Fall 12-1-2011

Distributed Algorithms for Energy Savings in the Core Network

Lin Liu
University of Nebraska-Lincoln, linliu@cse.unl.edu

DISTRIBUTED ALGORITHMS FOR ENERGY SAVINGS IN THE CORE NETWORK

by

Lin Liu

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Byrav Ramamurthy

Lincoln, Nebraska

December, 2011

DISTRIBUTED ALGORITHMS FOR ENERGY SAVINGS IN THE CORE NETWORK

Lin Liu, M. S.

University of Nebraska, 2011

Adviser: Byrav Ramamurthy

Recently, many efforts have been undertaken to reduce the energy consumption of core networks. Bundle link is a commonly deployed technique in core networks to combine several high-speed physical sublinks into a virtual connection to achieve bandwidth upgrade flexibility and network reliability. The traffic passing through a bundle link can be carried fully over the first few sublinks (bin packing) or evenly distributed over all sublinks (load balancing). In the current network when a bundle link is on, all of its sublinks are on, thus, selectively shutting down a few sublinks during periods of low traffic could save a large amount of energy while keeping the network topology stable. Previous green network research studies focused on centralized global-optimization techniques which intend to concentrate traffic into a small set of network nodes or links and shut down the other ones under the control of the network management system. Thus, they require frequent changes to the network topology and their solutions are not scalable even with the help of simplified heuristics.

We propose distributed local-optimized algorithms based on thresholds for both bin packing and load-balancing cases to dynamically adjust the number of active sublinks. In our algorithm the core routers rely on the link utilization during the previous time slot and use a threshold to trigger the sublinks' up or down operations. For each bundle link we always retain at least one active sublink to keep the network topology stable. We simulate an Internet2 based synthetic network using bundle links and conduct experiments for both bin-packing and load-balancing cases. The experiment results show that a great deal

of (up to 86%) energy consumed on core router ports could be saved with appropriate parameter value settings in both cases. Employing different parameter settings for different types of bursty links could greatly reduce congestion with limited loss of energy savings. Compared to previously proposed ILP (Integer linear programming) based centralized algorithms, our distributed algorithms can achieve high energy savings and result in fast, autonomous, topology-invariant and scalable solutions.

ACKNOWLEDGMENTS

First, I would like to express my appreciation to my advisor and committee chair, Professor Byrav Ramamurthy for his expertise, guidance and encouragement throughout the course of my master's study. His support has always been very generous in both time and research resources. This work would not have been done without his guidance. His technical and editorial advice and infinite patience were essential for the completion of this thesis. I feel privileged to have had the opportunity to study under him. I thank Professor Lisong Xu and Professor Hong Jiang for serving on my Master's Thesis defense committee; their involvement has greatly improved and clarified this work.

I thank all the CSE staff and friends, for their friendship and for all the memorable times in UNL. I would like to thank everyone who has helped me along the way.

I specially thank my wife without whose support none of this would have been possible. I thank my sons, who give me happiness and light up my life. Last, but not least, my deepest thanks go to my parents for their self-giving love and support throughout my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Energy Consumption of Network

Modern industry and society rely heavily on all kinds of electronic devices and these in turn are driven by scarce energy resources such as petroleum, gas and nuclear plant, etc. The energy consumption of the human world has increased rapidly for a long time and has reached 1.504 TWh in 2008 [1] and it is estimated to increase by 49% from 2007 to 2035 [2]. Specifically, electricity consumption has grown fastest and its rising speed (2.3%) is much greater than that of the worlds' energy demands (1.4%) [2]. ICT (Information and communication technology), as one of the major contributor of electricity consumption is responsible for 2% - 10% of the worldwide power consumption and the number is estimated to reach 50% in the future years [3]. Also ICT is estimated to account for approximately 2% global carbon dioxide emissions, more than that of the aviation industry [4]. Table 1.1 shows a rough statistic of electricity consumption status of five categories of equipment in the ICT industry [5]. From Table 1.1 we can see that Internet and its related infrastructure and equipments have occupied considerable volume of energy consumption of ICT.

Several other statistics also attest to the significance of network equipment energy con-

Table 1.1: Energy consumption for various equipment types of ICT

Equipment Type	Est. Consumption 2007 (GW)	Est. Annual growth rate
Data centers	26	12%
PCs	28	7.5%
Network Equipment	22	12%
TVs	40	9%
Others	40	5%
Total	156	

sumption in ICT. Powering wired networks in the United States alone costs an estimated 0.5-2.4 billion dollars per year and total energy consumption for networked systems reaches 150 TWh (at the cost of 15 billion dollars) [6] of which one fourth of the total energy consumption comes from networks and data centers [7]. A study estimated that the Internet equipment consumed roughly 8 % of the total energy (i.e. electricity) in the United States with the prediction of growth to 50 % within a decade. British Telecom, which consumes 0.7% of the total UK's energy in the winter of 2007 and was listed as the biggest single power consumer in UK [4].

There are several reasons for the increasing energy consumption by communication networks. First, the Internet is growing rapidly in terms of the number of users and available devices. Today's Internet has an enormous user base and it is continuously expanding and admitting a tremendous number of new users and electrical devices (such as smart phones, tablets, online gaming boxes and Internet TVs). Throughout the history of the Internet, people have focused on developing novel network techniques (such as 3G/4G mobile, WiFi and sensor network) and building up gigantic network infrastructures to try to enable us to access the Internet anytime, anywhere and at high speed. Also more and more electrical devices including phones, TVs, even watches and irrigation hoses are provided Internet-access functions for convenient data retrieval and remote control. The flourishing 3G/4G mobile network which enables its users to share photo, video and music in smartphones has poured a huge quantity of new streams of data onto the Internet. Second, the application modes and software architectures are changing. As cloud computing becomes the trend, more services rely on the network and the remote data centers to store and process data. Also some new applications such as P2P (Peer-to-Peer), Twitter and Facebook are generating multiple copies of information and broadcast them to multiple users. In particular the P2P applications inherently attempt to occupy as much bandwidth as possible to speed up the download. Third, the content of Internet itself is changing. Internet is no longer primarily transmitting kilobytes of text messages for each user. Current Internet services carry much more Mega or Giga byte sized voice and video data than ever before. The combination of above factors leads to the rapid increase of network bandwidth demands which forces ISPs (Internet Service Providers) to upgrade the network infrastructure to provide more and faster core and access switches, routers and links to accommodate the growing bandwidth demands. Thus, the network energy consumption especially in the core network has been growing rapidly.

The rapid increase of network energy consumption is closely related to the rapid increase of traffic demands due to the following reasons. First, the Internet bandwidth is growing fast and higher network bandwidth needs more powerful processing capabilities and cooling systems, thus, consuming more energy. The processing capabilities of routers and switches rely on the higher clock frequencies and design of high-speed ASIC chips in the switch boards and line cards. Although low-voltage chip technologies have been introduced, the overall power density in the chips is still increasing and the underlying power efficiency curve has started to became plateau [8]. Furthermore, the limits of traditional air cooling method to dissipate the heat of routers will soon be reached [8] while water cooling systems are much more costly. Second, current networks are designed to offer best-effort services and redundancy is provided everywhere. Over-provisioned link bandwidth and protection links are designed for peak traffic load or link failure which means that significant amounts of energy are wasted. Higher bandwidth also means greater variance of traffic [9]. Thus, network link bandwidth utilization can vary widely which also contributes to the low efficiency of network. In fact, the Internet inherently lacks power awareness and runs at low efficiency for a long time. One example of this is the pretty low network link utilization. On average, the link utilization of the Internet Service Provider (ISP) network is estimated at 30% - 40% [6]. As a result, the telecommunications industry is listed as one of the least efficient industries in the world. From the point of view of telecommunication carriers, energy consumption has more meaning than energy bills. Workloads exceeding the initial design capability of the energy system or the cooling system will force the ISP to upgrade or even physically reconstruct the whole site.

In conclusion, the power consumption of networks has increased rapidly and the situation has become more and more intolerable. Therefore, research on energy-aware network design or green networking has become popular in recent years. Realizing that current networks lack power awareness, all kinds of research attempts to save energy have been conducted during these years, such as low energy consumption hardware design, network protocol design, traffic scheduling, and green data center, etc. Among them, one of the major concerns to be addressed is the low link utilization of current networks.

Actually low link utilization is pretty common in core networks due to several practical reasons. First, it is normal for network traffic growth to deviate widely from what was projected. When an ISP commences constructing its network, it is hard to estimate how many customers it will service when the network is fully implemented. The network construction is a long process. Starting from network planing, bidding, purchasing, to the final system delivery and servicing, several years could pass by. Also, project implementations take lots of human resources and management time of ISP project departments. So, they mostly prefer to follow the step size of customer growth in a rough manner, that is, they hope to prepare a good enough infrastructure and reduce upgrade times to avoid repeated tedious work. Thereby, ISPs often deliberately overestimate the traffic demands and construct networks beyond the current demands. As a result, ISP networks commonly experience long periods of low utilization. The second reason for low network utilization comes from the coarse grained scalability of network bandwidth upgrades. When network traffic exceeds current network capacity, a bandwidth upgrade will be required. ISPs cannot upgrade the bandwidth to an arbitrary desired rate due to the limitation of current link techniques: only a limited number of interface rates are available. For example, upgrading 1 Gbps Ethernet connection to the next scale, will result in a 10 Gbps connection, and that possibly would produce low link utilization since the real traffic demand normally will not increase so quickly to ten times. Optical network techniques such as SDH (Synchronous optical networking) and SONET (Synchronous optical networking) have the same problem. The third and the most important reason for low link utilization is due to the unpredictable and bursty nature of Internet traffic. Internet traffic varies greatly according to the usrs' communication activities. The gap between the peak and the trough of the link traffic is huge within a day. As Internet works in a best-effort basis, usually the bandwidth of a specific link is set to be at least greater than the maximum expectation of future link traffic to reduce the possibilities of congestions.

To increase the flexibility of bandwidth upgrades, the bundle link or link aggregation technique is widely used in most core networks. The basic idea is to bundle several sublinks together to serve as a virtual link. That is, instead of upgrading to the expensive next higher rate link, we can gradually add more low-bandwidth sublinks to better follow the growth of rising traffic. In this way, the network utilization could be increased and the network capital and operational costs could be greatly reduced. A bundle link also helps to increase the reliability in case of link failure and the capability of disaster tolerance of a core network.

In the following, we first introduce some concepts such as core network and link aggregation technique. After that we discuss the motivation of this thesis.

1.2 Core Network

A core network is the central part of a telecom network that provides various services to customers who are connected by the access network [10]. Typically it refers to the high capacity communication facilities that connect multiple primary nodes in mesh or ring topologies. These topologies are designed based on the traffic patterns between the two nodes and the tradeoff between redundancy and cost optimization [11]. The connections between the nodes typically consist of wavelength-division multiplexed optical fiber links. In the largest core networks, between 40 and 80 wavelengths are used. [11]

The Internet backbone is the biggest and the most diverse core/backbone network in the world which has a set of principle routes between large, strategically interconnected networks and core routers in the whole world [12]. The Internet data routes are hosted by commercial, government, academic and other high-capacity network centers, the Internet exchange points and network access points, that interchange Internet traffic between the countries, continents and across the oceans of the world.

The devices and facilities in the core/backbone networks are switches and routers. The trend is to push the intelligence and decision making into access and edge devices and keep the core devices dumb and fast. Technologies used in the core and backbone facilities are data link layer and network layer technologies such as SONET, DWDM, ATM, IP, etc [10].

There were several major Internet backbone providers in the telecommunications industry such as Cable & Wireless Worldwide, UUNet, Sprint, AT&T and Verizon which own some of the largest Internet backbone networks and they sell their services to ISPs. Each ISP has its own contingency backbone network or outsourced backup. These networks are intertwined and criss-crossed to create a redundant network. Core routers are separately set up to navigate data through this diverse web that the backbone creates. Due to the high bandwidth demands, typically the trunk lines of backbone network consists of many fiber optic cables bundled together to increase the capacity [12]. As of 2011, the most common bandwidth of optical trunk link of backbone network is from 1Gbps to 40 Gbps. Some backbone networks are planning to increase their core connection bandwidth to 100 Gbps.

1.3 Link Aggregation and Over-provisioned Capacity

Link aggregation or bundle link is not a new concept. Also termed as trunk aggregation, it has several aliases depending on the standard organizations or equipment providers, such as Ether-Channel (Cisco), Link Aggregation Control Protocol (LACP, IEEE 802.3ad), Multi-link trunking (Nortel), Smartgroup (ZTE), EtherTrunk (Huawei), port channel, etc.

The formerly published bundle link standard is IEEE 802.3ad-2000 (LACP) which has now moved to the IEEE 802.1AX standard and most applications conform to LACP. LACP is a layer 2 control protocol that can be used to automatically detect, configure and manage multiple physical links between two adjacent LACP enabled devices into one bundle link while the aggregation technology actually can be implemented at any of the lowest three layers of the OSI model [13].

