Design and management of networks with low power consumption

Truong Khoa Phan

To cite this version:

Truong Khoa Phan. Design and management of networks with low power consumption. Other [cs.OH]. Université Nice Sophia Antipolis, 2014. English. ¡ NNT : 2014NICE4065 ¿.

HAL Id: tel-01127256
https://tel.archives-ouvertes.fr/tel-01127256

Submitted on 7 Mar 2015

UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS
ÉCOLE DOCTORALE DES SCIENCES ET TECHNOLOGIES DE
L'INFORMATION ET DE LA COMMUNICATION

PHD THESIS

to obtain the title of

Docteur en Sciences

de l'Université de Nice - Sophia Antipolis
Mention : Informatique

Defended by
Truong Khoa PHAN

Design and Management of Networks with Low Power Consumption

COATI Project
(INRIA, I3S (CNRS/UNS))
Advisors:
Joanna Moulierac
David Coudert

Defended on 25th September, 2014

Jury:

Reviewers:	Edoardo AMALDI	-	Politecnico di Milano (Milano, Italy)
	Laurent LEFÈVRE	-	INRIA (Lyon, France)
Examinators:	Bernadetta ADDIS	-	Ècole des Mines de Nancy
	David COUDERT	-	I3S and INRIA Sophia Antipolis(France)
	Arie M.C.A. KOSTER	-	RWTH Aachen (Germany)
	Joanna MOULIERAC	-	I3S and INRIA Sophia Antipolis(France)
	Guillaume Urvoy-Keller	-	I3S (Sophia Antipolis, France)

Acknowledgements

First of all, I would like to take the opportunity to thank my advisors Prof. Joanna Moulierac and Dr. David Coudert for helping me to improve my work through their advices and ideas.

I would like to thank Prof. Edoardo Amaldi and Dr. Laurent Lefèvre for reviewing this manuscript. This work would not have been done without the help of several coauthors. I also thank them very much.

I have special thanks to Patricia Lachaume who I consider as a mother. She took care all my administrative problems during my three years of PhD.

I would like to express my gratitude to my colleagues from COATI team. I also thanks my Vietnamese friends at Sophia, Antibes and Nice. With them, I have a feeling like to live in a family.

Last but not the least, I dedicate this thesis to parents, my sister and my brother. They always give support and encouragement during my years of study.

Design and Management of Networks with Low Power Consumption

Abstract: In this thesis, we study several models of energy-aware routing. For each model, we present a linear programming formulation to find the exact solution. Moreover, since energy-aware routing is NP-hard problem, we also propose efficient heuristic algorithms for large scale networks.

In the first part of this thesis, we deal with GreenRE - a new energy-aware routing model with the support of redundancy elimination. We first present a deterministic model in which we show how to combine energy-aware routing and redundancy elimination to improve energy efficiency for backbone networks. Then, we extend the model in order to take into account uncertainties in traffic volumes and redundancy rates.

The second part of this thesis is devoted to the deployment issues of energy-aware routing in practice. In detail, to avoid service deterioration for end-users, we limit changes of network configurations in multi-period traffic matrices in Open Shortest Path First (OSPF) protocol. Next, we address the problem of limited rule space in OpenFlow switches when installing energy-aware routing configurations.

Finally, we present in the appendix other works developed during this thesis: multicast network protocol and TCP congestion control algorithm.

Keywords: Energy-aware Routing, Redundancy Elimination, Open Shortest Path First, Software Defined Networks.

Conception et Gestion de Réseaux Efficaces en Énergie

Résumé :

Dans cette thèse, nous étudions plusieurs modèles de routage efficaces en énergie. Pour chaque modèle, nous présentons une formulation en programmation linéaire mixte permettant de trouver une solution exacte. En outre, comme il s'agit de problèmes NP-difficiles, nous proposons des heuristiques efficaces pour des réseaux de grande taille.

Dans la première partie de cette thèse, nous étudions une solution de routage efficace en énergie dans laquelle nous ajoutons la possibilité d'éliminer des redondances dans les paquets transmis sur le réseau. Nous montrons premièrement que l'ajout de l'élimination des redondances permet d'améliorer l'efficacité énergétique des réseaux en éteignant plus de liens. Ensuite, nous étendons le modèle afin qu'il prenne en compte un certain niveau d'incertitudes dans le volume de trafic et le taux de redondances.

La deuxième partie de cette thèse est consacrée aux problèmes qui se posent lors du déploiement de tels protocoles dans les réseaux. Plus particulièrement, nous proposons de minimiser les changements entre deux configurations réseaux consécutives lorsque plusieurs matrices de trafic sont considérées. Le routage des demandes étant alors assuré avec le protocole de routage OSPF (Open Shortest Path First). Ensuite, nous abordons le problème de la limitation du nombre de règles de routage dans les routeurs en utilisant une technologie de type SDN (Software Defined Networks). Enfin, nous présentons en annexe des travaux complémentaires réalisés au cours de cette thèse concernant le routage multicast et le contrôle de congestion TCP.

Mots clés : Energy-aware Routing, Redundancy Elimination, Open Shortest Path First, Software Defined Networks.

Contents

Acronym

CDF Cumulative Distribution Function

DFS Dynamic Frequency Scaling

DVS Dynamic Voltage Scaling

EAR Energy-aware Routing

ICT Information and Communication Technology

IGP Interior Gateway Protocol

IP Internet Protocol

ISP Internet Service Provider

ILP Integer Linear Programming

MILP Mixed Integer Linear Program

MPLS Multi-protocol Label Switching

OSPF Open Shortest Path First

PoP Point of Presence

QoS Quality of Service

RE Redundancy Elimination

SDN Software-defined Networking

TCAM Ternary Content Addressable Memory

TE Traffic Engineering

WAN Wide Area Network

WOC WAN Optimization Controller

Chapter 1

Introduction

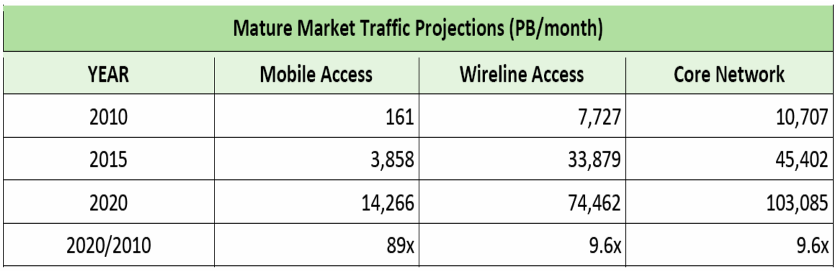
Contents

The concept of energy-efficient networking has gained increasing popularity in the past few years. Because of economical and environmental reasons, green networking has become a key issue for the industry as well as the research community. Recent studies have shown a continuously growing trend of the energy costs and electrical requirements for telecoms [BBDC11, BCRR12]. Since networks are often designed to endure peak load, they are normally under-utilized, leaving a large room for energy saving. For instance, data centers and Internet Service Provider (ISP) involve high-performance and high-availability computing. They therefore rely on powerful devices which are organized in a redundant architecture. While these redundancies greatly increase the network reliability, they also reduce the energy efficiency as all network devices are powered on at full capacity but highly under-utilized most of the time. For this reason, a potential energy saving approach is to put unused devices into sleep mode in off-peak hours without affecting network performance and reliability.

In the main part of this thesis, we study multiple approaches to optimize the power consumption for Internet Service Provider (ISP). Beside green networking, this thesis also presents additional work on multicast network and TCP congestion control. In this introduction, we motivate these approaches, mention the techniques used, and finally enumerate our main contributions.

1.1 Motivation

The impact of the Internet on our lives has become more and more important in recent years. According to a report by Cisco [Cisco13], the peak global throughput has increased by 41% through the year 2012 alone. The recent smart-phone, tablet and laptop revolutions have been contributing to this phenomena. As estimation by [GreenTouch13], over the decade 2010-2020, the global wire-line Internet traffic will increase by a factor of 16, to approach 250 exabytes per month. Moreover, the global mobile Internet traffic will grow even faster, approximately 150 times to reach 40 exabytes per month. Fig. 1.1 shows in details traffic projections of different kinds of networks in the Mature Market (consists of Japan, Northern America and Western Europe). As prediction, the traffic in the Mature Market is growing more slowly than the global traffic: for the period 2010-2020, traffic in mobile access network will increase by 89 times while it will be 9.6 times for the wire-line access and the core network. It is also noted that the traffic contribution to the core network coming from the mobile wireless back haul is small in comparison to the traffic from the wire-line network.



Mature Market Traffic Projections (PB/month)			
YEAR	Mobile Access	Wireline Access	Core Network
2010	161	7,727	10,707
2015	3,858	33,879	45,402
2020	14,266	74,462	103,085
2020/2010	89x	9.6x	9.6x

Figure 1.1: Traffic projections and corresponding multiplicative growth factors [GreenTouch13]

To keep pace, Internet Service Providers (ISP) have to rely on similar growth in bandwidth and capacities of routers and switches. This causes a significant rise in energy consumption and CO_2 emission. According to several studies, the emission of CO_2 produced by the Information and Communication Technology (ICT) could range from 2% up to 10% of the total man-made emissions by 2020 [Global07, Sma10, LHV$^+$12]. In this context, data centers and backbone networks will experience the highest energy consumption growth rates in the forthcoming years [LKWG11]. The studies also show that switches, hubs and routers account for 6 TWh ($\sim$ 0.5 - 2.4

billion dollars) per year in United States. In [Sma10], it is estimated that by 2020 the telecoms infrastructure and devices alone would account for about the 25% of the ICT sector's CO2 emissions. Therefore, the European Commission (EC) has set three key objectives for 2020 (which are known the "20-20-20" targets):

- A 20% reduction in EU greenhouse gas emissions from 1990 levels.
- Raising the share of EU energy consumption produced from renewable resources to 20%.
- A 20% improvement in the EU's energy efficiency.

Energy reduction techniques in the data center, backbone network or access network bear the promise of major cost reductions. In order to achieve these targets, the EC has financed more than 30 research projects working on energy efficiency in ICT [EUF14]. The reduction of energy is becoming an important field of research, not only on hardware but also on networking technology and protocol. Moreover, governments of many countries are starting to recognize the impact of telecommunication on the global energy consumption. This could bring future situations, where new regulations, e.g. increasing the cost of electricity, would be applied. As a result, it is necessary for ISPs to improve their network energy efficiency since it is a part of their operating expenses (OPEX).

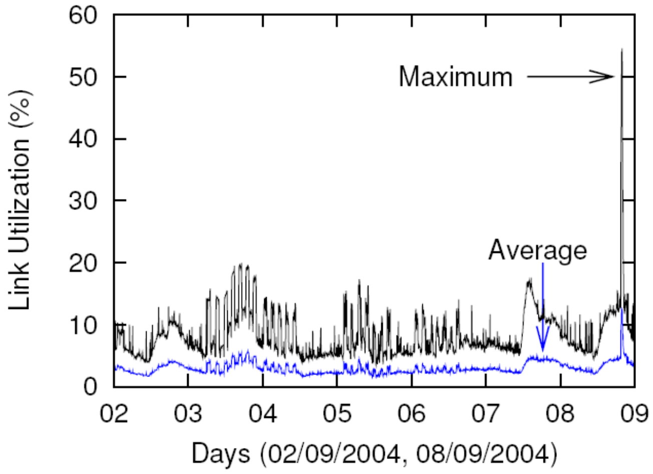


Figure 1.2: Maximum and average link utilization in Abilene network [ZYLZ10]

The work on Internet's energy consumption has been first evoked as a hypothetical working direction by Gupta et al. in 2003 [GS03]. As today's networks are designed and operated to carry traffic in the most reliable way, energy efficiency issue is not taken into account. As a result, a network is usually built with several redundant links and aggressive over-provision in bandwidth. While these redundancies greatly increase the network reliability, they also greatly reduce the network's energy efficiency as all network devices are powered on but highly under-utilized most of the time. Figure 1.2 shows the maximum and average link utilization in Abilene network, a large US education backbone, during a typical week [ZYLZ10]. The average link utilization is only about 2% while only one rare event pushes the maximum over 50%. Therefore, people proposed putting unused network elements into sleep mode to save energy. This research idea is usually called *energy-aware routing (EAR)* [BBDC11, BCRR12, GMMO10, ZYLZ10, CMN09, CSB+08]. However, from a practical point of view, there are a number of issues for EAR. First, the time required for sleeping/activating a link on router is substantial. However, there is work on Energy Efficient Ethernet (EEE) which is the standardization framework of the IEEE 802.3az [CRN+10, BBC+14]. The authors have shown that it is possible to put a link into sleep mode and wake it up in short time (e.g. less than $5\mu s$ for 10GBASE-T links). Thus, we believe that these advances will come in the future, especially if they offer big energy saving. Second, as traffic varies over time, EAR assumes to compute and apply a new routing configuration for each traffic matrix, making in a large number of applied configurations per day. Indeed, frequent changes in network configuration cause network oscillation, e.g. packets may arrive out of order, degrading the perceived QoS for end-users. One possible way to overcome this issue is to limit network reconfigurations. As shown in [CCRP13], with only few configurations, the energy saving can be close to the maximum one, in which a new configuration is applied for each traffic matrix.

Another research topic that has also been active recently is traffic redundancy elimination (RE) [ZA13, AGA+08, AMAR09, SGG10, ZC11]. Observing that traffic on the Internet contains a large fraction of redundancy (e.g. popular contents such as new movies are often downloaded several times subsequently), people propose to use RE techniques to reduce link load in backbone networks. It consists in splitting packets into small chunks, each is indexed with a small key, which are cached along traffic flows as long as they are popular. Then, keys are substituted to chunks in traffic flows to avoid sending multiple times the same content, and the original data are recovered on downstream routers based on the cache synchronization between the sending and the receiving routers. Therefore, traffic redundancy is removed and the capacities of network links are virtually increased. As a result, RE provides more room for aggregating traffic, which is useful for the energy-aware routing approach.

In this thesis, we study GreenRE - a new energy saving model which combines EAR and RE techniques (Chapter 3 and 4). Besides the GreenRE, we have also worked on the perspective of network management for energy-aware routing. That is we consider energy-aware traffic engineering applied in Open Shortest Path First (OSPF) protocol (Chapter 5) and Software-Defined Networking (SDN) (Chapter 6). Beyond the scope of green networking, this thesis also presents some additional works on multicast and TCP congestion control. We introduce in next Section state of the art of the research topics that we have used in this thesis.

1.2 Power Awareness in Network Design and Routing

In this thesis, we examine the problem of power-awareness of routers in backbone networks. Especially, there are three promising approaches, namely power-aware system design, power-aware network design and energy-aware routing. Note that the first two approaches, which are typically used by router manufacturers and network designers, are not the focus of this work. However, understanding these methods is necessary to have a completed picture on energy-efficient network. In this section, we first provide an overview of *power-aware system* and *power-aware network design*. Then, our main work on *energy-aware routing* is presented in detail.

1.2.1 Power-aware System Design

Depending on the area of application, power-aware system design can be classified in two different levels: circuit and equipment levels.

Circuit Level is based on new developments in CMOS technology. Decreasing feature sizes in semiconductor technology has contributed to performance gains as well as reducing the power per transmitted byte. In addition, standard techniques for power efficient design in router including clock gating, dynamic voltage scaling (DVS) and dynamic frequency scaling (DFS) have been used also. With DVS, the supply voltage is reduced when not needed, which results in lower power consumption (but also slower operation of the circuitry). Similarly, DFS reduces the number of processor instructions in a given amount of time (in low operation mode) to save energy.

Equipment Level In this level, there are a number of methods that can be applied to reduce energy. For example, electrical components can be replaced with their counterparts in the optical domain which are more energy efficient. In addition, optical technology innovation continues to evolve and should have an important impact on reducing power consumption in the future. Another approach can be used in the equipment level is multi-chassis router. This new architecture allows to physically separate components and to cluster them to form a single logical router. In particular, several line card chassis can be connected in a multi-chassis router. This architecture can solve the bandwidth scaling problem due to parallel processing of line cards. Although the aggregate power consumption increases on a single router, the heat load is easily spread over larger physical area. It therefore reduces the total cost of the required cooling system.

1.2.2 Power-aware Network Design

This approach is based on an efficient deployment of routers over a set of point of presences (PoPs) such that the total power demand is minimized while QoS requirements are satisfied. The authors in [CSB+08] have demonstrated that being aware of power consumption when designing network topologies can result in significant power reduction. For instance, with a given set of demands and link capacities, there are likely to have many router-level network topologies to satisfy a certain level of QoS. To this extend, the core is normally designed with dense connections using high bandwidth routers, while lower bandwidth routers (but also high connection density) are placed around the core. With power-aware network design, different chassis/line card configurations might be deployed in a network such that provisioning requirements are satisfied. Furthermore, power-hungry packet processing operations are limited to a subset of the routers. As an example, by minimizing energy-hungry components such as large IP routers, and transporting traffic at the lowest layer (more energy efficient) if possible can greatly reduce energy consumption. In addition, the long term objective is to replace power-hungry systems in the core with lower power systems. With such additional refinements in topology design, ISPs have opportunities to save energy costs and potentially reduce equipment footprints in PoPs.

1.2.3 Energy-aware Routing (EAR)

This approach aims at using network protocols to control the whole network, so as to minimize energy consumption while preserving QoS requirements. Before going into detail of EAR, we first present energy profile of router from a traffic load point of view. An energy profile is defined as the dependence of the energy consumption of router on its traffic load. There are two main energy profiles as shown in Fig. 1.3: "idleEnergy" and "fully proportional" models.

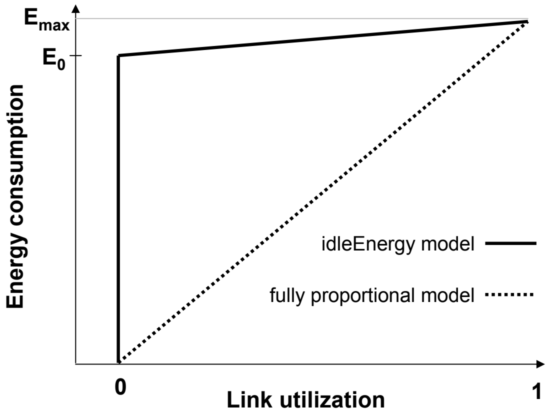


Figure 1.3: Energy profiles

Fully Proportional Model This model represents an ideal case where energy consumption varies linearly with the device utilization, between 0 and E_{max}. As stated in [BCL+10], network devices could present such a behavior if techniques like

Dynamic Voltage Scaling (DVS), modular switching fabrics, etc. are applied to the devices' components. In fact, this model is desired in green networking, however, today network devices are not power-proportional, and it is considered as a futuristic scenario.

idleEnergy Model This model is also named "on/off" energy profile. The influential paper of Chabarek et al. [CSB+08] has shown that the energy consumption of today network equipments is not proportional to the quantity of the transported traffic. In realistic, network device's energy consumption grows linearly between a minimum value E_0 and a maximum value E_{max} which correspond to the idle state and the maximum utilization state, respectively (Fig. 1.3). For more details, the website [Powerlib] lists a database of power consumption values for ICT network equipments.

In this thesis, we focus on the "idleEnergy model" to design and evaluate efficient energy-aware routing (EAR) protocol. We refer the readers to the references [GGS13, GNTS13] for more general work on energy-aware problem (with different energy profiles). In our work, the most basic notion of EAR includes mechanisms for turning off or putting components into sleep mode. In fact, numerous measurement campaigns have been set up to obtain accurate energy models for network equipments. For example, the authors in [CSB+08] have shown that power consumption of a router at a load of 75% is only 2% more than at an idle state (770W vs. 755W). Instead, the dominating factor is the number of active network elements on the network devices such as interfaces (or ports), line cards, base chassis, etc. [MSB09]. Therefore, in order to minimize energy consumption, as few network elements as possible should be used while preserving connectivity and Quality of Service (QoS). In general, networks are designed with redundant links and overprovisioning bandwidth to accommodate traffic bursts as well as to allow rerouting when links fail. As a result, the networks are under-utilized most of the time, leaving a large room for energy saving (Fig. 1.2). Intuitively, it is possible to have multiple paths between a pair of source-destination in a network. When traffic load is low, we can aggregate the traffic into a few links so that other links do not carry any traffic. Then, idle links of routers can be put into sleep mode for energy reduction. In fact, turning off entire routers can earn significant energy saving. However, it is difficult from a practical point of view as it takes time for turning on/off and also reduces life cycle of devices. Therefore, like existing works [CCRP13, GMPR12], we assume to turn off (or put into sleep mode) only links to save energy.

As an example of EAR, we consider a network topology as a grid 3×4 (Fig. 1.4). Each link of the network has capacity 4 Gbps. There are three traffic demands: $(0, 3)$, $(4, 7)$ and $(8, 11)$, each has a volume of 1 Gbps. For specific requirements, network operators can choose a routing protocol or apply different traffic engineering policies for their networks. Commonly, to guarantee QoS, a feasible routing solution should not cause any overloaded links. Therefore, all the four solutions in Fig. 1.4 are feasible. As shown in Fig. 1.4d, the shortest path routing uses 9 active links,

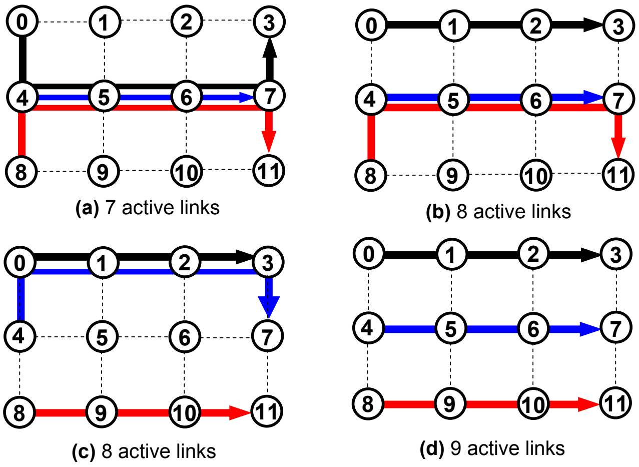


Figure 1.4: Example of energy-aware routing solutions.

then 8 idle links can be put into sleep mode to save energy. However, since there is enough capacity to aggregate the three traffic demands as in Fig. 1.4a, Fig. 1.4b and Fig. 1.4c, energy consumption can be further decreased. Indeed, the routing solution in Fig. 1.4a is the optimal energy-efficient one since it requires a minimum number of active links. The problem of minimizing the number of active links under capacity constraints can be precisely formulated using Mixed Integer Linear Programming (MILP). The authors in [GMMO10] proved that EAR is not in APX, that is there is no polynomial-time constant-factor approximation algorithm.

In this thesis, we propose a new energy-aware routing model call GreenRE. As the key point of EAR is that the network links must have enough capacity to aggregate traffic demands. We leverage a technique call traffic redundancy elimination (RE) to virtually increase capacity on links. From the view point of energy saving, RE allows to aggregate more traffic flows, increasing the energy efficiency of networks. We present in next section background on RE and we show how to use this technique to reduce traffic load as well as to improve energy efficiency for networks.

1.3 Reducing Traffic Load in Network

A significant amount of redundant traffic has been observed over the communication networks. As stated in many studies [AGA$^+$08, AMAR09, SW00], redundancy in Internet traffic is in a range of 15-60%. Typically, some contents on the Internet are highly popular objects, e.g. new movies, songs, etc. Due to many requests, these contents are transferred repeatedly across the network for a large number of users. Moreover, as common activities, a single user can repeatedly access or retrieve the same or similar contents over the Internet several times. It is clear that redundant traffic wastes network resources and even worsens the communication performance by saturating the network bandwidth. Furthermore, with the rapid growth of the Internet traffic, redundancy elimination has been arising as an urgent need. In recent years, there are several works from the academia [ZA13, AMAR09, SGG10, ASA09, AGA+08, SW00, WVS+99] and industries, such as Cisco [LRCBS05], Juniper [Juniper], BlueCoat [BlueCoat], and Riverbed [Riverbed]. These works are ranging from object-level to packet-level redundancy elimination.

- *Object-level approach:* the classical Web cache is an example of object level redundancy elimination [WVS+99]. In more detail, popular HTTP objects are stored in local caches (clients' computers or proxies of the network). Then, the cached contents are used to serve subsequent requests locally, without contacting remote severs. However, the object-level caching cannot eliminate all the redundant contents, especially for the contents that have been changed in only minor ways. Therefore, a better approach called packet-level redundancy elimination has recently been explored and investigated.

- *Packet-level approach:* Spring et al. [SW00] developed the first system which can remove redundant bytes from any traffic flow on the network. They call this approach as *protocol independent technique* as it operates below the application layer and attempts to remove any redundant bytes that appear on the network. Following this approach, several commercial vendors have introduced *Wide area Network (WAN) Optimization Controller (WOC)* - a device that can remove duplicate content from network transfer [BlueCoat, Riverbed, LRCBS05, Juniper]. WOCs are installed at individual sites of small ISPs and enterprises to offer end-to-end redundancy elimination between pairs of sites.

1.3.1 WAN Optimization Controller (WOC)

From the network perspective, a large enterprise usually has three different kinds of offices inside its network: branch offices, regional offices and data centers. The branch offices often connect to the regional offices and the data centers by WAN links with low bandwidth, high latency. Therefore branch offices are the ones suffering the most from a poor network performance. One solution to improve the performance over WAN connections is to pay more money to buy higher bandwidth for the WAN links. To this extent, WOC is another approach to overcome the transport and link capacity limitations. It includes many techniques working together such as application acceleration, TCP acceleration, data compression, data suppression, etc. [GC07]. In our work, focusing on traffic redundancy elimination, we present the two main techniques used in WOC to reduce traffic load: data compression and data suppression.

1.3.1.1 Data Compression

Traditional data compression is a technique used to encode data so that it consists of fewer bits than the original data representation. Each packet can be compressed by a compression algorithm such as Lempel-Ziv or DEFLATE [GC07]. For instance, the DEFLATE algorithm replaces repeated strings with pointers and further uses Huffman coding to efficiently encode symbols that frequently occur. This relies on the assumption that both the senders and the receivers use the same compression algorithm. However, it is well-known that DEFLATE does not compress small packets well [AMAR09], therefore it can not significantly reduce traffic load on the network. In this report, we focus on data suppression - an efficient technique used in WOC to eliminate redundant data traffic.

1.3.1.2 Data Suppression

Data suppression is also commonly called packet-level redundancy elimination. The main idea is to detect patterns of data that have been sent over the network. As shown in Fig. 1.5, the patterns of previously sent data are stored in the database of the accelerators (or WOC) at both the sending and the receiving side. Whenever the accelerator on the sending side notices the same kind of data pattern coming from the sending host, it sends a small signature instead of the original data. The receiving accelerator then recovers the original data by looking up the signature in its database. Because signatures are only a few bytes in size, sending signatures instead of actual data gives significant bandwidth saving.

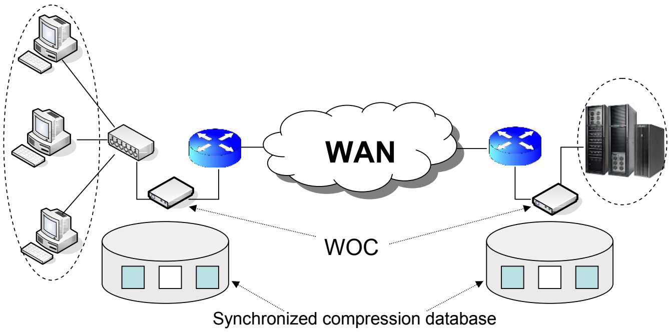


Figure 1.5: Reduction of end-to-end link load using WOC

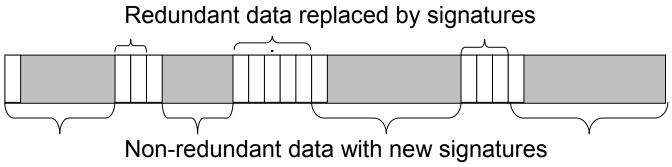

Redundant data replaced by signatures
Non-redundant data with new signatures

Figure 1.6: Encoded message with suppression technique [GC07]

When peering accelerators perform data suppression, only the signatures of the data segments which were stored in database are sent. Otherwise, the accelerators generate signatures for new data segments. As shown in Fig. 1.6, an encoded message consists of many small signatures. Besides, we can see some bigger pieces (with gray color) along with a small signature in the head. They are the new data segments and the new generated signatures which have not been stored yet in the database.

When an encoded message is received, the receiving accelerator begins to decode this message. This is done by replacing each signature sent without attached data with the corresponding data pattern in the local database. Any data pattern that has an accompanying signature is added to the local compression database and the signature is stripped from the encoded message. It is noteworthy that data suppression is commonly implemented in the network or transport layer, thus it does not differentiate among applications. For example, downloading an object from a website populates the local suppression database. The signatures related to this object can be used later for e-mail application when sending with the same (or modified) object. Furthermore, an object that is sent for the first time may be compressible because of some common data patterns from previous objects already stored in the data suppression database. To better explain, we consider a simple example as Fig. 1.7.

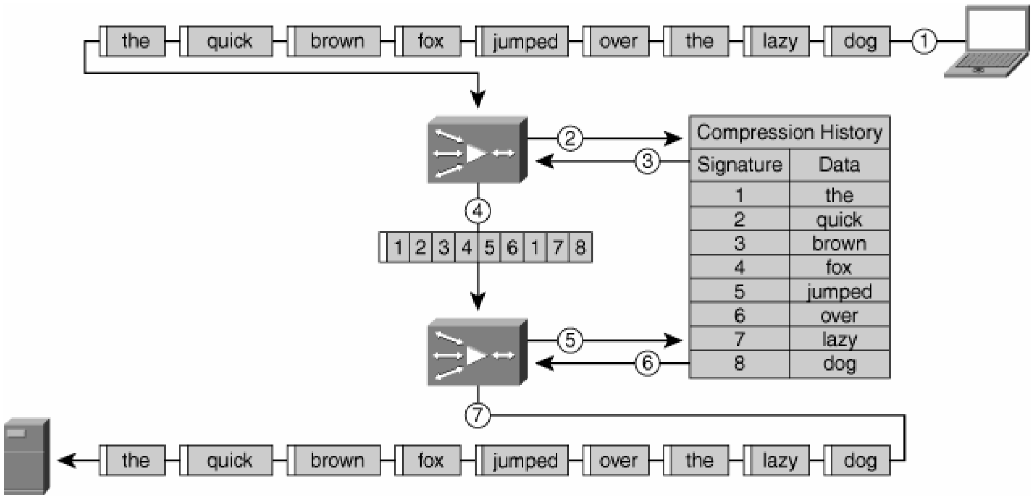


Figure 1.7: Step-by-step of packet suppression process [GC07]

As described in Fig. 1.7, the process of packet suppression is divided by 7 steps:

1. A stream of data is sent as a series of packets.
2. The WOC verifies the existence of each packet in its local database (the compression history).
3. Redundancy is eliminated and redundant segments are replaced by signatures.
4. The encoded packet is sent across the network and intercepted by a remote WOC.
5. The remote WOC compares the contents of the encoded packet with items in its local database.

6. The remote WOC replaces signatures with data patterns from its database.
7. The remote WOC reconstructs the original data and forwards it to the intended destination.

When deploying the accelerators, a best practice is to ensure that enough storage capacity is allocated for the compression history. We refer the readers to a detail calculation of the cache size in [GC07].

1.3.2 Packet Caches on Routers

Recently (2008 - present), the success of WOC deployment has motivated researchers to explore the potential of network-wide redundancy elimination (RE) (see the survey [ZA13]). For instance, Anand et al. [AGA+08][AMAR09][ASA09] have considered the benefits of deploying packet-level RE in routers across the entire Internet and they have shown that packet-level RE is more effective than the object-level one. The basic idea of this proposal is similar to data suppression (Section 1.3.1.2). We show a process of removing redundancy in data packet as Fig.1.8.

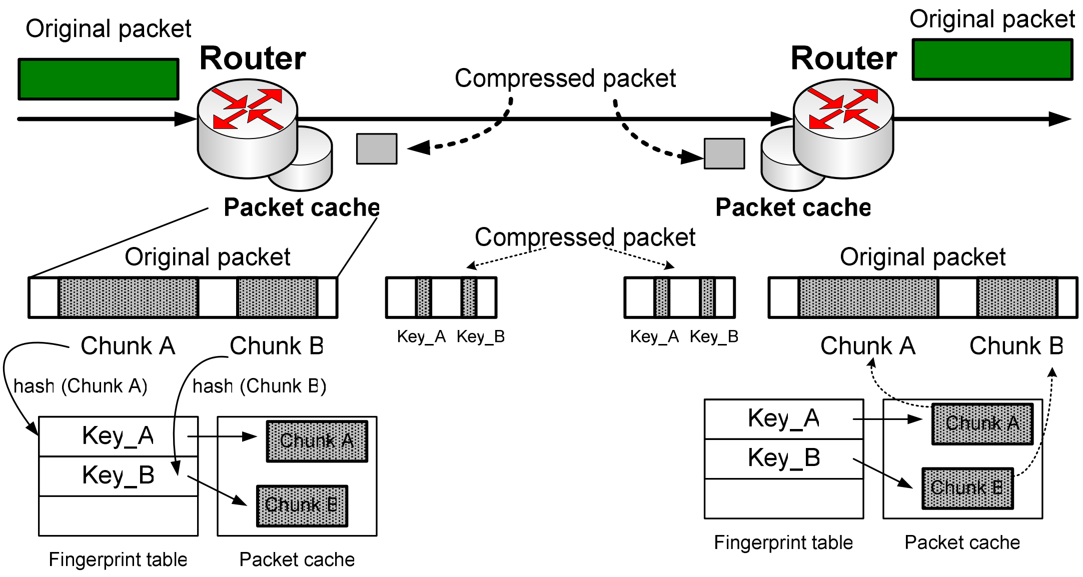


Figure 1.8: Packet-level data redundancy [ZA13]

Each router in an ISP network maintains a cache of recently forwarded packets. For every incoming packet, upstream routers run an algorithm to compute a set of fingerprints. Rather than using MD5 hash, the algorithm uses a sliding hash function which significantly cuts down the hash computation time per packet [ZA13, HWG12]. We refer the readers to the paper [HWG12] for more details on fingerprint algorithms. After computing fingerprints for an arriving packet, each fingerprint is checked against the fingerprint table. If a match is found, it means that the incoming packet has bytes in common with an in-cache packet. The algorithm will try to expand this packet to find a bigger matching region. Each matching region is then removed from the incoming packet and replaced by a small signature (or small key). These signatures then are used by downstream routers to reconstruct the original packet from their local cache. It is important to make sure that the cache on downstream router is consistent with the upstream one.

