACM Press, 2007.

[9] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S. Diot, "Packet-level traffic measurements from the sprint ip backbone," *Network, IEEE*, vol. 17, no. 6, pp. 6–16, 2003.

[10] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer*, vol. 40, pp. 33–37, 2007.

[11] V. Inc, "Reduce Energy Costs and Go Green With VMWare Green IT Solutions," March 2009. [Online; accessed 26-May-2013].

[12] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, "Managing energy and server resources in hosting centers," *SIGOPS Oper. Syst. Rev.*, vol. 35, pp. 103–116, Oct. 2001.

[13] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive internet services," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, (Berkeley, CA, USA), pp. 337–350, USENIX Association, 2008.

[14] D. Meisner, B. T. Gold, and T. F. Wenisch, "Powernap: eliminating server idle power," in *Proceedings of the 14th international conference on Architectural support for programming languages and operating systems*, ASPLOS XIV, (New York, NY, USA), pp. 205–216, ACM, 2009.

[15] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in *IEEE INFOCOM*, 2011.

[16] A. Qureshi, "Plugging Into Energy Market Diversity," in *7th ACM Workshop on Hot Topics in Networks (HotNets)*, (Calgary, Canada), October 2008.

[17] A. Vahdat, M. Al-Fares, N. Farrington, R. Mysore, G. Porter, and S. Radhakrishnan, "Scale-out networking in the data center," *Micro, IEEE*, vol. 30, no. 4, pp. 29–41, 2010.

[18] D. Abts and B. Felderman, "A guided tour of data-center networking," *Commun. ACM*, vol. 55, pp. 44–51, June 2012.

[19] P. Mockapetris, "Domain names - concepts and facilities." RFC 1034 (INTERNET STANDARD), Nov. 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936.

[20] P. Mockapetris, "Domain names - implementation and specification." RFC 1035 (INTERNET STANDARD), Nov. 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2673, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, 6604.

[21] T. Berners-Lee, R. Fielding, and H. Frystyk, "Hypertext Transfer Protocol – HTTP/1.0." RFC 1945 (Informational), May 1996.

[22] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A power benchmarking framework for network devices," in *Proceedings of the 8th International IFIP-TC 6 Networking Conference*, NETWORKING '09, (Berlin, Heidelberg), pp. 795–808, Springer-Verlag, 2009.

[23] A. Vishwanath, J. Zhu, K. Hinton, R. Ayre, and R. Tucker, "Estimating the energy consumption for packet processing, storage and switching in optical-ip routers," in *Opti-*

cal Fiber Communication Conference/National Fiber Optic Engineers Conference 2013, p. OM3A.6, Optical Society of America, 2013.

[24] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *In Proc. IEEE INFOCOM*, 2008.

[25] Wikipedia, "List of mobile network operators," 2013. [Online; accessed 09-July-2013].

[26] J. T. Louhi, "Energy efficiency of modern cellular base stations," in *IEEE INTELEC '07*, (New York, NY, USA), pp. 475–476, IEEE, 2007.

[27] E. Oh, B. Krishnamachari, X. Liu, and Z. Niu, "Toward dynamic energy-efficient operation of cellular network infrastructure," in *IEEE Communications Magazine*, June 2011.

[28] J. Lorincz, T. Garma, and G. Petrovic, "Measurements and modelling of base station power consumption under real traffic loads," *Sensors*, vol. 12, no. 4, pp. 4281–4310, 2012.

[29] B.-Y. Choi, S. Moon, Z.-L. Zhang, K. Papagiannaki, and C. Diot, "Analysis of point-to-point packet delay in an operational network," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1797–1807 vol.3, 2004.

[30] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer-Verlag, 2005.

[31] C. Chekuri and S. Khanna, "A ptas for the multiple knapsack problem," pp. 213–222, 1999.

[32] N. Padhy, "Unit commitment-a bibliographical survey," *Power Systems, IEEE Transactions on*, vol. 19, no. 2, pp. 1196–1205, 2004.

[33] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, 1960.

[34] K. G. Brill, "The invisible crisis in the data center: The economic meltdown of moore's law," tech. rep., The Uptime Institute, 2007.

[35] C. L. Belady, "In the data center, power and cooling costs more than the it equipment it supports," *Electronics Cooling*, vol. 23, Jan. 2007.