Core routers of major vendors support LACP, e.g., Cisco 10000 and 7600 series routers and Huawei Quidway S9300 Series Terabit Routing Switch. Cisco routers support bundle links across multiple chassis. The reasons to bundle multiple links together to work as one logical link are two-fold: bandwidth limitations and lack of resilience. As we discussed before, in Ethernet and optical networks, the bandwidth upgrades are by an order of fixed multiple such as 10. For example, Ethernet upgrades from 1 Gbits/s to 10 Gbits/s. Before traffic demands reach the next scale, carriers prefer to manually configure to add new Ethernet or SONET/SDH links alongside the existing ones and combine them into one logical link via link aggregation. For example, an ISP network which currently comprises of 10 Gbps connections can add three 10 Gbps links to each link to construct 40 Gbps bundle links. Also, considering a single link connection, the cable itself or ports can fail and hence a bundle link composed of multiple physical links could help to reduce the single points of failure. The current Internet backbone is composed of multiple networks and the trunk lines inside typically consist of many optical fibers bundled together to increase the capacity [12]. The bundle link technology is widely applied in current core network connections with the number of sublinks within one bundle link ranging from 2 to approximately 20 with most of them in the middle [6].

1.4 Motivation

This thesis focuses on exploring possible energy savings in the core routers of the backbone network using bundle links.

First, we would like to know how energy is consumed in the backbone network and core routers. Fig 1.1 shows a summary of the power consumption of core networks [11] derived from the data sheets of Juniper T series 4 routers. As we can see, links only con-

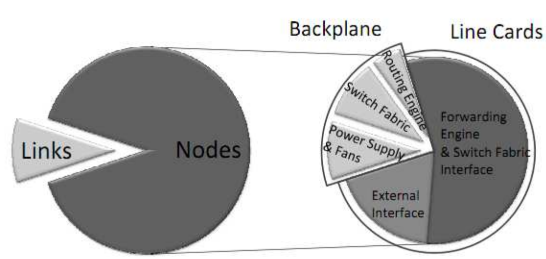


Figure 1.1: Summary of core power distribution

tribute approximately 10% to the total energy consumption while the remaining 90% is consumed in the core routers. Further, in the routers, 25% of the energy is consumed by the backplane which includes power supply and fans, routing engine and switch fabric. The remaining 75% is consumed in the line cards, divided between the forwarding engine and the switch fabric interface (75%) and the external interface (25%) [11]. Also the energy consumption of core equipments will include some overhead energy consumption which is from the cooling system, the Uninterruptible Power Supply (UPS) and some other facility equipments. The overhead energy consumption is closely related to the energy consumption of core ICT equipment and its amount ranges from 0.5 to 1 times of that of the core ICT equipment. From the above core power analysis, we know that saving energy consumption of core routers is meaningful and shutting down some core router ports could save a great deal of energy of the core network.

Many efforts related to green networking have been undertaken and some important research results have been obtained recently. Before the recent focus on green networking, researchers have applied sleep or frequency/voltage scaling mechanisms to save energy in traditional electrical devices. Adjusting the sleep mechanism achieves energy savings by powering down parts of system during idle periods while the latter achieved the target by lowering performance and voltage of system during active periods. Similar ideas have been proposed in network equipment such as routers and switches or their attached line cards, ports and links [7]. Some new technologies are viewed as revolutionary solutions for reducing the network energy consumption. For example, the applications of fiber have created extremely low power consumption in long-haul transmission systems comparing to that of traditional electrical-cable based transmission systems. Fiber based optical network has become the mainstream configuration of core networks. The ultimate target of optical networking, all-optical switching remains in the stage of theoretical research and the future development of this technology is expected to be able to fundamentally solve the network bandwidth and energy consumption problems. Currently the core network still relies on Optical-Electrical (O/E) and Electrical-Optical (E/O) conversion to assist in switching in the optical network. These fiber transmission systems together with the core routers and switches form the major energy consumers of the core network. Since energy consumed in long-haul fibers and optical transmission devices have been greatly reduced, the network equipment (core routers and switches) function as the a key factor of growing importance in terms of network energy savings.

In real network operations, the core switches, routers and their attached line cards and ports are deliberately kept active all the time and the core equipments (such as switches, routers, links and optical transmission systems) are commonly redundant-provisioned to provide the best network QoS (Quality of service) and the high availability of network, thus, great energy-saving opportunities could be explored in the energy-inefficient core network.

Our goal is to save energy in the ports of core routers with bundle links. When dealing with routers, there is an important issue which is different from dealing with network terminals, that is, whether or not to change the network topology while achieving network energy savings. As we know, shutting off a user computer or a server may not change the network topology and has very limited or no impact on the survivability or performance of the whole network; however, shutting down a core router or a bundle link attached to it, could have drastic consequences since it changes the network topology. From our understanding, changing network topology to achieve energy savings currently is not a good idea since a stable IP layer route is so critical that numerous upper-layer applications and protocols rely on it and frequent routing changes bring problems of packet loss, retransmission and serious delay, which are beyond the control of current routing protocols. Keeping network topology unchanged is a strict principle in our design of core network energy savings, which is different from some previous methods.

1.5 Opportunity

One interesting thing about a bundle link is that, although a bundle link consists of many physical sublinks, actually they can be shut down or brought up individually. Combined with the fact that core networks are capacity over-provisioned, it gives us a great opportunity to reduce network energy consumption if we can dynamically adjust the number of active sublinks within one bundle link, that is, shutting down some sublinks during idle periods or bringing up sublinks during busy periods according to the link traffic.

One presumption of this method is that the ports or line cards in the core routers should be able to be shut down or brought up quickly to avoid too much data loss, which actually requires hardware support from equipment vendors. Currently, the time taken for a port to shift from the idle state to the active state could be on the order of milliseconds [14], which is not quick enough. We believe that such quick switching technology will be developed soon if it could be shown to save a lot of energy.

Another issue we need to consider is about which mode we should shift the router ports to from the active state: low-rate mode or power down mode. The former is related to the rate scaling technique and the latter means totally shutting down the ports. Current research data shows that rate-scaling would not decrease energy consumption on the port significantly while shutting the ports down could bring the energy consumption down to zero [15]. That is the major reason we prefer shutting down the ports. Shutting down a line card should save more energy than shutting down its attached ports. Due to the reason that the number of ports in line cards differs in routers, to simplify the calculation, we use the number of ports shutdown instead of line cards to calculate the possible energy savings.

There are two kinds of traffic distribution methods in core network applying bundle link technique: bin-packing and load balancing. The former allows us to pack traffic into the minimum number of sublinks while the latter attempts to evenly distribute all the traffic across all the sublinks. In the bin-packing case, the utilization of the first several sublinks and the utilization of the last sublink could be very different, while in load balancing case, all the sublinks have almost the same link utilization. In this project, we design two different strategies for the energy savings for the two cases.

1.6 Contribution

In this thesis, we address the problem of reducing the energy consumption in core networks. We design several local threshold-based methods to automate the port operations of shutting down or bringing up sublinks to discover achievable energy savings. Our target is to investigate the trade offs between maximizing the energy savings from powering down as many sublinks as possible and minimizing the congestion cases by provisioning enough active sublinks to accommodate traffic shifts. Furthermore, we evaluate the performance of applying different parameter settings to the above algorithms. Finally, we separately apply the different parameter settings to different bundle links according to the historical traffic characteristics of these bundle links, in order to achieve a better tradeoff between energy savings and congestion risk. We conduct several experiments based on an Internet2-derived synthetic network. First, we study the maximum possible energy savings in an Internet2 derived synthetic network by locally shutting down sublinks during idle periods and bringing up sublinks during busy periods to get an idea of the upper bound of energy savings. Then we separately deploy our proposed algorithms to bin packing and load balancing cases. Furthermore, we evaluate the tradeoff performance in the two cases by changing the parameter settings and finally we achieve a better tradeoff by setting different parameter settings to different bundle links (based on their burstiness). The experiment results show that the locally optimized threshold based methods can save most of the port energy consumption in this synthetic core network and, by appropriately adjusting the parameter setting of each bundle link, the accompanying congestions can be greatly reduced with limited loss of energy savings. Compared to the slow, centralized, unscalable and topology-variant global-optimized green networking methods, our approach is a distributed, fast, autonomous, topology-invariant and scalable solution. Although the theoretical energy savings of our approach is less than those of the global-optimized methods, our approach still achieves most of the energy savings in a more practical way.

1.7 Outline

This thesis has been organized as follows. Chapter 2 first presents the general network energy saving methods in the literature and then discusses several global optimized methods proposed for green networks. Chapter 3 discusses our proposed local optimized threshold based methods and possible parameter settings. Chapter 4 describes the simulation experiments based on an Internet2-derived synthetic network and their results. Chapter 5 provides the conclusions and describes the possible future work.

Chapter 2

Related Work

Recently green networking research has drawn attention and become popular due to the daily rapidly increasing energy consumption and the resulting energy crisis. This research interest also comes from the requirements of the major core network carriers (such as AT&T and Sprint) and ISPs (Internet Service Providers) to reduce the cost of the network infrastructure and its corresponding energy consumption in order to increase profitability. The early methods focused on hardware redesign to reduce the network energy consumption. However, there is increasing realization that solely relying on developing low consumption silicon technologies is not enough to solve this problem; upper layer mechanisms should be involved in achieving great network energy savings [16]. Our thesis focuses on the energy savings of the core routers in the fixed core network with bundle links. In the following, first, we introduce the classification of some recent approaches to enabling green networking. Then, we discuss approaches particularly related to our work on core networks.

2.1 A General Taxonomy of Green Networking Approaches

In terms of energy savings, the research studies on different types of networks may have different objectives. For example, studies on sensor networks and cellular devices expend a great deal of effort on increasing battery life time of devices or deploying special antenna technologies to minimize the radiation power with fixed wireless signal coverage, thus, different methods and strategies could be applied in different types of networks and in different parts of the network. Here we only discuss the green networking concepts and approaches deployed in fixed networks, which can be divided into three categories: re-engineering, dynamic adaptation and sleeping/standby [16].

Re-engineering approaches aim at redesigning energy-efficient elements for network device architectures by optimizing the internal organization of devices and reducing their intrinsic complexity levels. These are usually hardware-based approaches consisting of new silicon technologies (e.g., for ASICs, FPGAs, network/packet processors, etc.), memory technologies (Ternary Content-Addressable Memory (TCAM), etc.) for packet processing engines, and novel media/interface technologies for network links (energy efficient lasers for fiber channels, etc.) [16]. The most challenging solution is the adoption of pure-optical switching architecture to replace the traditional electrical switching system. The study in [17] emphasizes that photonic technologies alone will not solve the looming energy bottle-neck problem. Some other typical re-engineering methods focus on decreasing voltages in chipsets, reducing the number of gates in the forwarding hardware, customizing silicon for packet forwarding and even synchronizing the operations of routers and scheduling traffic in advance [16].

Dynamic adaption approaches are designed to modulate capacities of network device resources (such as bandwidths used, computational capacities of packet processing engines, etc.) to meet actual traffic loads or service requirements. Two major approaches are power scaling and idle logic. The former allows dynamically reducing the working rate of processing engines or link interfaces while the latter allows rapidly turning off sub-components when no activities are performed, and bringing them up when the system receives new activities such as "Wake-on-packet" [16]. Generally these approaches need an optimization policy to configure and control the capabilities and states according to the estimated workload and service requirements and they are usually implemented as a software application. Many schemes and methods have been developed, however, to calculate the optimal solution requires significant computation and an estimation of the current workload which might not be feasible in all cases [16].

Sleeping/standby approaches are used to smartly and selectively drive unused network/device portions to low standby (with very low power consumption) modes, and to wake them up only if necessary. These approaches are characterized by higher energy savings and much longer wake-up time. The underlying fact of these approaches is that the sleeping equipment might lose its network connectivity and affect the availability of applications and services relying on it. For example, sleeping servers might lose their TCP connections. Some proxy methods such as Network Connectivity Proxy (NCP) are introduced to allow devices to overcome this problem. For example, an NCP can handle ARP, ICMP, DHCP, and other low-level network presence tasks for a group of network hosts [18].