Obviously, there are two key challenges that hinder the deployment of RE on routers. First, a significant number of memory accesses and heavy computation are required during various stages of RE. Second, a large amount of memory is required for fingerprints and packets stored at routers. Anand et al. [ASA09] introduces SmartRE which considers these challenges in the design. The authors show that on the desktop equipped with 2.4 GHz CPU and 1 GB RAM used for storing caches, the prototype can work at 2.2 Gbps for encoding packets (finding fingerprints and replacing matching regions by signatures) and at 10 Gbps for decoding or reconstructing the original packets. Moreover, they believe that a higher throughput can be attained if the prototype is implemented in hardware. Therefore, the key challenges of limitation in memory and CPU can be overcome.

Another interesting fact is how much load on network links can be reduced when deploying RE on routers? Several real traffic traces have been collected from many networks such as at 11 corporate enterprises in US [AMAR09], at a large university in US [AGA+08] and at 5 sites of a large corporate network in North America [SGG10]. The authors in [AMAR09][AGA+08][SGG10] conclude that the bandwidth saving by using RE can be up to 50%. In addition, a further 10-25% traffic load can be reduced when considering redundancy-aware routing - traffic flows from the same source are aggregated on the same links to achieve inter-flows RE [AGA+08].

In summary, by using RE technique, the volume of traffic demand can be significantly reduced. This is useful for traffic engineering and EAR as capacity on network links are virtually increased. However, from energy saving perspective, RE has a drawback since it increases energy consumption of routers as well [GMPR12]. To find a good trade-off, we have proposed GreenRE - a model that combines EAR and RE to increase energy efficiency for backbone network (Chapter 3 and 4).

1.4 Deployment of EAR in Real World

Beyond the scope of GreenRE model, in this thesis, we also study the impacts of energy-aware routing on network protocols. In particular, we consider real problems when deploying EAR on Open Shortest Path First (OSPF) and Software-defined network (SDN). We introduce in this section some backgrounds on these protocols.

1.4.1 EAR with Open Shortest Path First (OSPF)

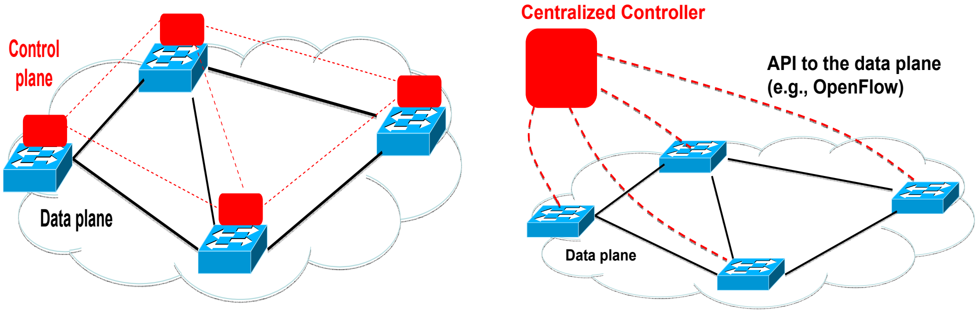
OSPF is a link-state routing protocol for Internet Protocol (IP) networks. It is perhaps the most widely used interior gateway protocol (IGP) in large enterprise networks. It gathers link state information from available routers and constructs a topology map of the network. It then computes the shortest path tree for each destination based on Dijkstra's algorithm. The weights of the links, and thereby the shortest path routes, can be changed by network operators. For instance, the weights could be set proportional to their physical distances, or as suggested by Cisco [Cisco05], the weights should be set inverse to their links' capacity without taking any knowledge of the demand into account. It is widely believed that the OSPF protocol is not flexible enough to do traffic engineering, for example to give a good load balancing routing solution. This is one of the reasons for introducing Multi-protocol Label Switching (MPLS). However, as shown in [FT00], the weights can be optimized for load balancing problem. The authors showed that the found OSPF weight setting can perform closely to that of the *optimal general routing* (MPLS-TE style) where the routing flow of each traffic demand is optimally distributed over all paths between source and destination.

Recently, we found a number of works that have been devoted to energy-aware traffic engineering using OSPF protocol [ACG13, FWMG13, SLX+12, CCGS13]. This link state approach performs a local calculation of shortest paths based on a set of link weights. This avoids optimizing routing on a per-flow basis (like MPLS-TE) which can be complex when a large number of traffic demands are considered. In summary, these works try to find an OSPF weight setting that routes the traffic in a shortest path manner while it minimizes the number of active routers/links. Then, inactive network elements are put into sleep mode to save energy.

From the perspective of traffic engineering, we argue that stability in routing configuration also plays an important role in QoS. In details, frequent changes in network configuration (link weights, slept and activated links) to adapt with traffic fluctuation in daily time cause network oscillation. We propose a novel optimization method of link weight so as to limit the changes in network configurations in multi-period traffic matrices (Chapter 5).

1.4.2 EAR with Software-defined Networking (SDN)

SDN in general, and OpenFlow in particular [MAB+08], has been attracting a growing attention in the networking research community in recent years. In traditional networks (Fig. 1.9a), network devices such as routers and switches act as "closed" systems. They work as "black boxes" with applications implemented on them. Users can only control them via limited and vendor-specific control interfaces. Moreover, the data plane (forwarding function) and control plane are integrated in each device, making them quite difficult to deploy new network protocols. SDN is a new networking paradigm that decouples the control plane from the data plane (Fig. 1.9b). It provides a flexibility to develop and test new network protocols and policies in real networks. OpenFlow has applications in a wide range of networked environments and over past few years, many applications have been built using the OpenFlow API [MAB+08]. For instance, the work in [JKM+13] describes B4 - one of the first and largest SDN deployments in Google data center network. B4 has been in deployment for three years and real lessons learned show that B4 can efficiently meet application bandwidth demands, supports rapid deployment of new network control services and is robust with failure conditions.



(a) Traditional network

(b) SDN network

Figure 1.9: Traditional network vs. SDN network

OpenFlow is a promising method to implement EAR in a network. Without setting entries manually, OpenFlow can collect traffic matrix, performs routing calculation and then installs new routing rules on routers. For instance, the authors in [HSM+10] have implemented and analyzed ElasticTree on a prototype testbed built with production OpenFlow switches. The idea is to use OpenFlow to control traffic flows so that it minimizes the number of used network elements to save energy. Similarly, the authors in [WYW+12] have set up a small testbed using OpenFlow switches to evaluate energy saving for their models. OpenFlow switches have also been mentioned in many existing works as an example of the traffic engineering method to implement the EAR idea [CMTY11, VNS+11].

In this thesis, we discover that the rule space at OpenFlow switches is also important as it can change the routing solution and affects QoS. We therefore propose an optimization method to minimize energy consumption for a backbone network while respecting capacity constraints on links and rule space constraints on routers (Chapter 6).

1.5 Research Methodology

1.5.1 Metrics studied

In this thesis, we are interested in optimizing and evaluating different metrics. In this section we briefly describe them in a clear and simple way. Throughout this thesis, we are focusing on the main concern, that is the energy consumption of backbone networks. Indeed, there are many elements on network that need to apply energy efficiency. In this work, we focus on energy consumption of routers, particularly on the interfaces of routers. The method used here is aggregating the traffic on few links, then putting unused links (or precisely, the two network interfaces connecting the two routers) into sleep mode. From practical point of view, it takes time for sleeping/waking up and also reduces life cycle of devices. We therefore consider that the routers are always on but only links can be put into sleep mode to save energy. As a result, the metric energy efficiency is proportional to the number of idle links on a network.

This sleep mode approach requires network-wide coordination of routers. The challenges are two-fold, namely how to manipulate the routing paths to make as many idle links as possible without significantly affecting network performance and reliability. Since power-aware traffic engineering uses less number of links, it is important to make sure that links are not overloaded and packets do not experience extra long delays. Therefore, beside energy saving, we also evaluate other metrics relating to QoS such as end-to-end delay and link utilization.

- End-to-end delay: as energy-aware routing aims at minimizing the number of active links, longer paths can be used to route traffic demands. However, as EAR should guarantee a certain level of QoS, long end-to-end delay can be a problem, especially for sensitive delay applications such as audio, video streaming. Therefore, in this work, we also evaluate the path length metric when designing our heuristic algorithms.

- Link utilization: since EAR attempts to *aggregate* traffic into a subset of network links, load balancing is sometimes ignored. However, link load is also an important factor to better QoS as it can efficiently handle unexpected surges in traffic demands. Therefore, link utilization is also a metric that we consider in this work.

1.5.2 Techniques used

Over the course of this thesis we faced different problems, calling for different solutions. The main techniques used, ordered from the more theoretical to more empirical, are:

- Mixed Integer Linear Programming (MILP) is the main technique that we use throughout the thesis (Chapter 3, 4, 5 and 6). Basic idea of linear programing (LP) and MILP are described in Chapter 2.

- Robust optimization and duality in linear programming, used in Chapter 4 and 5, described in Chapter 2.

- Greedy heuristic algorithm, used in Chapter 3, 4, 5 and 6, described in Chapter 2.

- Experiments, using commercial software (CPLEX) (Chapter 3, 4, 5 and 6), network simulator NS-2 (used in Appendix A), and live network emulation tool Dummynet (used in Appendix B).

1.6 Contributions

The remainder of this thesis is organized around my contributions. What follows in this section are their short descriptions. Following the regulation in our team, the alphabetic order of authors is employed for every paper except [PMTT12, PTNT12, PTM+10a, LKL+09].

Chapter 2: Preliminaries

In this chapter, we present some preliminaries that we use throughout the thesis. They include linear programming, duality, robust optimization and greedy heuristic strategy.

Chapter 3: Green Networking with Redundancy Elimination

In this chapter, we propose GreenRE - a new EAR model with the support of *data redundancy elimination (RE)*. This technique, enabled within routers, can virtually increase the capacity of network links. Based on real experiments on Orange Labs platform, we show that performing RE increases the energy consumption for routers. Therefore, it is important to determine which routers should enable RE and which links to put into sleep mode so that the power consumption of the network is minimized. We model the problem as Mixed Integer Linear Program (MILP), introduce cutset inequalities to speedup the MILP resolution and propose greedy heuristic algorithms based on shortest path routing for large networks. Simulations on several network topologies show that the GreenRE model can gain further 37% of energy saving compared to the *classical EAR* model.

The results of this chapter have been submitted and accepted for publication in [GMPR14, KPT13, GMPR12].

Chapter 4: Robust Optimization for GreenRE

Motivating from the GreenRE model, we propose a robust model in which fluctuation of traffic demands and redundancy elimination rates are considered. In details, we allow any set of a predefined size of traffic flows to deviate simultaneously from their nominal volumes or RE rates. Using this extra knowledge on the dynamics of the traffic pattern, we are able to significantly increase energy efficiency for backbone networks. We formally define the problem and model it as Mixed Integer Linear Program (MILP). We then propose an efficient heuristic algorithm that is suitable for large networks. Simulation results with real traffic traces show that our approach allows for $16 - 28\%$ extra energy saving with respect to the classical EAR model.

The results of this chapter have been submitted and accepted for publication in [CKP14a, CKPT13, CKP14c, CKP14b].

Chapter 5: Optimizing IGP Link Weights for Energy-efficiency

In this chapter, we consider to save energy with Open Shortest Path First (OSPF) protocol. From the perspective of traffic engineering, we argue that stability in routing configuration also plays an important role in QoS. In details, frequent changes in network configuration (link weights, slept and activated links) to adapt with traffic fluctuation in daily time cause network oscillation. We propose a novel optimization method of link weight so as to limit the changes in network configurations in multi-period traffic matrices. We formally define the problem and model it as Mixed Integer Linear Program (MILP). We then propose efficient heuristic algorithm that is suitable for large networks. Simulation results with real traffic traces on three different networks show that our approach achieves high energy saving and less pain for QoS (in term of less changes in network configuration).

The results of this chapter have been submitted in [MP14a, MP14b].

Chapter 6: Energy-aware Routing with Software-Defined Networks

In this chapter, we focus on using Software-Defined Network (SDN) for energy-aware routing (EAR). SDN can collect traffic matrix and then computes routing solutions satisfying QoS while being minimal in energy consumption. However, prior works on EAR have assumed that the table of OpenFlow switch can hold an infinite number of rules. In practice, this assumption does not hold since the flow table is implemented with Ternary Content Addressable Memory (TCAM) which is expensive and power-hungry. In this work, we propose an optimization method to minimize energy consumption for a backbone network while respecting capacity constraints on links and rule space constraints on routers. In details, we present an exact formulation using Integer Linear Program (ILP) and introduce efficient greedy heuristic algorithm. Based on simulations, we show that using this smart rule space allocation, it is possible to save almost as much power consumption as the classical EAR approach.

The results of this chapter have been accepted for publication in [GMP14a, GMP14b].

Appendix A: Xcast6 Treemap Islands

Due to the complexity and poor scalability, IP Multicast has not been used on the Internet. Recently, Xcast6 - a complementary protocol of IP Multicast has been proposed. However, the key limitation of Xcast6 is that it only supports small multicast sessions. To overcome this, we propose Xcast6 Treemap islands (X6Ti) - a hybrid model of Overlay Multicast and Xcast6. In summary, X6Ti has many advantages: support large multicast groups, simple and easy to deploy on the Internet, no router configuration, no restriction on the number of groups, no multicast routing protocol and no group management protocol. Based on simulation, we compare X6Ti with IP Multicast and NICE protocols to show the benefits of our new model.

The results of this chapter have been accepted for publication in [PMTT12, MPTT11, PTM+10a, LKL+09].

Appendix B: MaxNet TCP Congestion Control

Congestion control is a distributed algorithm to share network bandwidth among competing users on the Internet. In the common case, quick response time for mice traffic (HTTP traffic) is desired when mixed with elephant traffic (FTP traffic). As the current approach, loss-based with Additive Increase - Multiplicative Decrease (AIMD), is too greedy and eventually, most of the network bandwidth would be consumed by elephant traffic. As a result, it causes longer response time for mice traffic because there is no room left at the routers. MaxNet is a new TCP congestion control architecture using an explicit signal to control transmission rate at the source node. In this work, we show that MaxNet can control well the queue length at routers and therefore the response time to HTTP traffic is several times faster than with TCP Reno/RED.

The results of this chapter have been accepted for publication in [PTNT12]

1.7 Publications

We now list the publications that are included in this thesis.

Journals

1. [GMPR14] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut *Minimization of Network Power Consumption with Redundancy Elimination*, Submitted to Computer Communication, 2014 (in revision).

2. [MP14a] J. Moulierac, and T. K. Phan *Optimizing IGP Link Weights for Energy-efficiency in Multi-period Traffic Matrices*, Submitted to Computer Communication, 2014 (in revision).

3. [CKP14a] D. Coudert, A. Kodjo, and T. K. Phan *Robust Energy-aware Routing with Redundancy Elimination*, Submitted to Computers and Operations Research, 2014 (in revision).

Conferences and Workshops

1. [GMP14a] F. Giroire, J. Moulierac, and T. K. Phan *Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing*, in IEEE Global Communications Conference (GlobeCom), 2014.

2. [CKP14c] D. Coudert, A. Kodjo, and T. K. Phan *Robust Optimization for Energy-aware Routing with Redundancy Elimination*, in Algotel, 2014.

3. [CKPT13] D. Coudert, A. Koster, T. K. Phan, and M. Tieves *Robust Redundancy Elimination for Energy-aware Routing*, in IEEE International Conference on Green Computing and Communications (GreenCom), 2013.

4. [KPT13] A. Koster, T. K. Phan and M. Tieves *Extended Cutset Inequalities for the Network Power Consumption Problem*, in International Network Optimization Conference (INOC), Electronic Notes in Discrete Mathematics, **41** 2013, 69 – 76.

5. [GMPR12] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut *Minimization of Network Power Consumption with Redundancy Elimination*, in IFIP NETWORKING, Lecture Notes in Computer Science, **7289** (2012), 247–258.

6. [PMTT12] T. K. Phan, J. Moulierac, N. C. Tran, and N. Thoai *Xcast6 Treemap Islands - Revisiting Multicast Model*, in ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT) (Student Workshop), 2012.

7. [PTNT12] T. K. Phan, T. T. Tran, D. D. Nguyen, and N. Thoai *MaxNet and TCP Reno/RED on Mice Traffic*, in Modeling, Simulation and Optimization of Complex Processes (Springer), 2012.

8. [PTM+10a] K. T. Phan, N. Thoai, E. Muramoto, K. K. Ettikan, B. P. Lim, and P. Y. Tan *Treemap - the Fast Routing Convergence Method for Application Layer Multicast*, in IEEE Consumer Communications and Networking Conference (CCNC), 2010.

9. [LKL+09] B. P. Lim, E. K. Karrupiah, E. S. Lin, T. K. Phan, N. Thoai, E. Muramoto, and P. Y. Tan *Bandwidth fair application layer multicast for multi-party video conference application*, in IEEE Consumer Communications and Networking Conference (CCNC), 2009.

Research Reports

1. [MP14b] J. Moulierac and T. K. Phan *Optimizing IGP Link Weights for Energy-efficiency in a Changing World*, INRIA Research Report, 2014.

2. [CKP14b] D. Coudert, A. Kodjo, and T. K. Phan *Robust Energy-aware Routing with Redundancy Elimination*, INRIA Research Report, 2014.

3. [GMP14b] F. Giroire, J. Moulierac, and T. K. Phan *Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing*, INRIA Research Report, 2014.

4. [MPTT11] J. Moulierac, T. K. Phan, N. Thoai, and N. C. Tran *Xcast6 Treemap Islands - A Mixed Model of Application and Network Layer Multicast*, INRIA Research Report, 2011.

1.8 Bibliography

[ACG13] E. Amaldi, A. Capone, and L. .G. Gianoli, *"Energy-aware IP Traffic Engineering with Shortest Path Routing"*, Computer Networks **57** (2013), 1503–1517.

[AGA+08] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker, *"Packet Caches on Routers: the Implications of Universal Redundant Traffic Elimination"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2008, pp. 219–230.

[AMAR09] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, *“Redundancy in Network Traffic: Findings and Implications”*, ACM Special Interest Group for the Computer Systems Performance Evaluation Community (SIGMETRICS), 2009, pp. 37–48.

[ASA09] A. Anand, V. Sekar, and A. Akella, *“SmartRE: an Architecture for Coordinated Network-wide Redundancy Elimination”*, ACM Special Interest Group on Data Communication (SIGCOMM), 2009, pp. 87–98.

[BBC+14] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, and P. Lago, *“A Closed-Form Model for the IEEE 802.3az Network and Power Performance”*, IEEE Journal on Selected Areas in Communications **32** (2014), 16 – 27.

[BBDC11] 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 Communication Surveys and Tutorials **13** (2011), 223 – 244.

[BCL+10] A. P. Bianzino, C. Chaudet, F. Larroca, D. Rossi, and J. L. Rougier, *“Energy-aware Routing: A Reality Check”*, IEEE GLOBECOM Workshop, 2010.

[BCRR12] A. P. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, *“A Survey of Green Networking Research”*, IEEE Communication Surveys and Tutorials **14** (2012), 3 – 20.

[BlueCoat] *Bluecoat: Wan optimization*, http://www.bluecoat.com/.

[CCGS13] A. Capone, C. Cascone, L. G. Gianoli, and B. Sansò, *“OSPF Optimization via Dynamic Network Management for Green IP Networks”*, Sustainable Internet and ICT for Sustainability (SustainIT), 2013, pp. 1–9.

[CCRP13] L. Chiaraviglio, A. Cianfrani, E. L. Rouzic, and M. Polverini, *“Sleep Modes Effectiveness in Backbone Networks with Limited Configurations”*, Computer Networks **57** (2013), 2931–2948.

[Cisco05] *Cisco: Ospf design guide*, www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/7039-1.html, 2005.

[Cisco13] *Cisco visual networking index: Forecast and methodology, 2012 - 2017*, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html, 2013.

[CKP14a] D. Coudert, A. Kodjo, and T. K. Phan, *"Robust Energy-aware Routing with Redundancy Elimination"*, Computers and Operations Research (submitted), 2014.

[CKP14b] D. Coudert, A. Kodjo, and T. K. Phan, *"Robust Energy-aware Routing with Redundancy Elimination"*, http://hal.inria.fr/hal-00936745, 2014, INRIA Research Report.

[CKP14c] D. Coudert, A. Kodjo, and T. K. Phan, *"Robust Optimization for Energy-aware Routing with Redundancy Elimination"*, Algotel, 2014.

[CKPT13] D. Coudert, A. Koster, T. K. Phan, and M. Tieves, *"Robust Redundancy Elimination for Energy-aware Routing"*, IEEE International Conference on Green Computing and Communications (GreenCom), 2013, pp. 179–186.

[CMN09] L. Chiaraviglio, M. Mellia, and F. Neri, *"Energy-aware Backbone Networks: a Case Study"*, IEEE International Conference on Communications (ICC) Workshop, 2009.

[CMN11] L. Chiaraviglio, M. Mellia, and F. Neri, *"Minimizing ISP Network Energy Cost: Formulation and Solutions"*, IEEE/ACM Transaction in Networking **20** (2011), 463 – 476.

[CMTY11] A. R. Curtis, J. C. Mogul, J. Tourrilhes, and P. Yalagandula, *"DevoFlow: Scaling Flow Management for High-Performance Networks"*, ACM Special Interest Group on Data Communication (SIGCOMM), vol. 41, 2011, pp. 254 – 265.

[CRN+10] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. A. Maestro, *"IEEE 802.3az: The Road to Energy Efficient Ethernet"*, IEEE Communications Magazine **48** (2010), 50–56.

[CSB+08] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, *"Power Awareness in Network Design and Routing"*, IEEE International Conference on Computer Communications (INFOCOM), 2008.

[EUF14] *"European Union Funded Projects Information in Green Networks"*, http://cordis.europa.eu/fp7/ict/sustainable-growth/, 2014.

[FT00] B. Fortz and M. Thorup, *"Internet Traffic Engineering by Optimizing OSPF Weights"*, Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 2, 2000, pp. 519–528.

[FWMG13] F. Francois, N. Wang, K. Moessner, and S. Georgoulas, *“Green IGP Link Weights for Energy-efficiency and Load-balancing in IP Backbone Networks”*, International Conferences on Networking (IFIP NETWORKING), 2013, pp. 1–9.

[GC07] T. Jr. Grevers and J. Christner, *“Application Acceleration and WAN Optimization Fundamentals”*, Cisco Press, 2007.

[GGS13] R. G. Garroppo, S. Giordano, and G. Nencioniand M. G. Scutella, *“Mixed Integer Non-Linear Programming Models for Green Network Design”*, Computers and Operations Research **40** (2013), 273 – 281.

[Global07] *Global action plan*, http://globalactionplan.org.uk, 2007.

[GMMO10] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy, *“Minimizing Routing Energy Consumption: from Theoretical to Practical Results”*, IEEE/ACM Green Computing and Communications (GreenCom), 2010, pp. 252–259.

[GMP14a] F. Giroire, J. Moulierac, and T. K. Phan, *“Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing”*, IEEE Global Communications Conference (GlobeCom), 2014.

[GMP14b] F. Giroire, J. Moulierac, and T. K. Phan, *“Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing”*, http://hal.inria.fr/hal-00990038, 2014, INRIA Research Report.

[GMPR12] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, *“Minimization of Network Power Consumption with Redundancy Elimination”*, International Conferences on Networking (IFIP NETWORKING), **7289** (2012), 247–258.

[GMPR14] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, *“Minimization of Network Power Consumption with Redundancy Elimination”*, Computer Communications (submitted), 2014.

[GreenTouch13] *Greentouch green meter research study: Reducing the net energy consumption in communications networks by up to 90% by 2020*, http://www.greentouch.org/index.php?page=green-meter-research.

[GNTS13] R. Garroppo, G. Nencioni, L. Tavanti, and M. G. Scutella, *“Does Traffic Consolidation Always Lead to Network Energy saving?”*, IEEE Communications Letters **17** (2013), 1852 – 1855.

[GS03] M. Gupta and S. Singh, *“Greening of the Internet”*, ACM Special Interest Group on Data Communication (SIGCOMM), 2003, pp. 19–26.

[HSM+10] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, *"ElasticTree: Saving Energy in Data Center Networks"*, USENIX conference on Networked systems design and implementation (NSDI), 2010.

[HWG12] E. Halepovic, C. Williamson, and M. Ghaderi, *"Enhancing redundant network traffic elimination"*, Computer networks, 2012, pp. 795–809.

[JKM+13] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Holzle, S. Stuart, and A. Vahdat, *"B4: Experience with a Globally-Deployed Software Defined WAN"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2013.

[Juniper] *Juniper networks: Application acceleration*, http://www.juniper.net/us/en/products-services/application-acceleration.

[KPT13] A. Koster, T. K. Phan, and M. Tieves, *"Extended Cutset Inequalities for the Network Power Consumption Problem"*, International Network Optimization Conference (INOC), Electronic Notes in Discrete Mathematics, **41** (2013), 69–76.

[LHV+12] S. Lambert, W. V. Heddeghem, W. Vereecken, B. Lannoo, D. Colle, and M. Pickavet, *"Worldwide Electricity Consumption of Communication Networks"*, Optics Express **20** (2012), 513–524.

[LKL+09] B. P. Lim, E. K. Karrupiah, E. S. Lin, T. K. Phan, N. Thoai, E. Muramoto, and P. Y. Tan, *"Bandwidth fair application layer multicast for multi-party video conference application"*, IEEE Consumer Communications and Networking Conference (CCNC), 2009, pp. 1–5.

[LKWG11] C. Lange, D. Kosiankowski, R. Weidmann, and A. Gladisch, *"Energy Consumption of Telecommunication Networks and Related Improvement Options"*, IEEE Journal of Selected Topics in Quantum Electronics **17** (2011), 285 – 295.

[LRCBS05] E. Lev-Ran, I. Cidon, and I. Z. Ben-Shaul, *"Method and Apparatus for Reducing Network Traffic over Low Bandwidth Links"*, Patent US 7636767, 2005.

[MAB+08] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, *"Openflow: Enabling Innovation in Campus Networks"*, ACM Computer Communication Review **38** (2008), 69 – 74.

[MP14a] J. Moulierac and T. K. Phan, *"Optimizing IGP Link Weights for*

Energy-efficiency in Multi-period Traffic Matrices", Computer Communications (submitted), 2014.

[MP14b] J. Moulierac and T. K. Phan, *"Optimizing IGP Link Weights for Energy-efficiency in a Changing World"*, http://hal.inria.fr/hal-00988882, 2014, INRIA Research Report.

[MPTT11] J. Moulierac, T. K. Phan, N. Thoai, and N. C. Tran, *"Xcast6 Treemap Islands - A Mixed Model of Application and Network Layer Multicast"*, http://hal.inria.fr/inria-00637656/, 2011, INRIA Research Report.

[MSB09] P. Mahadevan, P. Sharma, and S. Banerjee, *"A Power Benchmarking Framework for Network Devices"*, International Conferences on Networking (IFIP NETWORKING), 2009, pp. 795–808.

[PMTT12] T. K. Phan, J. Moulierac, N. C. Tran, and N. Thoai, *"Xcast6 Treemap Islands - Revisiting Multicast Model"*, ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT) (Student Workshop), 2012, pp. 33–34.

[Powerlib] *Powerlib*, http://powerlib.intec.ugent.be/database/.

[PTM+10] K. T. Phan, N. Thoai, E. Muramoto, K. K. Ettikan, B. P. Lim, and P. Y. Tan, *"Treemap - the Fast Routing Convergence Method for Application Layer Multicast"*, IEEE Consumer Communications and Networking Conference (CCNC), 2010, pp. 1–5.

[PTNT12] T. K. Phan, T. T. Tran, D. D. Nguyen, and N. Thoai, *"MaxNet and TCP Reno/RED on Mice Traffic"*, Modeling, Simulation and Optimization of Complex Processes, 2012, pp. 247–255.

[Riverbed] http://www.riverbed.com/us/solutions/wan_optimization/.

[SGG10] Y. Song, K. Guo, and L. Gao, *"Redundancy-aware Routing with Limited Resources"*, International Conference on Computer Communications and Networks (ICCCN), 2010, pp. 1–6.

[SLX+12] M. Shen, H. Liu, K. Xu, N. Wang, and Y. Zhong, *"Routing On Demand: Toward the Energy-Aware Traffic Engineering with OSPF"*, International Conferences on Networking (IFIP NETWORKING), 2012, pp. 232–246.

[Sma10] *"SMART 2020: Enabling the low carbon economy in the information age"*, http://www.smart2020.org/_assets/files/02_Smart2020Report.pdf.

[SW00] N. T. Spring and D. Wetherall, *"A Protocol-Independent Technique for Eliminating Redundant Network Traffic"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2000.

[VNS+11] N. Vasic, D. Novakovic, S. Shekhar, P. Bhurat, M. Canini, and D. Kostic, *"Identifying and Using Energy-Critical Paths"*, ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT), 2011.

[WVS+99] A. Wolman, M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy, *"On the Scale and Performance of Cooperative Web Proxy Caching"*, ACM Symposium on Operating Systems Principles (1999), 16–31.

[WYW+12] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, *"CARPO: Correlation-Aware Power Optimization in Data Center Networks"*, IEEE INFOCOM, 2012.

[ZA13] Y. Zhang and N. Ansari, *"On Protocol-Independent Data Redundancy Elimination"*, IEEE Communications Surveys and Tutorials **16** (2013), 455 – 472.

[ZC11] E. Zohar and I. Cidon, *"The Power of Prediction: Cloud Bandwidth and Cost Reduction"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2011.

[ZYLZ10] M. Zhang, C. Yi, B. Liu, and B. Zhang, *"GreenTE: Power-aware Traffic Engineering"*, IEEE International Conference on Network Protocols (ICNP), 2010, pp. 21–30.

Chapter 2

Preliminaries

Contents

In this chapter, we present some preliminaries that we use throughout the thesis. They include linear programming, robust optimization and greedy heuristic strategy.

2.1 Preliminary: Linear Programming

Linear Programming (LP) is a general framework that can be used to model many combinatorial problems [Sch98, Chv83]. A linear program comprises a linear *objective* function, a set of linear inequality *constraints* and a set of *variables*, upon which the objective and the constraints are defined. The objective function can be either minimized or maximized. If the goal is just to justify whether the set of constraints is feasible or not, the objective function can be omitted. The constraints are inequalities comprising a linear combination of variables.

A LP can be written as:

$$\max\{\mathbf{c}^T\mathbf{x} : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}, \tag{2.1}$$

where A is a matrix and $\mathbf{c}$ and $\mathbf{b}$ are vectors of known coefficients and $\mathbf{x}$ is the vector of variables. If all the variables are real numbers, we simply call the linear program. However, if some variables are integers, we say we face a *Mixed Integer Linear Program* (MILP) (ILP if all the variables are integral).

An interesting property of linear program is its duality. For any LP of the form presented in the formulation 2.1, called the *primal* problem, its *dual* problem is:

$$\min\{\mathbf{b}^T\mathbf{y} : A^T\mathbf{y} \geq \mathbf{c}, \mathbf{y} \geq \mathbf{0}\} \tag{2.2}$$

Notice that the dual of the dual problem is the original primal problem. The objective function of the dual problem, at any feasible solution, is always greater than the value of the objective function of the primal, at any feasible solution.

Furthermore, if the primal has an optimal solution $\mathbf{x}^\star$, then its dual has an optimal solution $\mathbf{y}^\star$ given by:

$$\mathbf{c}^T\mathbf{x}^\star = \mathbf{b}^T\mathbf{y}^\star. \tag{2.3}$$

These properties are often used to find bounds on the objective function value. This can be useful for solving algorithms, or as a stopping criterion when a solution that is close enough to optimum is sufficient. In this thesis, we use duality as a basic for our $\Gamma-$robust network design (Chapter 4).

It is well known that MILP is NP-hard in general. Still, due to wide application over practical problems, there is a big interest in solving these models. Many exact methods have been proposed: cutting plane, branch and bound, column generation and row generation to name a few (see [Sch98, Chv83] for further reading). These methods are usually accessed through solvers – software packages which allow finding exact or approximate solutions of specified MILP. A brief overview of currently available solvers can be found in [LL10].

In this thesis, we use the multi-commodity flow model, a classical approach in routing problem (see [Min06, AMO93] for a survey). We present a network topology as an undirected graph $G = (V, E)$. The set of nodes V describe routers and the edges $(u, v) \in E$ describe connections between those routers. We note $N(u)$ the set of neighbors of u in the graph G. We denote f_{uv}^{st} the fraction of the flow on edge (u, v) flowing from u to v corresponding to the demand D^{st}. First, there is a set of constraints called flow conservation, that basically states that incoming flows must be equal to outgoing flows, unless they are at the endpoints:

$$\sum_{v\in N(u)} \left(f_{vu}^{st} - f_{uv}^{st}\right) = \begin{cases} -1 & \text{if } u = s, \\ 1 & \text{if } u = t, \ \forall u \in V, (s,t) \in \mathcal{D} \\ 0 & \text{else} \end{cases} \tag{2.4}$$

Then, for each link, the sum of values of flows flowing through it cannot exceed *link capacity* C:

$$\sum_{(s,t)\in\mathcal{D}} D^{st}\left(f_{uv}^{st} + f_{vu}^{st}\right) \le C \quad \forall uv \in E \tag{2.5}$$

Finally, we set the flow variable as fractional or binary depending on the routing model we are considering.

Based on the above multi-commodity flow model, we can obtain a number of useful variants. The capacity can be a constant (maybe given for each link), when doing routing over a given network, or some cost function, when doing network provisioning. Depending on the objective, there may be various optimization goals basing on different costs, or even no goal when the only interest is for finding a feasible routing. Later in this chapter, we extended this approach by taking into account compressed flows. Solving the ILP directly yields an exact solution, albeit the running time is exponential in the instance size. Limiting the time given to the solver may yield sub-optimal, but possibly acceptable solutions. We refer the readers to the book [AMO93] for further applications of network flows.