[36] S. Pelley, D. Meisner, T. F. Wenisch, and J. W. VanGilder, "Understanding and abstracting total data center power," *Proc. 2009 Workshop on Energy Efficient Design (WEED '09)*, June 2009.

[37] "Five ways to reduce data center server power consumption." White Paper, Apr. 2008.

[38] J. Li, Z. Li, K. Ren, and X. Liu, "Towards optimal electric demand management for internet data centers," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 183–192, 2012.

[39] M. Lin, A. Wierman, L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in *INFOCOM, 2011 Proceedings IEEE*, pp. 1098 –1106, april 2011.

[40] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing server energy and operational costs in hosting centers," in *ACM SIGMETRICS*, (New York, NY, USA), pp. 303–314, ACM, 2005.

[41] M. Mazzucco and D. Dyachuk, "Balancing electricity bill and performance in server farms with setup costs," *Future Generation Computer Systems*, vol. 28, no. 2, pp. 415 – 426, 2012.

[42] 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 2010 (INFOCOM'2010), (San Diego, USA), March 2010.

[43] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the Electric Bill for Internet-Scale Systems," in *ACM SIGCOMM*, (Barcelona, Spain), August 2009.

[44] E. Kayaaslan, B. B. Cambazoglu, R. Blanco, F. P. Junqueira, and C. Aykanat, "Energy-price-driven query processing in multi-center web search engines," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, (New York, NY, USA), pp. 983–992, ACM, 2011.

[45] N. Buchbinder, N. Jain, and I. Menache, "Online job-migration for reducing the electricity bill in the cloud," in *Proceedings of the 10th international IFIP TC 6 conference on Networking - Volume Part I*, NETWORKING'11, (Berlin, Heidelberg), pp. 172–185, Springer-Verlag, 2011.

[46] U. Bhaskar and L. Fleischer, "Online mixed packing and covering," *CoRR*, vol. abs/1203.6695, 2012.

[47] R. Urgaonkar, B. Urgaonkar, M. J. Neely, and A. Sivasubramaniam, "Optimal power cost management using stored energy in data centers," in *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, SIGMETRICS '11, (New York, NY, USA), pp. 221–232, ACM, 2011.

[48] M. Mazzucco, D. Dyachuk, and R. Deters, "Maximizing cloud providers revenues via energy aware allocation policies," *CoRR*, vol. abs/1102.3058, 2011.

[49] F. Y.-K. Oh, H. S. Kim, H. Eom, and H. Y. Yeom, "Enabling consolidation and scaling down to provide power management for cloud computing," in *Proceedings of the 3rd USENIX conference on Hot topics in cloud computing*, HotCloud'11, (Berkeley, CA, USA), pp. 14–14, USENIX Association, 2011.

[50] L. Rao, X. Liu, L. Xie, and Z. Pang, "Hedging against uncertainty: A tale of internet data center operations under smart grid environment," *Smart Grid, IEEE Transactions on*, vol. 2, pp. 555 –563, sept. 2011.

[51] 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 (HotPower '09)*, pp. 1–5, 2009.

[52] X. Zheng and Y. Cai, "Energy-aware load dispatching in geographically located internet data centers," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 4, pp. 275 – 285, 2011.

[53] G. Koutitas and P. Demestichas, "Challenges for energy efficiency in local and regional data centers," *Journal of Green Engineering*, vol. 1, pp. 1 – 32, October 2010.

[54] Z. Abbasi, T. Mukherjee, G. Varsamopoulos, and S. Gupta, "Dahm: A green and dynamic web application hosting manager across geographically distributed data centers12," *Atlanta*, vol. 60, p. 80, 2011.

[55] L. Chiaraviglio and I. Matta, "Greencoop: cooperative green routing with energy-efficient servers," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, pp. 191–194, ACM, 2010.

[56] D. H. Phan, J. Suzuki, R. Carroll, S. Balasubramaniam, W. Donnelly, and D. Botvich, "Evolutionary multiobjective optimization for green clouds," in *ACM GECCO 2012*, ACM, 2012.

[57] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. Andrew, "Greening geographical load balancing," in *Proceedings of the 2011 ACM SIGMETRICS*, SIGMETRICS '11, (San Jose, California, USA), ACM, 2011.