2.2 Related Work

Previous energy saving methods in ICT have focused on the re-engineering of silicon chip or low-power-consumption hardware design. The study in [19] makes the first breakthrough in promoting energy saving in the Internet. The authors discover that the inefficiency of the Internet is worse than a typical wireless LAN due to networking devices in

Internet consuming a great deal of energy even when idle [19]. They classify the possible energy saving approaches at three levels: at an individual switch or router level; at the network level; at the Internet level to allow route adaptation and consider putting the router or some subcomponents of the router to sleep as an appropriate method to save energy. They discuss the prerequisites and strategies to implement sleeping and the impact of the sleeping approach on the implementation of popular switching and routing (OSPF and BGP) protocols. In [20], the authors investigate the feasibility of the sleeping option in LAN switches. Also they design a DELS (Dynamic Ethernet Link Shutdown) algorithm to achieve significant energy savings in a LAN while keeping packet delays within reasonable bounds. The DELS algorithm makes sleeping decisions based on buffer occupancy, the behavior during previous packet arrival times and a configurable maximum bounded delay. In [21], they introduce a mechanism to predict the number of packets that may arrive in a given interval of time to allow shutting down the link temporarily. This new algorithm achieves significant energy savings (40% - 80%) in typical LAN traffic and it works even for highly bursty traffic but at the cost of increased packet loss and delay. The study in [22] proposes three schemes for power reduction in switches − Time Window Prediction, Power Save Mode and Lightweight Alternative and the authors propose a novel architecture for buffering ingress packets using shadow ports. The results show that up to 32% energy savings could be reached with minimal increase in latency or packet-loss. With the support of Wake-on-Packet features, shadow ports and fast transitioning of the ports between their high and low power states, these savings reach 90% of the maximum theoretical savings.

The authors in [15] suggest two kinds of power management schemes that reduce network energy consumption: sleeping during idle periods and adapting the rate of network interface based on the traffic. The sleeping method uses a lot of time and energy when many transitions between sleeping state and active state happen. To solve this problem, they propose a smart buffering mechanism at the interface which relies on edge routers to group packets between the same source-destination pair and transmits them in bursts to minimize the number of transitions and maximize the sleeping time. By applying simple power management algorithm, their experiment shows that with the right hardware support, there is the potential for saving much energy with a small and bounded impact on performance, e.g., a few milliseconds of delay [15] .

In [8], the authors propose a general model for router power consumption based on energy consumption measurement of different configurations of widely used core and edge routers. Along with mixed integer optimization techniques, they explore the potential impact of power-awareness in a set of example networks. Their experiments show power consumption in experiments can vary by as much as an order of magnitude indicating that there may be substantial opportunities for reducing power consumption in the short term [8].

The authors in [23] apply multiple energy saving approaches into their work. They propose two types of approaches for power saving routers: power-efficient designing and the power saving designing. Power-efficient designing basically uses a hardware-advanced low-power-consumption approach in routers which includes integrated ASICs/FPGAs of routers, developing a scalable central architecture and using new high-speed memories and high-speed interfaces such as a SerDes. The experiment results show that they successfully developed a router with a throughput of over 1Tbps while the power per throughput (W/Gbps) was reduced by over 50% compared to conventional routers. In power saving design, they proposed an approach to cutting down wasted power consumption which includes methods for static performance control and dynamic performance control. The former can shut down the unused slots and ports in a switch in a static way and requires halting the forwarding procedure and the static performance control also applies the method of frequency switching, which is claimed to have achieved a savings of 10-20% power reduction during the power saving mode. The authors introduce the dynamic performance control as a promising power saving approach for next-generation routers in which they attempt to control the router's performance dynamically according to the amount of received traffic. This method uses the dynamically performance-controllable router architecture/circuit and the traffic monitoring/predicting technology. The dynamically performance controllable router architecture/circuit enables a multi-level fine granularity performance control and enables the switch to avoid performance degradation. The authors propose turning On/Off the individual packet processing engine to control the performance of the router. The traffic monitoring/predicting technology uses Energy Efficient Ethernet (EEE) which is an IEEE standard [24] by changing the PHY speed according to the incoming traffic, from 10Base-T to 10GBase-T to save energy at the Ethernet interfaces.

In [25], the authors propose a novel approach to switch off network nodes and links while still guaranteeing full connectivity and maximum link utilization. By defining which is the minimum set of routers and links that have to be used in order to support a given traffic demand, they attempt to power off links and even full routers while guaranteeing QoS constraints, such as maximum link utilization. Simple algorithms have been presented to select which elements have to be powered off, and simple scenarios have been considered to assess the proposed heuristics and the achieved energy savings. The authors provide an ILP (integer linear programing) formulation of the problem in which the objective function is not to minimize cost or maximize performance, but to minimize the total power consumed by the network, while connectivity and maximum link utilization are taken as constraints. This ILP problem is shown to fall in the class of capacitated multi-commodity flow problems, and therefore it is NP-complete; they proposed some heuristic algorithms to solve it. Simulation results in a realistic scenario show that it is possible to reduce the number of links and nodes currently used by up to 30% and 50% respectively during off-peak hours, while offering the same service quality.

An experiment of applying multiple energy saving methods in a real IP backbone network with the real traffic profile was conducted in [26]. In this experiment, the authors create a hierarchically designed topology which is similar to the actual topology of the national ISPs. In the designed topology there are 8 core nodes connected by 50 Gbps links, 52 backbone nodes connected by 20 Gbps links, 52 metro nodes connected by 10 Gbps links and 260 feeders connected by 10 Gbps links to metro nodes. The matrix of traffic demands are generated from some basic knowledge of Internet traffic such as 30% of traffic is confined within the same ISP, while 70% of traffic is coming from and going to other ISPs. After that, they propose a simple algorithm to select the network equipments that must be powered on in order to guarantee the service. The basic idea is to sort the devices according to the amount of energy they consume, and then try to power off first the devices that consume more energy. Also they first try to power off the nodes and then try to power off the remaining links. Simulation results in a realistic scenario show that it is possible to reduce more than 23% of total energy consumption, which corresponds to a saving of 3GWh/year.

In [27], the authors view the traffic load as one of the most influencing factors of the total energy consumption of equipment and study the energy consumption from a traffic load point of view. They introduced the concept of energy profile (EP) as the dependence of the energy consumption (in Watt hour, Wh) as a function of the traffic load or traffic throughput of a particular network component and proposed several energy profiles for telecom equipments such as Linear energy profile, Log10 energy profile, Log100 energy profile, Cubic energy profile and On-off energy profile. Then they extend the energy aware routing (EAR) towards an energy profile aware routing (EPAR) and include energy profiles into dimensioning, routing and traffic-engineering decisions. Experiments based on core topology of 50 locations and 88 edges are conducted to check the performance of applying different energy profiles and route optimization approaches. The results show energy savings of more than 35% can be achieved by applying energy profile aware routing compared to shortest path routing.

The authors in [28] addressed the problem of achieving a tradeoff between conflicting objectives of using a sleeping mode: to minimize the total energy consumed by the network elements and to minimize the end-to-end delay experienced by the connections. They propose routing and scheduling algorithms in scenarios which allow only a zero-rate sleeping state and a full-rate state for network equipments. They present different scheduling strategies in frame-based periodic scheduling or on a single line topology and realize scheduling for arbitrary topology by partitioning the network into a collection of lines and separately apply the schedule on them. In case the routing is given, they compared the performance of two schedules: active period proportional to local traffic or global traffic. The study shows that the end-to-end delay of the latter is much shorter.

The authors in [6] investigated energy savings in bundle links for the first time. They argue that removing entire links from the topology might cost too much (in terms of capacity and connectivity) and can lead to transient disruptions of the routing protocol; instead they propose shutting down some sublinks in a bundle link to avoid the problem. Given a topology, bundle size and traffic matrix, they attempted to identify the optimal set of cables to shut down and this problem is formulated as an Integer Linear Programming (ILP) which is NP-complete problem. They proposed three heuristics based on linear optimization techniques: Fast Greedy Heuristic (FGH), Exhaustive Greedy Heuristic (EGH), Bi-level Greedy Heuristic (BGH). Their methods are centralized and conducted in the network management system by using the traffic matrix and the network topology. The heuristics remove cables in a certain order until no further cables can be removed and differ in the order selection and the numbers of removed sublinks [6]. Experiments based on both synthetic and realistic topologies are conducted to evaluate the performance of three heuristic algorithms and the results show that their energy-saving performances are almost indistinguishable while the Fast Greedy Heuristic (FGH) algorithm takes much less running time than the other two.

Table 2.1: Summary of running times

Topology	FGH	EGH	BGH
Abilene	8±2 sec	50±8 sec	5±2 min
Waxman	50±20	17±5 hr	*
Hierarchical	14±4 min	*	*

Obviously, FGH is the preferred heuristic. Table 2.1 shows the running times of the three heuristics in the three network topologies used in the experiments. The asterisk represents an unacceptable running time.

From the experiment results, we can see that with more nodes, edges, and demands, the computation time for heuristics are increasing quickly and a scalability problem could arise there. Even for the quickest heuristic, FGH takes 8 ± 2 seconds in the Abilene, 5 ± 20 mins in the Waxman and 14 ± 4 mins in the Hierarchical topology respectively, which is too long to be practical. Imagine all the routers having to wait for at least 30 minutes to get the link scheduling solution from the network management in the Waxman topology. A lot of congestion could occur during this period. Therefore, these heuristics are hard to realize for real-time control and operation on bundle links and face serious scalability problems which will greatly limit the upgrading of the core network. Another disadvantage of these heuristics is the presumption that network demands are known in advance while in reality traffic demands in the backbone are shifting all the time and remain hard to predict. The differences between the presumed demands and the real traffic could totally change the calculation results and some bundle links might end up shutting down too many sublinks. The corresponding risks of congestion might offset the energy-saving benefit. Also the authors ignore exploiting the historical traffic information to assist in making decisions on port operations and their methods save energy at the cost of changing topology, which might negatively impact the performance of upper layer applications. We argue that, although combining several links' traffic together and conducting a global optimized precise calculation might save more energy, due to the inscrutability of link traffic patterns in the core network, it might be more reasonable to allow the router to shut down or bring up ports by itself based on the historical traffic information. This approach might not achieve the maximum energy savings but it should work faster, autonomously and avoid the scalability problem. As we discussed, previous centralized green core network research studies focused on global-optimization techniques which intend to concentrate traffic into a small set of network nodes or links and shut down the other ones. Thus, they require frequent changes to the network topology. Most of these global techniques actually are NP-complete problems whose solutions are not scalable even with the help of simplified heuristics. Considering this, our approach involves a distributed and local optimization algorithm.

2.3 Summary

In this chapter we first introduced the general classification of approaches to enable green networking. Then we discussed the green networking techniques applied in the fixed core network which are related to our work. Specifically we discussed several centralized global-optimized methods in green networks and a particular paper [6] which first addresses the topic of green core network operation in bundle link scenario. We analyzed the major problems of these centralized algorithms. In the next chapter, we discuss our proposed distributed local-optimized methods in the green core network.

Chapter 3

Problem Setting and Our Approach

In this chapter we formalize the problem setting and discuss our proposed distributed locally-optimized algorithms for energy savings in the core network. Basically these are link utilization threshold based algorithms to enable the core router to direct its port operations (shutting down or bringing up sublinks) while at the same time keeping the network topology unchanged. For the two bundle link traffic distribution cases (bin packing and load balancing), we separately developed two slightly different algorithms. Also we consider the strategy of deploying different parameter settings of these algorithms to different bursty bundle links to achieve a better tradeoff between the energy savings and the congestion risk.

3.1 Problem Setting

The problem is to determine at each time slot for each bundle link how many of its sublinks should be active. A centralized globally-optimized method [6] was previously proposed which however could result in some problems. The first problem is that their method needs the knowledge of traffic requirements of the whole network which is hard to estimate. The second problem is that their method is ILP (Integer Linear Programming) based which is an NP-complete problem and to solve it could be time consuming depending on the size and the complexity of the network, which makes their method not scalable. The disadvantages of their method were fully discussed in chapter 2.

We would like to propose a distributed algorithm which makes the sublink operation decision at the local core routers. That is, for each bundle link, its number of active sublinks is decided independently and locally. Ideally, the traffic demand of bundle links at each time slot are known in advance. Let us say for one bundle link L, the traffic demand in the next time slot is D. The number of sublinks in it is N and currently the number of active sublinks is C. Each sublink has bandwidth B and the utilization threshold of L is set as T. Then the formula for sublink determination for next time slot is shown in equation 3.1

$$A = \lceil \frac{D}{B * T} - C \rceil \tag{3.1}$$

If A is positive, it means that A sublinks need to be brought up; zero means nothing to do; otherwise, A sublinks could be shut down. One problem with this method is that D is hard to predict precisely and it is expected that there is variance between the estimated traffic and the real traffic. So equation 3.1 can be applied to calculate the optimum value but not be practical in calculation of real case. Actually since we don't know how much traffic increase or decrease in the next time slot maximumly, we can only speculate the traffic demand of the next time slot based on the actual traffic in the previous time slots. Here, we simply assume that only the traffic data in the last time slot are given as input.