2.2 Preliminary: Robust Optimization

Over the past years, robust optimization has been established as a special branch of mathematical optimization allowing to handle uncertain data [BTGN09, BTN02]. A specialization of robust optimization, which is particularly attractive by its computational tractability, is the so-called Γ-robustness concept introduced by Bertsimas and Sim [BS03, BS04]. Instead of deterministic coefficients, the coefficients a_j of a constraint $\sum_j a_j x_j \leq b$ are assumed to be random variables. Bertsimas and Sim have shown that in case all random variables are independent and have a symmetric distribution of the form $a_j \in [\bar{a}_j - \hat{a}_j, \bar{a}_j + \hat{a}_j]$ (with $\bar{a}_j$ the average and $\hat{a}_j$ the maximum deviation), it can be guaranteed that the constraint is satisfied with high probability by defining an appropriate integer Γ and replacing the constraint by

$$\sum_j \bar{a}_j x_j + \max_{J:|J|\leq\Gamma} \sum_{j\in J} \hat{a}_j x_j \leq b. \tag{2.6}$$

This constraint models that for each realization of the uncertainties at most Γ many (but arbitrary) coefficients can deviate from their nominal values. Given an arbitrary realization, it is shown in [BS03, BS04], that the probability that (2.6) is violated, is about $1 - \Phi(\frac{\Gamma-1}{\sqrt{n}})$, where Φ is the cumulative distribution function of the standard normal distribution and n equals the number of uncertain coefficients. This result is independent of the actual distribution of a_j.

Note, that constraint (2.6) is deterministic and the complete problem can be reformulated as a standard mixed integer problem. So the model including uncertainty can be solved by the same means as the original problem, again see [BS03, BS04] for details. From a practical perspective, by varying the parameter Γ, different solutions can be obtained with different levels of robustness (the higher Γ the more robust, but also more expensive, the solution is). This concept has already been applied to several network optimization problems [AABP07, KKR11, DKK$^+$13].

In this thesis, we use Γ−robustness to deal with the uncertainties of traffic demands and redundancy elimination rates. We show that the Γ−robustness is well-suited to our problem since in real traffic traces, only a few of the demands are simultaneously at their peaks. Then network operators can choose a suitable Γ parameter to fit with their networks.

2.3 Preliminary: Greedy Heuristic Algorithm

A greedy algorithm is an algorithm that follows the locally optimal choice at each stage with the hope of finding a global optimum. In many problems, greedy strategies fail to find the globally optimal solution, because they usually do not operate exhaustively on all the data. Nevertheless, they are useful because they may yield solutions that approximate the global optimal solution in a reasonable time.

There are several examples of using heuristic algorithms. For instance, the *traveling salesman problem (TSP)* asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city? This problem is known to be NP-hard, the worst-case running time of finding optimal solution increases exponentially with the number of cities. From a practical point of view, a greedy strategy should be used to quickly solve the TSP, that is "At each stage, visit an unvisited city that is the nearest one to the current city". Obviously, this heuristic approach does not guarantee to find a globally optimal solution, however it can find a feasible solution in a reasonable time. It is noted that we can find a 2-approximation algorithm for metric TSP based on minimum spanning tree. Moreover, an improvement called Christofides' algorithm can achieve a 3/2-approximation algorithm for metric TSP [TSP].

In general, greedy algorithms have the following components:

- A candidate set, from which a solution is created.
- A feasibility function, which is used to determine if a candidate (from the candidate set) can be used to contribute to a solution.
- An objective function, which is used to assign a value to a solution. Then, these values are used by a selection function to choose the best candidate to be added to the solution.
- A stop condition, which indicates when the algorithm should stop.

For further reading on greedy heuristic, we refer the readers to the book [CLRS09]. Throughout this thesis, we are working on the energy-aware routing (EAR) problem. The goal of EAR is to find a feasible routing solution (without overloaded link) that minimizes the number of active links. The authors in [GMMO10] proved that EAR is not in APX (and so it is an NP-hard problem), that is there is no polynomial-time constant-factor approximation algorithm. In addition, the authors also proposed greedy heuristic algorithms to find efficient solutions for large networks [GMMO10]. In this thesis, we propose a generic heuristic strategy based on [GMMO10] as follows (Fig. 2.1):

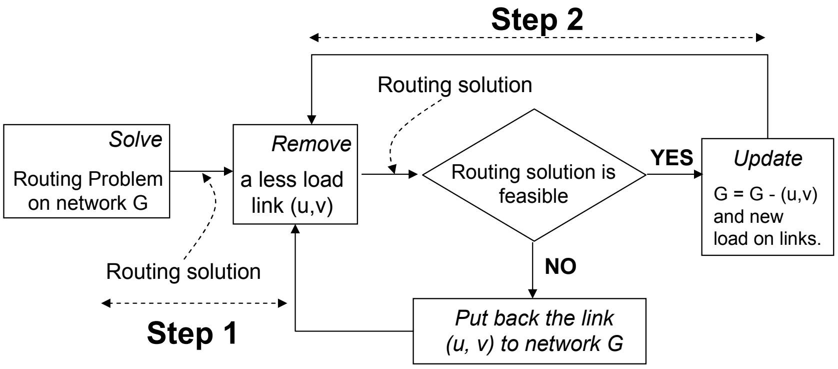

Step 2
Routing solution
Solve
Routing Problem on network G
Remove
a less load link (u,v)
Routing solution is feasible
YES
Update
G = G - (u,v) and new load on links.
NO
Routing solution
Put back the link (u, v) to network G
Step 1

Figure 2.1: Diagram of heuristic algorithm

- Step 1: given an input which are a network topology G and a set of traffic demand, find a feasible routing solution.
- Step 2: based on the routing in step 1, compute load for each link on the network. Then, remove the least load link from the network. After that, check if we still can find feasible routing solution. If yes, update new load on links based on new routing solution and continue the removing link process. Otherwise, we put back the removed link and choose the next less load one to remove. The algorithm will terminate when no more links can be removed.

Based on this generic heuristic, we develop in detail the algorithms for specific problems presented in Chapter 3, 4, 5 and 6.

2.4 Bibliography

[AABP07] A. Altin, E. Amaldi, P. Belotti, and M. C. Pinar, *"Provisioning Virtual Private Networks under Traffic Uncertainty"*, Networks **49** (2007), 100–115.

[BS03] D. Bertsimas and M. Sim, *"Robust Discrete Optimization and Network Flows"*, Mathematical Programming **98** (2003), 49 – 71.

[BS04] D. Bertsimas and M. Sim, *"The Price of Robustness"*, Operations Research **52** (2004), 35 – 53.

[BTGN09] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *"Robust optimization"*, Princeton Series in Applied Mathematics, Princeton University Press, 2009.

[BTN02] A. Ben-Tal and A. Nemirovski, *"Robust Optimization - Methodology and Application"*, Mathematical Programming **92** (2002), 453 – 480.

[CLRS09] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *"Introduction to Algorithms"*, MIT press, 2009.

[DKK+13] S. Duhovniko, A. M. C. A. Koster, M. Kutschka, F. Rambach, and D. Schupke, *"*T*-Robust Network Design for Mixed-Line-Rate-Planning of Optical Networks"*, National Fiber Optic Engineers Conference (OFC/NFOEC), 2013.

[GMMO10] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy, *"Minimizing Routing Energy Consumption: from Theoretical to Practical Results"*, IEEE/ACM Green Computing and Communications (GreenCom), 2010, pp. 252–259.

[KKR11] A. M. C. A. Koster, M. Kutschka, and C. Raack, *"On the Robustness of Optimal Network Designs"*, IEEE International Conference on Communications

(ICC), 2011, pp. 1 – 5.

[LL10] J. T. Linderoth and A. Lodi, *“MILP software”*, Wiley Encyclopedia of Operations Research and Management Science, 2010.

[Min06] Michel Minoux, *“Multicommodity network flow models and algorithms in telecommunications”*, Handbook of Optimization in Telecommunications, Springer, 2006, pp. 163–184.

[Sch98] A. Schrijver, *“Theory of Linear and Integer Programming”*, John Wiley & Sons, Inc., 1998.

[TSP] *“Approximation Algorithms for the Traveling Salesman Problem”*, http://people.orie.cornell.edu/dpw/orie6300/Recitations/rec11.pdf.

Chapter 3

Green Networking with Redundancy Elimination

Contents

In this chapter, we propose GreenRE - a new energy-aware routing (EAR) model with the support of *data redundancy elimination (RE)*. This technique, enabled within routers, can virtually increase the capacity of network links. However, as RE requires additional energy consumption, the model should determine which routers should enable RE and which links to put into sleep mode to minimize the total power consumption.

3.1 Publications

The remainder of this chapter corresponds to *Minimization of Network Power Consumption with Redundancy Elimination* by F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut which has been submitted to the journal of Computer Communications, 2014. This is an extended version of the work with the same title and authors accepted for publication in the proceedings of IFIP NETWORKING, Lecture Notes in

Computer Science, 2012. Additionally, an enhance ILP formulation (section 3.4.2) is followed the paper *Extended Cutset Inequalities for the Network Power Consumption Problem* by A. Koster, T. K. Phan and M. Tieves which is published in the proceedings of International Network Optimization Conference (INOC), Electronic Notes in Discrete Mathematics, 2013.

3.2 Introduction

Recent studies exhibit that traffic load on routers has a small influence on their energy consumption [CSB+08, MSB09]. Instead, the dominating factor is the number of active elements on routers such as ports, line cards, base chassis, etc. The basic idea of energy-aware routing (EAR) is that, during low traffic periods (e.g. at night), traffic demands can be routed over a subset of the network links while preserving connectivity and QoS. In this way, the links excluded by the routing paths can be put into sleep mode (or more precisely, two network interfaces on the two routers will sleep) to save energy.

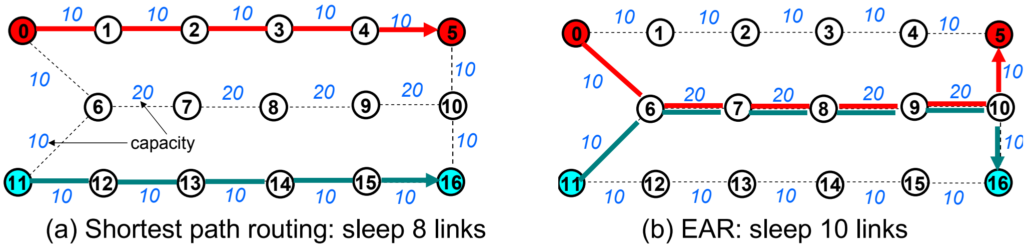
In general, link capacity is the main constraint of the EAR problem. In this work, we use an assumption that routers can eliminate redundant data traffic and hence, virtually increase capacity of network links. As a result, more traffic flows can be redirected and more links can be put into sleep mode to save energy. Although routers nowadays cannot remove repeated content from network transfers, there exists WAN Optimization Controller (WOC) - a commercial device used in enterprises or small ISPs to eliminate traffic redundancy [BlueCoat, GC07, Riverbed]. In order to identify the power consumption directly induced by RE, we perform real experiments on the WOC. Because the main idea RE is similar to the WOC functionality (see Section 3.3.2), we believe that when a router eliminates traffic redundancy, it also consumes additional energy like the WOC. In summary, the contributions of this chapter are the following:

- We do real experiments to exhibit the power consumption of a WOC.
- We define and formulate GreenRE - a new EAR model as Mixed Integer Linear Program (MILP).
- We propose and evaluate a greedy heuristic algorithms that can be used for large-scale networks.
- We evaluate energy saving of the GreenRE model on real network topologies.

The rest of this chapter is structured as follows. We summarize related works in Section 3.3. In Section 3.4, we model GreenRE as MILP, then propose a greedy heuristic algorithm. Evaluation results are presented in Section 3.5. Finally, we conclude the work in Section 3.6.

3.3 Related Works

3.3.1 Classical Energy-aware Routing (EAR)



(a) Shortest path routing: sleep 8 links

(b) EAR: sleep 10 links

Figure 3.1: Example of shortest path routing and EAR.

As an example of EAR, we refer to Fig. 3.1. There are two traffic demands 0 → 5 and 10 → 15, both with a volume $D = 10$ Gbps. As shown in Fig. 3.1a, the shortest path routing is feasible because the links have enough capacity to route all the demands. In this solution, 8 links can be put into sleep mode to save energy. However, we can do better with EAR solution in Fig. 3.1b where we allows 10 links to sleep to further reduce energy consumption. Energy-aware routing is known to be NP-Hard problem [GMMO10]. There are many work in literature proposing the exact formulation and also the heuristic algorithms to find admissible solutions for large networks [GMMO10, CMN11].

3.3.2 Traffic redundancy elimination (RE)

Internet traffic exhibits a large amount of redundancy when different users access the same or similar contents. Therefore, several works [AGA$^+$08, AMAR09, ASA09, SGG10] have explored how to eliminate traffic redundancy on the network. Spring et al. [SW00] developed the first system to remove redundant bytes from any traffic flows. Following this approach, several commercial vendors have introduced WAN Optimization Controller (WOC) - a device that can remove duplicate content from network transfers [BlueCoat, GC07, Riverbed]. WOCs are installed at individual sites of small ISPs or enterprises to offer end-to-end RE between pairs of sites.

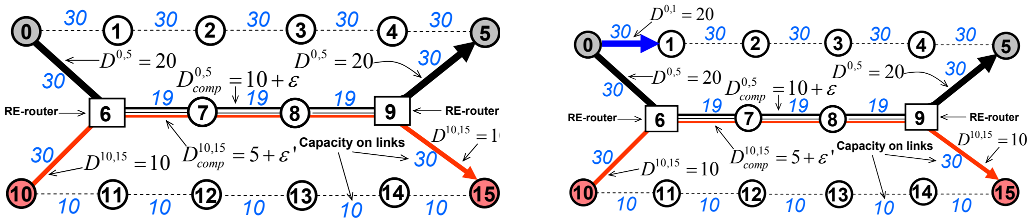
Recently, the success of WOC deployment has motivated researchers to explore the benefits of deploying RE in routers across the entire Internet [AMAR09, ASA09, SGG10]. The core techniques used here are similar to those used by the WOC: each router on the network has a local cache to store previously sent data used to encode and decode data packets later on. Obviously, this technique requires heavy computation and large memory for the local cache. However, Anand et al. have shown that on a desktop 2.4 GHz CPU with 1 GB RAM, the prototype can work at 2.2 Gbps for encoding and at 10 Gbps for decoding packets [ASA09]. Moreover, they believe that higher throughput can be attained if the prototype is implemented in hardware. Several real traffic traces have been collected to show that up to 50%

of the traffic load can be reduced with RE support [AMAR09, ASA09, SGG10].

In the next section, we introduce GreenRE - the first model of *energy-aware routing* with RE support. We show that RE, which was initially designed for bandwidth saving, is also potential to reduce network power consumption.

3.4 Energy-aware Routing with RE

In the GreenRE model, RE is used to virtually increase capacity of the network links. A drawback is that, as shown in [GMPR12], when a router performs RE, it consumes more energy than usual. This introduces a trade-off between enabling RE on routers and putting links into sleep mode. We show that it is a non-trivial task to find which routers should perform RE and which links should sleep to minimize energy consumption for a backbone network.



(a) sleep 10 links, enable 2 RE-routers

(b) sleep 9 links, enable 2 RE-routers

Figure 3.2: GreenRE with 50% of traffic redundancy

As an example, we refer to Fig. 3.2a with two traffic demands $D^{0,5} = 20$ Gb and $D^{10,15} = 10$ Gb. Let a RE-router cost 30 Watts (see Section 3.5.1) and a link consume 200 Watts [CMN11]. Assume that 50% of the traffic is redundant and RE service is enabled at the router 6 and router 9. As shown in Fig. 3.2a, the traffic flows $0 \to 5$ and $10 \to 15$ are compressed at router 6 (and decompressed later at router 9) to $(10+\varepsilon)$ Gb and $(5+\varepsilon')$ Gb, respectively where $\varepsilon, \varepsilon'$ denotes the total size of the signatures used for each flow. In reality, each signature is only a few bytes in size [GC07], therefore $\varepsilon, \varepsilon'$ are small and the routing in Fig. 3.2a is feasible without any congestion. As a result, the routing in Fig. 3.2a is feasible and the GreenRE solution allows 10 links to be in sleep mode while enabling 2 RE-routers. Energy saving can be computed as $(10 \times 200 - 2 \times 30) = 1940$ Watts. It is noted that, in some extreme cases, GreenRE even helps to find feasible routing solution meanwhile it is impossible for the classical EAR. For example, if we add a third demand from router 0 to 1 with volume 20 Gb, then Fig. 3.2b is a feasible solution. However, without RE-routers, no feasible solution is found because there is not enough capacity to route all the three demands.

3.4.1 Mixed Integer Linear Program (MILP) Formulation

The GreenRE model can be formulated as MILP. We present a network topology as an undirected graph $G = (V, E)$. The set of nodes V describe routers and the edges $(u, v) \in E$ describe connections between those routers. We note $N(u)$ as a set of neighbor nodes of u in the graph G. For each link $(u, v) \in E$, we use a binary variable x_{uv} to determine if the link is used or not. If link (u, v) is active, two network interfaces at router u and router v are enabled, this consumes PE_{uv} Watts. We define γ^{st} as the percentage of unique (non redundant) traffic. For example, with 40% of redundancy ($\gamma^{st} = 0.6$), instead of sending a traffic demand 10 Gb, we are sending only $(6 + \varepsilon)$ Gb after removing redundancy. For simplicity, since ε is small, we can ignore it in the formulation and a traffic flow from which redundancy has been removed is called a *compressed flow*. It is noted that, the notion γ^{st} only captures the intra-flow redundancy (and not the inter-flow redundancy as presented in [AGA$^+$08]). We note f^{st}_{uv} (resp. g^{st}_{uv}) be the fraction of normal flow (resp. compressed flow) on edge (u, v) corresponding to the demand (s, t) flowing from u to v. We define a binary variable w_u which is equal to 1 if router u performs RE (called RE-router and it consumes additional PN_u Watts). For ease of reading, we recall the meaning of notations in following table:

$G = (V, E)$	a network with a set of routers V and a set of links E
$N(u)$	a set of neighbor nodes of u in the network
C_{uv}	capacity of the link (u, v)
μ	maximum link utilization
PE_{uv}	power consumption of the link (u, v)
PN_u	power consumption of the RE-router u
$\mathcal{D}$	a set of all demands
D^{st}	volume of the demand (s, t)
γ^{st}	percentage of unique traffic of the demand (s, t)
x_{uv}	binary variable indicates if a link (u, v) is used or not
w_u	binary variable indicates if RE is enable at router u or not
f^{st}_{uv}	fraction of normal flow from s to t on edge (u, v)
g^{st}_{uv}	fraction of compressed flow from s to t on edge (u, v)

Table 3.1: Summary of notations

We consider three different scenarios of the problem: (1) all routers on the network can perform RE, we can enable or disable RE service on routers; (2) only a predefined set of routers on the network have RE capability, other routers are normal routers and (3) there is a limited number of RE-routers, the network operators should find where to place them to increase energy efficiency for the network. We formulate the three scenarios of the GreenRE problem as follows:

3.4.1.1 Scenario 1: All Routers are RE-capable Routers

$$\min \sum_{uv\in E} PE_{uv}x_{uv} + \sum_{u\in V} PN_u w_u \tag{3.1}$$

$$\text{s.t.} \sum_{v\in N(u)} \left(f^{st}_{vu} + g^{st}_{vu} - f^{st}_{uv} - g^{st}_{uv}\right) = \begin{cases} -1 & \text{if } u = s, \\ 1 & \text{if } u = t, \; \forall u \in V, (s,t) \in \mathcal{D} \\ 0 & \text{else} \end{cases} \tag{3.2}$$

$$\sum_{v\in N(u)} (g^{st}_{uv} - g^{st}_{vu}) \leq w_u \qquad \forall u \in V, (s,t) \in \mathcal{D} \tag{3.3}$$

$$\sum_{v\in N(u)} (g^{st}_{vu} - g^{st}_{uv}) \leq w_u \qquad \forall u \in V, (s,t) \in \mathcal{D} \tag{3.4}$$

$$\sum_{(s,t)\in\mathcal{D}} D^{st}\left(f^{st}_{uv} + f^{st}_{vu} + \gamma^{st}(g^{st}_{uv} + g^{st}_{vu})\right) \leq \mu C_{uv}x_{uv} \quad \forall uv \in E \tag{3.5}$$

$$f^{st}_{uv}, \quad g^{st}_{uv} \in \; [0,1], \qquad w_u, \quad x_{uv} \in \{0,1\} \tag{3.6}$$

The objective function (3.1) is to minimize the power consumption of the network represented by the number of active links and RE-routers. Constraints (3.2) establish flow conservation constraints. Constraints (3.3)-(3.4) are used to determine whether RE service is enabled at router u or not. If it is not ($w_u = 0$), the router u only forwards flows without compression or decompression, then the amount of compressed flows incoming and outgoing the router u is unchanged. It is noted that if a flow is compressed, it needs to be decompressed somewhere on the way to its destination. This requirement is implicitly embedded in the constraints (3.4). For instance, assume that a destination node t is not a RE-router ($w_t = 0$). When a compressed flow g^{st} reaches its destination, because t is the last node on its path, the flow can not be decompressed. Consider the constraints (3.4), we have $u = t$, then $\sum_{v\in N(u)} g^{st}_{vt} > 0$ (the compressed flow enters node t) and $\sum_{v\in N(u)} g^{st}_{tv} = 0$ (t is the destination node). Therefore, the constraint (3.4) is violated and the flow should be decompressed before or at least at the destination node ($w_t = 1$). We consider an undirected link capacity model [RKOW11] in which the capacity of a link is shared between the traffic in both directions. We use constraints (3.5), where μ denotes the link utilization in percentage, to limit the available capacity of a link.

3.4.1.2 Scenario 2: a Predefined Set of RE-capable Routers

We define the following constraints:

$$w_u = 0 \qquad \forall u \notin V', \quad V' \subset V, \tag{3.7}$$

where V' is a predefined subset of routers that have RE-capability, we force all other routers to be normal routers ($w_u = 0$). By adding (3.7) to the first scenario (3.1)–(3.6), we have the second scenario of the GreenRE problem.

3.4.1.3 Scenario 3: a Limited Numbers of RE-routers

We add the following constraints to the first scenario (3.1)–(3.6):

$$\sum_{u \in V} w_u \leq M, \tag{3.8}$$

where M is the maximum number of RE-routers that can be placed on the network. By using constraints (3.8), we allow any router to perform RE. However, the total number of RE-routers on the network should be less than M.

3.4.2 Extended Cutset Inequalities for GreenRE problem

The content of this Section is from the paper [KPT13]. We enhance the MILP formulation of the GreenRE model by deriving cutting planes to speedup the MILP resolution. This enhancement can be applied for all the three above scenarios.

3.4.2.1 Valid Inequalities

The authors in [RKOW11] have studied valid inequalities for the capacitated network design problem. Following their work, we present inequalities for strengthening the GreenRE formulation. Given a set $S \subset V$, the total demand, which needs to be routed between S and $V \setminus S =: \overline{S}$ is denoted by

$$D^S := \sum_{v \in S} \sum_{w \in \overline{S}} D^{vw} + \sum_{v \in \overline{S}} \sum_{w \in S} D^{vw}.$$

Further, let $\delta(S, \overline{S})$ be the corresponding cut between both sets. Now, the well known cutset inequality for network design [Ata02, MMV93] can be adapted.

Theorem 1. *Let $S, \overline{S} \subset V$ be a partition of V. Then the **cutset** inequality*

$$\sum_{uv \in \delta(S,\overline{S})} x_{uv} \geq \left\lceil \frac{D^S}{c\beta} \right\rceil. \tag{3.9}$$

holds for GreenRE where $\beta = 1/\gamma$ (in (3.5)).

This inequality assumes that at least one RE-router is available in each of the two subsets. Assuming the contrary, we could increase the right-hand side. The following result takes the actual number of RE-routers into account.

Theorem 2. *Let $S, \overline{S} \subset V$ be a partition of V. Then the **extended cutset** inequality*

$$\left(\left\lceil \frac{D^S}{c} \right\rceil - \left\lceil \frac{D^S}{c\beta} \right\rceil \right) \sum_{u \in S} w_u + \sum_{uv \in \delta(S,\overline{S})} x_{uv} \geq \left\lceil \frac{D^S}{c} \right\rceil \tag{3.10}$$

holds for GreenRE.

Proof. Let S be given. Since w_u is binary, we distinguish two cases:

Case 1: Let $w_u = 0$ for all $u \in S$. Then (3.10) becomes

$$\sum_{uv \in \delta(S,\overline{S})} x_{uv} \geq \left\lceil \frac{D^S}{c} \right\rceil,$$

which is equivalent to the cutset inequality in absence of RE-routers.

Case 2: Let $\sum_{u \in S} w_u \geq 1$. We obtain

$$\begin{aligned} & \left(\left\lceil \tfrac{D^S}{c} \right\rceil - \left\lceil \tfrac{D^S}{c\beta} \right\rceil\right) \textstyle\sum_{u \in S} w_u + \sum_{uv \in \delta(S,\overline{S})} x_{uv} \\ \geq \quad & \left(\left\lceil \tfrac{D^S}{c} \right\rceil - \left\lceil \tfrac{D^S}{c\beta} \right\rceil\right) + \textstyle\sum_{uv \in \delta(S,\overline{S})} x_{uv} \geq \left\lceil \tfrac{D^S}{c} \right\rceil \end{aligned}$$

which holds, because of the cutset inequality (3.9). □

Comparing inequality (3.9) with (3.10), we conclude: both inequalities are equal if exactly one RE-router is deployed (in S). If no RE-router is deployed, the latter one strictly dominates the first and vice versa if more than one RE-router is available. In fractional solutions, no dominance relation can be given: the latter inequality has a weaker left-hand side while the first inequality has a weaker right-hand side. If S contains RE-routers but $\overline{S}$ does not, an exchange of S and $\overline{S}$ yields the stronger inequality.

3.4.2.2 Recognizing violated Inequalities

Employing all inequalities for all possible cuts in the GreenRE formulation is not a realistic option. In the following, we will present a straightforward approach for separating these inequalities via an integer linear program generalizing the separation of cutset inequalities. While the objective function 'rebuilds' the extended cut inequality (3.10) for the current LP solution (w^*, x^*), we need the following variables:

For each $v \neq w \in V$, let $z_{vw} \in \{0,1\}$ denote, whether v and w are in separate sides of the cut. Further, let $d, d_\beta \in \mathbb{Z}_{\geq 0}$, which represent the values $\lceil \frac{D^S}{c} \rceil$ and $\lceil \frac{D^S}{c\beta} \rceil$, respectively. For every node $v \in V$, $\alpha_v \in \{0,1\}$ denotes whether v is in S or in $\overline{S}$. Finally, given a sufficiently large constant $M \in \mathbb{N}$, for $k = 1, \ldots, M$, and $v \in V$, let $\alpha_v^k \in \{0,1\}$ denote if k equals $\lceil \frac{D^S}{c} \rceil - \lceil \frac{D^S}{c\beta} \rceil$ and v is in S, i.e., α_v^k is an enumeration of all possible coefficients of the RE-router variable coefficients in the extended cut. Consequentially, we have that

$$\sum_{k=1}^{M} k \alpha_v^k w_v^* = \left(\left\lceil \frac{D^S}{c} \right\rceil - \left\lceil \frac{D^S}{c\beta} \right\rceil\right) \sum_{v \in V} \alpha_v w_v^* = \left(\left\lceil \frac{D^S}{c} \right\rceil - \left\lceil \frac{D^S}{c\beta} \right\rceil\right) \sum_{v \in S} w_v^*.$$

Then the problem of finding a violated extended cut can be written as an extension of the triangle-formulation of the cut-polytope [BM86] as

$$
\begin{aligned}
\min \ & \sum_{k=1}^{M} k w_v^* \alpha_v^k + \sum_{vw \in E} x_{vw}^* z_{vw} - d \\
\text{s.t.} \ & -1+\varepsilon \le \frac{1}{c} \sum_{v\in V} \sum_{\substack{w\in V\\ w\neq v}} D^{vw} z_{vw} - d \le 0 && && (3.11)\\
& -1+\varepsilon \le \frac{1}{c\beta} \sum_{v\in V} \sum_{\substack{w\in V\\ w\neq v}} D^{vw} z_{vw} - d_\beta \le 0 && && (3.12)\\
& z_{vw} + z_{uw} + z_{uv} \le 2 && \forall\ \{u,v,w\} \subset V && (3.13)\\
& z_{vw} + z_{uw} \ge z_{uv} && \forall\ \{u,v,w\} \subset V && (3.14)\\
& \alpha_v + \alpha_w \le 2 - z_{vw} && \forall\, v,w \in V, v \neq w && (3.15)\\
& \alpha_v + \alpha_w \ge z_{vw} && \forall\, v,w \in V, v \neq w && (3.16)\\
& \sum_{k=1}^{M} \alpha_v^k = \alpha_v && \forall\, v \in V && (3.17)\\
& M(1-\alpha_v) + \sum_{k=1}^{M} k\alpha_v^k \ge d - d_\beta && \forall\, v \in V && (3.18)\\
& z_{vw}, \alpha_v, \alpha_v^k \in \{0,1\}, d, d_\beta \in \mathbb{Z}_{\ge 0}
\end{aligned}
$$

In this model, the inequalities (3.13), (3.14) establish a feasible cut and the inequalities (3.11) and (3.12) recognize the rounded traffic across this cut. (3.15) and (3.16) determine which nodes are within the one side (S) of the cut, and the inequalities (3.17) - (3.18) choose the correct α_v^k values for each α_v.

If the resulting objective value is strictly smaller than zero, the optimal solution of the corresponding MIP describes a partition of V, violating an extended cutset inequality (the z_{vw} correspond to the edge variables and the α_v to the RE-router variables in S). If the contrary holds, none of these extended cuts is violated.

This integer program can be adapted to separate cutset inequalities (3.9) easily by omitting the unnecessary parts (d, α_v, α_v^k) and restricting to the constraints (3.13), (3.14) and (3.12).

3.4.3 Heuristic Algorithm

Energy-aware routing problem is known to be NP-Hard [GMMO10, KPT13]. Solving the MILP to find optimal solution is time consuming and it only works for small network. We therefore present in this section two heuristic algorithms to quickly find feasible solutions, called *H-GreenRE* (presented in [GMPR12]) and H_{ILP}-*GreenRE* (presented in [GMPR14]).

3.4.3.1 H-GreenRE

In the first step of the heuristic algorithm, we assume that all routers on the network are RE-routers. Therefore volumes of traffic demands are virtually decreased to $Dr_{st} = \gamma D_{st},\ \gamma \in (0,1]$. Based on this, we try to find a feasible routing solution so that it minimizes the number of active links. Then for the second step, we try to disable RE on routers to save energy while we guarantee that the routing found in the previous step is still feasible.

Algorithm 1: Finding a feasible routing
Input: An undirected weighted graph $G = (V, E)$ where each edge e has a capacity C_e, a residual capacity R_e, an initial metric w_e and a set of demands $Dr_{st} \in D$.

```
∀$e \in E, R_e = C_e,\ w_e =$ total number of demands
Sort the demands in random order
while $Dr_{st}$ has no assigned route do
    compute the shortest path $SP_{st}$ with the metric $(w_e)$
    assign the routing $SP_{st}$ to the demand $Dr_{st}$
    $\forall e \in SP_{st}, R_e = R_e - Dr_{st}, w_e = w_e - 1$
end
return the routing (if it exists) assigned to the demands in $D$
```

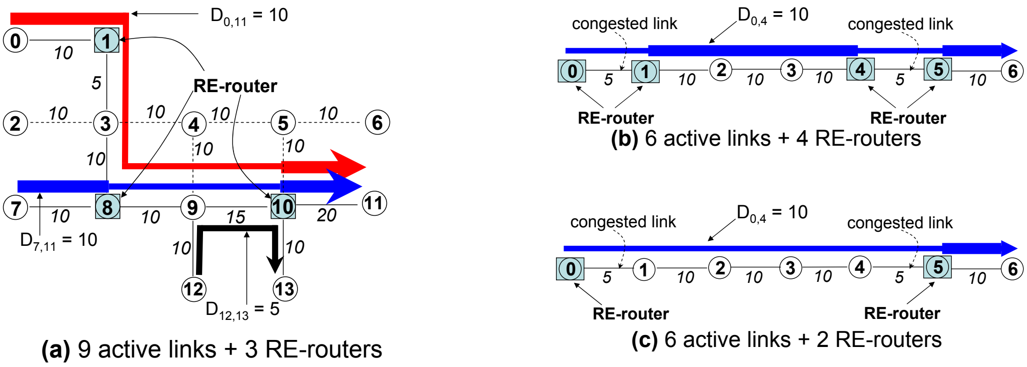
Starting with the *Algorithm 1*, we compute a feasible routing for the RE-demands. Initially, all links on the network are set up with the same metric w_e which is equal to the total number of demands. We compute the shortest path for each demand with the metric w_e on links. Then, the links that have carried the shortest path is updated with metric: $w_e = w_e - 1$. Using this metric in the shortest path, we implicitly set high priority to reuse links that have already been selected. Then, the *Algorithm 2 - Step 1* is used to remove in priority links that are less loaded. C_e/R_e is used as the load on a link where R_e is the residual capacity on link e when previous demands have been routed.

In *Step 2*, we use the routing solution found in the *Step 1* as the input of the algorithm. Then, we consider all the traffic demands as normal demands without RE. Hence, some links can be congested because the total traffic volume of demands may exceed the link capacity. The heuristic we use in *Step 2* is based on following observations:

(a) Which demand to perform RE first? In Fig. 3.3a, we can see that when performing RE for $D_{0,11}$ on router 1 and 10, the amount of traffic passing all the congested links (links (1, 3), (8, 9) and (9, 10)) is also reduced. Then, the heuristic in step 2 decides to perform RE for this flow first. Assume that the redundancy factor $\gamma = 0.5$, then the links (8, 9) and (9, 10) are still congested. After removing $Dr_{0,11}$, the available capacity of links (8, 9) and (9, 10) are 5 and

Algorithm 2: *Input:* An undirected weighted graph $G = (V, E)$ where each edge $e \in E$ has a capacity C_e and a residual capacity R_e

```
Step 1 - Removing less loaded links:
while edges can be removed do
    remove the edge $e$ that has not been chosen and has smallest value $C_e/R_e$
    compute a feasible routing with the Algorithm 1
    if no feasible routing exists, put $e$ back to G
end
return the feasible routing if it exists.
Step 2 - Enabling set of RE-routers:
Consider normal demands (without RE) and routing solution in Step 1
while network is congested do
    find demands to perform RE first (details in part (a))
    enable set of RE-routers using end points congestion (details in part (b))
end
```



(a) 9 active links + 3 RE-routers

(b) 6 active links + 4 RE-routers

(c) 6 active links + 2 RE-routers

Figure 3.3: Congested links and RE-routers

10, respectively. Applying the same rule, the next demand to perform RE is $D_{7,11}$. Then, there is no congested link on the network since the link (9, 10) is also released from congestion. Finally, only the routers 1, 8 and 10 are needed to enable RE. In summary, the algorithm will perform RE for the flows that pass through most of the congested links first.