[58] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. Andrew, "Geographical load balancing with renewables," in *Proceedings of the 2011 ACM GREENMETRICS*, GREENMETRICS '11, ACM, 2011.

[59] A. Sucevic, L. L. Andrew, and T. T. Nguyen, "Powering down for energy efficient peer-to-peer file distribution," *SIGMETRICS Perform. Eval. Rev.*, vol. 39, pp. 72–76, Dec. 2011.

[60] Z. Fang, L. Zhao, R. R. Iyer, C. F. Fajardo, G. F. Garcia, S. E. Lee, B. Li, S. R. King, X. Jiang, and S. Makineni, "Cost-effectively offering private buffers in socs and cmps," in *Proceedings of the international conference on Supercomputing*, ICS '11, (New York, NY, USA), pp. 275–284, ACM, 2011.

[61] F. Javed and N. Arshad, "On the use of linear programming in optimizing energy costs," in *Proceedings of the 3rd International Workshop on Self-Organizing Systems*, IWSOS '08, (Berlin, Heidelberg), pp. 305–310, Springer-Verlag, 2008.

[62] T. Logenthiran, D. Srinivasan, and A. M. Khambadkone, "Multi-agent system for energy resource scheduling of integrated microgrids in a distributed system," *Electric Power Systems Research*, vol. 81, no. 1, pp. 138 – 148, 2011.

[63] G. Celli and F. Pilo, "Optimal distributed generation allocation in mv distribution networks," in *Power Industry Computer Applications, 2001. PICA 2001. Innovative Computing for Power - Electric Energy Meets the Market. 22nd IEEE Power Engineering Society International Conference on*, pp. 81 –86, 2001.

[64] F. Javed and N. Arshad, "Adopt: An adaptive optimization framework for large-scale power distribution systems," *Self-Adaptive and Self-Organizing Systems, IEEE International Conference on*, vol. 0, pp. 254–264, 2009.

[65] M. A. Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, "A simple analytical model for the energy-efficient activation of access points in dense wlans," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, e-Energy '10, (New York, NY, USA), pp. 159–168, ACM, 2010.

[66] C. Cavdar, A. Yayimli, and L. Wosinska, "How to cut the electric bill in optical wdm networks with time-zones and time-of-use prices," in *Optical Communication (ECOC), 2011 37th European Conference and Exhibition on*, pp. 1 –3, sept. 2011.

[67] S. Lee and S. Sahu, "Efficient server consolidation considering intra-cluster traffic," in *GLOBECOM*, pp. 1–6, 2011.

[68] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath, "Load balancing and unbalancing for power and performance in cluster-based systems," 2001.

[69] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *INFOCOM, 2012 Proceedings IEEE*, pp. 1431 –1439, march 2012.

[70] H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, F. B. Cetin, and S. Babu, "Starfish: A self-tuning system for big data analytics," in *5th Biennial Conference on Innovative Data Systems Research (CIDR) 2011*, pp. 261 –272, January 2011.

[71] H. Herodotou, F. Dong, and S. Babu, "No one (cluster) size fits all: automatic cluster sizing for data-intensive analytics," in *Proceedings of the 2nd ACM Symposium on Cloud Computing*, SOCC '11, (New York, NY, USA), pp. 18:1–18:14, ACM, 2011.

[72] D. Aikema and R. Simmonds, "Electrical cost savings and clean energy usage potential for hpc workloads," in *Sustainable Systems and Technology (ISSST), 2011 IEEE International Symposium on*, pp. 1 –6, may 2011.

[73] J. Pang, A. Akella, A. Shaikh, B. Krishnamurthy, and S. Seshan, "On the responsiveness of DNS-based netowrk control," in *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pp. 21–26, ACM, 2004.

[74] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet Routing Convergence," in *Proc. of the ACM SIGCOMM*, August 2000.

[75] A. Nazir, S. Raza, and C.-N. Chuah, "Unveiling facebook: a measurement study of social network based applications," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, IMC '08, (New York, NY, USA), pp. 43–56, ACM, 2008.

[76] W. H. Kwon and S. H. Han, *Receding Horizon Control*. Elsevier, 2005.

[77] G. Box, G. M. J. , and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. Prentice-Hall, 1994.