Finally, the problem can be formalized as follows: in the core network, for each bundle link L, given its link traffic data D' in the last time slot and its number of sublinks N, find a good setting of the number of active sublinks C' for the next time slot to achieve a better tradeoff P between energy savings E and possible additional cost K. Here, the possible additional cost K is referred to as two issues. The first issue is the impact of congestion. Currently we count the occurrences of congestion to quantify their impact. In the future we want to count the total size of data loss due to congestion as the impact of congestion since it could give us a better sense of its influence to the network QoS (Quality of Service). The second issue is the cost of port operations (shutting down or bringing up router ports). A port operation consumes energy and might result in some data transmission delay. Due to different core routers might have different energy consumption and time delay in port operations, currently, we only count the number of port operations to show their different influence to energy savings in different parameter configurations. With more concrete data of port operations from core routers' providers, we can quantify the impact of port operations and introduce it into our tradeoff consideration. But that will be a part of our future work. In this study, the tradeoff is mainly between the energy savings and the occurrences of congestion. Also in some experiments we provide the number of port operations as a reference.

3.2 Our Approach

The basic idea of saving energy in a bundle link is that during each time slot we try to use the minimum number of sublinks to carry the whole link traffic to save energy. That is, when the traffic is low, we shut down some sublinks and the associated ports of routers, while retaining the Internet2 layer3 adjacencies; when traffic rises high enough, we bring up a few sublinks to satisfy the increased bandwidth requirement. To maintain an unchanged network topology, at any time, each bundle link should keep at least one sublink active.

To predict current traffic based on previous one, a simple way is setting an alarm line just like the flood alarm line, such that only if the traffic rises to or beyond the line, the alarm signal will be triggered to activate the corresponding operations. Better traffic speculation methods based on some statistical models could be developed to improve the performance. Also in practice service providers care about the link utilization more than the amount of link traffic passing through. So the first step of our calculation is converting the link traffic to the link utilization. Then some utilization headroom should be reserved for the bursty traffic, thereby, a special link utilization threshold is set up. When the link utilization of the last sublink in previous time slot reaches to or beyond the utilization threshold, the signal of adding sublinks is triggered, and in the next time slot, one or more new sublinks will be activated to adapt to the rising traffic. The number of new added sublinks can be one, or two or more depending on the character of link traffic and how aggressive or conservative the strategy is adopted. If the traffic is always very bursty, the increasing traffic could easily exceed the load capabilities of the active sublinks which we call congestion or overflow. In that case, each “add-sublink” signal should activate more sublinks to allow rising traffic to reduce the possibility of congestion, otherwise, each time adding one sublink might be enough. In the Internet2 case, due to the fact that some links are reserved for education and research purposes, their link traffic loads turn out to be much more bursty and unpredictable than those of the normal ISP links.

The basis of the method is that a router has the best knowledge of its current link traffic and setting appropriate sampling frequency and utilization threshold of link traffic could make the router sensitive and agile enough to respond to the varying traffic. In our method, the utilization threshold is set as the major parameter to decide the number of active sublinks for each bundle link. Also as we discussed earlier, there are two types of bundle link traffic distribution methods deployed in the real network. In the bin packing case, the traffic passing through a bundle link can be carried fully over the first few sublinks; while in the load balancing case, the bundle link traffic are evenly distributed over all sublinks. We deployed two algorithms for the two cases which have the same principle but slightly different strategies. Now we separately discuss the strategies of these two cases.

3.2.1 Strategy for bin packing case

In the bin packing case, we only care about the utilization of the latest-activated sublink and assume the other active sublinks are fully loaded. Only when the utilization of latest-activated sublink goes beyond the threshold, does the router bring up a new sublink to allow for increased traffic and when the link utilization of latest-activated sublink decreases to zero, the sublink is shutdown to save energy. Thus there are two thresholds which are used by a core router. A high threshold (t_{high}) which triggers adding a sublink and a low threshold (t_{low}) which triggers shutting down a sublink. In our study, we always set t_{low} to be zero, so that we do not accidentally lose any traffic. However, we set t_{high} threshold to a high value (such as 90%) to study its impact on the energy savings. In the remainder of the paper, we refer to the t_{high} threshold as simply threshold.

3.2.2 Strategy for load balancing case

In the load balancing case, each active sublink reserves headroom, thus, the risk of congestion is reduced. The algorithm for the load balancing case is similar to that of the bin packing case with two improvements. First, since the number of active sublinks varies within a day, we improve the balance of bursty cushion capability by using float utilization: according to the number of active sublinks in the previous time slot and currently deployed utilization threshold, we slightly regulate the threshold up or down. Also we conduct some statistical investigations which leads to our second improvement. In our Internet2 traffic based experiment, we found that it is common that the congestions occur as a group in a short period. That is, two or more congestions could happen in a short period. We are not sure about the deep reasons for this phenomenon thus far. We speculate that it might come from the burst mode of the edge routers or because of the Internet caching mechanism. The normal traffic waves appear as a controllable gentle shape just like waves on a peaceful lake while some specific events could stimulate the rapid increase of traffic. And for each event, its impact form is not a simple spike but a group of them which occur within a short time period, just like the ripples generated by throwing a stone in the lake. This phenomenon could imply that the bursty traffic in the core network contains some unknown internal patterns. We can utilize this result by deploying a time delay parameter to constrain the operations of shutting down sublinks within the next ten time slots. That is, as long as a congestion occurs, a ten-time-slot delay signal is enabled which will stop the operations of shutting down sublinks within the next ten time slots even though the utilization of the latest-activate sublink decreases to zero. Currently, we only apply the floating threshold and the delay mechanism in the load balancing case. But we argue that they also can be applied in bin packing case, since the size of the headroom space in bin packing also could vary with the time and the number of active links and the delay mechanism still works when a group of traffic spikes appear. In this study, we show that some feasible methods or improvements could help in improving the tradeoff. We only test these methods in some special-case experiments to show their validity and we claim that they can be deployed in other scenarios as well.

3.2.3 Tradeoff

As we can see, the principle of our distributed locally-optimized algorithm is to save energy by rightsizing the bundle link capacities and by increasing their link utilizations at the cost of increased congestion.

Actually, congestions happen sporadically and cannot be 100% guaranteed to be removed in all networks due to the unpredictability and bursty nature of the network traffic. The congestion could bring the problems of data loss and affect the performance of higher layer applications. Some methods and techniques have already been applied to reduce or remove the impact of congestion. One of them is the congestion avoidance mechanism in the TCP protocol. Another common method in ISP networks is utilizing the monitoring function of network management system to periodically gauge the work load and utilization of each link and deploy some rerouting mechanisms to redistribute the excessive traffic through load balancing. With these efforts, the impact of congestion could be greatly reduced or removed if the congestions happen in a low amplitude and last for a short period. Otherwise, the situation could become even disastrous if the congestions occur frequently in large amplitude or last for a long time, especially when that link is a high speed link. Locally-optimized algorithms could increase the possibilities of congestion since the headroom of bundle links are greatly compressed and the bursty traffic can more easily break through the roof.

When congestion happens, the response time of the distributed locally-optimized algorithms depends on two factors: how quickly it detects the congestion and how fast it calculates the solution. The calculation time of distributed locally-optimized algorithm is almost negligible since it uses simplified local-decision-making strategy and does not need to solve the ILP-based NP-complete problems. Actually the maximum congestion detection time in our locally-optimized algorithm almost equals the traffic sampling interval. The sublink scheduling decisions rely on the core routers instead of the management system, thus, it could increase the link traffic sampling frequency to reduce its response time and the local decisions can be made at the same rate as the sampling frequency. In that case, the response time of locally-optimized solution could be greatly reduced. And we note that increasing sampling frequency might greatly increase the workload of routers. For example, in the Internet2 network, the core routers sample link traffic at a frequency of every 10 seconds.

When we are exploring the possible energy savings in the core network, however, we cannot simply pursue as much energy savings as possible and ignore the possible impact from too frequent congestion; also since the network impact from small congestion could be negligible, we might accept them without losing the chance of greater energy savings. The final objective of this study is to look for good tradeoffs in this case. In our algorithms, two parameters mostly contribute to the control of the height of dynamic roof and the compression rate of headroom: the link utilization threshold and the add-sublink strategy, which is, how aggressive is the strategy to add new subinks when it needs more bandwidth. To reach a better tradeoff between the energy savings and the congestion risk, we consider making these two parameters dynamic intuitively. We note that, a lower utilization threshold strategy could reduce the possibilities of congestion at the cost of lower energy savings since now the headroom for bundle link is greater. Also a more aggressive strategy (activating more sublinks every time) works along the same way: reducing the energy-saving benefit to decrease the congestion risk. In the experiments, we would like to separately evaluate their performances in different parameter settings. We could further improve the tradeoff by setting different value combinations of these two parameters to different bundle links according to the burstiness of their traffic.

Algorithm 1 FLHT algorithm

Require: In bundle link set L, for each bundle link L_x, its latest-activated sublink's link utilization at the previous time slot $i-1$ (U_x^{i-1}) and its number of active sublinks at the previous time slot $i-1$ (N_x^{i-1}).

Ensure: The number of active sublinks for bundle link L_x at time slot i (N_x^i)

```
for each time slot i do
    for each bundle link L_x do
        if U_x^{i-1} ≥ 90% then
            N_x^i = N_x^{i-1} + 1.
        end if
        if U_x^{i-1} ≤ 0% then
            N_x^i = N_x^{i-1} - 1.
        end if
return N_x^i
```

For the bin packing case, we develop a distributed locally-optimized algorithm with

Algorithm 2 DLHT algorithm

Require: In bundle link set L, for each bundle link L_x, its latest-activated sublink's link utilization at the previous time slot $i-1$ (U_x^{i-1}), its number of active sublinks at the previous time slot $i-1$ (N_x^{i-1}) and its add-sublink strategy S_x^{add}.

Ensure: The number of active sublinks for bundle link L_x at time slot i (N_x^i)

for each time slot i **do**
  for each bundle link L_x **do**
    if $U_x^{i-1} \geq 90\%$ **then**
      if S_x^{add} is Low Strategy **then**
        $N_x^i = N_x^{i-1} + 1$.
      end if
      if S_x^{add} is Medium Strategy **then**
        $N_x^i = N_x^{i-1} + 2$.
      end if
      if S_x^{add} is High Strategy **then**
        $N_x^i = N_x^{i-1} + 3$.
      end if
    end if
    if $U_x^{i-1} \leq 0\%$ **then**
      $N_x^i = N_x^{i-1} - 1$.
    end if
return N_x^i

fixed threshold and fixed add-sublink strategy which is called FLHT (Fixed Local Heuristic Threshold-based) algorithm as shown in Algorithm 1. Then we propose another algorithm which deploys dynamic parameters which we refer to as DLHT (Dynamic Local Heuristic Threshold-based) algorithm as shown in Algorithm 2. We consider three types of DLHT algorithms with different add-sublink strategies: highly aggressive, moderately aggressive and least aggressive. We separately call them DLHT-High, DLHT-Medium and DLHT-Low algorithms. The differences among them are: DLHT-High algorithm always activates three sublinks when the current link utilization exceeds the threshold; while DLHT-Medium and DLHT-Low activate two and one respectively. The similarity among them is that: they all shut down only one sublink each time the utilization of latest-activated sublink goes down to zero. The reason for shutting down only one sublink when necessary is that we observed that traffic often increases quickly and drops quickly and the strategy which reduces the number of sublinks aggressively will lead to frequent port operations which not only consume considerable energy but also add to time delays, thus, offsets the benefit.

Actually the DLHT-Low algorithm is identical to our previous FLHT algorithm. Another reason to design the DLHT-High and DLHT-Medium algorithms is that in the experiments we found that there are cases of very bursty link traffic. For example, sometimes some bundle link are so bursty that their link traffic increases to more than double or triple of the last time slot. We call that kind of congestion a super congestion and each time activating one sublink is definitely not enough in that case. Super congestion does not happen in every bundle link and can be viewed as a characteristic feature of the bursty links. For example, Internet2 is a US national research and educational network whose core network comprises of 9 sites connected by 26 links and for each pair of connected sites there are two separate links. Observing the long-term traffic of Internet2 core network links, we found that traffic shapes of those links are very different: some of them are always very bursty while the others are much smoother. Fig. 3.1 shows a typical example which presents the traffic shapes of two Internet2 links (No. 5567 and No. 5568) between Chicago and Kansas City. According to the different traffic shapes, we classified all the Internet2 links into two groups: the very bursty link group and the normal bursty link group. What we found is that the super congestion all happened in the very bursty link group and the occurrence of normal congestion in the very bursty link group is much more than that of the normal bursty link group. This is reasonable as burstiness is the major reason for producing congestion. Intuitively, it is more difficult to use rerouting or congestion control mechanisms to reduce or remove the impact of super congestion than doing that in the normal congestion.