(b) End Points Congestion: in Fig. 3.3b and Fig. 3.3c, the demand (0,6) has traffic volume of 10 and the number on links indicates link capacity. Therefore, the two links (0, 1) and (4, 5) are congested. Hence, a naive solution is to enable RE at the two end-point routers of each congested link as shown in Fig. 3b. However, a better solution with less RE-routers should be to enable RE only at the starting (router 0) and ending point (router 5) of all the congested links (Fig. 3c). In summary, the algorithm will look for the longest congested part of the flow to enable RE-routers.

The main problem of *H-GreenRE* is that, to find a feasible routing, we pick up demand one by one and try to route it using shortest path routing. Thus, in case the feasible routing is not a shortest path, *H-GreenRE* can not find it. We therefore present an improvement version of the heuristic, called H_{ILP}*-GreenRE*.

3.4.3.2 H_{ILP}-GreenRE

Using the ILP formulation, H_{ILP}*-GreenRE* tries all the possibilities to find a feasible routing if any, thus it is more efficient than the *H-GreenRE* (see Section 3.5.3.1). In summary, the heuristic algorithm has two steps: the first step is to use as few active links as possible, and then we minimize the number of RE-routers in the second step.

Algorithm 3: *Inputs:* A graph $G = (V, E)$ with link capacity C_e, a set of traffic demands and non-redundant rates.

```
1  Step 1 - Minimize number of active links by removing low loaded links:
2  Find a feasible routing solution using the MILP called $P\_current$;
3  Let $S$ be an ordered list initialized with the links of $G$ sorted by increasing
     traffic load in $P\_current$;
4  Let $R := \emptyset$ be the set of links that cannot be removed;
5  repeat
6      $e := S.lowest\_loaded\_link()$ such that $e \notin R$;
7      $S := S \backslash \{e\}$;
8      if a feasible routing P_new on $E \backslash \{e\}$ is found then
9          if $P\_new$ has less active links than $P\_current$ then
10             $P\_current := P\_new$;
11             $S :=$ list of links sorted by increasing traffic load in $P\_new$;
12             $E := E \backslash \{e\}$;
13         end
14     else
15         $R := R \cup \{e\}$;
16     end
17 until $(S = \emptyset)$ or $(R = S)$;
18 Return the final feasible routing solution (if any);
19 Step 2 - Find feasible solution minimizing the number of RE-routers on the
     set of active links $E$ found in Step 1.
```

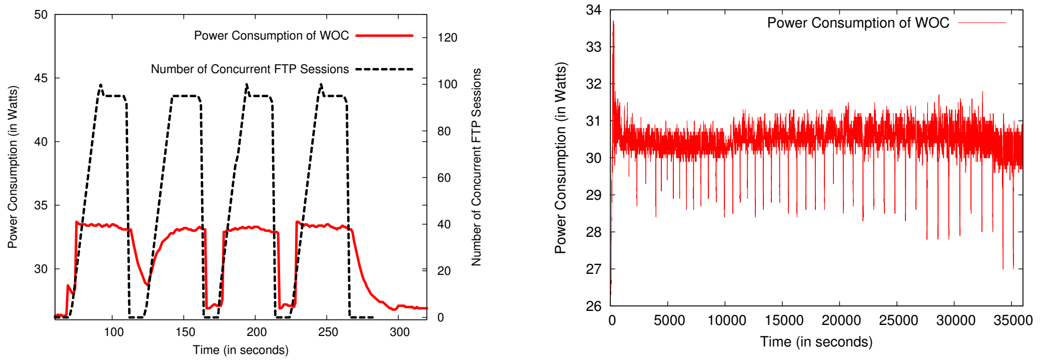
Step 1 of Algorithm 3 is a constraint satisfaction problem returning a feasible routing solution. We use the same framework for the three scenarios of the GreenRE problem. For details, to find feasible solutions ($P_{current}$ - line 2 and P_{new} - line 8), we set the objective function to *min* 0 and use the constraints (3.2)–(3.6) for scenario 1. Similarly, scenario 2 (resp. scenario 3) uses the constraints (3.2)–(3.7) (resp. constraints (3.2)–(3.6), (3.8)) and the objective is *min* 0. In each round of the algorithm, we try to remove a link with low load (line 6 - 7) and then find a new feasible routing (P_{new} - line 8) using less active links. The idea behind this algorithm is that we try to put into sleep mode the low loaded links and to accommodate their traffic on other links in order to reduce the total number of active links. Observe that unused links (i.e. links that do not carry traffic) are not considered in the set S since the removal of such a link will result in a routing P_new equal to $P\_current$. To further reduce the computation time, we can consider additional heuristic. For instance, while removing a low loaded link (line 6 - 7), we can also set the variable x_{uv} associated to a heavily loaded link to 1 so that it can speed up the resolution for finding P_{new} (line 8). Indeed, such high loaded link will certainly be part of the final solution. Since we relax the objective function and the goal is just to justify whether a set of constraints is feasible or not, it is quite fast to find $P_{current}$ and P_{new}. In our evaluations, the execution time of Algorithm 3 (including the two steps) is less than one hour for the tested network topologies.

After *Step 1*, if a feasible routing is found, and so a set of active links, we proceed to *Step 2* to minimize the number of enabled RE-routers. More precisely, we use again the MILP formulation (of the scenario we want to solve) in which the objective function is set to min $\sum_{u \in V} w_u$. Furthermore, we set all binary variables associated to active links to 1 and the others to 0 (this speed-up the resolution of the MILP).

3.5 Experiment and Evaluation Results

3.5.1 Energy Consumption with WOC

Several results of bandwidth saving using WOC can be found in [GC07]. We have also performed experiments on the network platform of the project Network Boost at Orange Labs (the full figure of the test-bed can be found in [Report11]). We installed two WOCs, each at the access link of the two sites (let's call them site A and site B). These two sites are connected via a backbone composed of 4 routers. We setup FTP connections for uploading files from site A to site B. As shown in



(a) With concurrent FTP sessions

(b) With increasing size of uploaded files

Figure 3.4: Power consumption of the WOC

Fig. 3.4a, power consumption of the WOC is increased (from 26 Watts to 34 Watts) with the number of concurrent FTP sessions. For the next experiment, we keep only one FTP session and let the WOC perform RE for 10 hours in which the sizes of uploaded files are increased. The results show that the WOC consumes around 30 Watts on average (Fig. 3.4b). Therefore, for sake of simplicity, we use an average value of power consumption (30 Watts) to represent additional cost for the router to perform RE.

3.5.2 Cutset Inequalities vs. Standard MIP-solution Process

In this section, we want to show benefits of incorporating the inequalities (3.9) and (3.10) within the (standard) MIP-solution process. In this preliminary study, for every LP solution, the inequalities (3.9) are separated. Only if no violated cut is found, the inequalities (3.10) are separated. We used modified instances of the SNDlib [OWPT10]. Taking care of the great variety within that library, all demands have been scaled such that a routing (without compression) is feasible at a capacity of $10,000$ per link and infeasible at a capacity of $9,000$. In this study, we used the instances Abilene (scaling factor: 102), Atlanta (2.6), Dfn-bwin (4.5), France (1.1) and Polska (0.17).

All computations have been done with CPLEX 12.4, with CPU-/thread-usage limited to one. For obtaining clear results, CPLEX internal cutting-planes have been disabled. We report on progress after the root node and compare ourselves to the usage of plain CPLEX (with the same settings). All scenarios have been tested with capacities $5,000$ (halved), $10,000$ (normal) and $20,000$ (doubled) and a compression factor of $\gamma = 0.5$. The price for an edge has been determined as $200W$ [CMN11] and the price of an RE-router as $30W$. Success of the separation routine is measured by the relatively closed gap at the root, i.e. let DB denote the LP relaxation, DB_s the best dual bound obtained by our separation approach and PB the best primal bound available. Then we define the gap closed as $GC := \frac{DB_s - DB}{PB - DB}$. Clearly, an improvement is given, as soon as this value is positive and a higher value corresponds to a bigger improvement.

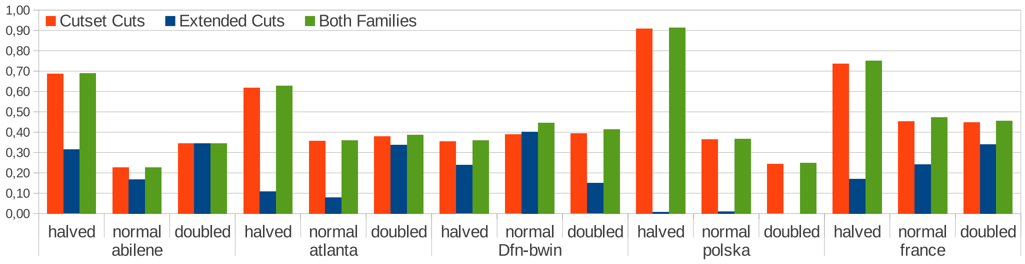


Figure 3.5: Relative Gap Closing

The results presented in Figure 3.5 are throughout positive. The gain from separating one or both of the two classes of inequalities is shown. The total improvement amounts to an average gap closed of 46.1%. This improvement was achieved by an average amount of 45 cuts per instance, 33 cutset inequalities (3.9) and 12 extended cuts (3.10). While the relation between the amounts of both cutting plane families is clear by the lazy separation approach of the extended cuts, it appears that the extended cuts (3.10) can still improve on the cutset inequalities (3.9). However, the success and the relation between both classes is highly dependent on the underlying network topology (and the edge/RE-router prices). For example the Polska (doubled) instance seems to be very unfortunate for extended cuts, while the Dfn-bwin (normal) seems to favor exactly these. On the Abilene (normal) instance, both behave equal.

The drawback of this separation approach is that it is time consumption. The time needed for finding violated cuts and re-optimizing the linear relaxation is significant. Depending on the amount of cuts found, this procedure can amount to an increase of time consumption of more than 100%.

3.5.3 Computation Results with GreenRE

We solve the GreenRE model with IBM CPLEX 12.4 solver [IBM]. All computations were carried out on a computer equipped with 2.7 Ghz Intel Core i7 and 8 GB RAM. We studied ten classical real network topologies extracted from SNDLib [OWPT10]. Their sizes span from 15 to 54 nodes and from 22 to 89 edges, as summarized in Table 3.2. According to the results of the works mentioned in Section 3.3.2, we use redundancy rates equal to 50% (high redundancy, $\gamma = 50\%$), 25% (medium redundancy, $\gamma = 75\%$) and 10% (low redundancy, $\gamma = 90\%$). For worst-case scenario and for comparison with previous work [GMMO10, GMPR12], all links are set up with the same capacity C and the demands are all-to-all (one router has to send traffic to all remaining routers on the network) with the same traffic volume D for each demand.

3.5.3.1 Heuristic H-GreenRE vs. H_{ILP}-GreenRE

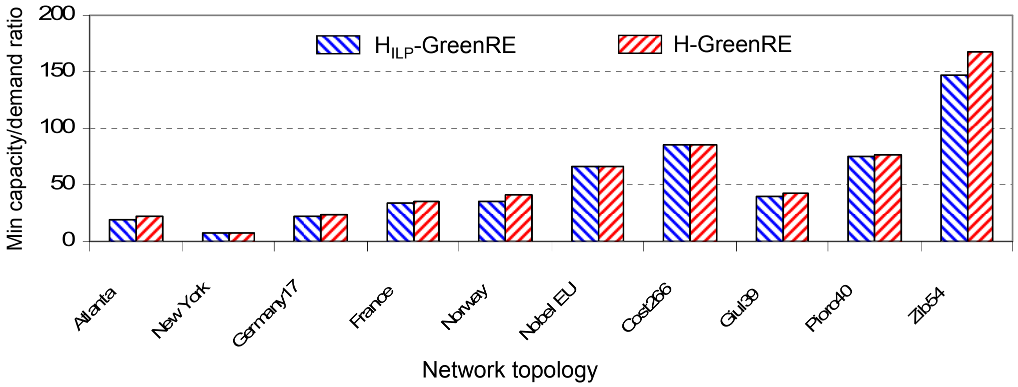


Figure 3.6: Comparison of λ_{minRE} between H_{ILP}-GreenRE and H-GreenRE

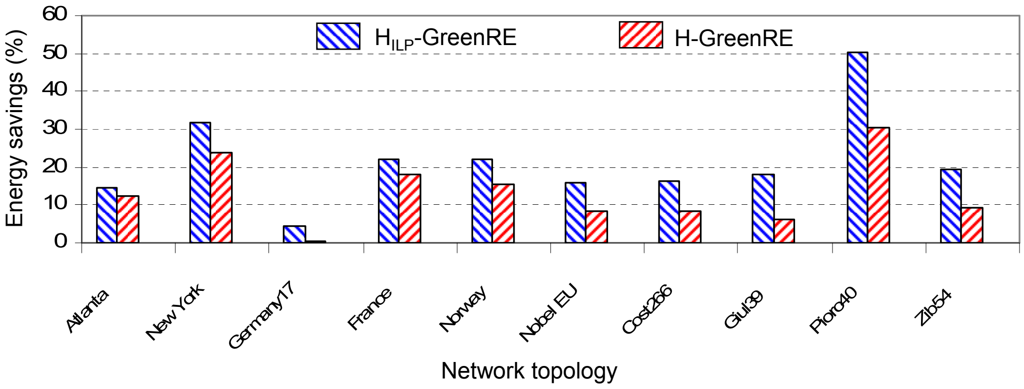


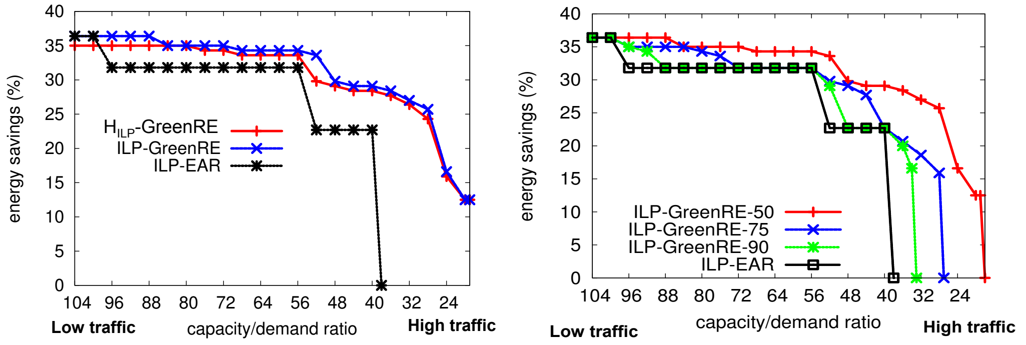
Figure 3.7: Comparison of energy saving between H_{ILP}-GreenRE and H-GreenRE

To compare between *H_{ILP}-GreenRE* and *H-GreenRE* heuristic algorithms, two evaluation scenarios (with $\gamma = 50\%$) have been done for the ten network topologies (we sort the networks in increasing order of the number of nodes).

First, we find the minimum values of capacity/demand ratio λ_{minRE} that allow for each heuristic algorithm to find a feasible routing solution *with the support of RE-routers*. Note that, λ represents the level of traffic load on the network. Small value of λ means that the traffic load on the network is high (e.g. traffic at peak hours), thus it is hard to find feasible solution because of the lack of capacity (refer to the example in Fig. 3.2b). Therefore, the heuristic algorithm that can find feasible routing with smaller value of λ_{minRE} is the better one. To compute λ_{minRE}, we first fix the demand value, e.g. $D = 1$. Then, starting with a large capacity value, e.g. $C = 1000$, we decrease the value of C and test the heuristic until we get the minimum value of C that is still possible to find a feasible solution. Let's call this value is C_{min}, then we have $\lambda_{minRE} = C_{min}$. Fig. 3.6 shows that H_{ILP}-GreenRE can find feasible solutions with smaller values of λ_{minRE} than H-GreenRE. For example, for the Atlanta network, H_{ILP}-GreenRE finds a solution with $\lambda_{minRE} = 19$ while H-GreenRE is with $\lambda_{minRE} = 22$: that is, for example, for a link capacity of 10 Gbit/sec, the first heuristic succeeds in routing an all-to-all demand of 10/19 =0.53 Gbit/sec for each demand and the second heuristic, a demand of only 10/22 = 0.45 Gbit/sec. In summary, H_{ILP}-GreenRE finds feasible solutions close to the lower bounds of λ_{min} found in [GMMO10]. The best improvement is on Zib54 network: $\lambda_{minRE} = 147$ (for H_{ILP}-GreenRE) in comparison with $\lambda_{minRE} = 168$ (for H-GreenRE).

We show next the energy saving for the ten networks. We use the value of λ_{minRE} that allows for H-GreenRE to find feasible routing solution for each network (the second column in Fig. 3.6). If a network has dense links, there are more chances to redirect traffic and put links into sleep mode, thus more energy can be saved. As shown in Fig. 3.7, H_{ILP}-GreenRE again outperforms H-GreenRE for all the networks. Energy efficiency can be increased from 2% (Atlanta network) to 19.8% (Pioro40 network).

3.5.3.2 Energy Saving for Atlanta Network



(a) Heuristic vs. ILP-GreenRE vs. ILP-EAR

(b) with different RE rates

Figure 3.8: Evaluation results for Atlanta network

In this subsection, we present evaluation results for Atlanta network. In Fig. 3.8a, with the same redundancy rate ($\gamma = 50\%$), we vary capacity/demand ratios and compare between the *ILP-EAR without* RE-routers (given by [GMMO10]), the ILP with RE-routers (the formulation (3.1)-(3.6): *ILP-GreenRE*) and the heuristic with RE-routers (H_{ILP}-*GreenRE*). Even for small network like Atlanta, CPLEX also takes some hours to find an optimal solution when the capacity/demand ratios are high (e.g. $\lambda \geq 48$). It is noted that when $\lambda < 48$, it is possible to find an optimal solution within one hour. We limit the solving time to one hour for all instances of Atlanta network corresponding to different capacity/demand ratios. In average, the optimality gap is within 10% for all the best solutions. The heuristic is quite fast, it takes less than 10 seconds to find a solution. The $x-$axis in Fig. 3.8a represents the capacity/demand ratio λ and the $y-$axis is energy saving in percentage. As shown in Fig. 3.8a, without RE-router (ILP-EAR), there is no feasible routing solution and hence, no energy is saved if $\lambda < 38$. When λ increases, links have more bandwidth to aggregate traffic, the solutions with and without RE-router converge to the same amount of energy saving. In general, the heuristic with RE-routers works well and approximates to the results of ILP-GreenRE (the max gap is 3.8%).

In Fig. 3.8b, we evaluate energy saving for Atlanta network with different level of redundancy. It is clear that when traffic redundancy is high, e.g. $\gamma = 50\%$, more traffic flows are aggregated and thus, more links can be turned off to save energy. Similarly, when $\gamma = 75\%$ and $\gamma = 90\%$ (corresponding to 25% and 10% of traffic redundancy), less energy can be saved. These remarks can be seen in Fig. 3.8b where the gaps between *ILP-GreenRE* and *ILP-EAR* are reducing when γ is increasing. It is noted that *ILP-GreenRE* should be at least as good as *ILP-EAR*. It is because the objective of *ILP-GreenRE* is to minimize energy consumption for the network. In case the redundancy elimination does not help to turn off more links, *ILP-GreenRE* does not enable RE service on router (even it helps to reduce the traffic load). Therefore, in the worst case scenario (redundancy rate is zero or

$\gamma = 100\%$), *ILP-GreenRE* has no RE-enabled router and the routing solution is the same as in *ILP-EAR*.

3.5.3.3 Energy Saving for the Ten Classical Networks

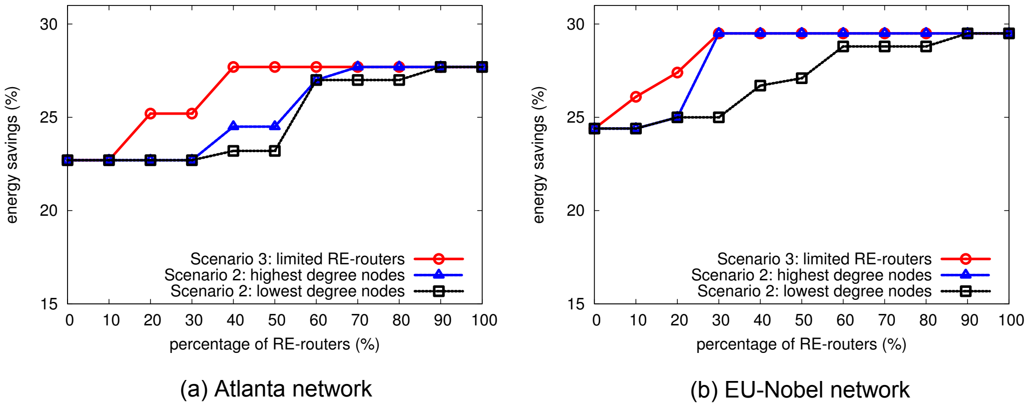
Network	$\mid V \mid$	$\mid E \mid$	λ_{min}	Traffic volume (capacity/demand ratio λ)					
				with RE-router			without RE-router		
				λ_{min}	$2\lambda_{min}$	$3\lambda_{min}$	λ_{min}	$2\lambda_{min}$	$3\lambda_{min}$
Atlanta	15	22	38	27.7%	34.3%	36.4%	0%	32%	36%
New York	16	49	15	52.2%	62.9%	65.8%	2%	59%	63%
Germany17	17	26	44	30.6%	36.7%	37.3%	0%	35%	39%
France	25	45	67	39.2%	43.4%	46%	0%	42%	44%
Norway	27	51	75	37.7%	45.6%	47.8%	12%	43%	47%
Nobel EU	28	41	131	29.2%	33.1%	34.2%	12%	32%	34%
Cost266	37	57	175	30.6%	35%	36.3%	3.5%	32%	35%
Giul39	39	86	85	42.5%	50.5%	53.3%	0%	45%	50%
Pioro40	40	89	153	50.5%	53.7%	55.2%	0%	53%	54%
Zib54	54	80	294	27.5%	30.8%	32.8%	0%	30%	33%

Table 3.2: Gain of energy consumption (in %)

We present in Table 3.2 energy gain for ten classical network topologies using H_{ILP}-GreenRE and H-EAR - the heuristic *without* RE-routers found in [GMMO10]. We use λ_{min} here which are the smallest value of capacity/demand ratios that allow to find feasible route for all the demands *without RE-router* (found in [GMMO10]). In the evaluation, a range of $\lambda = \{\lambda_{min}, 2\lambda_{min}, 3\lambda_{min}\}$ is used to represent high (e.g. traffic at peak hours), medium and low traffic load (e.g. traffic at night) on the networks. As shown in Table 3.2, with RE-routers, it starts to save a large amount of energy (in average 37%) even with $\lambda = \lambda_{min}$. Recall that routing with RE-routers is possible even with $\lambda < \lambda_{min}$ while no feasible solution is found without RE-router. When λ is large enough, it is not necessary to have RE-routers on the network, therefore both the solutions (with and without RE-router) converge to almost the same value of gains in energy saving.

3.5.3.4 Energy Saving for Scenario 2 and Scenario 3 of the GreenRE Problem

In this section, we evaluate energy saving of scenario 2 (a predefined subset of RE-capable routers) and scenario 3 (a limited numbers of RE-capable routers). We set link capacity and demand corresponding to λ_{min} in Section 3.5.3.3. The x-axis of Fig. 3.9 is the percentage of RE-capable routers on the network. For instance, with scenario 3, we find the routing solution that minimizes energy consumption while there are at most $(x \times |V|)$ RE-routers on the network. For scenario 2, we place $(x \times |V|)$ RE-capable routers on (1) highest degree nodes or (2) lowest degree nodes in graph G. As shown in Fig. 3.9, the scenario 3 always outperforms the scenario 2



(a) Atlanta network

(b) EU-Nobel network

Figure 3.9: Energy saving with limited RE-routers vs. a subset of capable RE-routers

since it can find best positions to place RE-capable routers. For instance, in Atlanta network with a maximum of 6 RE-routers, the max gap is 4.5% and there are 4 RE-routers at the highest degree nodes and the two others are at the medium and the lowest degree nodes. Another important observation we found in the scenario 2 is that, placing RE-routers on high degree nodes gives better results in energy saving. It is because placing RE-capable routers on high degree nodes helps to reduce traffic load and gives more chances to redirect traffic on a few links, allowing other links on these nodes to sleep.

3.6 Conclusion

To the best of our knowledge, GreenRE is the first work considering redundancy elimination as a complementary help for energy-aware routing problem. We formulate the problem as Mixed Integer Linear Program and propose greedy heuristic algorithms. The evaluations on several network topologies show a significant gain in energy saving with GreenRE. For future work, we will consider a more realistic model in which data redundancy rates and traffic demand volumes fluctuate based on real life traffic traces. Moreover, we plan to study the inter-flow redundancy as it could further reduce network traffic.

3.7 Bibliography

[AGA+08] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker, *"Packet Caches on Routers: the Implications of Universal Redundant Traffic Elimination"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2008, pp. 219–230.

[AMAR09] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, *"Redundancy in Network Traffic: Findings and Implications"*, ACM Special Interest Group for the Computer Systems Performance Evaluation Community (SIGMETRICS), 2009, pp. 37–48.

[ASA09] A. Anand, V. Sekar, and A. Akella, *"SmartRE: an Architecture for Coordinated Network-wide Redundancy Elimination"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2009, pp. 87–98.

[Ata02] A. Atamtürk, *"On Capacitated Network Design Cut-set Polyhedra"*, Mathematical Programming **60** (2002), 425–437.

[BlueCoat] *BlueCoat: WAN Optimization*, http://www.bluecoat.com/.

[BM86] F. Barahona and A. Mahjoub, *"On the Cut Polytope"*, Mathematical Programming **36** (1986), 157–173.

[CMN11] L. Chiaraviglio, M. Mellia, and F. Neri, *"Minimizing ISP Network Energy Cost: Formulation and Solutions"*, IEEE/ACM Transaction in Networking **20** (2011), 463 – 476.

[CSB+08] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, *"Power Awareness in Network Design and Routing"*, IEEE International Conference on Computer Communications (INFOCOM), 2008.

[GC07] T. Jr. Grevers and J. Christner, *"Application Acceleration and WAN Optimization Fundamentals"*, Cisco Press, 2007.

[GMMO10] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy, *"Minimizing Routing Energy Consumption: from Theoretical to Practical Results"*, IEEE/ACM Green Computing and Communications (GreenCom), 2010, pp. 252–259.

[GMPR12] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, *"Minimization of Network Power Consumption with Redundancy Elimination"*, International Conferences on Networking (IFIP NETWORKING), 2012, pp. 247–258.

[GMPR14] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, *"Minimization of Network Power Consumption with Redundancy Elimination"*, Computer Communications (submitted), 2014.

[IBM] IBM ILOG, *CPLEX Optimization Studio 12.4.*

[KPT13] A. Koster, T. K. Phan, and M. Tieves, *"Extended Cutset Inequalities for the Network Power Consumption Problem"*, International Network Optimization Conference (INOC), Electronic Notes in Discrete Mathematics, 2013, pp. 69–76.

[Report11] www-sop.inria.fr/mascotte/rapports_stages/KhoaPhan_internship-2011.pdf.

[MMV93] T. Magnanti, P. Mirchandani, and R. Vachani, *"The Convex Hull of Two Core Capacitated Network Design Problems"*, Mathematical Programming **60** (1993), 233–250.

[MSB09] P. Mahadevan, P. Sharma, and S. Banerjee, *"A Power Benchmarking Framework for Network Devices"*, International Conferences on Networking (IFIP NETWORKING), 2009, pp. 795–808.

[OWPT10] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, *SNDlib 1.0 - survivable network design library*, Networks **55** (2010), no. 3, 276–286.

[Riverbed] http://www.riverbed.com/us/solutions/wan_optimization/.

[RKOW11] C. Raack, A. M. C. A. Koster, S. Orlowski, and R. Wessäly, *"On Cut-based Inequalities for Capacitated Network Design Polyhedra"*, Networks, vol. 57, 2011, pp. 141 – 156.

[SGG10] Y. Song, K. Guo, and L. Gao, *"Redundancy-aware Routing with Limited Resources"*, International Conference on Computer Communications and Networks (ICCCN), 2010, pp. 1–6.

[SW00] N. T. Spring and D. Wetherall, *"A Protocol-Independent Technique for Eliminating Redundant Network Traffic"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2000.

Chapter 4

Robust Optimization for GreenRE

Contents

The GreenRE model which we have presented in Chapter 3 allows us to improve energy efficiency of networks. In this chapter, we extend this study to take into account fluctuations in traffic demand volumes and redundancy rates. Using this extra knowledge on the dynamics of the traffic pattern, we are able to significantly improve energy efficiency for the network.

4.1 Publications

The first part of this chapter (Section 4.2) corresponds to *Robust Redundancy Elimination for Energy-aware Routing* by D. Coudert, A. Koster, T. K. Phan, and M. Tieves which has been accepted for publication in the proceedings of IEEE International Conference on Green Computing and Communications (GreenCom), 2013. In this first part, only uncertainty on redundancy elimination rates is considered. We then present an extended version to deal with the fluctuation of both traffic demand volumes and RE rates in the second part of this chapter (Section 4.3). This extension corresponds to *Robust Optimization for Energy-aware Routing with Redundancy Elimination* (in the proceedings of Algotel, 2014) and *Robust Energy-aware Routing* *with Redundancy Elimination* (submitted to Computers and Operations Research, 2014.) by D. Coudert, A. Kodjo, and T. K. Phan.

4.2 Redundancy Elimination Fluctuation

4.2.1 Introduction

In modern communication infrastructures, energy-consumption is one of the most critical aspects for designing network topologies. Routing has not only to be feasible with respect to congestion, but as energy efficient as possible. Therefore, the classical energy-aware routing (EAR) problem aims at minimizing the active elements of routers (the most influencing factor of energy consumption), while all traffic demands are routed without any overloaded links [CSB$^+$08, CMN11, GS03, ZYLZ10]. In chapter 3, we have introduced GreenRE, a combining model of EAR and Redundancy Elimination (RE) that increases energy efficiency of a backbone network.

Although solving the GreenRE model is already a complex task [KPT13], it does not take traffic redundancy fluctuations into account. Instead, each of the demands contains a constant factor of redundant traffic. This assumption may lead to infeasible or inefficient network designs, i.e., a high value of estimated traffic redundancy causes overloading, whereas using an underestimated value wastes energy saving.

The contribution of this section is an extension of the GreenRE model as state-of-the-art technique to include uncertainty of traffic redundancy as well. Therefore, a mean of dealing with uncertainties has to be chosen carefully. While a general worst-case analysis is inefficient in applications, the Γ-robustness concept [BS03] models uncertainties in a more realistic way. This technology-independent concept has already been successfully applied to, for example the network design problem under demand uncertainty [AABP07, KKR11]. Given a parameter $\Gamma \geq 0$, the problem considers any simultaneous deviation of at most Γ traffic pairs from their nominal traffic volumes.

In this section, we extend the GreenRE model by applying the idea of Γ-robustness to uncertain data redundancy. We propose GreenRobustRE - a model that includes uncertainty of redundancy elimination rates. Accordingly, contributions are structured as:

- In Section 4.2.3, we define and formulate the GreenRobustRE problem as mixed integer linear program. To the best of our knowledge, this is the first work considering robustness on redundancy elimination for traffic flows.

- In Section 4.2.4, we exemplarily evaluate energy saving for two networks based on real-life traffic traces and estimated redundancy fluctuation. The results show a significant increase of energy saving by the GreenRobustRE model, compared to previous models.

As central point of this chapter, we show the superiority of the GreenRobustRE model in both, being closer to reality (model wise - Section 4.2.3) and yielding

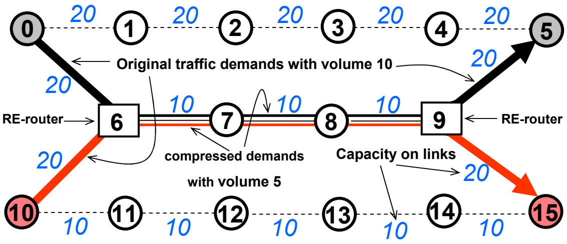


Figure 4.1: GreenRE routing with 50% of traffic redundancy: turn off 10 links, enable 2 RE-routers.

better solutions (Section 4.2.4). A representation as mixed integer linear program offers an accurate description of the potential of the proposed concept. For large networks, a more refined solution approach is necessary.

We start with a review of already known, related work (especially the GreenRE model). We repeat the complementary concepts of energy aware routing before presenting the GreenRE as a combination of both ideas. Furthermore, we introduce the Γ-robust optimization approach as the background of the Green-RobustRE problem. In following, we will explain the GreenRobustRE problem, concluding with exemplary computations and a conclusion/evaluation.

4.2.2 Background: An evolution of models

4.2.2.1 GreenRE Model

The GreenRE model is an extension of EAR, i.e., a combination of RE and EAR. In this model, the redundancy elimination technique virtually increases the capacity of the network. A drawback is that the caching process increases the energy consumption on routers. We have shown in [GMPR12], that a router performing RE consumes more energy than usual. This introduces a trade-off between enabling RE on routers (increasing their power consumption) and turning off links (saving their expenses), such that designing an optimal network topology is not trivial.

As proof of concept, we refer to Fig. 4.1. Let a RE-router cost 30 Watts [GMPR12] and a link consume 200 Watts [CMN11]. Assume that 50% of the traffic is redundant and RE-service is enabled at router 6 and router 9. Hence, all traffic flows passing between the routers 6, 7, 8, 9 can be compressed to 5 Gbps at router 6 and are decompressed to full size at router 9. So, the routing as shown in Fig. 4.1 is feasible (without any congestion). As a result, the GreenRE solution allows to turn off 10 links and enables 2 RE-routers which saves $(10 \times 200 - 2 \times 30) = 1940$ Watts.