Since the possibilities of super congestion are very small, always activating some extra idle links to prepare for the rare chances of congestion would definitely reduce the amount of energy savings. More aggressive algorithms lead to more loss of energy savings. We

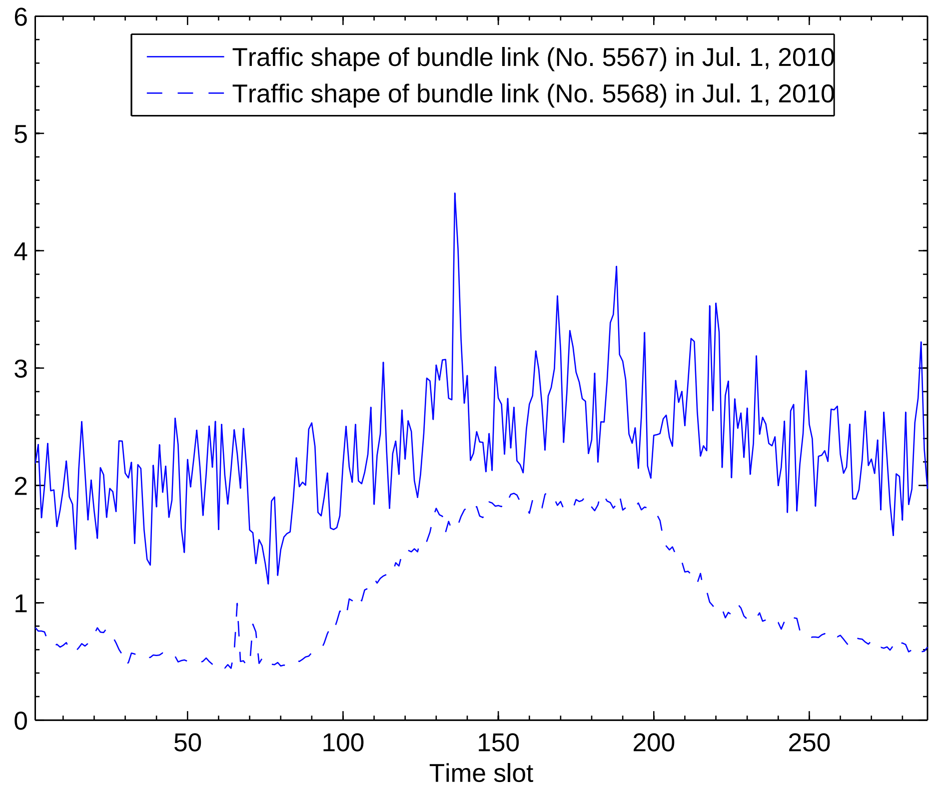


Figure 3.1: Traffic shape of bursty link and smooth link in Internet2

prefer to reduce congestion as much as possible with limited loss of energy savings.

For load balancing case, we use a floating threshold. The major reason is to balance the congestion cushion capability of the bundle link within a day. Two factors affect this capability: the number of current active sublinks and the current deployed threshold. According to the two factors, the threshold in the load balancing case is automatically adjusted (slightly moved up or down) to attempt to keep a consistent headroom space. Another important improvement in load balancing case is the delay remove-sublink operation mechanism. Previously we have discussed the reason for deploying this strategy and we note that different networks might have different traffic characteristics, and hence the setting of delay time on them might be different. In our Internet2-based simulation experiment, setting the delay time as 10 seconds is good enough to remove many possible congestion events with limited loss of energy savings.

Finally, we reach a better tradeoff by assigning different parameter settings to different bursty links. Since ISPs care about the possible network congestions more than energy savings, in this study, the tradeoff is inclined to reduce the network congestion as much as possible with limited loss of energy savings. Also, in this study, we did not consider the possible time delay from the port operations and provide the comparison of number of port operations for different solutions in some experiments.

3.3 Summary

In this chapter, we discussed the problem statement and our proposed distributed locally-optimized algorithms in both bin packing and load balancing cases. Specially, we discussed the different methods and strategies to reach better tradeoffs between energy savings and congestion risk. In the next chapter, we discuss the synthetic network based simulation experiments based on our algorithms and evaluate their performance.

Chapter 4

Experiment and Results

In this chapter we discuss our simulation experiments and analyze the results based on our distributed locally-optimized algorithms.

4.1 Experiment Environment Setup

To check the performance of our proposed algorithms, we construct a synthetic network which simulates the network topology of the Internet2 core network. Also we utilize the Internet2 link traffic data which are collected and stored by RRDtools in the synthetic network to best simulate the real network traffic.

4.1.1 Internet2

Internet2 is an U.S. research and education network which is designed to provide dynamic, innovative and cost-effective hybrid optical and packet network. The core network of Internet2 is composed of 9 sites connected by 26 links and for each pair of connected sites there are two separate links. In comparison to a commercial ISP network, Internet2 releases the information about its network topology and provides its complete link traffic data in the form of RRD (Round Robin Database) file to download.

4.1.2 RRDtool

The RRD tool fetches requisite data in a round robin manner from multiple DS (data sources) and consolidates the data in multiple methods and stores them in RRA (Round Robin Archives) as RRD files. Within one RRD file, several archives might be included: an archive for primary data point (PDP) and the other archives for several consolidated data point (CDP) derived from PDP. We can use command similar as the following to extract the traffic data of PDP of Internet2.

rrdtool fetch samplelink.rrd AVERAGE –start 1212296400 –end 1212296400+86390 > traffic.data

In above example, “samplelink.rrd” is the name of the RRD file for a specific sample link. RRDtool extracts the traffic data within the specified period (with specified start time and end time) in the sample link and stores them in the file of “traffic.data”. Internet2 RRD files store the latest three years’ link traffic data of Internet2 at a minimum interval of 10 seconds, and we consolidated them to the traffic data in interval of 5 minutes for our experiments. Currently, we just want to show the performance of our methods and to simplify the calculation, we select the longer sampling interval. Also, 5 minutes interval for sampling traffic is commonly used by ISPs. Based on that, Internet2 link utilization can be calculated by the standard formula. In fact, in each time slot (5 minutes), a specific bidirectional link will be sampled twice separately in both end sites (site A and site B) and each sampling data includes traffic data in two directions: outgoing and incoming. At a specific time slot, the outgoing traffic data of site A might not equal to the incoming traffic data of site B, is because both sites’ sampling time might not be exactly the same. Our rule is to use the maximum value of the four sample data as the link traffic data at that time slot. Since we are trying to reduce the possible congestion, the maximum value seems to be more reasonable than the average value in terms of acting as the link traffic data.

4.1.3 Synthetic network

Our synthetic network utilizes almost the same network topology as Internet2 except that we assume the existence of a bundle link in each link. Internet2 does not apply bundle link technique since a 10GE link is enough to carry its traffic; while in commercial ISP network bundle links are commonly applied. To simulate a large ISP network, in our synthetic network, we replace each link of Internet2 by a bundle link composed of ten 1GE optical sublinks, i.e., the same bandwidth but 10 times as many links as before. And we assume each sublink carries one-tenth of the link traffic of the original Internet2 link. In this way, the total traffic of each bundle link is the same as that of the original Internet2 link. The

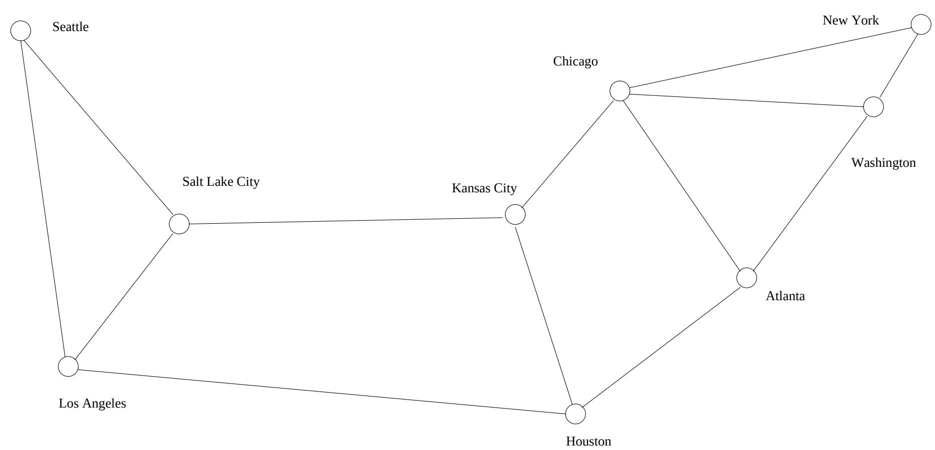


Figure 4.1: Topology of Internet2

synthetic network topology is shown in Fig. 4.1 which includes 9 sites and 26 bundle links (totally 260 pairs of 1GE bidirectional fibers). In original Internet2 topology, between each two directly-connected nodes, there are two separate links to serve different purposes and their link traffic shapes generally have big differences. We keep the same configuration in

Table 4.1: Bundle links in synthetic network

Site1	Site2	Bundle Link ID	Number of fibers
CHIC	ATLA	COMPOSITE_LINKCL5419	10
CHIC	ATLA	COMPOSITE_LINKCL5638	10
HOUS	ATLA	COMPOSITE_LINKCL5423	10
HOUS	ATLA	COMPOSITE_LINKCL5562	10
WASH	ATLA	COMPOSITE_LINKCL5133	10
WASH	ATLA	COMPOSITE_LINKCL5251	10
KANS	CHIC	COMPOSITE_LINKCL5567	10
KANS	CHIC	COMPOSITE_LINKCL5568	10
NEWY	CHIC	COMPOSITE_LINKCL5239	10
NEWY	CHIC	COMPOSITE_LINKCL5667	10
WASH	CHIC	COMPOSITE_LINKCL5250	10
WASH	CHIC	COMPOSITE_LINKCL5637	10
KANS	HOUS	COMPOSITE_LINKCL5560	10
KANS	HOUS	COMPOSITE_LINKCL5561	10
LOSA	HOUS	COMPOSITE_LINKCL5559	10
LOSA	HOUS	COMPOSITE_LINKCL5581	10
SALT	KANS	COMPOSITE_LINKCL5138	10
SALT	KANS	COMPOSITE_LINKCL5566	10
SALT	LOSA	COMPOSITE_LINKCL5563	10
SALT	LOSA	COMPOSITE_LINKCL5571	10
SEAT	LOSA	COMPOSITE_LINKCL5564	10
SEAT	LOSA	COMPOSITE_LINKCL5572	10
WASH	NEWY	COMPOSITE_LINKCL4643	10
WASH	NEWY	COMPOSITE_LINKCL5242	10
SEAT	SALT	COMPOSITE_LINKCL5565	10
SEAT	SALT	COMPOSITE_LINKCL5573	10

our synthetic network, thus, actually there are 26 bundle links in the topology.

Table 4.1 shows the information of all bundle links in our synthetic network.

Thirty days (from July 1, 2010 to July 30, 2010) worth of link utilization data are

Table 4.2: Typical link traffic information extracted from Internet2 RRD file

Timestamp	Input	Output
1212296410	6.0812216354e-07	5.1619134628e-07
1212296420	5.0450804956e-07	4.5207642781e-07
1212296430	5.8053464978e-07	5.6639656430e-07
1212296440	5.6483092482e-07	5.5529660413e-07
1212296450	4.9683281349e-07	4.6534343965e-07
1212296460	5.2271588971e-07	4.9615828455e-07

extracted from Internet2 for experimental analysis and a typical link utilization data set is shown in Table 4.2.

4.2 Pre-experiment Observations

We start from the following observations: (1) A bundle link usually consists of several physical (SONET) links, which may have very low utilization. For example, bundle link 5423 from Atlanta to Houston consists of 10 physical links and their utilization is commonly less than 1%. (2) The energy consumed at the router ports due to a physical (SONET) link is related to its state (ON/OFF) and not related to its utilization. The method of rate adaptation does not contribute much energy saving here; ports must be shut down [15].

To avoid the problem of energy consumption differing in products from different vendors, we use port-hour or port-5minutes as energy-saving unit instead of watts for evaluation. Although we only investigate shutting off the pair of router ports for a sublink, in an actual network, the energy consumption due to the other devices such as transponders and amplifiers, can also be reduced. Specifically, the transponders consume a lot of energy and cost in optical networks. Furthermore, if all the ports in a line card are shut down, the whole line card could be shut down and the energy savings of line card will be greater than that of shutting down its attached ports. To fully uncover the energy savings in these scenarios might introduce complicated and diverse cases. To simply the calculation, we just count the number of ports shut down in core routers and prefer using percentage of shut down ports vs total ports to evaluate the energy saving performance of our method.

4.3 Simulation Experiment Procedure and Results

4.3.1 Simulation experiments in bin packing case

As discussed before, our proposed algorithm in bin packing case normally would generate more energy savings. In our experiment, we first consider the energy savings in bin packing case and we would like to know the maximum value of potential energy saving in our synthetic network within 30 days, that is, the upper bound of our method. In bin packing case, assigning an appropriate threshold value for each bundle link is tricky. First the threshold cannot be too high (very close to 100%), since bringing up a sublink takes time (on the order of milliseconds) and the new sublink may not be ready to carry traffic and a large quantity of data might be lost. Secondly it is inefficient if the threshold is set too low (such as 40%). A tradeoff is made between the two extreme scenarios to keep the system sensitive and efficient. In our experiment, a preliminary threshold was set as 90% for each bundle link to check the energy saving and risk since 90% is close to the link capacity while still leaving some headroom to adapt to the bursty traffic. Also we conduct the same experiment but set the threshold as 80% for comparison. An optimal threshold value could be calculated from the statistical analysis of long-term historical utilization data.

In addition to setting the threshold to reasonable values (such as 90%), we set the threshold to 100% and assuming that the link traffic demand of each time slot is known in advance in order to calculate the theoretical optimum (least) number of sublinks needed for each bundle link to carry all of its traffic in each time slot. Note that as pointed out earlier, this optimum cannot be reached in practical settings due to high traffic variation and risk of data loss. The upper bound is shown in the figures (Fig. 4.2, Fig. 4.3, Fig. 4.4) as dotted lines.

Then, an experiment using our FLHT algorithm was conducted to check the energy saving performance. In the experiment, the threshold is set as 90% for each bundle link. We would like to know whether setting the threshold as 90% is safe enough or not to handle the varying traffic in this scenario, that is, we wish to verify whether one more active sublink is enough or not to carry the new traffic. Imagine that the traffic increases too much in the next time slot, thus, one new sublink is not enough to carry the new traffic. Adding two or more active sublinks each time we add new sublinks can help reduce the possibilities of traffic congestion at the cost of decreased energy savings.

Fig. 4.2 shows the energy saving of the last time slot (23:55:00) for 30 days. The energy saving in one day is shown in Fig. 4.3. The energy saving in 30 days is shown in Fig. 4.4. During almost the whole month of July 2010, all bundle links in the synthetic network can save around 86% of total ports energy which is more than 160,000 port*hours in total.

As a result of applying FLHT algorithm and setting 90% as the threshold and deploying low add-sublink strategy, traffic congestion occurs in less than 6111 of the 224,640 time slots in the 30 days period across all the bundle links in the network (or 2.7% of the time slots); while changing the setting to the medium aggressive add-sublink strategy, the time slots of traffic congestion reduced to 0.18%. Even in the latter setting, the energy savings are still considerable. Fig. 4.2. shows that in a single time slot (the last time slot of a day) of 30 days energy of 188 port*5minutes could be saved. Fig. 4.3. shows that on a particular day energy of 63522 port*5minutes could be saved. Fig. 4.4. shows that on a particular 30 day period energy of 1,926,650 port*5minutes could be saved. We can see in all three example cases most ports can be shut down and significant energy can be saved. Results from additional experiments (with 80% as threshold) provide similar but decreased savings in energy.

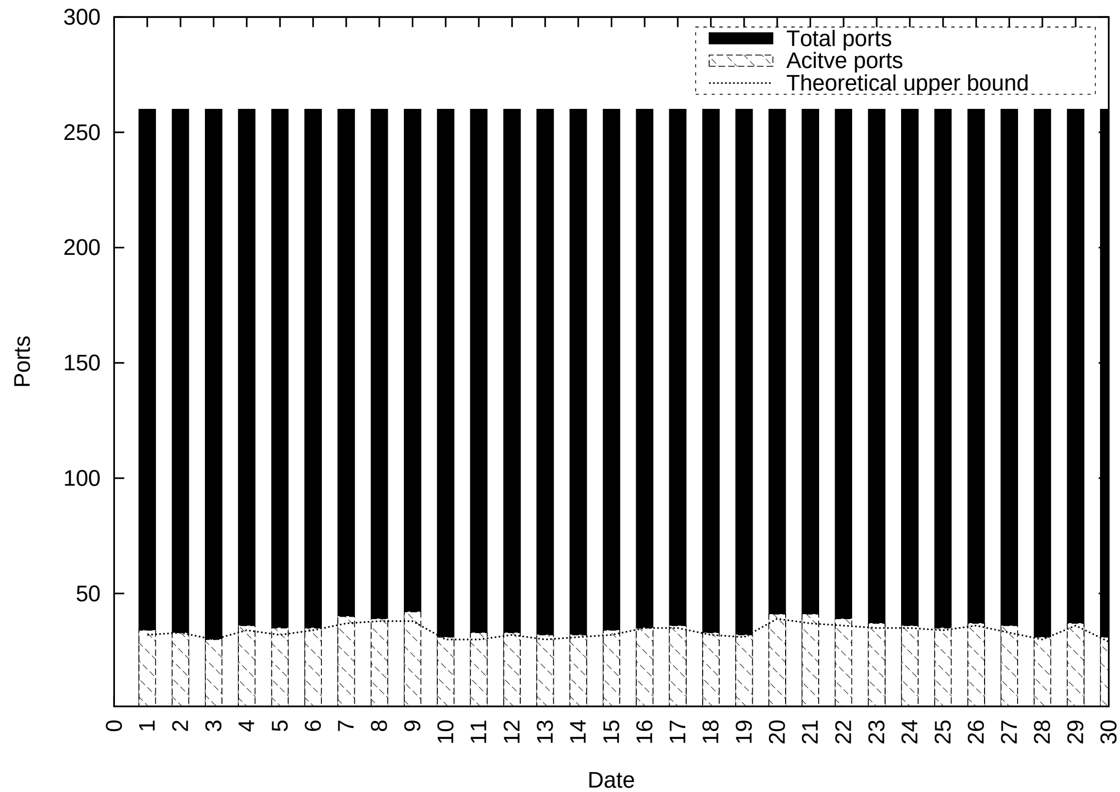


Figure 4.2: Energy savings of one time slot setting 90% as the threshold

4.3.2 Simulation experiments for different parameter settings

In previous experiments, we utilized the fixed threshold and the fixed add-sublink strategy to evaluate the energy-saving performance of our FLHT algorithm in bin packing case. We found that most of the port energy can be saved with small cases of congestion. Now we attempt to deploy different parameter settings to reduce the cases of congestion to reach a better tradeoff. As we discussed in chapter 3, DLHT means the FLHT algorithm with dynamic parameter settings. The DLHT-Low, DLHT-Medium, DLHT-High algorithms respectively mean deploying the add-sublink strategy of adding one or two or three sublinks each time it needs more bandwidth. In this experiment, we select three link utilization thresholds (90%, 80%, 60%) to be applied in our DLHT algorithm of bin packing case and we compare their performances. First, we deployed DLHT-Low algorithm on our synthetic

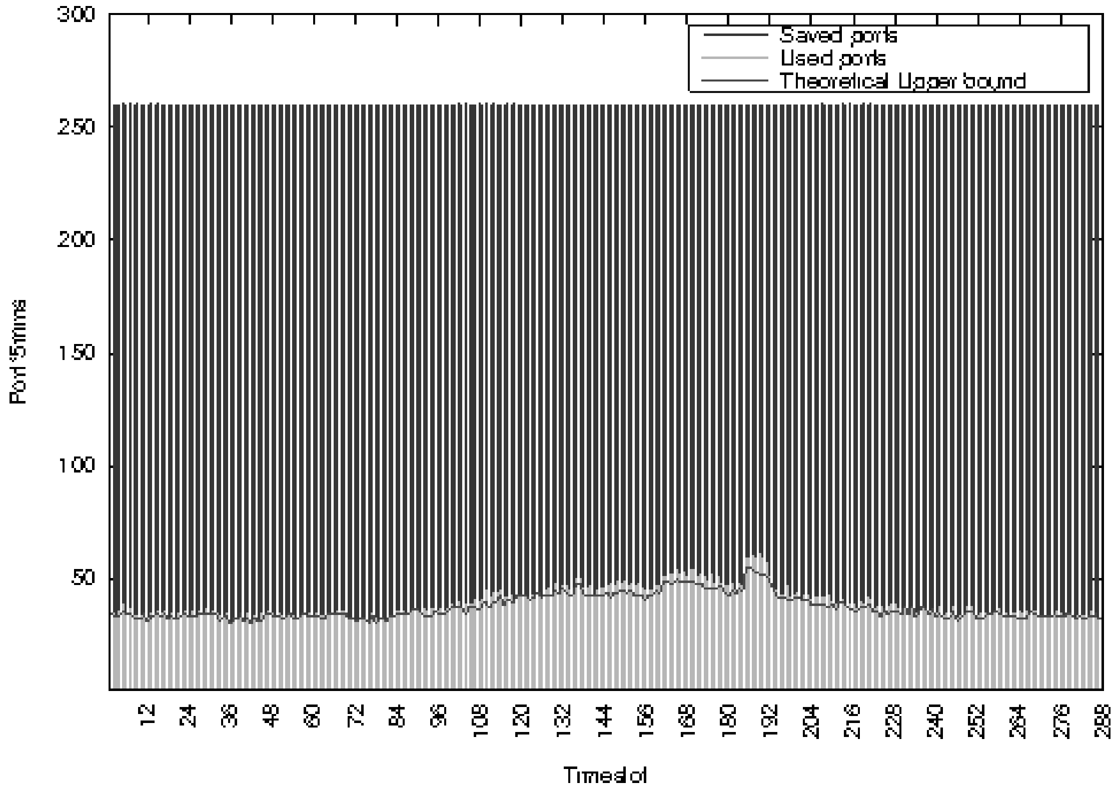


Figure 4.3: Energy savings in one day setting 90% as the threshold

network and check how different thresholds affect the energy-savings. The result is shown in Fig. 4.5. As we can see, changing the threshold from 90% to 60% only decreases (around 2.3%) the energy savings slightly. Now we are interested in how the occurrences of congestion and super congestion are affected in above scenarios. In this experiment, we care about two link groups' performance. The result is shown in Fig. 4.6. To make the graph clear, we only show the result of setting 60% and 90% as the utilization thresholds. The x-axis is an order of bundle links in which the first 13 links are the bundle links from the very bursty group while the latter 13 links are the bundle links from the normal bursty link group. First, we noticed that the very bursty link group incurs many more cases of congestion and super congestion than the normal bursty link group. Actually the latter one has close to zero cases of congestion and super congestion; so there is almost no difference

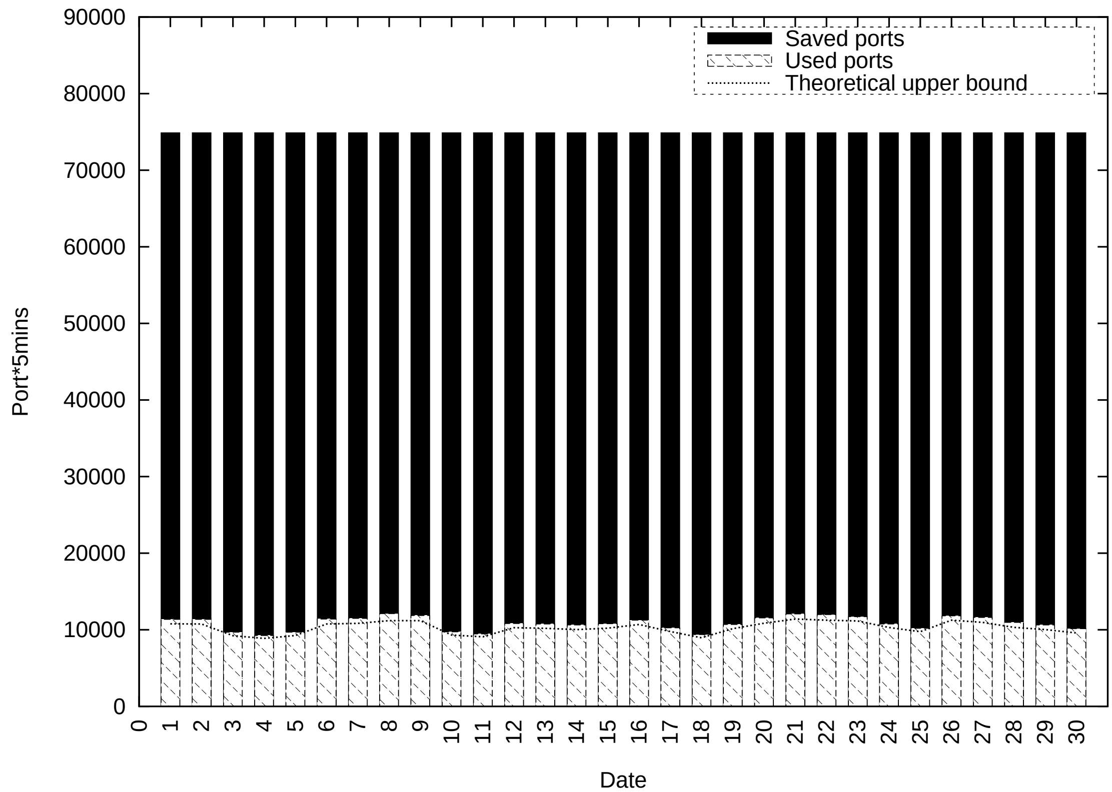


Figure 4.4: Energy savings over 30 days setting 90% as the threshold

for the normal bursty link group by setting different utilization thresholds. While in the very bursty link group, whether setting 60% or 90% as thresholds, the results are very different. By setting the lower threshold, the number of congestion events is greatly decreased. The number of super congestion almost remains the same in different utilization settings. This is reasonable, since for the latest-activated sublink, changing utilization from 60% to 90% will not change much the situation when the link traffic more than doubles. It seems that the utilization setting is very effective in reducing the number of normal congestion in bursty links.