More precisely, the GreenRE problem is defined on an undirected graph $G = (V, E)$, where C_e denotes the capacity of link $e \in E$. The set of demands is given by $\mathcal{D} = \{(s,t) \in V \times V : s \neq t\}$ and $D^{st} \geq 0$ denotes the amount of traffic requested from target t of source s. Let $PE_e, PN_u \geq 0$ be the power consumption of an active link / RE-router. The constant $\lambda^{st} \in [0, 1)$ denotes the percentage of traffic redundancy of a demand (s,t). Corresponding to λ^{st}, we define $\gamma^{st} := (1-\lambda^{st})$, which represents the percentage of unique (non redundant) traffic. For instance, for a 10 Gbps traffic demand with $\lambda^{st} = 40\%$ of redundancy, its volume can be reduced by GreenRE to $10\gamma^{st} = 6$ Gbps of non-redundant traffic. For simplicity, a traffic flow, from which redundancy has been removed, is called a *compressed flow.*

Binary variables x_{uv} and w_u denote the activated links / RE-routers. We use variables f^{st}_{uv}, $g^{st}_{uv} \geq 0$ $\forall (s,t) \in \mathcal{D}, uv \in E$ describing the fraction of normal and compressed flows of demand (s,t), routed directly from u to v. For ease of reading, we recall the meaning of notations in following table:

$G=(V,E)$	a network with a set of routers V and a set of links E
$N(u)$	a set of neighbor nodes of u in the network
C_e	capacity of the link e
μ	maximum link utilization
PE_e	power consumption of the link e
PN_u	power consumption of the RE-router u
$\mathcal{D}$	a set of all demands
D^{st}	volume of the demand (s,t)
γ^{st}	percentage of unique traffic of the demand (s,t)
x_e	binary variable indicates if a link e is used or not
w_u	binary variable indicates if RE is enable at router u or not
f^{st}_{uv}	fraction of normal flow from s to t on edge (u,v)
g^{st}_{uv}	fraction of compressed flow from s to t on edge (u,v)

Table 4.1: Summary of notations

We formulate the GreenRE model as follows:

$$\min \sum_{e \in E} PE_e x_e + \sum_{u \in V} PN_u w_u \tag{4.1}$$

$$\text{s.t.} \sum_{v \in N(u)} \left(f^{st}_{vu} + g^{st}_{vu} - f^{st}_{uv} - g^{st}_{uv}\right) = \begin{cases} -1 & \text{if } u = s, \\ 1 & \text{if } u = t, \\ 0 & \text{else} \end{cases}$$
$$\forall u \in V, (s,t) \in \mathcal{D} \tag{4.2}$$

$$\sum_{(s,t) \in \mathcal{D}} D^{st}\left(f^{st}_e + \gamma^{st} g^{st}_e\right) \leq \mu C_e x_e \qquad \forall e \in E \tag{4.3}$$

$$\sum_{v \in N(u)} \left(g^{st}_{uv} - g^{st}_{vu}\right) \leq w_u \qquad \forall u \in V, (s,t) \in \mathcal{D} \tag{4.4}$$

$$\sum_{v \in N(u)} \left(g^{st}_{vu} - g^{st}_{uv}\right) \leq w_u \qquad \forall u \in V, (s,t) \in \mathcal{D} \tag{4.5}$$

$$x_e \in \{0,1\},\ w_u \in \{0,1\},\ f_{uv}^{st} \in [0,1],\ g_{uv}^{st} \in [0,1] \tag{4.6}$$

where $f_e^{st} = f_{uv}^{st} + f_{vu}^{st}$ and $g_e^{st} = g_{uv}^{st} + g_{vu}^{st}$. The objective function (4.1) minimizes the power consumption of the network represented by the number of active links and RE-routers. The equations (4.2) establish flow conservation, whereas the constraints (4.3) limit the available capacity (where μ denotes the maximum link utilization). The constraints (4.4) and (4.5) determine, whether decoding/encoding is necessary at a node u, such that RE-service is activated ($w_u = 1$) or not. Compression is necessary/takes place when the sum of incoming compressed flow is bigger than the sum of outgoing compressed flow (4.4) or vise versa (4.5). So, if u is a normal router, it only forwards flows without compression or decompression and if the percentage of compressed flow changes in a node, a RE-router is required. For the sake of notation, we assume that, all routers have the capability to perform RE-service, so we can enable it when needed.

It is also noted, in a feasible solution of GreenRE, a compressed flow is decompressed somewhere on the way to its destination. Otherwise, one node (latest at the target) would receive more incoming compressed traffic as outgoing (without being a RE-router), violating constraints (4.5). Consequently, in every optimal solution, there will be at least two active RE-routers or none at all. Clearly, employing more RE-routers (or links) than absolutely needed is feasible but not optimal.

In an aggregated perspective, the above described models are a range of more and more fine-tuned concepts to model energy efficient networks. Automatically, this leads to questions related to quality measures of these models, which again is dependent on precise data. Since in most cases, data is uncertain by nature, we believe that this uncertainty has to be included within these models. Our contribution is a proposal of including uncertainties within the GreenRE model as state-of-the art concept.

4.2.3 GreenRobustRE Model

As state-of-the-art model for energy-aware routing, the deterministic GreenRE model assumes that each traffic demand has a constant non-redundant value γ^{st}. This assumption leads to an inaccurate evaluation of energy saving, since the actual traffic redundancy rate fluctuates and is not known in advance. In practice, avoiding congestion is the most pressing matter, such that modeling has to be very close to worst-case analysis. By the above mentioned Γ-robustness and its probability bound, the conservatism of modeling can be alleviated by employing this concept. If the Γ is chosen appropriate, the probability of feasibility is high enough and as we show in Section 4.2.4, a significant improvement over the worst-case solution is still possible.

In the following, we propose the GreenRobustRE model, which addresses fluctuations by optimizing against a certain amount of uncertainties. As a consequence, the link capacity constraint (4.3) is deterministically satisfied, if this amount of uncertainty is realized (and satisfied by a very high probability for any other realization). Therefore, we adapt the approach of Bertsimas and Sim[BS03, BS04] as follows: For each demand pair, two values describe the potential (or “realized” in the sense of random variables) redundancy elimination: a (nominal) default value $\overline{\gamma}^{st} \in (0,1]$ and a maximal deviation $\widehat{\gamma}^{st} \geq 0$, $(\overline{\gamma}^{st} + \widehat{\gamma}^{st} \leq 1)$, such that the actual redundancy value γ^{st} is known to be within $[\overline{\gamma}^{st}, \overline{\gamma}^{st} + \widehat{\gamma}^{st}]$. So, whereas γ^{st} is a deterministic value in the GreenRE, it is now a random variable, symmetric distributed on an interval and defined by the two values $\overline{\gamma}^{st}$ and $\widehat{\gamma}^{st}$. Potentially, each demand can be compressed by its default ratio to $\overline{\gamma}^{st} D^{st}$. Applying Γ-robustness, we consider that at most Γ redundancy ratios fluctuate simultaneously. This means, the affected demand volumes have a lower compression potential, i.e., a higher value of γ^{st}. Consequentially, in Γ many cases the compressed flow can amount to a value as high as $(\overline{\gamma}^{st} + \widehat{\gamma}^{st})D^{st}$.

For instance, based on historical traces, a demand (s,t) seems to contain 60% of non-redundant (unique) traffic on average. Hence, we assume a nominal non-redundant ratio of $\overline{\gamma}^{st} = 0.6$. Assuming at most 90% of the traffic at any time is non-redundant as an upper bound, we can protect ourselves against wrong assumptions by adding $\widehat{\gamma}^{st} = 0.3$. Depending on the desired level of protection of our solution, we choose a Γ-value, such that our solution is still feasible (and optimal) if at most Γ many redundancy ratios deviate their assumptions, without specifying which ones.

Given a parameter $0 \leq \Gamma \leq |\mathcal{D}|$, the GreenRobustRE problem is to find a feasible routing at minimal energy costs, while the link capacity constraints are satisfied if at most Γ traffic pairs deviate from their $\overline{\gamma}^{st}$ values simultaneously. Note that $\Gamma = |\mathcal{D}|$ amounts to worst-case optimization, whereas $\Gamma = 0$ models the opportunistic case without uncertainty. The straightforward (but nonlinear) robust capacity constraint for a given Γ and an edge $e \in E$ is:

$$\sum_{(s,t)\in\mathcal{D}} D^{st}\left(f_e^{st} + \overline{\gamma}^{st} g_e^{st}\right) + \max_{\substack{Q\subseteq\mathcal{D}\\ |Q|\leq\Gamma}} \left\{ \sum_{(s,t)\in Q} \widehat{\gamma}^{st} D^{st} g_e^{st} \right\} \leq \ \mu C_e x_e \qquad \forall e \in E \quad (4.7)$$

Given g_e^{st}, the maximum part of (4.7) can be computed by:

$$\begin{aligned} \beta(g,\Gamma) := \max & \sum_{(s,t)\in\mathcal{D}} \widehat{\gamma}^{st} D^{st} g_e^{st} z_e^{st} & \\ \text{s.t.} & \sum_{(s,t)\in\mathcal{D}} z_e^{st} \leq \Gamma & [\pi_e] \\ & z_e^{st} \in \{0,1\} & [\rho_e^{st}] \end{aligned}$$

Based on [BS03], a *compact* reformulation can be obtained by employing total-unimodularity and LP duality of $\beta(g,\Gamma)$:

$$\begin{aligned} \beta(g,\Gamma) = \min \Gamma\pi_e + & \sum_{(s,t)\in\mathcal{D}} \rho_e^{st} & \\ \text{s.t. } \pi_e + \rho_e^{st} & \geq \widehat{\gamma}^{st} D^{st} g_e^{st} & \forall (s,t) \in \mathcal{D} \\ \rho_e^{st}, \pi_e & \geq 0 & \forall (s,t) \in \mathcal{D} \end{aligned}$$

where the primal binary variables z_e^{st} denote whether or not g_e^{st} is part of the subset $Q \subseteq \mathcal{D}$. The dual variables π_e and ρ_e^{st} corresponds to the constraint $\sum_{(s,t)\in\mathcal{D}} z_e^{st} \leq \Gamma$

and $z_e^{st} \leq 1$ (in the linear relaxation), respectively. Embedding this into (4.1)–(4.6), the GreenRobustRE can be compactly formulated by replacing the constraint (4.7) by:

$$\sum_{(s,t)\in\mathcal{D}} \left(D^{st}(f_e^{st} + \overline{\gamma}^{st} g_e^{st}) + \rho_e^{st}\right) + \Gamma\pi_e \leq \mu x_e C_e \quad \forall e \in E$$

$$\begin{aligned} &\pi_e + \rho_e^{st} \geq \widehat{\gamma}^{st} D^{st} g_e^{st} && \forall (s,t) \in \mathcal{D}, \forall e \in E \\ &\rho_e^{st}, \pi_e \geq 0 && \forall (s,t) \in \mathcal{D}, \forall e \in E \end{aligned}$$

Compared to the deterministic model GreenRE which has $|E| + |V| + 4|E||\mathcal{D}|$ variables and $|E| + 3|V||\mathcal{D}|$ constraints, this GreenRobustRE model has $|E| + |E||\mathcal{D}|$ additional variables and $|E||\mathcal{D}|$ additional constraints.

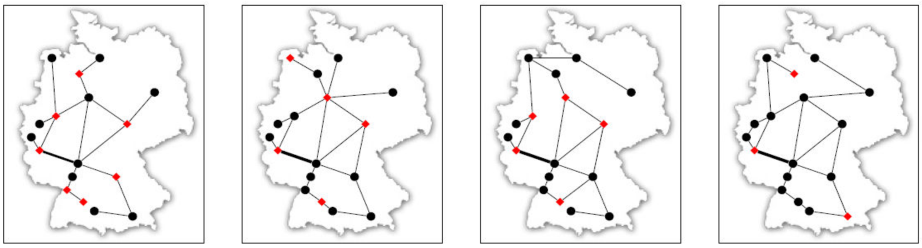
Note, that by the above reformulation, we can obtain a new (deterministic) mixed integer problem (called GreenRobustRE), protecting against uncertainties with high probability. While we believe that the theoretical improvement of this model is apparent by the above explanations, we will give a computational/practical evaluation in the next section.

4.2.4 Computational Evaluation

4.2.4.1 Test instances and Experimental settings

We solve the GreenRobustRE model with IBM ILOG CPLEX 12.4 solver [IBM]. All computations were carried out on a computer equipped with 2.7 Ghz Intel Core i7 and 8 GB RAM. We consider real-life traffic traces collected from the SNDlib [OWPT10]: the U.S. Internet2 Network (Abilene) ($|V| = 12$, $|E| = 15$, $|\mathcal{D}| = 130$) and the national research backbone network Germany17 ($|V| = 17$, $|E| = 26$, $|\mathcal{D}| = 251$). Capacity is set to $C = 10$ Gbps and $\mu = 0.5$ [CMN11] for each link.

In our computations, we use a single traffic matrix consisting of the mean volume for each traffic demand during a one day period. To achieve a network with high link utilization, all traffic was scaled with a factor four, while to avoid individual bottlenecks, we use four parallel links for (Köln, Frankfurt) in the Germany17 network and use double links for four links in the Abilene network: (ATLAng, HSTNng), (ATLAng, WASHng), (CHINng, IPLSng) and (HSTNng, LOSAng). For each network, 9 scenarios are generated by combining three nominal values $\overline{\gamma}$ and three deviation values $\widehat{\gamma}$. In our tests, we assume that in every scenario the ranges of the compression values are independent of the node-pair, i.e., $\overline{\gamma}^{st} = \overline{\gamma}$ and $\widehat{\gamma}^{st} = \widehat{\gamma}$ for all $s, t \in \mathcal{D}$. According to [AGA$^+$08, AMAR09] an upper bound on traffic redundancy of 50% can be assumed. Therefore, we assume that $\overline{\gamma} \geq 0.5$. In fact, we use three scenarios ($\overline{\gamma} = 0.5, \widehat{\gamma} = 0.1$), ($\overline{\gamma} = 0.5, \widehat{\gamma} = 0.25$) and ($\overline{\gamma} = 0.5, \widehat{\gamma} = 0.5$) to represent traffic demands with high redundant ratio ($\overline{\gamma} = 0.5$) and low ($\widehat{\gamma} = 0.1$), medium ($\widehat{\gamma} = 0.25$) or high ($\widehat{\gamma} = 0.5$) deviation. Similarly, the other scenarios are ($\overline{\gamma} = 0.7, \widehat{\gamma} = 0.1$), ($\overline{\gamma} = 0.7, \widehat{\gamma} = 0.2$), ($\overline{\gamma} = 0.7, \widehat{\gamma} = 0.3$) and ($\overline{\gamma} = 0.8, \widehat{\gamma} = 0.05$),



$\Gamma = 0$
#RE-routers= 7
#links= 20

$\Gamma = 10$
#RE-routers= 5
#links= 23

$\Gamma = 15$
#RE-routers= 5
#links= 23

$\Gamma = 251$
#RE-routers= 3
#links= 24

Figure 4.2: Routing Solutions on Germany17, $\overline{\gamma} = 0.7, \widehat{\gamma} = 0.2$

($\overline{\gamma} = 0.8, \widehat{\gamma} = 0.1$), ($\overline{\gamma} = 0.8, \widehat{\gamma} = 0.2$). For each scenario, we vary the robustness parameter Γ between 0 and $|\mathcal{D}|$.

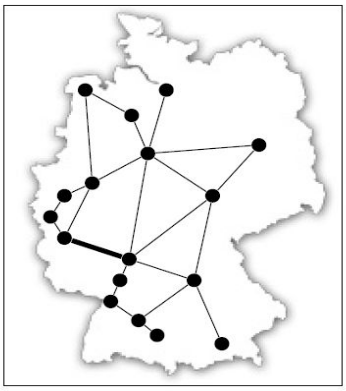
4.2.4.2 Results & Discussion

Before discussing particular trends or characteristics of solutions, we want to give a visualization of a typical solution of GreenRobustRE. In Fig. 4.2, we present solutions (within 10% optimality gap) for the Germany17 instance with $\overline{\gamma} = 0.7$ and $\widehat{\gamma} = 0.2$ ($\Gamma \in \{0, 10, 15, 251\}$). The figure indicates by line thickness, that the edge Koeln-Frankfurt is always employed multiple times (3, 4, 4, 4). It is noted, that the $\Gamma = 0$ case mirrors the GreenRE model ($\gamma = 0.7$) and the $\Gamma = 251$ case equals to the GreenRE model with $\gamma = 0.9$. As above, $\gamma^{st} = \gamma$ for all demands $s, t \in D$. The subset of chosen edges is printed black and the activated RE-routers are displayed as red squares. For comparisons sake, Fig. 4.3 presents the corresponding EAR solution, i.e., routing without any compression/decompression (note that the edge between Frankfurt and Koeln has to be used 4 times).

Energy saving vs. robustness In this section, we investigate the relation between energy saving and the level of robustness regarding the parameter Γ. All instances of the Abilene network can be solved to optimality in less than 10 minutes. For the Germany17 network, we limit the solving time to one hour and all best solutions are within 10% of optimality.

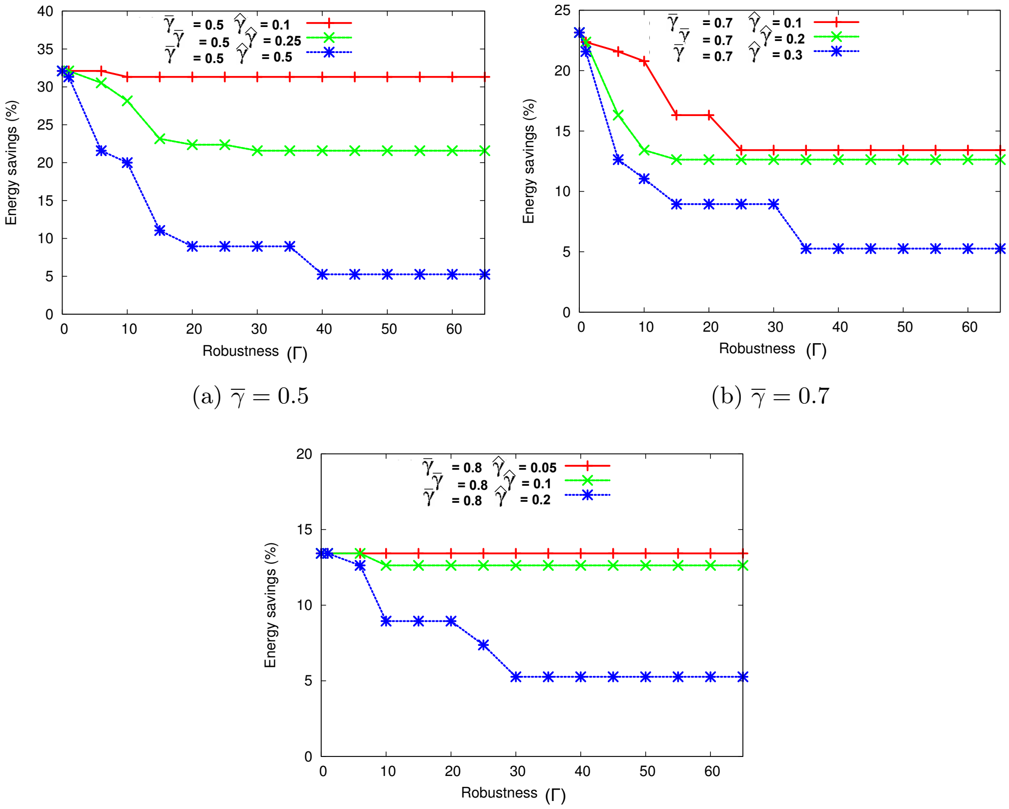
In a typical solution between three and seven RE-routers are activated. We observed that this number can changed independently of the Γ value. A prognosis is difficult to give, since the number of RE-routers is highly dependent on the traffic volumes, the capacity, and the network topology. Clearly, the same holds for the employed edges and depending on the demands and the employed RE-routers. However, at least a spanning tree has to be contained in any solution (since every node requests traffic from any other nodes).

Fig. 4.4a– Fig. 4.5c show the trade-off between energy saving vs. the value of Γ for each pair of ($\overline{\gamma}$, $\widehat{\gamma}$). The percentage of energy saving of the GreenRobustRE



#links= 26

Figure 4.3: EAR Solution Germany17



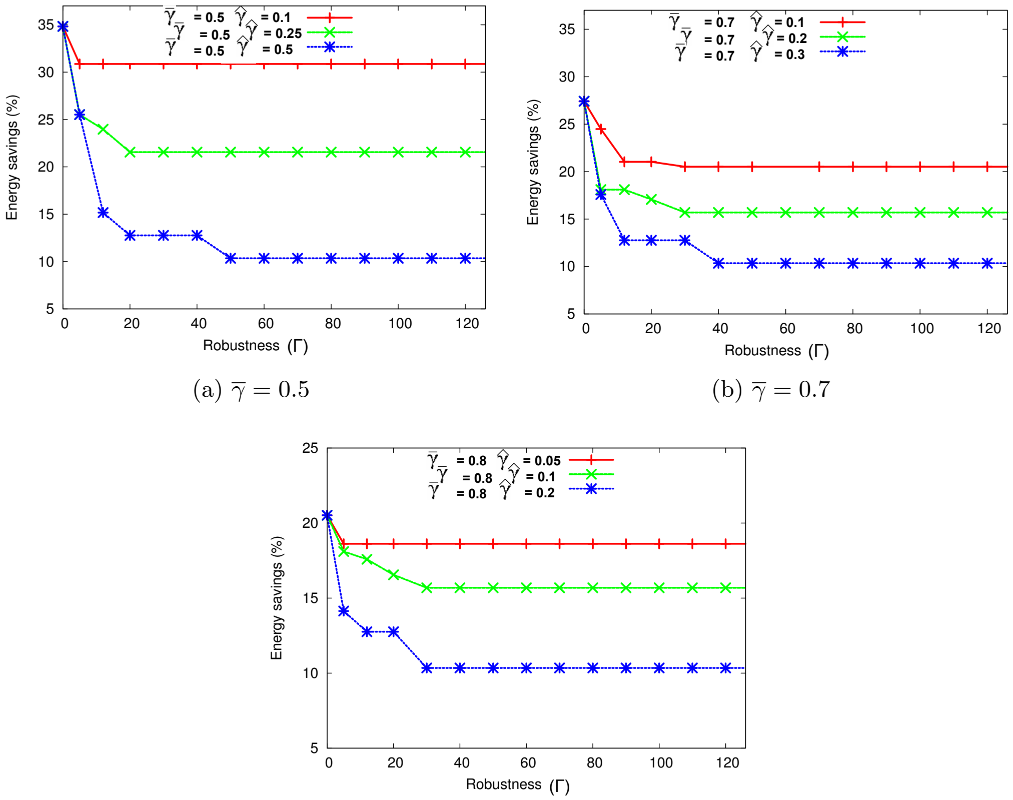
(a) $\overline{\gamma} = 0.5$

(b) $\overline{\gamma} = 0.7$

(c) $\overline{\gamma} = 0.8$

Figure 4.4: Energy saving vs. Γ-robustness for Abilene network

is computed in comparison with the case, that all links on the network are active (non-EAR solution). In both the Abilene and the Germany17 network, the solutions



(a) $\overline{\gamma} = 0.5$

(b) $\overline{\gamma} = 0.7$

(c) $\overline{\gamma} = 0.8$

Figure 4.5: Energy saving vs. Γ-robustness for Germany17 network

do not change when $\Gamma \geq \frac{|\mathcal{D}|}{2}$, and thus the x-axis is cut at $\Gamma = \frac{|\mathcal{D}|}{2}$. Clearly, a high value of Γ reduces the amount of energy saving for the network. From a technical point of view, increasing Γ leads to higher compression multipliers in (4.3) which are directly linked to bigger coefficients in the same constraint. Thus, more capacity is needed and energy consumption increases (potential energy saving decrease).

However, we observe that the energy saving are only reduced at low values of Γ. The energy level becomes constant after a certain level of robustness is requested. For example in Fig. 4.4a the amount of energy saving does not change when $\Gamma \geq 10$, $\Gamma \geq 30$ and $\Gamma \geq 40$, respectively. Similar observations can be drawn from Fig. 4.4b – Fig. 4.5c. An explanation of this phenomenon can be found in the distribution of the demand volumes. A fraction of the demands is dominating the others in volume. Hence, when the value of Γ covers all of these dominating demands, increasing Γ does not affect the routing solution and the percentage of energy saving remains stable. Fig. 4.4c ($\overline{\gamma} = 0.8$, $\widehat{\gamma} = 0.05$) shows the extreme case, where the solution is already fully robust for $\Gamma = 0$, i. e., it is identical to the solution of $\Gamma = |\mathcal{D}|$. This means the routing for a certain Γ has enough (spare) capacities to cover additional fluctuations without employing more links / RE routers.

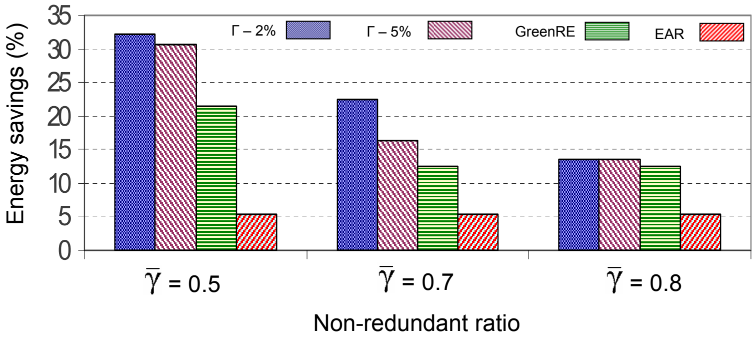


Figure 4.6: Abilene network - GreenRobustRE vs. GreenRE vs. Classical EAR.

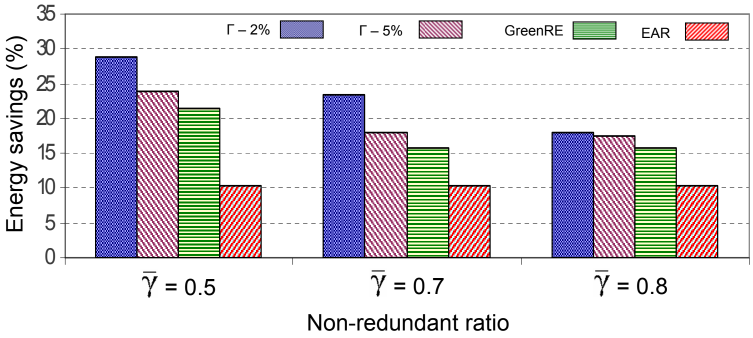


Figure 4.7: Germany17 - GreenRobustRE vs. GreenRE vs. Classical EAR.

GreenRobustRE vs. GreenRE vs. Classical EAR In this section, we exemplarily show that GreenRobustRE outperforms GreenRE and the classical EAR in case only a few traffic pairs deviate their redundancy elimination values simultaneously. Bertsimas and Sim [BS03] proved that already for small values of Γ, the capacity constraints are not violated with high probability. Experiments in [KKR11] for uncertain demands confirm and even strengthen this result (but $\Gamma = 0$ yields infeasible solutions over almost realizations). Therefore, we compare EAR and GreenRE with Γ as low as 2% and 5% of the traffic pairs. Since we do not know the fluctuation of redundancy elimination in the (deterministic) GreenRE model, γ^{st} needs to be underestimated by the worst case realization, i. e., $\gamma^{st} = \overline{\gamma}^{st} + \widehat{\gamma}^{st}$. Note that by this choice of γ^{st}, the GreenRE is equivalent to the GreenRobustRE model with $\Gamma = |\mathcal{D}|$.

The estimated values of unique traffic and its deviation used in this section are $(\overline{\gamma} = 0.5, \widehat{\gamma} = 0.25)$, $(\overline{\gamma} = 0.7, \widehat{\gamma} = 0.2)$ and $(\overline{\gamma} = 0.8, \widehat{\gamma} = 0.1)$. On the x-axis of Fig. 4.6 and Fig. 4.7, four columns for each value of $\overline{\gamma}$ represent the GreenRobustRE (with $\Gamma = 2\%$ and 5% of the total traffic pairs), the GreenRE, and the classical EAR.

We observe that the lowest energy saving are achieved by EAR. The energy efficiency for the network is improved when combining redundancy elimination and EAR (GreenRE). More importantly, the GreenRobustRE always outperforms the GreenRE, in many cases even by a considerable amount. Referring to Section 4.2.4.2, the GreenRobustRE model converges to the GreenRE model if the robustness level is increased (e.g. more than 50% of the traffic pairs).

Altogether, we observe that in cases where a worst-case analysis is not necessary, but rather the congestion should be avoided with high probability, the Γ robustness approach yields a significant improvement over previously proposed models. By the GreenRobustRE model, network operators can draw more accurate estimations (both in quality and feasibility) of energy saving for their network depending on the level of desired robustness. In this context, solutions for different Γ can support a well-reasoned decision making.

In this section, we have proposed a concept for embedding data uncertainty into state-of-the-art models for minimizing energy consumption of backbone networks. Taking traffic redundancy uncertainties into account, the GreenRobustRE model provides an accurate model for potential energy saving in backbone networks. Based on a case study with real-life traffic demands, we show the relation between energy saving and the desired robustness for the network. Further, we give insights in the relation between this and earlier proposed models, showing that the GreenRobustRE model is clearly superior in our test-cases. In next section, our model is expanded to include more general data uncertainties, i.e., both RE rate and demand volume fluctuations are considered as well.

4.3 Redundancy Elimination and Demand Volume Fluctuation

4.3.1 Robust-GreenRE Model

As shown in the previous Section, solving the GreenRobustRE model is already a complex task, however, it does not take traffic volumes fluctuations into account. Instead, each of the demands contains a constant factor of volume. This assumption leads to inefficient network designs and wastes energy saving. In this section, we present an extended model to deal with the fluctuation of both traffic demand volumes and RE rates, namely the Robust-GreenRE model.

In the Robust-GreenRE model, two values determining percentage of non-redundant traffic are given for each traffic demand: a nominal (default) value $\overline{\gamma}^{st} \in (0, 1]$ and a deviation $\widehat{\gamma}^{st}$ such that $0 \leq \widehat{\gamma}^{st}, \overline{\gamma}^{st} + \widehat{\gamma}^{st} \leq 1$ and the actual non-redundant rate $\gamma^{st} \in [\overline{\gamma}^{st}, \overline{\gamma}^{st} + \widehat{\gamma}^{st}]$. Similarly, each traffic demand is given by a nominal value $\overline{D}^{st} \geq 0$ and a deviation $\widehat{D}^{st} \geq 0$ so that the actual demand volume $D^{st} \in [\overline{D}^{st}, \overline{D}^{st} + \widehat{D}^{st}]$.

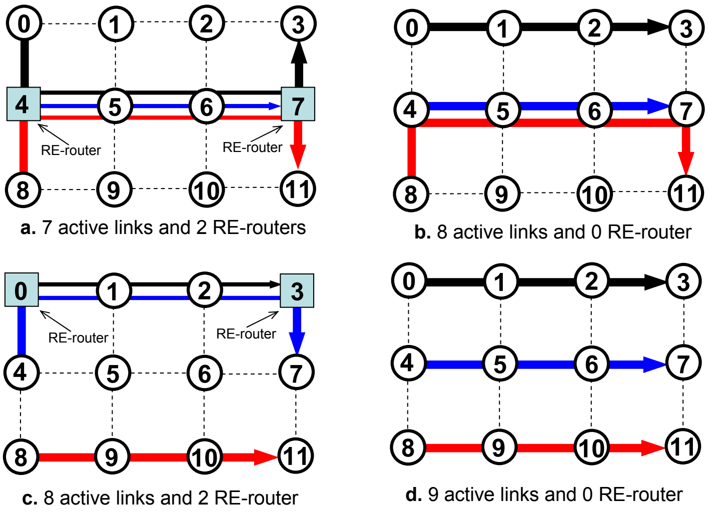
Potentially, each demand is expressed with its default value: $D^{st} = \overline{D}^{st}$ and $D^{st}_{comp} = \overline{\gamma}^{st} \times \overline{D}^{st}$. In the worst case realization, the peak values should be used and each traffic pair is expressed by $D^{st} = (\overline{D}^{st} + \widehat{D}^{st})$ and $D^{st}_{comp} = (\overline{\gamma}^{st} + \widehat{\gamma}^{st}) \times (\overline{D}^{st} + \widehat{D}^{st})$. Given two integral parameters $0 \leq \Gamma_d, \Gamma_\gamma \leq |\mathcal{D}|$ ($|\mathcal{D}|$ is the total number of demands), we denote $Q \subseteq \mathcal{D}$, $|Q| \leq \Gamma_d$, a set of traffic pairs allowed to deviate simultaneously from their nominal traffic volumes. Similarly, $Q' \subseteq \mathcal{D}$,

$|Q'| \leq \Gamma_\gamma$, is a set of demands in which all RE rates can deviate simultaneously. Observe that demands in $Q \cap Q'$ are simultaneously at their peak traffic and lowest RE rates. Given $(\Gamma_d, \Gamma_\gamma)$ as the desired robustness of the network, the Robust-GreenRE problem is to minimize the energy consumption of the network while satisfying the link capacity constraints whenever at most Γ_d demand volumes and Γ_γ RE rates deviate simultaneously from their nominal values.