Next we would like to evaluate the performance of different add-sublink strategies in terms of reducing the number of super congestion events. Fig. 4.7 shows for a threshold of 60%, the number of super congestion for DLHT-High, DLHT-Midium and DLHT-Low

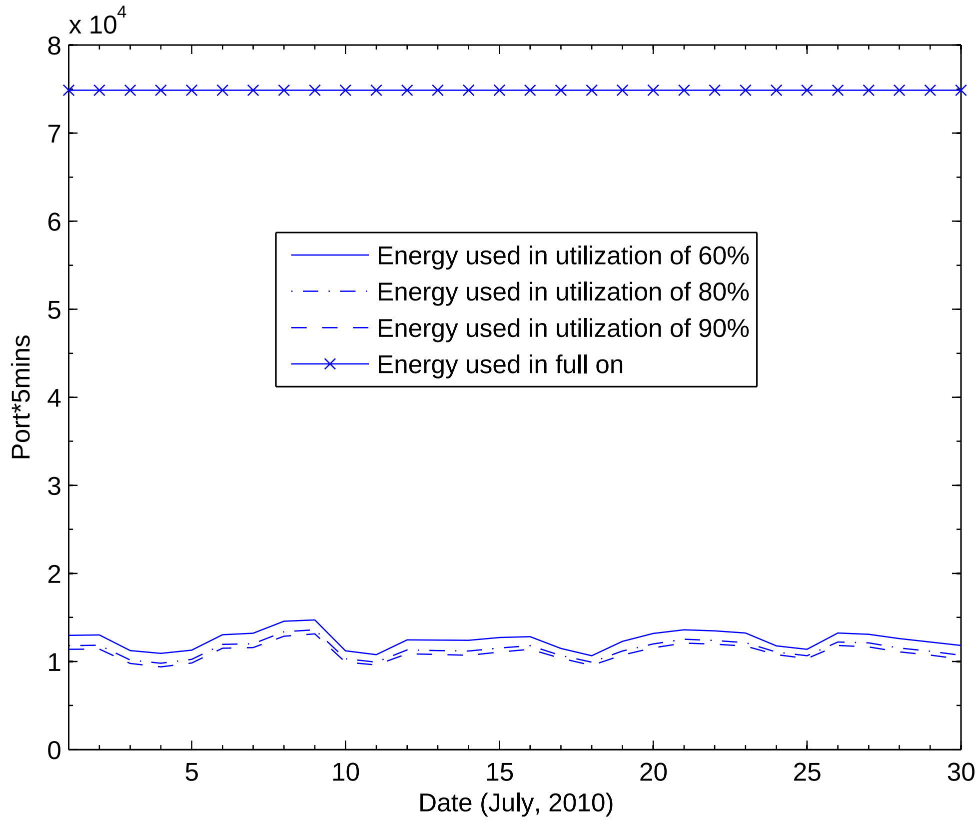


Figure 4.5: Comparison of energy savings deploying DLHT-Low algorithm

across 30 days. From Fig. 4.7, we can see that more aggressive strategy of DLHT algorithm could greatly reduce the cases of super congestion for the bursty link group.

From above analysis, we know that for normal bursty link group deploying the low aggressive DLHT algorithm and setting 90% as the utilization threshold is good enough; while for the very bursty link group, it had better use more aggressive DLHT algorithms to reduce the cases of super congestion and setting lower utilization threshold to decrease the cases of normal congestion. It seems that the corresponding loss in energy saving is almost negligible, but in fact we are missing one consideration which is the energy consumption for shutting down and bringing up sublinks.

Table 4.3 shows the sublink operations in several parameter combinations. It shows that

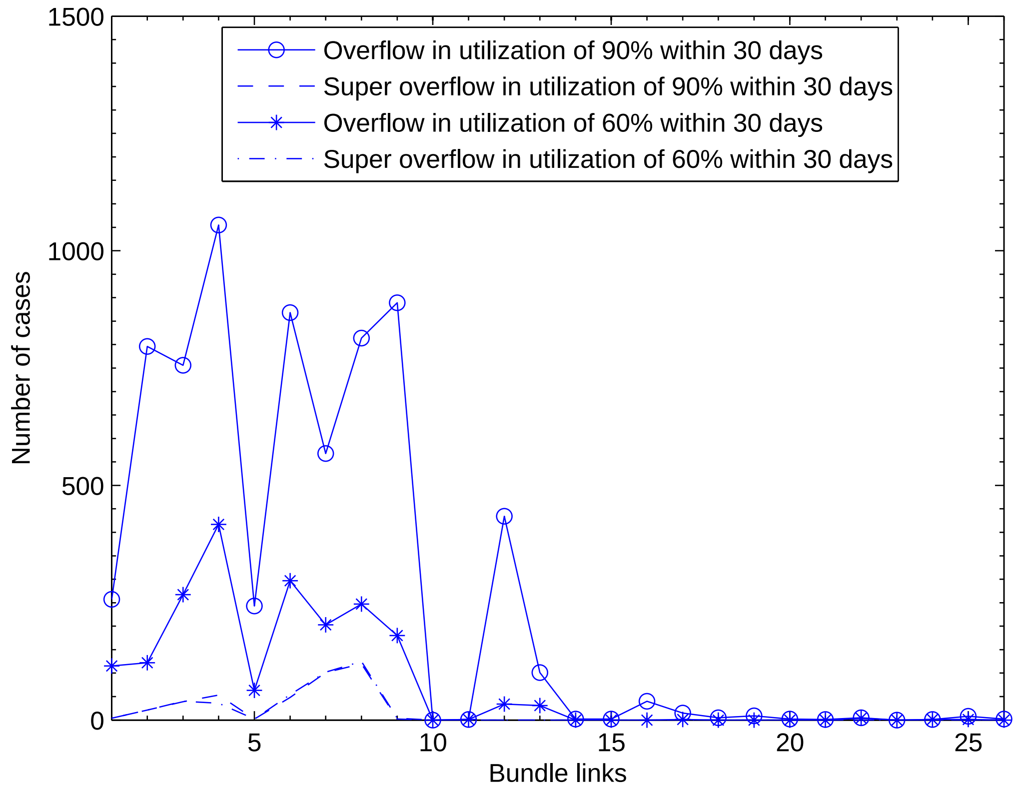


Figure 4.6: Comparison of congestion by 60% and 90%

Table 4.3: Comparison of number of sublink operations in 9 parameter combinations

Strategy//Utilization	60%	80%	90%
DLHT-HighAggressive	126255	98859	76264
DLHT-MediumAggressive	73202	58518	47380
DLHT-LowAggressive	30267	26600	24303

a more aggressive DLHT algorithm and lower utilization threshold would result in more sublink operations, thus, consuming more energy. The increased energy consumption in sublink operations might offset the benefit. However, this is an area of of future work. By above experimental result analysis, we can set different parameter combinations to different bundle links according to how bursty they are. We incorporate such hybrid parameters

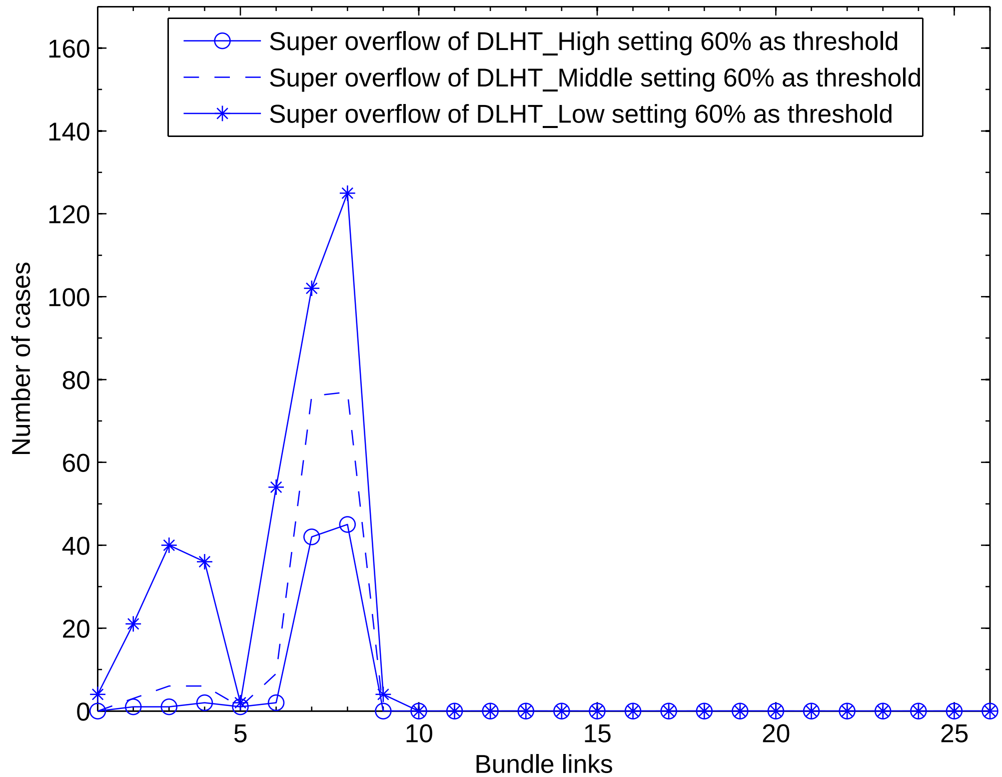


Figure 4.7: Comparison of super congestion of three DLHT algorithms

setting into our DLHT algorithms and refer to them as HDLHT (Hybrid DLHT) algorithms. Deploying HDLHT could reduce the cases of congestion and super congestion and keep limited loss of energy savings. To simply the calculation, we use a simple way to evaluate our idea: just use the parameter combination of DLHT-High algorithm and setting threshold as 60% for the very bursty bundle link group and DLHT-Low algorithm and setting threshold as 90% for the normal bursty bundle link group. Then we compare the results of the HDLHT with results from previous work to measure the improvement. The comparison results are shown in Table 4.4. Table 4.4 shows that with the HDLHT algorithm, the number of congestion and super congestion events are greatly reduced with very limited (3.2%) decrease of energy savings. Although there are still a lot of cases of congestion and

Table 4.4: Performance comparison between Hybrid DLHT and LHT algorithms

Algorithm	congestion	super congestion	energy savings
Hybrid DLHT	1701 (0.76%)	94 (0.04%)	1855168 (82.58%)
LHT	6874 (3.1%)	390 (0.17%)	1916646 (85.32%)

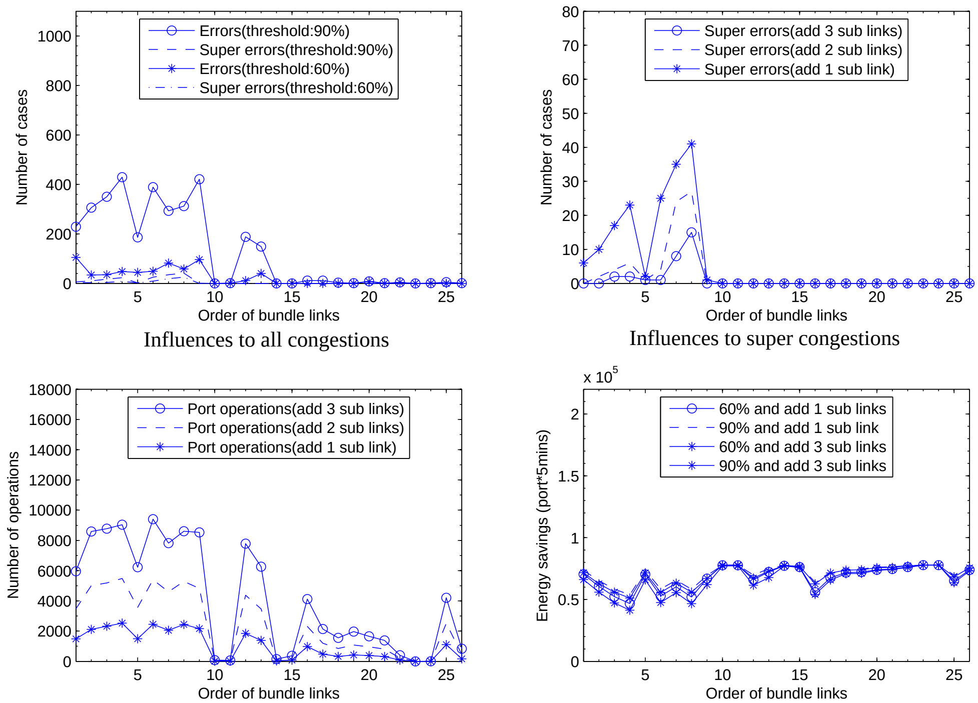
super congestion, even deploying HDLHT algorithm, actually most of them come from the very bursty link group which is much more bursty than the normal commercial ISP link. For example, in our synthetic network case, the very bursty link group produces 1609 congestion and 94 super congestion while the normal bursty link group only produces 92 congestion and 0 super congestion. So this algorithm should have much better performance when deployed in pure commercial networks.