Table 4.2: Demands and redundancy rates variation

Demand (s, t)	$\overline{D}^{st}$	$\widehat{D}^{st}$	$\overline{\gamma}^{st}$	$\widehat{\gamma}^{st}$
(0, 3)	3	1	0.5	0.3
(4, 7)	2	1	0.6	0.3
(8, 11)	1	2	0.7	0.3

Let us analyze the example of Fig. 4.8 to see that it is non-trivial to solve the Robust-GreenRE problem. We consider a (3×4) grid with a capacity of 4 Mbps per direction of each links. There are three traffic demands to be routed: $(0, 3)$, $(4, 7)$ and $(8, 11)$, each with respective nominal traffic volumes $\overline{D}^{st}$ and deviation $\widehat{D}^{st}$ (resp. nominal RE rates $\overline{\gamma}^{st}$ and deviation $\widehat{\gamma}^{st}$) as shown in Table 4.2. As shown in Fig. 4.8a, this is the optimal solution for the case in which no uncertainty is defined ($\Gamma_d = \Gamma_\gamma = 0$). In this solution, we activate two RE-routers at nodes 4 and 7 and the total traffic passing through links $(4 - 5 - 6 - 7)$ is equal to $\overline{D}^{0,3} \times \overline{\gamma}^{0,3} + \overline{D}^{4,7} \times \overline{\gamma}^{4,7} + \overline{D}^{8,11} \times \overline{\gamma}^{8,11} = 3 \times 0.5 + 2 \times 0.6 + 1 \times 0.7 = 3.4 < 4$.



a. 7 active links and 2 RE-routers

b. 8 active links and 0 RE-router

c. 8 active links and 2 RE-router

d. 9 active links and 0 RE-router

Figure 4.8: Example of robustness

Consider now the robust case in which $\Gamma_d = \Gamma_\gamma = 1$. There are 9 possible cases for the combinations of deviation in traffic volumes and RE rate as reported in Table 4.3. In Case 1, demand $(0, 3)$ deviates both on its traffic volume and RE

Table 4.3: 9 cases of the robustness

Case	Q	Q'	Best solution	Link load l_{uv} (Mbps)
1	(0,3)	(0,3)	Fig. 1b (1600 Watts)	$l_{0,1,2,3} = 4$, $l_{4,5,6,7} = 3$, $l_{8,4} = l_{7,11} = 1$
2	(0,3)	(4,7)	Fig. 1b (1600 Watts)	$l_{0,1,2,3} = 4$, $l_{4,5,6,7} = 3$, $l_{8,4} = l_{7,11} = 1$
3	(0,3)	(8,11)	Fig. 1b (1600 Watts)	$l_{0,1,2,3} = 4$, $l_{4,5,6,7} = 3$, $l_{8,4} = l_{7,11} = 1$
4	(4,7)	(0,3)	Fig. 1b (1600 Watts)	$l_{0,1,2,3} = 3$, $l_{4,5,6,7} = 4$, $l_{8,4} = l_{7,11} = 1$
5	(4,7)	(4,7)	Fig. 1b (1600 Watts)	$l_{0,1,2,3} = 3$, $l_{4,5,6,7} = 4$, $l_{8,4} = l_{7,11} = 1$
6	(4,7)	(8,11)	Fig. 1b (1600 Watts)	$l_{0,1,2,3} = 3$, $l_{4,5,6,7} = 4$, $l_{8,4} = l_{7,11} = 1$
7	(8,11)	(0,3)	Fig. 1c (1660 Watts)	$l_{0,1,2,3} = 3.6$, $l_{4,0} = 2$, $l_{8,9,10,11} = 3$, $l_{3,7} = 2$
8	(8,11)	(4,7)	Fig. 1c (1660 Watts)	$l_{0,1,2,3} = 3.3$, $l_{4,0} = 2$, $l_{8,9,10,11} = 3$, $l_{3,7} = 2$
9	(8,11)	(8,11)	Fig. 1c (1660 Watts)	$l_{0,1,2,3} = 2.7$, $l_{4,0} = 2$, $l_{8,9,10,11} = 3$, $l_{3,7} = 2$

rate. Thus the solution of Fig. 4.8a is infeasible because the traffic volume passing through links (4–5–6–7) is $(\overline{D}^{0,3}+\widehat{D}^{0,3})\times(\overline{\gamma}^{0,3}+\widehat{\gamma}^{0,3})+\overline{D}^{4,7}\times\overline{\gamma}^{4,7}+\overline{D}^{8,11}\times\overline{\gamma}^{8,11} = (3+1)\times(0.5+0.3)+2\times 0.6+1\times 0.7 = 5.1 > 4$. The optimal solution in this case is presented in Fig. 4.8b in which 8 links are activated and no RE-router is used. The power consumption is $8 \times 200 = 1600$ Watts.

In Case 9, both the traffic volume and the RE rate of demand $(8, 11)$ deviate simultaneously. The solution in Fig. 4.8b is infeasible in this case even if we enable RE-routers at node 4 and 7 since the total traffic passing through links $(4-5-6-7)$ will be $\overline{D}^{4,7}\times\overline{\gamma}^{4,7}+(\overline{D}^{8,11}+\widehat{D}^{8,11})\times(\overline{\gamma}^{8,11}+\widehat{\gamma}^{8,11}) = 2\times 0.6+(1+2)\times(0.7+0.3) = 4.2 > 4$. In Case 9, the optimal solution is the one of Fig. 4.8c with 8 active links and 2 RE-routers. However, in the Robust-GreenRE model with $\Gamma_d = \Gamma_\gamma = 1$, any demand can deviate from its nominal volume or RE rate, as long as at most one demand and one RE rate deviate their volumes at the same time. Consequently, a solution is feasible if and only if it satisfies all of the 9 cases. Hence, Fig. 4.8d is the only feasible solution since Fig. 4.8c is infeasible for Case 1 of Table 4.3.

The idea of robustness is that we should reserve some space in the link capacity to accommodate the fluctuation in the traffic volumes and RE rates. To do so, we define a function $\delta(f, g, \Gamma_d, \Gamma_\gamma)$ such that the capacity constraints satisfy:

$$\sum_{(s,t)\in\mathcal{D}} \overline{D}^{st} \left(f_{uv}^{st} + \overline{\gamma}^{st} g_{uv}^{st}\right) + \delta(f, g, \Gamma_d, \Gamma_\gamma) \leq \mu C_{uv} x_{uv} \tag{4.3'}$$

The problem now is how to find the value of the function $\delta(f, g, \Gamma_d, \Gamma_\gamma)$. To answer this question, we use the notations $Q_d = Q \backslash Q'$, $Q_\gamma = Q' \backslash Q$ and $Q_{d\gamma} = Q \cap Q'$ as independent sets such that: $Q_{d\gamma}$ contains demands in which both traffic volumes and RE rates can deviate, Q_d (resp. Q_γ) contains demands in which only traffic volumes (resp. RE rates) can deviate from their nominal values. Indeed, we can formulate the problem using the two sets Q (demands variation) and Q' (RE rates variation). However, the final formulation will be non-linear. Therefore the three sets Q_d, Q_γ and $Q_{d\gamma}$ have to be used to overcome this problem. For simplicity, we use the notation e instead of uv, $\forall\ \{uv\} \in E$. Then the worst case scenario when considering fluctuation on an arc e is given by:

$$\begin{aligned}
\sum_{(s,t)\in\mathcal{D}} \overline{D}^{st} f_e^{st} + \max_{Q\subseteq\mathcal{D}} \Big\{ \sum_{(s,t)\in Q} \widehat{D}^{st} f_e^{st} \Big\} + \sum_{(s,t)\in\mathcal{D}} \overline{D}^{st} \overline{\gamma}^{st} g_e^{st} \\
+ \max_{Q_\gamma = Q'\backslash Q} \Big\{ \sum_{(s,t)\in Q_\gamma} \overline{D}^{st} \widehat{\gamma}^{st} g_e^{st} \Big\} + \max_{Q_{d\gamma} = Q\cap Q'} \Big\{ \sum_{(s,t)\in Q_{d\gamma}} (\widehat{D}^{st} \widehat{\gamma}^{st} \\
+ \widehat{D}^{st} \overline{\gamma}^{st} + \overline{D}^{st} \widehat{\gamma}^{st}) g_e^{st} \Big\} + \max_{Q_d = Q\backslash Q'} \Big\{ \sum_{(s,t)\in Q_d} \widehat{D}^{st} \overline{\gamma}^{st} g_e^{st} \Big\} \ \leq\ \mu C_e x_e
\end{aligned} \tag{4.3''}$$

Obviously, Constraints (4.3') and (4.3") are equivalent if $\delta(f, g, \Gamma_d, \Gamma_\gamma)$ is the maximum part of Constraint (4.3"). Constraint (4.3") can be rewritten as a set of many constraints corresponding to all possible sets Q_d, Q_γ and $Q_{d\gamma}$, but the resulting model has an exponential number of constraints. We thus propose three methods to overcome this difficulty.

4.3.2 Compact formulation

Given f_e^{st}, g_e^{st}, Γ_d, and Γ_γ, the function $\delta(f, g, \Gamma_d, \Gamma_\gamma)$ can be computed by:

$$\begin{aligned}
\textit{(primal)} \qquad \delta(f, g, \Gamma_d, \Gamma_\gamma) = \max \sum_{(s,t)\in\mathcal{D}} \Big(\widehat{D}^{st} f_e^{st} (z_{e,Q_d}^{st} + z_{e,Q_{d\gamma}}^{st}) \\
+ \overline{D}^{st} \widehat{\gamma}^{st} g_e^{st} z_{e,Q_\gamma}^{st} + (\widehat{D}^{st} \widehat{\gamma}^{st} + \widehat{D}^{st} \overline{\gamma}^{st} + \overline{D}^{st} \widehat{\gamma}^{st}) g_e^{st} z_{e,Q_{d\gamma}}^{st} + \widehat{D}^{st} \overline{\gamma}^{st} g_e^{st} z_{e,Q_d}^{st} \Big)
\end{aligned}$$

$$\text{s.t.} \quad \sum_{(s,t)\in\mathcal{D}} \left(z^{st}_{e,Q_d} + z^{st}_{e,Q_{d\gamma}}\right) \le \Gamma_d \qquad \forall e \in E \qquad [\pi_{e,d}] \tag{4.3a}$$

$$\sum_{(s,t)\in\mathcal{D}} (z^{st}_{e,Q_\gamma} + z^{st}_{e,Q_{d\gamma}}) \le \Gamma_\gamma \qquad \forall e \in E \qquad [\pi_{e,\gamma}] \tag{4.3b}$$

$$z^{st}_{e,Q_d} + z^{st}_{e,Q_{d\gamma}} + z^{st}_{e,Q_\gamma} \le 1 \qquad \forall e \in E,\ (s,t) \in \mathcal{D} \qquad [\sigma^{st}_e] \tag{4.3c}$$

$$z^{st}_{e,Q_d} \in \{0,1\} \qquad \forall e \in E \qquad [\rho^{st}_{e,d}] \tag{4.3d}$$

$$z^{st}_{e,Q_\gamma} \in \{0,1\} \qquad \forall e \in E \qquad [\rho^{st}_{e,\gamma}] \tag{4.3e}$$

$$z^{st}_{e,Q_{d\gamma}} \in \{0,1\} \qquad \forall e \in E \qquad [\rho^{st}_{e,d\gamma}] \tag{4.3f}$$

where binary variables z^{st}_{e,Q_d}, z^{st}_{e,Q_γ} and $z^{st}_{e,Q_{d\gamma}}$ denote whether a traffic pair (s,t) belongs respectively to the sets Q_d, Q_γ, $Q_{d\gamma}$ or not. Note that, a traffic demand (s,t) can belong exactly to one and only one of the three sets Q_d, Q_γ and $Q_{d\gamma}$. Constraints (4.3a) and (4.3b) are used to limit size of the set $|Q| = |Q_d \cup Q_{d\gamma}| \le \Gamma_d$ and $|Q'| = |Q_\gamma \cup Q_{d\gamma}| \le \Gamma_\gamma$. Constraint (4.3c) indicates that no traffic pair (s,t) can belong to more than one of the three sets Q_d, Q_γ and $Q_{d\gamma}$.

We now need to find LP duality of the above primal problem using dual variables $\pi_{e,d}$, $\pi_{e,\gamma}$, σ^{st}_e, $\rho^{st}_{e,d}$, $\rho^{st}_{e,d\gamma}$ and $\rho^{st}_{e,\gamma}$. To do so, we first relax the last three constraints (4.3d), (4.3e) and (4.3f) to real variables: $0 \le z^{st}_{e,Q_d}, z^{st}_{e,Q_{d\gamma}}, z^{st}_{e,Q_\gamma} \le 1$. By employing LP duality for the relaxation of the primal, we obtain:

$$\textit{(dual)} \qquad \delta_{relax}(f,g,\Gamma_d,\Gamma_\gamma) = \min\ \Gamma_d\pi_{e,d} + \Gamma_\gamma\pi_{e,\gamma} + \sum_{(s,t)\in\mathcal{D}} (\sigma^{st}_e + \rho^{st}_{e,d} + \rho^{st}_{e,\gamma} + \rho^{st}_{e,d\gamma})$$

$$\text{s.t.} \quad \pi_{e,d} + \sigma^{st}_e + \rho^{st}_{e,d} \ge \widehat{D}^{st}(f^{st}_e + \overline{\gamma}^{st}g^{st}_e) \quad \forall (s,t) \in \mathcal{D} \tag{4.3a'}$$

$$\pi_{e,d} + \pi_{e,\gamma} + \sigma^{st}_e + \rho^{st}_{e,d\gamma} \ge \widehat{D}^{st}f^{st}_e + \left(\widehat{D}^{st}\widehat{\gamma}^{st} + \widehat{D}^{st}\overline{\gamma}^{st} + \overline{D}^{st}\widehat{\gamma}^{st}\right)g^{st}_e \quad \forall (s,t) \in \mathcal{D} \tag{4.3b'}$$

$$\pi_{e,\gamma} + \sigma^{st}_e + \rho^{st}_{e,\gamma} \ge \overline{D}^{st}\widehat{\gamma}^{st}g^{st}_e \quad \forall (s,t) \in \mathcal{D} \tag{4.3c'}$$

$$\pi_{e,d}, \pi_{e,\gamma}, \sigma^{st}_e, \rho^{st}_{e,d}, \rho^{st}_{e,\gamma}, \rho^{st}_{e,d\gamma} \ge 0 \quad \forall (s,t) \in \mathcal{D} \tag{4.3d'}$$

Since the primal problem is a *max* problem, the optimal value of the relaxation of the primal $\delta_{relax}(f,g,\Gamma_d,\Gamma_\gamma)$ is greater or equal to the original one $\delta(f,g,\Gamma_d,\Gamma_\gamma)$. As a result, the objective of the duality of the relaxation is also greater or equal to $\delta(f,g,\Gamma_d,\Gamma_\gamma)$ and it makes the capacity constraint strongly robust. By embedding this duality of the relaxation into (4.1)–(4.6), the (strong) Robust-GreenRE problem can be compactly formulated by replacing Constraint (4.3) with:

$$\sum_{(s,t)\in\mathcal{D}} (\sigma^{st}_e + \rho^{st}_{e,d} + \rho^{st}_{e,\gamma} + \rho^{st}_{e,d\gamma}) + \sum_{(s,t)\in\mathcal{D}} \overline{D}^{st}(f^{st}_e + \overline{\gamma}^{st}g^{st}_e) + \Gamma_d\pi_{e,d}$$
$$+\Gamma_\gamma\pi_{e,\gamma} \le \mu C_e x_e \qquad \forall e \in E$$

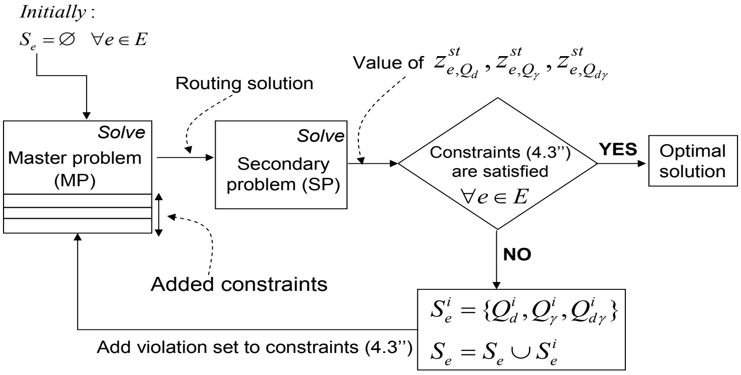
and adding constraints (4.3a'), (4.3b'), (4.3c') and (4.3d') to the deterministic model (4.1)–(4.6).

4.3.2.1 Constraint generation (Exact Algorithm)

The compact formulation in some cases gives a stronger robustness than what we need. Therefore, we pay more and the result obtained is a lower bound on energy saving. In this section, we present an algorithm that aims at finding the exact solution of the Robust-GreenRE model. We refer the reader to the explanation in [KKR13] for a similar method applied for the case in which only demand variation is considered. The main idea is to generate iteratively subsets of traffic demands representing demands which traffic volumes and/or RE rates may deviate from their nominal values. Let us call:

- *Master Problem (MP)*: deterministic ILP formulated with Constraints (4.1)–(4.6);
- *Secondary Problem (SP)*: primal model of the compact formulation, so Constraints (4.3a)–(4.3f) with the primal objective function.

We define for each link e of the network a set $S_e^i = \{Q_d^i, Q_{d\gamma}^i, Q_\gamma^i\}$ of demands which does not satisfy the constraints (4.3") (or (4.3')) where $S_e = \{S_e^i\}$, for all $e \in E$ at each iteration i of the algorithm (Fig. 4.9).



Initially :
$S_e = \emptyset \quad \forall e \in E$
Routing solution
Value of $z^{st}_{e,Q_d}, z^{st}_{e,Q_\gamma}, z^{st}_{e,Q_{d\gamma}}$
Solve
Master problem (MP)
Solve
Secondary problem (SP)
Constraints (4.3") are satisfied $\forall e \in E$
YES
Optimal solution
NO
Added constraints
$S_e^i = \{Q_d^i, Q_\gamma^i, Q_{d\gamma}^i\}$
$S_e = S_e \cup S_e^i$
Add violation set to constraints (4.3")

Figure 4.9: Diagram of constraint generation method

Initially, we set $S_e = \emptyset$ for all $e \in E$. We start the algorithm by solving the MP to find a feasible routing. Then, we use the values of f_e^{st} and g_e^{st} given by the routing solution as inputs for determining $\delta(f, g, \Gamma_d, \Gamma_\gamma)$ using the SP. Based on the objective value of the SP, we check if constraints (4.3") are satisfied or not for each link. As soon as we find a capacity violation on a link, we use the values of z^{st}_{e,Q_d}, $z^{st}_{e,Q_{d\gamma}}$ and z^{st}_{e,Q_γ} to determine Q_d^i, $Q_{d\gamma}^i$, Q_γ^i. We define S_e^i and update $S_e = S_e \cup S_e^i$. Finally, we add a new constraint corresponding to the violated contraint (4.3") and S_e^i to the Master Problem. This process is repeated until there is no more violation. If at one step, the Master Problem is infeasible, we conclude that there is no solution satisfying the robustness. Otherwise, the final solution is optimal for Robust-GreenRE.

4.3.2.2 Heuristic Algorithm

Energy-aware routing problem is known to be NP-Hard [GMMO10]. Also we now present a heuristic algorithm based on the compact ILP formulation to quickly find efficient solutions for large networks. Since the power consumption of a link (200 Watts [CMN11]) is much more than an enabled RE-router (30 Watts [GMPR12]), the heuristic gives priority to the minimization of the number of active links. In summary, the heuristic algorithm has two steps: the first step is to use as few active links as possible, and then we minimize the number of RE-routers in the second step.

Algorithm 4: *Inputs:* A graph $G = (V, E)$ modeling the network with link capacity C_{uv}; the robust parameters $(\Gamma_d, \Gamma_\gamma)$; a set of demands $\mathcal{D}$.

```
Step 1 - Minimize the number of active links by removing low loaded links:
Find a feasible routing solution called $P\_current$;
Let $S$ be an ordered list initialized with the links of $G$ sorted by increasing
  traffic load in $P\_current$;
Let $R := \emptyset$ be the set of links that cannot be removed;
repeat
    $e := S.lowest\_loaded\_link()$ such that $e \notin R$;
    $S := S \setminus \{e\}$;
    if a feasible robust routing P_new on $E \setminus \{e\}$ is found then
        $S\_new :=$ list of links sorted by increasing traffic load in $P\_new$;
        if $P\_new$ has less active links than $P\_current$ then
            $P\_current := P\_new$;
            $S := S\_new$; $E := E \setminus \{e\}$;
        end
    else
        $R := R \cup \{e\}$;
    end
until $(S = \emptyset)$ or $(R = S)$;
Return the final feasible routing solution (if any);
Step 2 - Find feasible solution minimizing the number of RE-routers on the
  set of active links $E$ found in Step 1.
```

Step 1 of Algorithm 4 is a constraints satisfaction problem returning a feasible routing. Hence, we use the MILP of the compact formulation without objective function. Our simulations show that it is quite fast to find such a feasible routing solution even for large networks. In each round of the algorithm, we try to remove a link with low load and then to find and evaluate a new feasible routing using less active links. The idea behind this algorithm is that we try to turn off low loaded links and to accommodate their traffic on other links in order to reduce the total number of active links. Observe that unused links (i.e. links that are not carrying traffic) are not considered in the set S since the removal of such a link will result in a routing *P_new* equal to the routing *P_current*.

If a feasible routing is found in *Step 1*, and so a set of active links, we proceed in *Step 2* to minimize the number of enabled RE-routers. More precisely, we use the compact MILP formulation in which the objective function is set to $\min \sum_{u \in V} w_u$. Furthermore, we set all binary variables associated to active links to 1 and the others to 0 (this speed-up the resolution of the MILP).

To further reduce the computation time of Algorithm 4, we can consider additional heuristic. For instance, in *Step 1*, while removing a low loaded link (and so setting a binary variable to 0) we can also set the variable $x_{\{uv\}}$ associated to a heavily loaded link to 1. Indeed, such link will certainly be part of the final solution.

4.3.3 Computational Evaluation

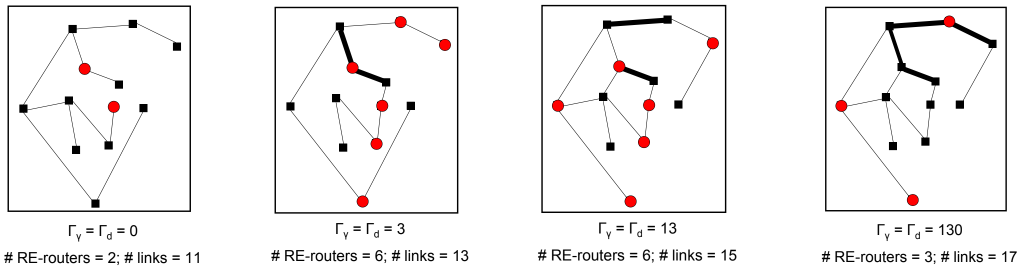
4.3.4 Test instances and Experimental settings

We solved the Robust-GreenRE model with IBM ILOG Cplex 12.4 solver [IBM]. All computations were carried out on a computer equipped with a 2.7 Ghz CPU and 8 GB RAM. We consider real-life traffic traces collected from the SNDlib [OWPT10]: the U.S. Internet2 Network (Abilene) ($|V| = 12$, $|E| = 15$, $|\mathcal{D}| = 130$), the Geant network ($|V| = 22$, $|E| = 36$, $|\mathcal{D}| = 387$) and the Germany50 ($|V| = 50$, $|E| = 88$, $|\mathcal{D}| = 1595$). Note that, in section 4.3.5.1, we use a simplified Abilene network in which only a half of demands are used (65 demands, randomly chosen). It is because an exponential number of constraints can be added to the constraint generation model and so the overall computation time is more than 10 hours. Capacity is set to $C_{uv} = 5/10/20$ Gbps for each arc of the Abilene/ Germany50/ Geant network, respectively.

In our test instances, each traffic demand has two values: the nominal and the peak volumes during one day period. These values can be collected using the dynamic traffic from the SNDlib. To achieve a network with high link utilization, all traffic was scaled with a factor of three. To avoid individual bottlenecks, we add parallel links to increase capacity on some specific links. To find parallel links, we first relax the variables $x_{\{uv\}}$ to integer variables in the Master Problem. Then, we find the routing solution for the worst case scenario ($\Gamma_d = \Gamma_\gamma = 100\%$) using the relaxed Master Problem. The links (u, v) in which $x_{\{uv\}} > 1$ would be the congested links, so we add more capacity on these links and call them as parallel links. According to [AGA$^+$08, AMAR09], based on real traffic traces, an upper bound on traffic redundancy is assumed to 50%. In the simulations, we use $\overline{\gamma} = 0.5$ and $\widehat{\gamma} = 0.3$ and for each scenario, we vary the robust parameters (Γ_d, Γ_γ) in between 0 and the total demands ($|\mathcal{D}|$).

4.3.5 Results and Discussion

Before discussing particular trends or characteristics of solutions, we want to give a visualization of a typical solution of Robust-GreenRE. In Fig. 4.10, we present solutions for the Abilene network. The figure indicates by line thickness, that the



$\Gamma_\gamma = \Gamma_d = 0$
# RE-routers = 2; # links = 11

$\Gamma_\gamma = \Gamma_d = 3$
# RE-routers = 6; # links = 13

$\Gamma_\gamma = \Gamma_d = 13$
# RE-routers = 6; # links = 15

$\Gamma_\gamma = \Gamma_d = 130$
# RE-routers = 3; # links = 17

Figure 4.10: Routing and RE-router placement on Abilene network

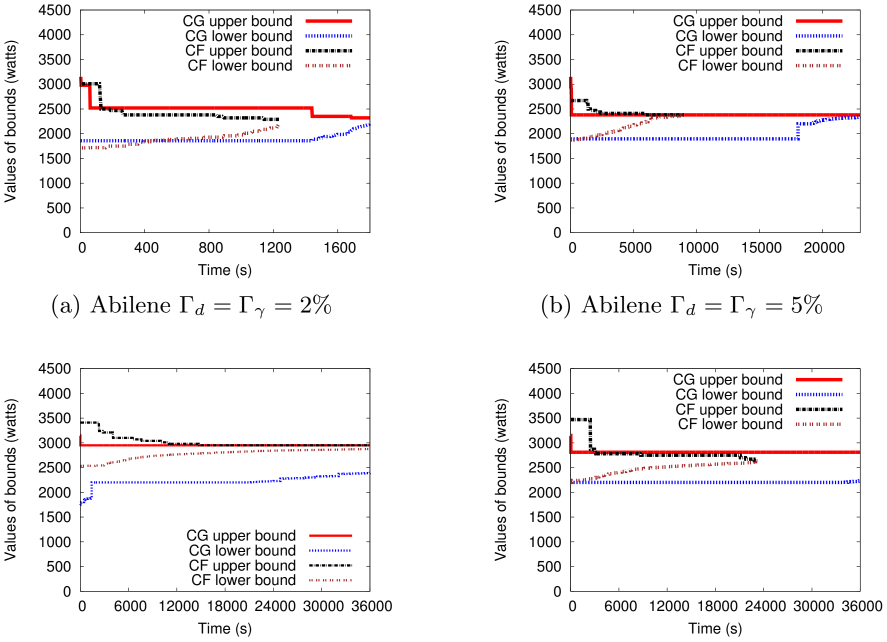
edge is employed with parallel links. It is noted, that the $\Gamma_\gamma = \Gamma_d = 0$ case mirrors the GreenRE model with nominal demands and RE rates while the $\Gamma_\gamma = \Gamma_d = 130$ case equals to the GreenRE model with all peak values of traffic demands and RE rates. The subset of chosen edges is printed black and the activated RE-routers are displayed as circles. In a typical solution, between two and six RE-routers are activated. We observed that this number can change independently of the Γ value. For instance, 2 RE-routers are needed when $\Gamma_\gamma = \Gamma_d = 0$. This number increases to 6 when $\Gamma_\gamma = \Gamma_d = 3$ or 13. However, the number of RE-routers reduces to 3 when $\Gamma_\gamma = \Gamma_d = 130$. A prognosis is difficult to give, since the number of RE-routers is highly dependent on the traffic volumes, the capacity, and the network topology. Clearly, the same holds for the employed edges and depending on the demands and the employed RE-routers. However, in general, an increase in Γ leads to higher capacity requirement and more links and/or RE-routers need to be used.

4.3.5.1 Gap between different methods

Table 4.4: Constraint Generation (CG) vs. Compact Formulation (CF) vs. Heuristic

$\Gamma_\gamma, \Gamma_d(\%)$	CG method			CF method		Heuristic	
	# violations	gap opt (%)	time (s)	gap opt (%)	time (s)	gap opt (%)	time (s)
2	5870	0	1800	0	1240	4	≤ 50
5	12981	0	23000	0	9000	5	≤ 50
10	64841	18.9	36.10^3	2.5	36.10^3	24	≤ 50
20	64433	20.6	36.10^3	0	22.10^3	27	≤ 50
100	65576	0	36.10^3	0	1400	7	≤ 50

In this section, we compare the energy saving offered by the three proposed methods: Constraints Generation (CG), Compact Formulation (CF) and Heuristic. We present in detail the comparison between the three methods in Table 4.4 for the simplified Abilene network. For CG method, an increase in level of robustness (representing by Γ_γ, Γ_d) leads to higher number of violations. CG can find optimal solution in less than 10 hours in case of small Γ_γ, Γ_d. However, for large values of Γ_γ, Γ_d, the computation time is increasing and the solution is still far from the optimality estimated by CPLEX. For instance, after 10 hours of computation, the



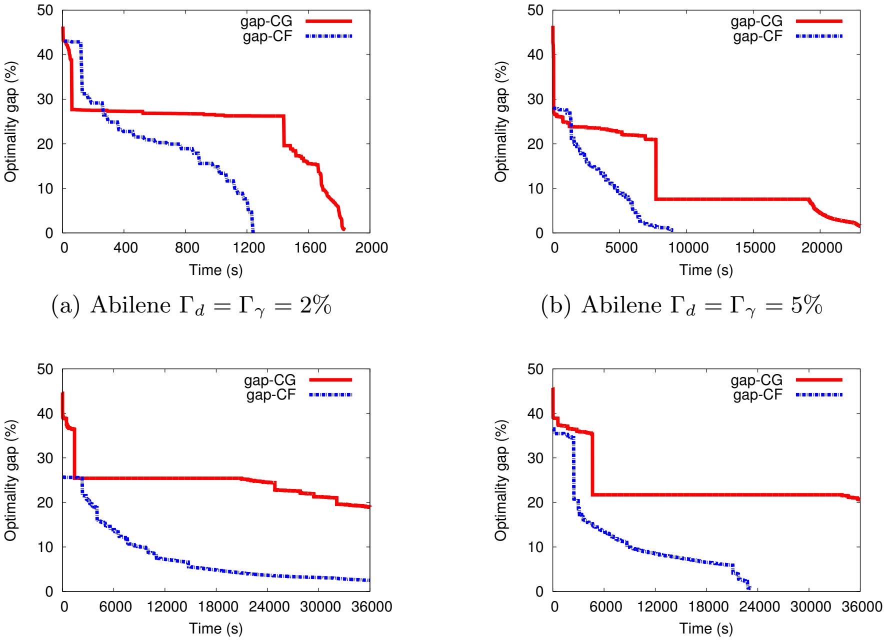
(a) Abilene $\Gamma_d = \Gamma_\gamma = 2\%$

(b) Abilene $\Gamma_d = \Gamma_\gamma = 5\%$

(c) Abilene $\Gamma_d = \Gamma_\gamma = 10\%$

(d) Abilene $\Gamma_d = \Gamma_\gamma = 20\%$

Figure 4.11: Upper bound and lower bound: *Compact Formulation (CF)* vs. *Constraint Generation (CG)*



(a) Abilene $\Gamma_d = \Gamma_\gamma = 2\%$

(b) Abilene $\Gamma_d = \Gamma_\gamma = 5\%$

(c) Abilene $\Gamma_d = \Gamma_\gamma = 10\%$

(d) Abilene $\Gamma_d = \Gamma_\gamma = 20\%$

Figure 4.12: Optimality gaps: *Compact Formulation (CF)* vs. *Constraint Generation (CG)*

optimality gap is 18.9% in case $\Gamma_\gamma = \Gamma_d = 10\%$ of total demands. The CF method is quite fast except in case $\Gamma_\gamma = \Gamma_d = 10\%$ of total demands, the optimality gap is 2.5% after 10 hours of computation. As expected, the heuristic algorithm is the fastest method. All feasible solutions can be found in less than 50 seconds.

To better see the evolution of the Constraints Generation (CG) and Compact Formulation (CF) methods, we show in Fig. 4.11 - 4.12, respectively the upper bound, the lower bound and the optimality gap obtained by CPLEX. The evolution of the CF method is much better than the CG method. As shown in Fig. 4.11a - 4.11d, in CF method, both the upper and lower bounds are improving meanwhile it seems only the lower bound in CG method is improving. As shown in Fig. 4.11c, both methods do not reach the optimality after 10 hours of computation, however, the gap of CF method is quite small (2.5%) with respect to the CG method (18.9%). Fig. 4.12 shows another view of the evolution: the gap between current feasible solution and optimal solution. This gap equals to zero mean the solution is the optimal one. Again, the CF method outperforms the CG method in term of improving optimality gap. However, it is noted that we can only find the exact solution using the CG method. The optimal point obtained by the CF method is only a lower bound of energy saving (see section 4.3.2.1).

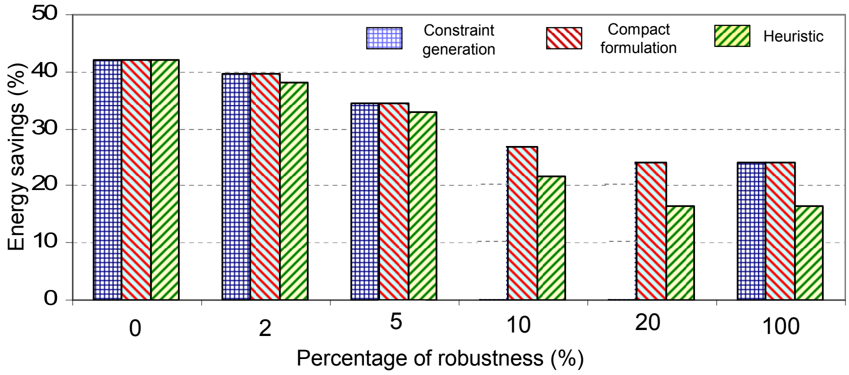
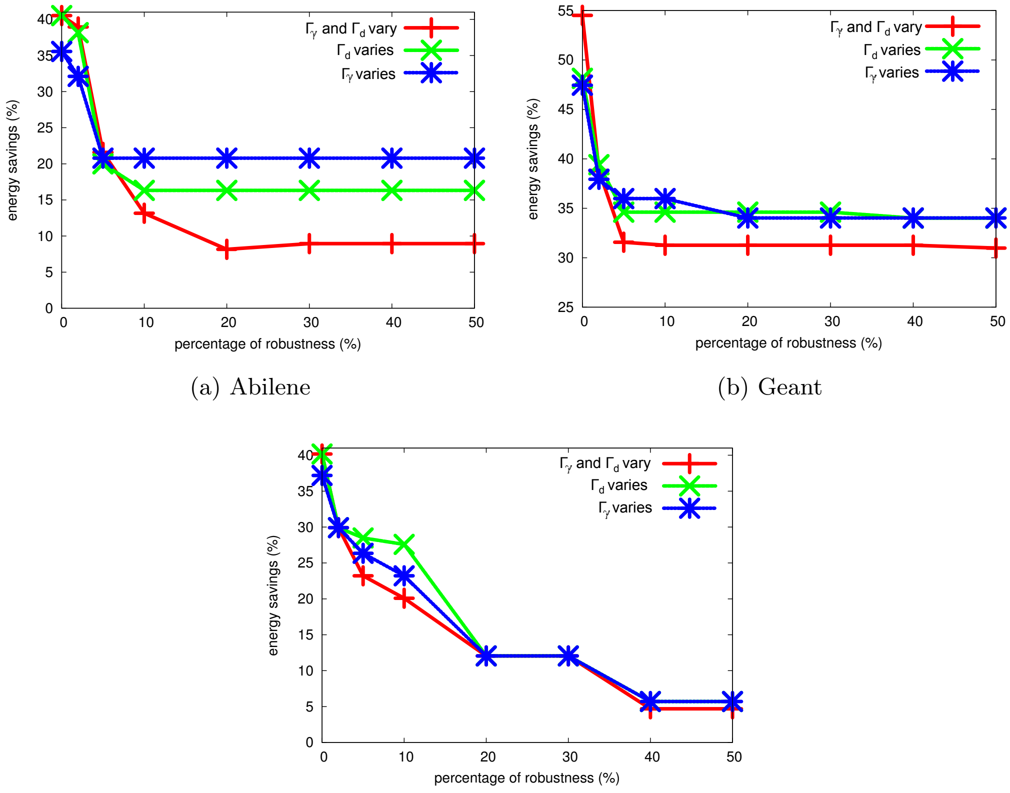


Figure 4.13: Comparison of the proposed methods on Abilene.

We show in Fig. 4.13 a comparison of performance between the three methods. The y-axis is the percentage of energy saving and the x-axis is the percentage of robustness over the total demand ($\Gamma/|\mathcal{D}|$). Both Γ_d and Γ_γ vary with the same value, e.g. robustness = 10% means $\Gamma_d = \Gamma_\gamma = 0.1 \times |\mathcal{D}|$. We observe that the maximum gap reported in Fig. 4.13 between the heuristic and the CG (and CF) method is 7.63%, and this gap decreases for small values of Γ_d and Γ_γ. Recall that measurements performed on real networks have shown that only a small fraction of the traffic demands deviate simultaneously from their nominal values [KKR13]. Furthermore, the aim of robust optimization is precisely to take benefit of that fact in order to improve the design of the network, and in our case to save more energy. We have seen that our heuristic algorithm offers good performances both in terms of running time and quality of the solution in this setting. Thus in the sequel, we will use our heuristic to evaluate the Robust-GreenRE model on larger instances.

4.3.5.2 Energy saving vs. robustness



(a) Abilene

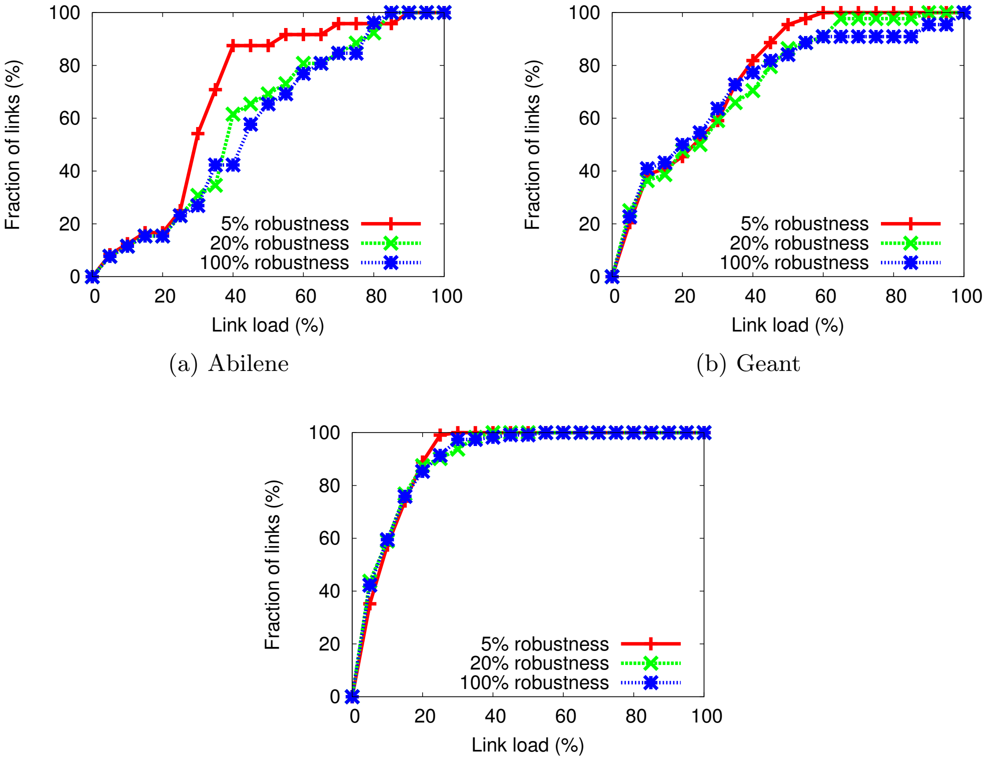
(b) Geant

(c) Germany

Figure 4.14: Energy saving vs. robustness for Abilene, Geant and Germany network

Fig. 4.14 shows the trade-off between energy saving and the level of robustness regarding the parameters $(\Gamma_d, \Gamma_\gamma)$. We consider three test cases (1) both Γ_d and Γ_γ, (2) only Γ_γ and (3) only Γ_d vary their values. In the Case 1, both Γ_d and Γ_γ vary with the same value of robustness. Note that, when $\Gamma_\gamma = \Gamma_d = 100\%$, all demands and compression rates are at the worst case, therefore the Robust-GreenRE is equivalent to the deterministic GreenRE. In Case 2 (resp. Case 3), while Γ_γ (resp. Γ_d) varies, Γ_d (resp. Γ_γ) is set to 2% of the total demands. In all the three networks, the solutions do not change when $\Gamma_d, \Gamma_\gamma \geq \frac{|\mathcal{D}|}{2}$, thus the x-axis is cut at 50%. We observe that energy saving are proportional to $1/\Gamma$. Indeed, large values of Γ reduces the interest for robust optimization. More precisely, when $\Gamma_d, \Gamma_\gamma \geq 30\%$, energy saving offered by the Robust-GreenRE model are almost the same as the GreenRE model, while when $\Gamma_d, \Gamma_\gamma \leq 20\%$ the Robust-GreenRE model allows for significant energy saving. An explanation of this phenomenon can be found in the distribution of the demand volumes. A small fraction of the demands dominates the others in volume. Hence, when the values of Γ_d, Γ_γ covers all of these dominating demands, increasing Γ_d, Γ_γ does not affect the routing solution and the percentage of energy saving remains stable. In Case 2 and Case 3, when only Γ_d or Γ_γ varies its value, the same phenomenon is observed. It means Γ_d and Γ_γ have almost the same role in contributing to the robustness of the network.

4.3.5.3 Link load



(a) Abilene (b) Geant

(c) Germany

Figure 4.15: CDF load on Abilene, Geant and Germany networks.

Intuitively, Robust-GreenRE would affect the utilization of links as fewer links are used to carry traffic. In this subsection, we evaluate the impact of Robust-GreenRE on link utilization. Specifically, we vary the level of robustness and see how the link utilization of the network is affected. We draw cumulative distribution function (CDF) of link load of Abilene, Geant and Germany networks in Fig. 4.15. For ease of observation, we only show three cases of robustness for each network, the other cases follow similar curve patterns. As shown in Fig. 4.15, Geant and Germany networks have light traffic load. For instance, 80% of links of Geant and Germany networks are under 40% and 20% of link utilization, respectively. Traffic on Abilene network is heavier, however there is no overloaded link and 80% of links are less than 70% of utilization. It is noted that when we consider a certain value of

Γ_d and Γ_γ, the traffic load is computed as the case where Γ_d demand volumes and Γ_γ RE rates are at their peak values. In fact, this is the worst case scenario in the range of the allowing fluctuation defined by Γ_d and Γ_γ. In this case, the total traffic on a link is computed as the value of the left hand side of the constraint (4.3'). For this reason, the computed traffic load is low when the level of robustness is low. For example, in Abilene network, for the case of 5% robustness, 85% of links are under 40% utilization meanwhile for the case 20% (resp. 100%) robustness, it is only 60% (resp. 40%) of links are under 40% utilization.

4.3.5.4 Robust-GreenRE vs. GreenRE vs. Classical EAR

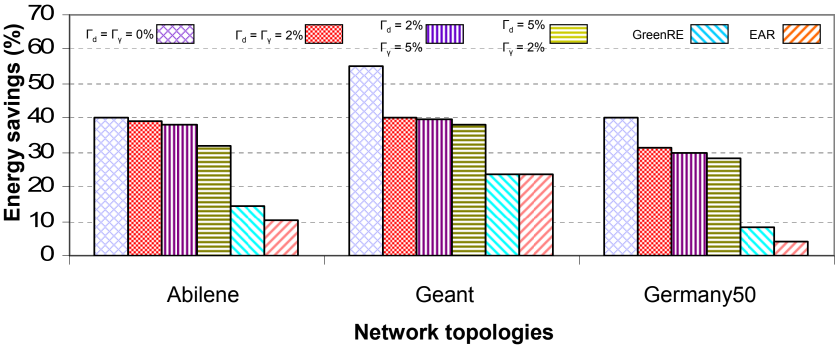


Figure 4.16: Robust-GreenRE vs. GreenRE vs. EAR.

In Fig. 4.16, we compare the Robust-GreenRE model with the GreenRE and the classical EAR (no compression) models for small values of Γ_d and Γ_γ. Since the GreenRE model does not take into account RE rate deviation, we set $\gamma^{st} = 0.8$ (20% of traffic is redundant) and for EAR model , γ^{st} is set to 1.0 (no compression). Furthermore, since traffic volume variations are not handled by GreenRE and EAR models, all demands are at peak. When $\Gamma_d = \Gamma_\gamma = 0\%$, all traffic demands are at their nominal values, the Robust-GreenRE model becomes the GreenRE model with nominal traffic demands, namely the $GreenRE_{nominal}$. Therefore, energy saving of the Robust-GreenRE model is in between that of the GreenRE and the $GreenRE_{nominal}$ models. We observe that, in Germany50 network, the EAR and the GreenRE models offer a small amount of energy saving. A prognosis is difficult to give, since energy saving is depended on both the network topology and the traffic matrix. One point can be used to explain the phenomenon is that the volume of peak traffic in Germany50 network is much bigger than the nominal one (the average ratio of the peak over the nominal traffic is around 6). That is why the Robust-GreenRE model can save much higher energy consumption than the EAR and the GreenRE model. It is noted that the Robust-GreenRE is more efficient than the GreenRE when only few traffic demands fluctuate their volumes and RE rates (Γ is relatively small). When Γ is quite big, e.g. $\Gamma >= 20\%$, the Robust-GreenRE and the GreenRE models yield almost the same amount of energy saving (as shown in Fig. 4.14). However, this result does not invalidate the benefit of Γ-robustness because in real-life traffic, only a few demands can vary their traffic simultaneously [KKR13]. In summary, when $\Gamma = 2-5\%$, the Robust-GreenRE model outperforms the other models and allows for $16 - 28\%$ additional energy saving in all the considered networks.

4.4 Conclusion

In this chapter, we formally define and model the Robust-GreenRE problem. Taking into account the uncertainties of traffic volumes and redundancy elimination rates, the Robust-GreenRE model provides a more accurate evaluation of energy saving for backbone networks. Based on real-life traffic traces, we have shown a significant improvement of energy saving compared with other models. As future work, we shall investigate implementation issues and impacts of Robust-GreenRE model on QoS and fault tolerance.

4.5 Bibliography

[AABP07] A. Altin, E. Amaldi, P. Belotti, and M. C. Pinar, *“Provisioning Virtual Private Networks under Traffic Uncertainty”*, Networks **49** (2007), 100–115.

[AGA+08] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker, *“Packet Caches on Routers: the Implications of Universal Redundant Traffic Elimination”*, ACM Special Interest Group on Data Communication (SIGCOMM), 2008, pp. 219–230.

[AMAR09] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, *“Redundancy in Network Traffic: Findings and Implications”*, ACM Special Interest Group for the Computer Systems Performance Evaluation Community (SIGMETRICS), 2009, pp. 37–48.

[BS03] D. Bertsimas and M. Sim, *“Robust Discrete Optimization and Network Flows”*, Mathematical Programming **98** (2003), 49 – 71.

[BS04] D. Bertsimas and M. Sim, *“The Price of Robustness”*, Operations Research **52** (2004), 35 – 53.

[CMN11] L. Chiaraviglio, M. Mellia, and F. Neri, *“Minimizing ISP Network Energy Cost: Formulation and Solutions”*, IEEE/ACM Transaction in Networking **20** (2011), 463 – 476.

[CSB+08] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, *“Power Awareness in Network Design and Routing”*, IEEE International

Conference on Computer Communications (INFOCOM), 2008.

[GMMO10] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy, *“Minimizing Routing Energy Consumption: from Theoretical to Practical Results”*, IEEE/ACM Green Computing and Communications (GreenCom), 2010, pp. 252–259.

[GMPR12] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, *“Minimization of Network Power Consumption with Redundancy Elimination”*, International Conferences on Networking (IFIP NETWORKING), 2012, pp. 247–258.

[GS03] M. Gupta and S. Singh, *“Greening of the Internet”*, ACM Special Interest Group on Data Communication (SIGCOMM), 2003, pp. 19–26.

[IBM] IBM ILOG, *CPLEX Optimization Studio 12.4*.

[KKR11] A. M. C. A. Koster, M. Kutschka, and C. Raack, *“On the Robustness of Optimal Network Designs”*, IEEE International Conference on Communications (ICC), 2011, pp. 1 – 5.

[KKR13] A. M. C. A. Koster, M. Kutschka, and C. Raack, *“Robust Network Design: Formulation, Valid Inequalities, and Computations”*, Networks **61** (2013), 128 – 149.

[KPT13] A. Koster, T. K. Phan, and M. Tieves, *“Extended Cutset Inequalities for the Network Power Consumption Problem”*, International Network Optimization Conference (INOC), Electronic Notes in Discrete Mathematics, 2013, pp. 69–76.

[OWPT10] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, *SNDlib 1.0 - survivable network design library*, Networks **55** (2010), no. 3, 276–286.

[ZYLZ10] M. Zhang, C. Yi, B. Liu, and B. Zhang, *“GreenTE: Power-aware Traffic Engineering”*, IEEE International Conference on Network Protocols (ICNP), 2010, pp. 21–30.

Chapter 5

Optimizing IGP Link Weights for Energy-efficiency

Contents

In this chapter, we consider to save energy with Open Shortest Path First (OSPF) protocol. From the perspective of traffic engineering, we argue that stability in routing configuration also plays an important role in QoS. In details, frequent changes in network configuration (link weights, slept and activated links) to adapt with traffic fluctuation in daily time cause network oscillation. We propose a novel optimization method of link weight so as to limit the changes in network configurations in multi-period traffic matrices. We formally define the problem and model it as Mixed Integer Linear Program (MILP). We then propose efficient heuristic algorithm that is suitable for large networks. Simulation results with real traffic traces on three different networks show that our approach achieves high energy saving and less pain for QoS (in term of less changes in network configuration).

5.1 Publications

This chapter corresponds to *Optimizing IGP Link Weights for Energy-efficiency in a Changing World* by J. Moulierac, and T. K. Phan which has been submitted to Computer Communication, 2014.

5.2 Introduction

Although green networking has been attracting a growing attention during the last years (see the surveys [BBDC11, BCRR12]), we found a limited number of recent works that have been devoted to both energy-aware routing (EAR) and shortest path routing [ACG13, FWMG13, SLX$^+$12, CCGS13, LTC12, CEL$^+$12, CEL$^+$10, CELP11, ACGM11]. These works consider the most widely used Internal Gateway Protocol (IGP) in IP networks, namely the Open Shortest Path First (OSPF) protocol. For energy efficiency, a set of link weights should be found so that its induced shortest paths use a minimal number of active links. Then, inactive network elements are put into sleep mode to save energy.

To deal with traffic variation, daily time periods are characterized by different traffic levels (e.g. morning, afternoon and night) and in each period, a single traffic matrix is assumed to be accurately collected. Then, each traffic matrix is associated with a corresponding weight setting configuration. As assumed in existing works [ACG13, FWMG13, SLX$^+$12, ACC$^+$14], as long as the network capacity is sufficient to handle all traffic demands, energy can be saved without causing service deteriorations to end users. Recall that [CCGS13] proposes to integrate [ACG13] in an off-line/on-line framework to guarantee both network responsiveness and prevent frequent oscillations. However, as explained in [CCRP13, FT02], frequent changes to link weights are highly undesirable and should be avoided as much as possible. First, applying a large number of configurations may result in frequent transitions between active and sleep modes of network links. This reduces the life cycle of network devices, since they are designed to be always powered on. Second, routing protocol convergence at the IP layer is affected. The weight changes have to be flooded in the network via control messages. The routers then recompute the shortest paths and update their routing tables. This may take seconds before all routers agree on the new shortest paths. Meanwhile, in this transient time, packets may arrive out of order, degrading the perceived QoS for end-users. We refer the reader to [BR01] for a detailed analysis of the stability issues in OSPF. In general, the more weight changes we try to flood simultaneously, the more chaos we introduce in the network [FT02].

In this chapter, we propose some methods to reduce the number of changes in weight setting for the multi-period energy-aware traffic engineering problem. In summary, we make the following contributions:

- We formally define and formulate the stable weight setting for multi-period traffic matrices using Mixed Integer Linear Program (MILP). The objective is to limit the changes in weight setting in the transition between traffic matrices.

- We also present a MILP robust formulation so that even a single weight setting can be feasible for a set of traffic matrices. Different from the stable weight setting, the robust one avoids any weight changes for the multi-period traffic matrices with the assumption that only a limited number of traffic demands are at their peaks simultaneously.
- We propose heuristic algorithms that are effective for large networks for both the stable weight setting and the robust methods.
- Using real-life data traffic traces, we show that our methods achieve high energy saving while reducing a large number of network reconfigurations in daily traffic variation.

The rest of this chapter is structured as follows. We summarize related works in Section 5.3. Then, our approaches to deal with traffic variation are introduced in Section 5.4. Evaluation results are presented in Section 5.5. Finally, we conclude the work in Section 5.6.

5.3 Related Work

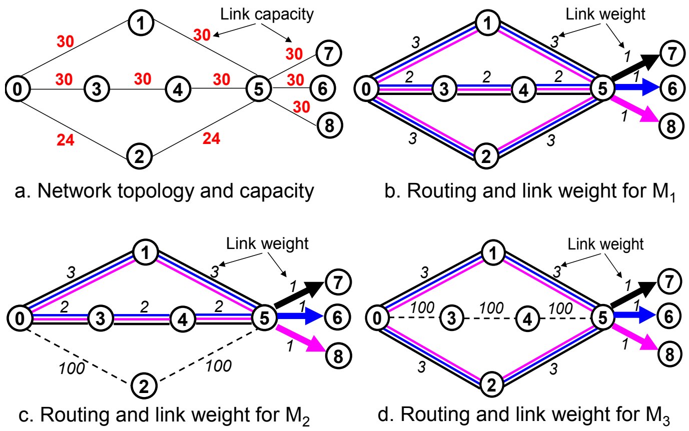
5.3.1 Optimizing Weight Setting for EAR

The problem of optimizing the OSPF weight setting is known to be NP-hard, exact formulation and heuristic algorithm have been proposed in literature [FT02, FT00]. EAR routing can be applied to a network by setting an appropriate link weight setting. By assigning high weights to a set of links, no traffic passes through them and these links can be put into sleep mode to save energy.

Table 5.1: Traffic matrices for OSPF/ECMP

Traffic matrix	Traffic demand		
	$(0,6)$	$(0,7)$	$(0,8)$
M_1	30	30	10
M_2	20	20	10
M_3	20	10	10

To better explain, we consider an example of a network topology with capacity on links as shown in Fig. 5.1a. There are 3 traffic demands and we collect their values at 3 different periods, leading to 3 traffic matrices M_1, M_2 and M_3 (Table 5.1). The routing solutions in Fig. 5.1 follow OSPF/ECMP (Equal-cost multi-path) policy: a traffic demand flowing through a node i is equally split among all the interfaces connected to i which belong to at least one shortest path toward the considered destination. As shown in Fig. 5.1b, the three traffic demands are split into 3 different paths from 0 to 5, each path carries $(30 + 30 + 10)/3 = 70/3 < 24$. So, this routing is feasible but zero link can sleep. When traffic decreases, we can have better solutions. For example, 2 and 3 links are put in sleep mode for M_2 (Fig. 5.1c) and M_3 (Fig. 5.1d), respectively.



a. Network topology and capacity

b. Routing and link weight for M$_1$

c. Routing and link weight for M$_2$

d. Routing and link weight for M$_3$

Figure 5.1: Example of OSPF/ECMP for EAR

5.3.1.1 Mixed Integer Linear Program (MILP)

The MILP of optimizing the OSPF weight setting problem was proposed in [ACG13]. In this section, we reformulate the MILP to fit with our robust model. The MILP uses the notations detailed on Table 5.2.

Table 5.2: Notations

w_{max}	the maximum value of a link weight.
M	a large enough constant. It can be set $M = 2w_{max}$.
D	a set of all traffic demands to be routed.
D_t	a set containing all destination nodes.
$\mathcal{D}^{st} \in D$	demand of the traffic flow from s to t.
C_{uv}	capacity of a link (u, v).
$\mu \in (0, 1]$	maximum link utilization that can be tolerated. It is normally set to a small value, e.g. $\mu = 0.5$.
$N(u)$	the set of neighbors of u in the network graph G.
k^t_{uv}	binary variable to determine if the link (u, v) belongs to one of the shortest paths from node u to node t.
z^{st}_u	variable to represent fraction of the flow (s, t) to be routed on outgoing node u using ECMP.
r^t_u	cost of the shortest path from u to t.
x_{uv}	binary variable to indicate if the link (u, v) is active or not.
f^{st}_{uv}	a flow (s, t) that is routed on the link (u, v).

$$\min \sum_{(u,v)\in E} x_{uv} \tag{5.1}$$

$$\text{s.t.} \sum_{v \in N(u)} \left(f_{vu}^{st} - f_{uv}^{st}\right) = \begin{cases} -1 & \text{if } u = s, \\ 1 & \text{if } u = t, \\ 0 & \text{else} \end{cases} \quad \forall u \in V, (s,t) \in \mathcal{D} \tag{5.2}$$

$$\sum_{(s,t) \in \mathcal{D}} D^{st}(f_{uv}^{st} + f_{vu}^{st}) \leq \mu C_{uv} x_{uv} \qquad \forall (u,v) \in E \tag{5.3}$$

$$0 \leq z_u^{st} - f_{uv}^{st} \leq 1 - k_{uv}^t \qquad \forall (s,t) \in \mathcal{D}; (u,v) \in E \tag{5.4}$$

$$f_{uv}^{st} - k_{uv}^t \leq 0 \qquad \forall (s,t) \in \mathcal{D}; (u,v) \in E \tag{5.5}$$

$$1 - k_{uv}^t \leq r_v^t + w_{uv} - r_u^t \leq (1 - k_{uv}^t)M \qquad \forall t \in D_t; (u,v) \in E \tag{5.6}$$

$$k_{uv}^t - x_{uv} \leq 0 \qquad \forall t \in D_t; (u,v) \in E \tag{5.7}$$

$$w_{uv} \geq (1 - x_{uv}) w_{max} \qquad \forall (u,v) \in E \tag{5.8}$$

$$x_{uv} + w_{uv} \leq w_{max} \qquad \forall (u,v) \in E \tag{5.9}$$

$$1 \leq w_{uv} \leq w_{max} \qquad \forall (u,v) \in E \tag{5.10}$$

$$x_{uv}, k_{uv}^t \in \{0,1\}; \quad f_{uv}^{st}, z_u^{st} \in [0,1]; \quad r_u^t \geq 0 \tag{5.11}$$

where w_{max} is the maximum value of a link weight. M is a large enough constant, it can be set $M = 2w_{max}$. D_t is a set containing all destination nodes. The objective function (5.1) minimizes the power consumption of the network represented by the number of active links. Constraints (5.2) establish the classical flow conservation constraints. We consider an undirected link capacity model [RKOW11] in which the capacity of a link is shared between the traffic in both directions. Constraints (5.3) limit the available capacity of a link (where μ denotes the maximum link utilization). The binary variable $k_{uv}^t = 1$ if and only if the link (u,v) belongs to one of the shortest paths from node u to node t. Constraints (5.4) are for ECMP routing. It makes sure that if $k_{uv}^t = 1$ then the flow f_{vu}^{st} destined to node t is equal to z_u^{st}, which is the common value of the flow assigned to all links outgoing from u and belonging to one of the shortest paths from u to t. Constraints (5.5) force $f_{uv}^{st} = 0$ for all links (u,v) that do not belong to a shortest path to node t. The variable r_u^t represents the cost of the shortest path from u to t. Constraints (5.6) compute weight of the link (u,v) if it belongs to the shortest path from u to t. Constraints (5.7) force link (u,v) to be on if it belongs to the shortest path from u to t. Note that, we do not force $x_{uv} = 0$ when $k_{uv}^t = 0$ because if (u,v) belongs another shortest path to t_1 (in this case $k_{uv}^{t_1} = 1$ and x_{uv} should be equal to 1). Constraints (5.8)-(5.10) guarantee that if a link weight is equal to w_{max}, then this link should be put into sleep mode.

5.3.1.2 Heuristic Algorithm

Finding optimal OSPF weight setting that deals with energy saving and/or traffic engineering issues is very challenging. We found in literature many works trying to solve this problem using heuristic approaches. For example, the authors in [FT00, ACG13] have proposed to use local search by iteratively modifying the OSPF weights so as to achieve the objective. The authors in [FWMG13] have used genetic algorithms to find the link weights for the joint-optimization of load-balancing and energy efficiency. As the traffic matrices are considered independently, these algorithms can find different sets of link weights in the optimization process for each traffic matrix. We call these methods as *freely changed weight setting*.

5.3.2 Γ-Robust Network Design

Over the past years, robust optimization has been established as a special branch of mathematical optimization allowing to handle uncertain data [BTGN09]. A specialization of robust optimization, which is particularly attractive by its computational tractability, is the so-called Γ-robustness concept introduced by Bertsimas and Sim [BS04]. Based on an observation that in real traffic traces, at a given time, only few of the demands are simultaneously at their peaks [KKR13, ZWLW14, WYW+12]. For instance, Fig. 5.2 shows real traffic traces of the three source-destination pairs: (a) Washington D.C. - Los Angeles, (b) Seattle - Indianapolis, and (c) Seattle - Chicago in the US Abilene Internet2 network in intervals of 5 mins during the first 10 days of July 2004 [KKR13]. We observe that there is no point that all the three demands are at the peak values at the same time. Thus, it confirms the assumption: it is unlikely that all the traffic demands assume their peak values simultaneously.

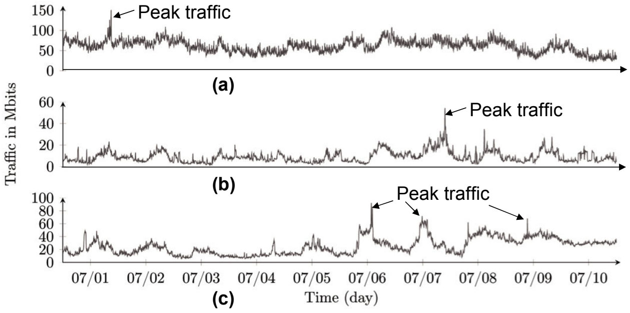
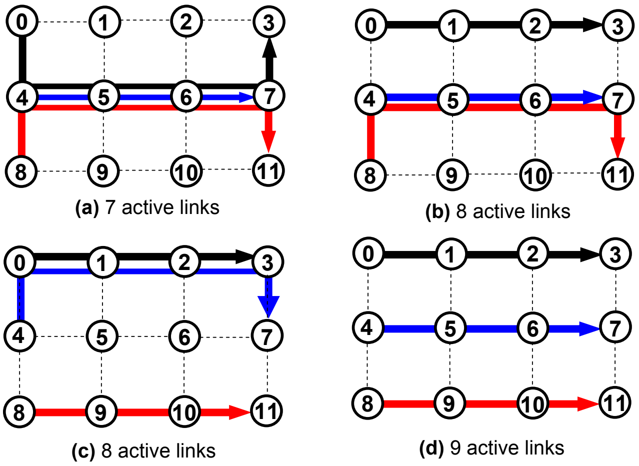


Figure 5.2: Traffic demands in Abilene network [KKR13]

Γ-robust network design allows to choose an integer parameter $\Gamma \geq 0$ so that at most Γ traffic demands can be at their peak values simultaneously. Note that, the model only limits a number of traffic demands (but not exactly which ones) that can be at their peaks at the same time. Therefore, from a practical perspective, by varying the parameter Γ, different solutions can be obtained with different levels of robustness. This concept has already been applied to several network optimization problems [KKR11, ACC+13, CKPT13].

To better explain, we consider an example in Fig. 5.3. We use a grid 3×4, each link has a capacity 4 Gbps. There are 3 traffic demands, each has a nominal and peak values (in Gbps) as shown in Table 5.3.

As an example, assume that $\Gamma = 2$, meaning that zero, one or two traffic demands can be at their peak values simultaneously. This leads to a combination of seven



(a) 7 active links **(b)** 8 active links **(c)** 8 active links **(d)** 9 active links

Figure 5.3: Example of EAR and Γ-Robust network

Table 5.3: Traffic demand variation

Demand (s, t)	Nominal value	Peak value
(0, 3)	1	4
(4, 7)	1	3
(8, 11)	1	2

Table 5.4: Example of robustness: $\Gamma = 2$

Case	Q	Best solution	Link load l_{uv} (Gbps)
1	{}	Fig. 1a (7 links)	$l_{0,4} = l_{7,3} = 1$, $l_{4,5,6,7} = 3$, $l_{8,4} = l_{7,11} = 1$
2	{(0, 3)}	Fig. 1b (8 links)	$l_{0,1,2,3} = 4$, $l_{4,5,6,7} = 2$, $l_{8,4} = l_{7,11} = 1$
3	{(4, 7)}	Fig. 1b or 1c (8 links)	$l_{0,1,2,3} = 1$, $l_{4,5,6,7} = 4$, $l_{8,4} = l_{7,11} = 1$ (Fig. 1b)
4	{(8, 11)}	Fig. 1a (7 links)	$l_{0,4} = l_{7,3} = 1$, $l_{4,5,6,7} = 4$, $l_{8,4} = l_{7,11} = 2$
5	{(0, 3), (4, 7)}	Fig. 1b (8 links)	$l_{0,1,2,3} = 4$, $l_{4,5,6,7} = 4$, $l_{8,4} = l_{7,11} = 1$
6	{(0, 3), (8, 11)}	Fig. 1b (8 links)	$l_{0,1,2,3} = 4$, $l_{4,5,6,7} = 3$, $l_{8,4} = l_{7,11} = 2$
7	{(4, 7), (8, 11)}	Fig. 1c (8 links)	$l_{0,1,2,3} = 4$, $l_{4,0} = l_{3,7} = 3$, $l_{8,9,10,11} = 2$

possibilities that are shown in Table 5.4. The set Q includes these seven cases.

It is easy to see that in Case 1, all traffic demands are at nominal values, hence as EAR, Fig. 5.3a is the best solution with only 7 active links. In Case 2, when $(0, 3)$ is at peak (4 Gbps), solutions in Fig. 5.3b and Fig. 5.3d are feasible. However, Fig. 5.3b is the best solution since only 8 active links are used. Similarly, in Case 7, solutions in Fig. 5.3c and Fig. 5.3d are feasible and Fig. 5.3c is the best one. In summary, Table 5.4 shows the complete possibilities of traffic variation and the corresponding best solution when $\Gamma = 2$. However, since we just limit the number demands (but not any specific demands) to be deviated, a feasible solution should be the one that satisfies all the seven cases. Therefore, Fig. 5.3d is the only feasible solution for $\Gamma = 2$. It is also easy to check that, if we limit $\Gamma = 1$ (less robust), Fig. 5.3b is the best solution. From these examples, we can see that, depending on the desired robustness of a network, a single routing solution can be feasible for many traffic matrices.

In this work, our goal is to avoid weight changes as much as possible between multi-period traffic matrices while minimizing energy consumption for the networks. The main contributions are presented in Section 5.4 where we proposed some methods to stabilize the OSPF weight setting (called *stable weight setting* and Γ-*Robust approach*). To give an idea of energy saving, we have implemented a simple *freely changed weight setting* algorithm (in Section 5.4.1.3) to compare with the *stable weight setting* and the Γ-*Robust approach.*

5.4 Optimizing OSPF Weight in Multi-period Traffic Matrices

5.4.1 Stable Weight Setting

In this approach, multi-period traffic matrices are used to capture the daily traffic pattern. However, these traffic matrices are not considered independently. The idea is that, when changing from a high to a lower traffic matrix (traffic load is reducing), we only consider to sleep unused links. In other words, any set of active links for low traffic is included in that of higher traffic. In addition, the weight setting of remaining links are unchanged. The reason is to limit changes in routing configuration and reduce network oscillations that affect QoS. As we add restrictions, *stable weight setting* has less potential in saving energy than the *freely changed weight setting* approach. For instance, as the example in Fig. 5.1, when traffic changes from M_2 to M_3, the *stable weight setting* can not have solution like Fig. 5.1d as both turning on and off links are necessary.

Since we try to stabilize the weight setting based on the previously used one, a question is how to find an initial weight setting that will be used for all the matrices of the multi-period traffic matrices. In fact, there are many ways to set link weights in practice. For instance, Cisco uses the inverse of link capacity [Cisco05]; or more complicated load-balancing traffic engineering methods can be found in [FT02, ACG13, FWMG13]. Actually, the initial set of weight has an impact on the energy saving in daily time. We can use the freely changed weight method to find a good configuration for a traffic matrix. However, this configuration may not be good for subsequent traffic matrices and how to find a good starter for the whole day traffic variation is beyond the scope of this chapter. In this work, the network operators are free to choose their own weight setting. Anytime they would like to start energy saving mode, the stable algorithm can be applied directly using the current weight setting configuration as the initial one. We propose an optimization formulation and heuristic algorithms for the *stable weight setting* method as follows:

5.4.1.1 Stable Weight MILP

The inputs are network topology $G = (V, E)$, traffic matrix D and a set of current link weights W. The output is a routing solution that minimizes the number of active links so that it satisfies constraints (5.2) - (5.11). Meanwhile, as the weight setting W should not be modified, following constraints should be added to the model (5.1) - (5.11):

$$w_{uv} - w^*_{uv} \geq (1 - x_{uv})(w_{max} - w^*_{uv}) \quad \forall (u, v) \in E \tag{5.12}$$

$$w^*_{uv} - w_{uv} \geq (x_{uv} - 1)w_{max} \quad \forall (u, v) \in E \tag{5.13}$$

We note w^*_{uv} as the current weight of the link (u, v). Constraints (5.12) - (5.13) are used to force the new link weight w_{uv} to be equal to w^*_{uv} if the link (u, v) is still used in the new routing solution. That is, if $x_{uv} = 1$, then $w_{uv} = w^*_{uv}$. Otherwise, when $x_{uv} = 0$, w_{uv} is set to w_{max}. This means that the link (u, v) does not belong to a shortest path routing. Thus, there is no traffic on the link (u, v) and it can be put into sleep mode.