4.3.3 Simulation experiments in load balancing case

As we discussed before, in load balancing case, each sublink in synthetic network has the same link utilization, thus, more headroom is reserved. In this experiment, specifically, we increase the link traffic by 50% to reserve the bandwidth capacity for the backup paths which is important for survivable optical networks and consider the annual increase of network traffic. Similar to the bin packing case, by a standard formula, the traffic data are transferred to the corresponding link utilization then DLHT algorithm is deployed in this scenario and we can test the performance of it in several different parameter settings. As we discussed in Chapter 3, two strategies are added in load balancing case to improve the DLHT algorithm which are floating threshold and delay remove-sublink operation mechanism. In our experiment, we enable the threshold to be floating by slightly adjusting the threshold value based on the value of the base threshold (we refer to the original planned threshold as base threshold) and the current number of active sublinks. When the current number of active sublinks is more than 4 and the base threshold is more than 80%, then we float the base threshold up by 5% and limit the adjusted floating threshold to be no more than 95%. For example, currently there are five active sublinks and the base threshold is 85%, then the threshold will be automatically adjusted to 90%. When the current number of active sublinks is more than 4 and the base threshold is no more than 80%, then we float the base threshold up by 10% and still limit the adjusted floating threshold to be no more than 95%. When the current number of active sublinks is no more than 2 and the base threshold is more than 80%, then we float the base threshold down by 10%. When the current number of active sublinks is no more than 2 and the base threshold is no more than 80%, then we float the base threshold down by 5%. Another strategy we deployed in the experiment is the delay remove-sublink operation strategy, which we already discussed in Chapter 3. It means when a congestion or super congestion occurs, the router will forbid the operation of removing sublinks in the latter 10 time slots. Both of the two improvements aim at reducing the congestion with a limited cost of energy savings.

Fig. 4.8 shows the comparisons of different parameter settings in load balancing case and it indicates some interesting information. First, the lower threshold greatly reduces the congestion but has limited influence in reducing the super congestion. Second, the more aggressive strategy greatly reduces the super congestion cases but at the cost of increased port operations which consume a lot of energy. Third, deploying different parameter settings results in very little difference in energy savings if we ignore the energy consumed by port operations.

Table 4.5 shows the performance comparisons of three solutions when setting 80% as the basic threshold and using the least aggressive strategy. As we can see, the first solution performs best in terms of energy savings. The other two lose very limited energy savings while greatly reducing the occurrences of congestion or super congestion. The third solution achieves a much better tradeoff than the previous two. The results prove that introducing the floating threshold and the delay remove-sublink mechanism in load



Influences to all congestions

Influences to super congestions

Influences to port operations

Influences to literal energy savings

Figure 4.8: Comparisons of different parameter settings in load balancing case

Table 4.5: Performance comparisons of three solutions

Solutions	Energy savings	congestion	Super congestion	Port operations
Fixed threshold in bin packing	1904241 (84.77%)	4540 (2.02%)	396 (0.18%)	26600
Fixed threshold in load balancing	1855331 (82.59%)	3840 (1.71%)	224 (0.1%)	35190
Floating threshold with delay in load balancing	1824134 (81.20%)	1960 (0.87%)	132 (0.06%)	36529

balancing case is useful to achieve better tradeoff.

The simulation experiments in load balancing case show that with different parameter settings, the improved algorithm works well and achieves better tradeoff than the previous algorithms. We also note that in reality the load balancing technique cannot evenly distribute the traffic among sublinks which might impair the efficiency of our algorithm.

4.4 Summary

In this chapter, we discussed the simulation experiments deploying our proposed distributed locally optimized approaches based on a synthetic network. We described the network model and the simulation environment considered for running the experiments and calculated the energy saving upper bound. We tested FLHT algorithm in bin packing case. Then we separately evaluated the effectiveness of different settings of two major parameters by deploying DLHT algorithms and different thresholds in the synthetic network. Finally we assigned different parameter settings to different bursty link groups. In load balancing case, we conducted experiments with the improvement of deploying the floating threshold and the delay remove-sublink mechanism and the experiments showed similar results and proved the validity of the two improvements. The simulation experiment results showed that most of router port energy consumption could be saved and with appropriate setting, the occurrences of congestion can be greatly reduced with very limited loss of energy savings. The conclusions and future work are discussed in the next chapter.

Chapter 5

Conclusions and Future Work

In this chapter we present the conclusions and future work of our project.

5.1 Conclusions

In this thesis, we proposed distributed locally optimized algorithms to achieve energy savings in core networks deploying the bundle link technique. For the two cases of traffic distribution in bundle link (bin packing and load balancing), we separately proposed slightly different strategies to trigger port operations of shutting down or bringing up sublinks. To reach a better tradeoff between energy savings and congestion risk, we evaluate the performance of different parameter settings in both these cases. Finally, we apply different parameter settings to different bundle links according to how bursty they are to greatly reduce cases of congestion. A group of simulation experiments were conducted on an Internet2 based synthetic network.

The simulation experiments deploying fixed threshold (90%) and least aggressive add-sublink strategy in bin packing case show that substantial energy savings could be achieved with a limited amount of congestion. Later we consider different value settings of two major parameters and appropriate assignments to different bursty links to achieve a better tradeoff. Then for experiments in the load balancing case, the algorithm deploying a floating threshold and delay mechanism greatly reduced cases of congestion with limited energy savings loss. All of the above experiments show that although traffic is unpredictable, with appropriate settings, our distributed methods work well and achieve energy savings close to optimal without substantial data loss. Ignoring the cost of port operations, the energy savings achievable is very promising, i.e., 86% of the energy can be saved in bin packing case. Due to local decision making, our method responds quickly and is more effective than global optimized methods and can be deployed in a distributed manner without scalability problems. We note that the theoretical maximal energy saving in global-optimized methods is higher than that of our local-optimized methods. The detailed performance comparison of these two types of algorithms is left as future work.

5.2 Future Work

Several tasks are left as future work such as deploying the distributed local-optimized algorithm online in pure commercial networks to show more realistic performance and deploying Round-Robin algorithm on router port allocation. Also we can consider the offset effect of the increased energy savings from rising port operations. A protocol to synchronize the steps for two routers' port operations could be developed. The alarm mechanism in network management needs to be modified to deal with the issues related to port operations of rightsizing bundle links since they could generate unnecessary alarms for network management system and the subsequent response or operations such as rerouting could be triggered if the management system is not aware of the "fakeness" of the alarms. Also dynamically adjusting the capacity of the bundle links modifies the capability set of the router and this information should be synchronized with the network management system. We did not consider the time effect in this work. Across a day or during weekdays or weekends, the burstiness of links could be different. So we could introduce a relationship between the burstiness and the time element into the algorithm design such as the approaches in [**?**]. Currently, our algorithms only use the traffic data information in the previous time slot. We could utilize the traffic information from several previous time slots and assign different weights to them to better exploit the historical traffic information to save energy. Also, shutting down some sublinks could impact the network survivability. We did not deeply investigate this problem now. We plan to investigate the influence of our methods on the network survivability and on the traffic over the backup path. We would like to have a performance comparison for global-optimized methods and local-optimized methods. Finally, we could investigate the dynamic threshold which is dynamically adjusting the threshold according to the burstiness of the bundle link in different time periods. For example, we can set the threshold higher when the bundle link traffic is smooth and set it lower when the bundle link traffic is bursty.

Bibliography

[1] Wikipedia, "Energy consumption." http://en.wikipedia.org/wiki/Energy_consumption. 1.1

[2] E. I. A. (US) and U. E. I. Administration, "International energy outlook 2010," 2010. 1.1

[3] T. Restorick, "An inefficient truth," *Global Action Plan Report*, 2007. 1.1

[4] R. Bolla, R. Bruschi, F. Davoli, and A. Ranieri, "Energy-aware performance optimization for next-generation green network equipment," in *Proceedings of the 2nd ACM SIGCOMM workshop on Programmable routers for extensible services of tomorrow*, pp. 49–54, ACM, 2009. 1.1, 1.1

[5] W. Vereecken, L. Deboosere, D. Colle, B. Vermeulen, M. Pickavet, B. Dhoedt, and P. Demeester, "Energy efficiency in telecommunication networks," p. 44, 2008. 1.1

[6] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks: reducing energy consumption by shutting off cables in bundled links," in *Proceedings of the first ACM SIGCOMM workshop on Green networking*, pp. 29–34, ACM, 2010. 1.1, 1.3, 2.2, 2.3, 3.1

[7] S. Nedevschi, J. Chandrashekar, J. Liu, B. Nordman, S. Ratnasamy, and N. Taft, "Skilled in the art of being idle: reducing energy waste in networked systems," in

Proceedings of the 6th USENIX symposium on Networked systems design and implementation (NSDI), pp. 381–394, 2009. 1.1, 1.4

[8] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *INFOCOM*, IEEE, 2008. 1.1, 2.2

[9] R. Morris and D. Lin, "Variance of aggregated web traffic," in *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.*, vol. 1, pp. 360–366, INFOCOM, IEEE, 2000. 1.1

[10] Wikipedia, "core network." http://en.wikipedia.org/wiki/core_network. 1.2

[11] W. Vereecken, W. Van Heddeghem, B. Puype, D. Colle, M. Pickavet, and P. Demeester, "Optical networks: how much power do they consume and how can we optimize this," in *European Conference (ECOC) and Exhibition on*, pp. 1–4, 2010. 1.2, 1.4, 1.4

[12] Wikipedia, "Internet backbone." http://en.wikipedia.org/wiki/Internet_backbone. 1.2, 1.3

[13] Wikipedia, "Link aggregation." http://en.wikipedia.org/wiki/Link_aggregation. 1.3

[14] M. Andrews, A. Anta, L. Zhang, and W. Zhao, "Routing and scheduling for energy and delay mimization in the powerdown model," in *INFOCOM*, IEEE, 2010. 1.5

[15] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*, pp. 323–336, 2008. 1.5, 2.2, 4.2

[16] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy efficiency in the future internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures," *IEEE Communications Surveys & Tutorials*, no. 99, pp. 1–22. 2, 2.1

[17] J. Baliga, R. Ayre, K. Hinton, and R. Tucker, "Photonic switching and the energy bottleneck," in *IEEE Photonics in Switching*, pp. 125–126, 2007. 2.1

[18] M. Jimeno, K. Christensen, and B. Nordman, "A network connection proxy to enable hosts to sleep and save energy," in *Performance, Computing and Communications Conference, 2008. IPCCC 2008*, pp. 101–110, 2008. 2.1

[19] M. Gupta and S. Singh, "Greening of the internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003. 2.2

[20] M. Gupta, S. Grover, and S. Singh, "A feasibility study for power management in lan switches," pp. 361–371, 2004. 2.2

[21] M. Gupta and S. Singh, "Using low-power modes for energy conservation in ethernet lans," pp. 2451–2455, INFOCOM, IEEE, 2007. 2.2

[22] G. Ananthanarayanan and R. Katz, "Greening the switch," in *Proceedings of the 2008 conference on Power aware computing and systems*, pp. 7–7, 2008. 2.2

[23] M. Yamada, T. Yazaki, N. Matsuyama, and T. Hayashi, "Power efficient approach and performance control for routers," in *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, pp. 1–5. 2.2

[24] IEEE, "Ieee p802.3az energy efficient ethernet task force." http://www.ieee802.org/3/az/index.html. 2.2

[25] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *Communications, 2009. ICC'09. IEEE International Conference on*, pp. 1–6, 2009. 2.2

[26] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware backbone networks: a case study," in *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, pp. 1–5, 2009. 2.2

[27] J. Restrepo, C. Gruber, and C. Machuca, "Energy profile aware routing," in *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, pp. 1–5, IEEE, 2009. 2.2

[28] M. Andrews, A. Anta, L. Zhang, and W. Zhao, "Routing and scheduling for energy and delay minimization in the powerdown model," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–5, IEEE, 2009. 2.2

[29] D. Seo, A. Featherstone, D. Weisman, and Y. Gao, "Market consolidation and productivity growth in us wireline telecommunications: Stochastic frontier analysis vs. malmquist index," *Review of Industrial Organization*, vol. 36, no. 3, pp. 271–294, 2010.