5.4.1.2 Stable Weight Heuristic

The stable weight setting problem is also very challenging for large networks. We propose in this section heuristic algorithms that can find feasible solution in an acceptable time. In brief, the heuristic algorithm includes *sleeping step* and *feasible routing check step* (Fig. 5.4).

There are many criteria to choose a link (u, v) to sleep (see in [ACG13, CMN11]). In this chapter, we propose to choose the *min load link* to sleep since this approach has been successfully applied in literature [ACG13, GMMO10, GMPR12, CMN11]. After the *sleeping step*, the *feasible routing check* step has inputs which are a subgraph G', a subset of link weights W' and the same traffic matrix D. We perform OSPF/ECMP routing for D on G' and check if some links are overloaded. If yes, the routing is not feasible, we mark the slept link as checked and go back to the *sleeping step* to find another link to make sleeping (the checked links will not be chosen). If the routing is feasible, we update the inputs and go back to the *sleeping step* to continue. This procedure is repeated until all links on the network are checked.

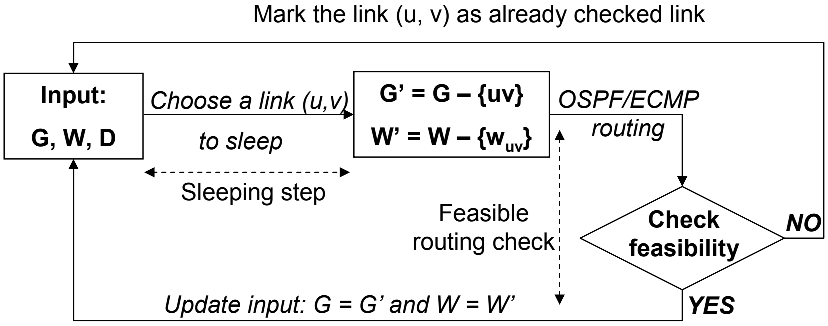


Figure 5.4: Heuristic diagram

To deal with multi-period traffic matrices, we first sort the traffic matrices in non-increasing order of traffic load, that is from D_n to D_1. The traffic load is computed as the sum of the volumes of all traffic demands in a traffic matrix. Then, we run the MILP or heuristic algorithm for D_n to find a feasible network configuration (link weight setting, set of links to sleep). Given this configuration as the inputs, we find new feasible configuration for D_{n-1} in which we consider only to sleep links and the remaining links keep the same weight setting. This process is repeated until we reach D_1. Following the traffic variation of daily time, from a low to higher traffic matrices (e.g. D_i to D_{i+1}), we simply apply the configuration that has been found (from D_{i+1} to D_i). In this scenario, only slept links are woken up and the remaining links keep the same weight setting.

5.4.1.3 Freely Changed Weight Heuristic

In order to compare the energy saving of the *stable weight heuristic*, we implemented a simple *freely changed weight heuristic* algorithm. Using the same diagram as in Fig. 5.4, at the *feasible routing check* step, we follow the idea of local search used in [FT00]. If the routing is infeasible, instead of marking the link as “checked” (impossible to sleep), we repeat the local search step, trying to find another feasible weight setting. The main idea in each iteration is that we increase the weights of the overloaded links to redirect traffic to other links with the hope that a new feasible routing solution can be found. Depending on the execution time of the algorithm, we can define a maximum number of loops for the local search. If there is still no feasible solution at the end of the iteration, we mark that link as “checked”, then the algorithm repeats the sleeping step described in Fig. 5.4 with another chosen link.

5.4.2 Γ-Robust Approach: One Network Configuration for All

We propose in this sub-section a method to find a single network configuration with a set of active links and weight setting that is feasible for all the considered traffic matrices.

5.4.2.1 Robust MILP

Assume that each traffic demand has a nominal $\overline{D}^{st}$ and a deviated value $\widehat{D}^{st}$ so as the peak value is $(\overline{D}^{st} + \widehat{D}^{st})$. Given a parameter $0 \leq \Gamma \leq |D|$, the robust model tries to find a feasible routing at minimal energy costs, while the link capacity constraints are satisfied if at most Γ traffic pairs simultaneously deviate from their nominal values $\overline{D}^{st}$. Note that $\Gamma = |D|$ amounts to worst-case optimization where all demands are at peak values. The straightforward robust capacity constraint for a given Γ and an edge $e \in E$ is:

$$\sum_{(s,t)\in D} \overline{D}^{st} f_e^{st} + \max_{\substack{Q\subseteq D\\ |Q|\leq\Gamma}} \Big\{ \sum_{(s,t)\in Q} \widehat{D}^{st} f_e^{st} \Big\} \qquad \leq \ \mu C_e x_e \qquad \forall e \in E \tag{5.14}$$

where $f_e^{st} = f_{uv}^{st} + f_{vu}^{st}$; Q is a subset containing demands that can be at peaks at the same time. The constraints (5.14) is non-linear since it contains the *max* notation. A trivial way to make it linear is to explicitly write down all the possibilities of the constraints, that is:

$$\sum_{(s,t)\in D} \overline{D}^{st} f_e^{st} + \sum_{(s,t)\in Q_i} \widehat{D}^{st} f_e^{st} \leq \ \mu C_e x_e \qquad \forall Q_i \subseteq D; |Q_i| \leq \Gamma; e \in E \tag{5.14$'$}$$

Obviously, the constraints (5.14′) is a combination of all possibilities of a subset Q_i which has the size $|Q_i| \leq \Gamma$. Therefore, it is impossible to put all the constraints into the MILP model at one time when the set of demand D is large. To overcome this problem, we apply the method Γ-robustness (introduced by Bertsimas and Sim [BS04]). The main idea of this method is to use LP duality to make a *compact* formulation, so that it is possible to solve the MILP. We present step-by-step the procedure to form the *compact* formulation as follows.

Assume that we know the value of f_e^{st} (then they are constants), the maximum part of (5.14) can be computed by the following ILP:

$$\beta(f, \Gamma) := \max \sum_{(s,t)\in D} \widehat{D}^{st} f_e^{st} z_e^{st} \tag{5.15}$$

$$\text{s.t.} \sum_{(s,t)\in D} z_e^{st} \leq \Gamma \qquad [\pi_e] \tag{5.16}$$

$$z_e^{st} \in \{0, 1\} \qquad [\rho_e^{st}] \tag{5.17}$$

where the primal binary variables z_e^{st} denote whether or not f_e^{st} is part of the subset $Q \subseteq D$. As proposed by Bertsimas and Sim, we employ LP duality with the dual variables π_e and ρ_e^{st} corresponds to the constraint $\sum_{(s,t)\in D} z_e^{st} \leq \Gamma$ and $z_e^{st} \leq 1$, respectively. The LP duality for $\beta(g, \Gamma)$ is as follows:

$$\beta(g, \Gamma) := \min \Big(\Gamma\pi_e + \sum_{(s,t)\in D} \rho_e^{st} \Big) \tag{5.18}$$

$$\text{s.t. } \pi_e + \rho_e^{st} \geq \widehat{D}^{st} f_e^{st} \qquad \forall (s,t) \in D \tag{5.19}$$

$$\rho_e^{st}, \pi_e \geq 0 \qquad \forall (s,t) \in D \tag{5.20}$$

Since the constraints (5.18)–(5.20) are linear. A *compact* reformulation can be obtained by embedding them into (5.1)–(5.13). As a result, the robust stable weight setting can be compactly formulated as (5.1)–(5.2), (5.4)–(5.13) and replace (5.3) by:

$$\sum_{(s,t)\in D} (\overline{D}^{st} f_e^{st} + \rho_e^{st}) + \Gamma \pi_e \leq \mu x_e C_e \qquad \forall e \in E \tag{5.21}$$

$$\pi_e + \rho_e^{st} \geq \widehat{D}^{st} f_e^{st} \qquad \forall (s,t) \in D; \forall e \in E \tag{5.22}$$

$$\rho_e^{st}, \pi_e \geq 0 \qquad \forall (s,t) \in D; \forall e \in E \tag{5.23}$$

5.4.2.2 Robust Heuristic Algorithm

The main idea of the heuristic algorithm is similar to the diagram in Fig. 5.4. However, it is difficult to check routing feasibility since we do not know explicitly which traffic demands are at peak values. To deal with this problem, we use the ILP constraints (5.21)–(5.23) for the *feasible routing check step* as they represent the robust capacity constraints. In details, consider a simplified MILP of the *robust weight setting* in which we only keep constraints (5.11) (remove variables k_{uv}^t, z_u^{st}, r_u^t), and (5.21)–(5.23) with $x_e \in \{0,1\}$ and $f_e^{st} \in [0,1]$ $\forall (s,t) \in D$; $\forall e \in E$. The OSPF/ECMP routing on G' with a set of link weight W' implicitly satisfies the flow conservation constraint. In addition, we have in hand a set of link weight W', therefore all the constraints (5.2) – (5.10) are not needed in the simplified ILP. When performing OSPF/ECMP routing for the subgraph G' (after the *sleeping step*), we can get all the values of f_e^{st} and x_e ($x_e = 0$ if $f_e^{st} = 0$ $\forall (s,t) \in D$, otherwise $x_e = 1$). Given them as the inputs, the variables x_e and f_e^{st} in the simplified MILP are now fixed, only ρ_e^{st} and π_e remain variables. Since the simplified MILP is used only to verify routing solution, we ignore the objective function and simply set it to *min* 0. To check routing feasibility, we run the simplified MILP with inputs: G', D, Γ, f_e^{st} and x_e, if a feasible solution is returned, it means that the routing solution satisfies the robust capacity constraints. Then, we go back to the *sleeping step* and continue the algorithm as in Fig. 5.4.

5.5 Computational Evaluation

We solved the MILP models with IBM CPLEX 12.4 solver [IBM]. All computations were carried out on a computer equipped with 2.7 Ghz Intel Core i7 and 8 GB RAM. We consider real-life traffic traces collected from the SNDlib [OWPT10]: the U.S. Internet2 Network (Abilene) ($|V| = 12, |E| = 15, |D| = 130$), the Geant network ($|V| = 22, |E| = 36, |D| = 387$) and the Germany50 ($|V| = 50, |E| = 88, |D| = 1595$).

In our test instances, five traffic matrices ($D1 - D5$) are used to represent daily traffic pattern (Fig. 5.5). From the SNDlib, we collect the mean and max traffic matrices (all traffic demands are at their mean and maximum values). Since traffic load is low, we use the mean traffic matrix as $D1$. To achieve a network with high

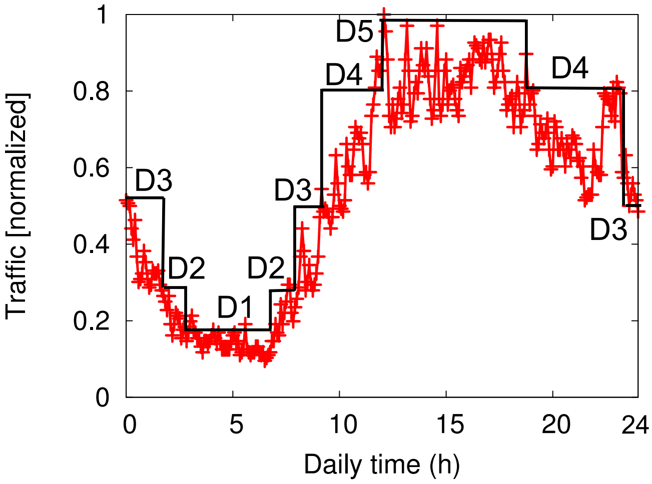


Figure 5.5: Daily traffic

link utilization, we scale the max traffic matrix with a factor of 1.3, 1.5, 1.8 and 2.0, and they form $D2-D5$, respectively. As a result, we represent $D5$ as the worst case scenario of highest traffic load. It is noted that, in realistic, even at peak hour, not all the traffic demands are at their maximum values as the case $D5$. In all test cases, as an approach of traffic engineering, we use a local search heuristic to find a set of link weights that minimize the maximum link load [FT00] for the traffic matrix $D5$. This weight setting is used as the initial one in the *stable weight* approaches.

5.5.1 Computation time

Table 5.5: Abilene network - optimal solutions

	Execution time (s)				
	D1	D2	D3	D4	D5
Stable weight MILP	≤ 5	≤ 5	≤ 5	≤ 5	≤ 5
Robust MILP	90				
Freely changed weight MILP	95	874	100	12900	20700

Table 5.6: Geant network - heuristic solutions

	Execution time (s)				
	D1	D2	D3	D4	D5
Stable weight	139	140	160	182	256
Robust stable weight	283				
Freely changed weight	62	157	1596	2115	3600

Table 5.7: Germany network - heuristic solutions

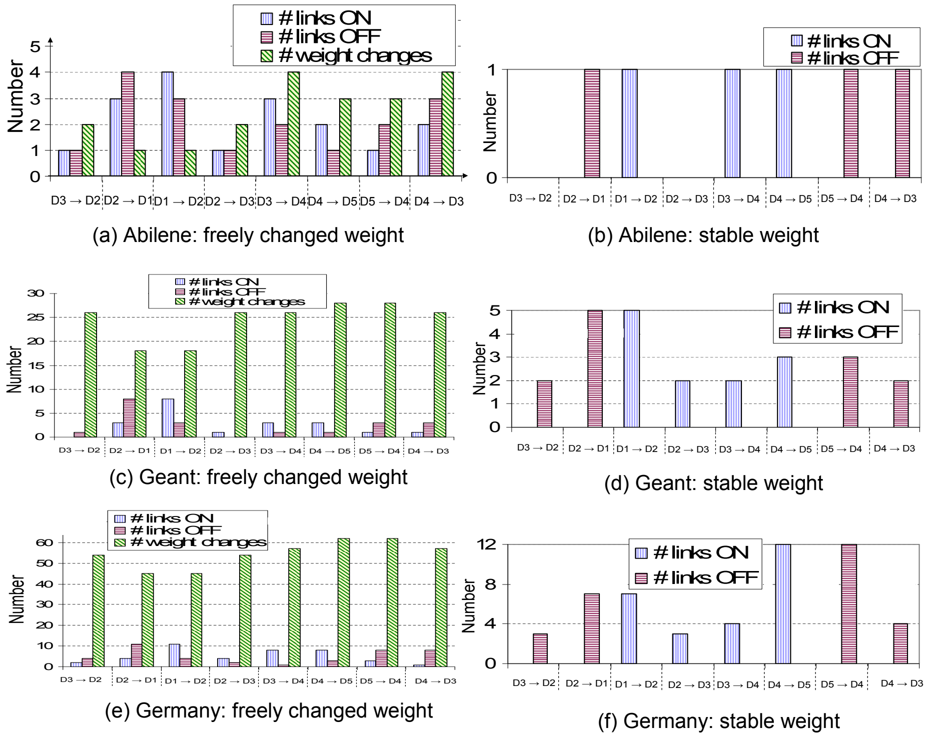
	Execution time (s)				
	D1	D2	D3	D4	D5
Stable weight	468	1586	1787	2108	3108
Robust stable weight	3090				
Freely changed weight	1739	3600	3600	3600	3600

For Abilene network, we can find *optimal solution using the MILP* for the three methods (stable, robust and freely changed weight). For larger networks (e.g. Geant, Germany50), only the heuristic algorithms are used to find solutions. The execution time of the *freely changed weight* heuristic is limited to one hour by varying the number of loops in the local search. For the *robust stable weight*, we run with different values of Γ and get an average running time.

It is clear that the *stable weight* and *robust* methods win a lot in running time. This is because these methods are based on an initial weight setting and we limit the change. Note that, we also use an initial weight setting for the *robust* case to limit network reconfiguration when changing from the normal (currently used) mode to the energy-aware mode. Thus, solution search space is small and optimal solutions can be found quite fast. Similar observation can be found for the heuristic approaches (Tables 5.6 and 5.7): the *stable weight* and *robust* methods take less than 1 hour for all test cases, meanwhile the execution time of the *freely changed weight* reaches the time limit set to 1 hour.

5.5.2 Stability of routing solutions

Fig. 5.6 shows changes in routing when shifting between periods of traffic during daily time. For the three tested networks, the *stable weight* approach always outperforms the *freely changed weight*. The former approach only allows to sleep links (resp. only wake up links) when changing from a high traffic matrix to a lower one (resp. from a low to a higher traffic matrix). However, for *freely changed weight*, there is no restriction, link can be turned on and off and also the weight setting of remaining links can be changed. For instance, in Abilene network, from $D3$ to $D2$, even the energy saving (and the number of active links) is unchanged, the solution allows one link to turn off, one link to turn on and two active links change their weights. Similar observation can be found for Geant and Germany networks (Fig. 5.6c - Fig. 5.6f). The larger the network we consider, the more chaos we introduce as more changes happen between multi-period traffic matrices.



(a) Abilene: freely changed weight

(b) Abilene: stable weight

(c) Geant: freely changed weight

(d) Geant: stable weight

(e) Germany: freely changed weight

(f) Germany: stable weight

Figure 5.6: Changes in freely changed weight vs. stable weight methods

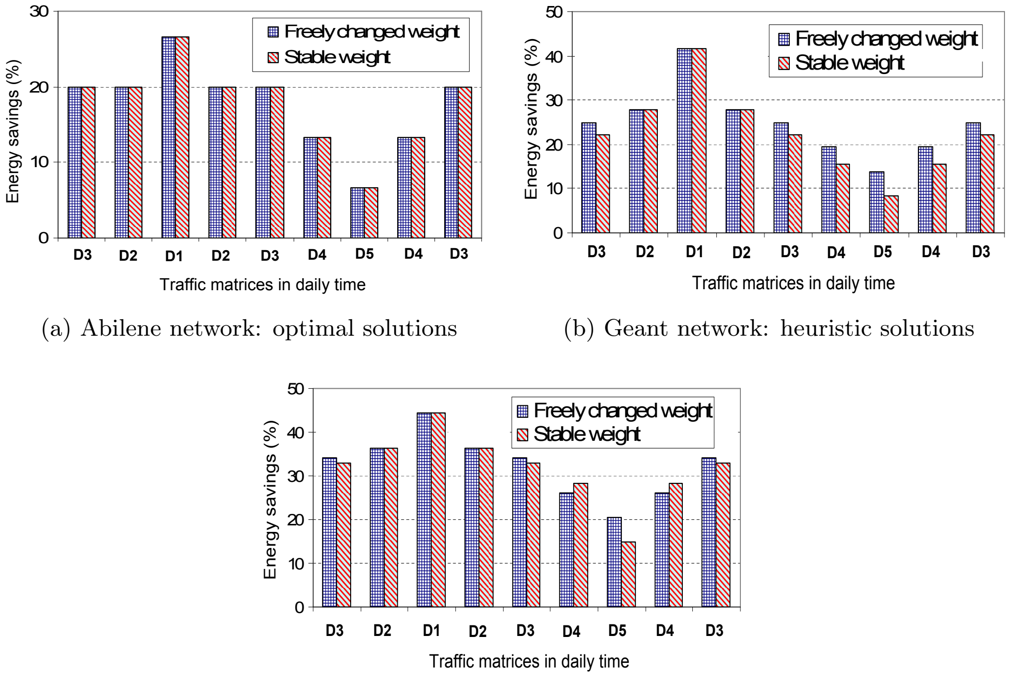
5.5.3 Energy saving in daily time

5.5.3.1 Stable weight vs. freely changed weight

Follow the curve of daily traffic, we show energy saving of the three networks in Fig. 5.7. It is clear that energy saving is high when traffic load is low since more links can be put into sleep mode to save energy. To compare between *stable weight* and *freely changed weight* approaches, the latter one can save more energy because it is flexible to change the weight setting. This can be observed in $D3, D4, D5$ in Fig. 5.7b and $D3, D5$ in Fig. 5.7c. Abilene network is small and only a few links (from 1 to 4 links) can sleep, thus the solutions between the two methods are similar. It is noted that, in $D4$ (Fig. 5.7c), *stable weight* method even has better result. It is because we limit the number of loops so that the heuristic algorithm is finished after one hour. Thus, it is possible for the *freely changed weight* heuristic to stop before finding a better solution than the *stable weight* approach. It can happen when the network is large, the algorithm needs to do several loops to find a good solution.

5.5.3.2 Robust vs. stable weight approaches

Fig. 5.8 shows energy saving of the *stable weight setting* vs. the $\Gamma-$*robustness* (with different value of Γ) in daily time traffic variation. Following the pattern of daily traffic, the stable weight can turns off many links and save much energy when traffic



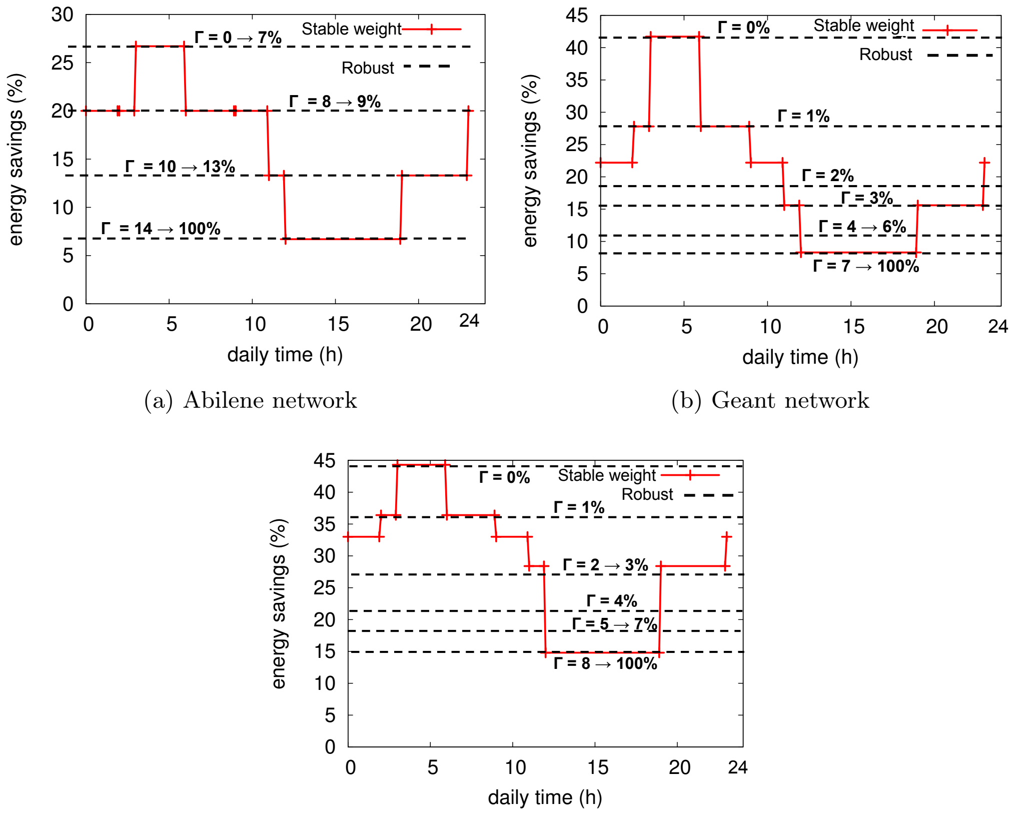
(a) Abilene network: optimal solutions

(b) Geant network: heuristic solutions

(c) Germany network: heuristic solutions

Figure 5.7: Energy saving in multi-period traffic matrices

load is low and vice versa. For the robust solution, defending on the γ-value, we assume only one configuration for all the traffic matrices. That is why the robust solution keeps the same amount of energy saving for the whole day. Simulation results confirm that the higher Γ is, the more robust, but the less power saving the solution is. Note that, when $\Gamma = 100\%$, the robust model becomes the worst case of the deterministic - the case with $D5$ (all traffic demands are at their peak values). In all the three networks, the solutions do not change when Γ is large enough (e.g. $\Gamma = 14\%$ for Abilene network). It is because in real traffic, only a small fraction of the demands dominates the others in volume. Hence, when the values of Γ covers all of these dominating demands, increasing Γ does not affect the routing solution and the percentage of energy saving remains the same. To give a visualized comparison, we also draw energy saving of the *stable weight* method in daily time. For instance, from Fig. 5.8a, if $\Gamma = 9\%$, it is possible to have only one weight setting that gives feasible routing if at most 9% of traffic demands are at their peaks simultaneously. Moreover, this single weight setting allows to save the same amount of energy as when we apply the *stable weight* method for $D2$ or $D3$ matrices. Similar observations can be found for Geant (Fig. 5.8b) and Germany network (Fig. 5.8c). However, Geant and Germany networks are more sensitive with traffic variation, significant energy saving is found only with small Γ.



(a) Abilene network

(b) Geant network

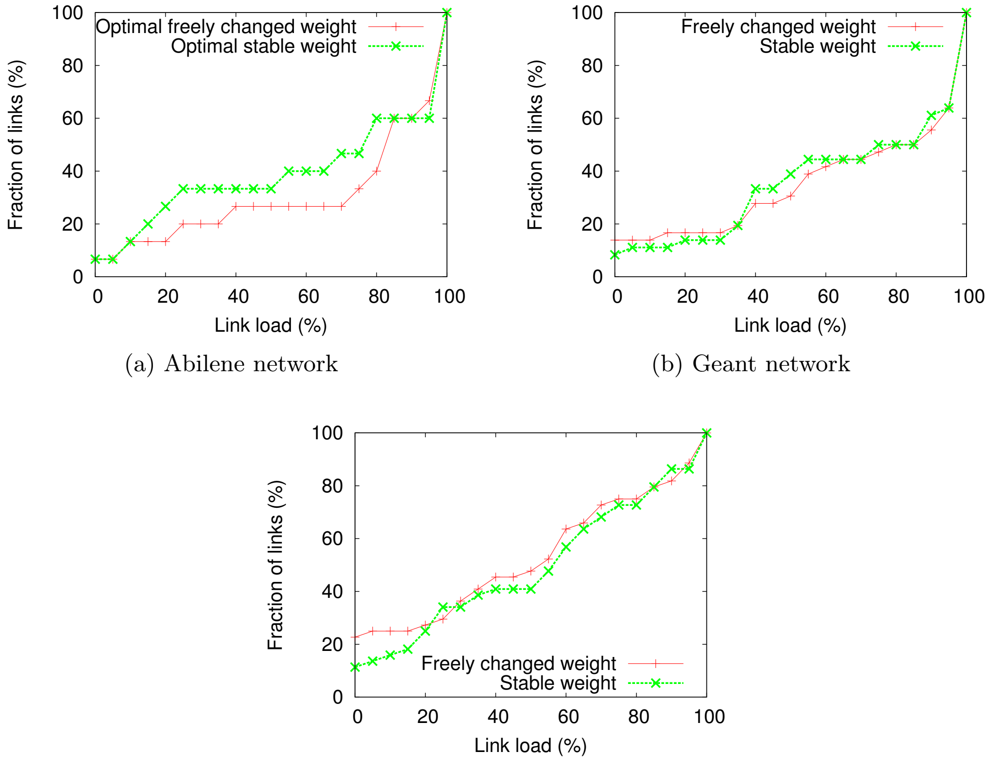
(c) Germany network

Figure 5.8: Robust weight vs. stable weight

5.5.4 Traffic load

5.5.4.1 Stable weight vs. freely changed weight

In the simulation, we set the maximum link utilization $\mu = 100\%$. Intuitively, EAR would affect the utilization of links as fewer links are used to carry traffic. In this subsection, we evaluate the impact of EAR on link utilization. We draw the cumulative distribution function (CDF) of link load of Abilene, Geant and Germany networks in Fig. 5.9. To test the worst case scenario, we use the highest traffic matrix ($D5$). Since we guarantee capacity constraints, no link is overloaded. Our goal is not load balancing, thus it is not easy to validate the *freely changed weight* and the *stable weight* methods, which one is better. However, from Fig. 5.9a, the *stable weight* method is slightly better, e.g. 60% of links have link utilization less than 80%, meanwhile it is only 40% of links for the *freely changed weight* method. This can be explained as the *stable weight* method uses an initial load-balancing link weight which is the one that minimizes the maximum link load.



(a) Abilene network

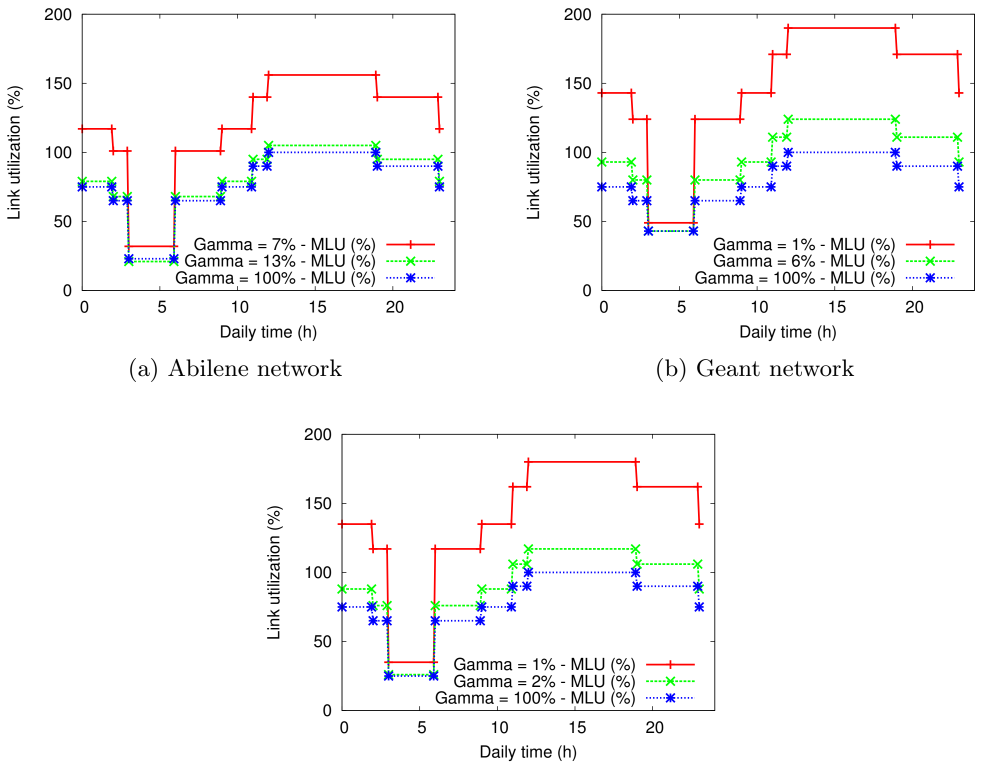
(b) Geant network

(c) Germany network

Figure 5.9: Link load of robust weight vs. stable weight

5.5.4.2 Robust approach

For each value of Γ, we find a link weight setting that satisfies the capacity constraint if at most Γ demands are at peaks at the same time. However, we would like to test what will happen if we use a single robust network configuration while traffic is varied in daily time. Fig. 5.10 shows the maximum link utilization over all active links in the network for different values of Γ. Obviously, if we use $\Gamma = 100\%$, we can find a single network configuration that is feasible (no overloaded link) for all-day traffic variation. However, the price of this solution is too expensive: e.g. only 6% of energy can be saved for the Abilene network (like the case $D5$). However we observe that, even with $\Gamma = 1\%$, the maximum link utilization of the three networks is less than 200%. It means that if we carefully set the value of μ in the capacity constraints (e.g. $\mu = 50\%$), then the robust solution with $\Gamma = 1\%$ can be feasible for all-day traffic variation. Moreover, if one network configuration for the whole day is too conservative, a daily traffic can be divided into few periods. Then, each period is applied with a single robust configuration. For instance, in Abilene network, we can use 4 periods: $2h - 9h$ ($\Gamma = 7\%$); $9h - 11h$ ($\Gamma = 13\%$); $11h - 19h$ ($\Gamma = 100\%$ - $D5$ traffic matrix) and $19h - 2h$ ($\Gamma = 13\%$). However, it may save less energy with respect to the *stable weight* approach as daily traffic is divided into 9 periods which



(a) Abilene network

(b) Geant network

(c) Germany network

Figure 5.10: Maximum link utilization (MLU) of robust solution in daily traffic

allows to find more flexible routing solutions.

5.6 Conclusion

To the best of our knowledge, this is the first study considering the stability of routing solution in energy-aware traffic engineering using OSPF protocol. We argue that, in addition to capacity constraints, the requirements on routing stability also play an important role in QoS. Moreover, using real traffic traces in the simulations, we show that our *stable weight* and *robust* methods are able to save a significant amount of energy. For future work, we will focus on how to find a good initial weight setting. Moreover, efficient heuristic algorithms with different policies for putting links into sleep mode should be considered.

5.7 Bibliography

[ACC+13] B. Addis, A. Capone, G. Carello, L. G. Gianoli, and B. Sansò, *"A Robust Optimization Approach for Energy-aware Routing in MPLS Networks"*, International Conference on Computing, Networking and Communications (ICNC), 2013, pp. 567–572.

[ACC+14] B. Addis, A. Capone, G. Carello, L. G. Gianoli, and B. Sansò, *"Energy Management Through Optimized Routing and Device Powering for Greener Communication Networks"*, IEEE/ACM Transactions on Networking **22** (2014), 313 – 325.

[ACG13] E. Amaldi, A. Capone, and L. .G. Gianoli, *"Energy-aware IP Traffic Engineering with Shortest Path Routing"*, Computer Networks **57** (2013), 1503–1517.

[ACGM11] E. Amaldi, A. Capone, L. G. Gianoli, and L. Mascetti, *"Energy Management in IP Traffic Engineering with Shortest Path Routing"*, IEEE WoWMoM, 2011, pp. 1–6.

[BBDC11] 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 Communication Surveys and Tutorials **13** (2011), 223 – 244.

[BCRR12] A. P. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, *"A Survey of Green Networking Research"*, IEEE Communication Surveys and Tutorials **14** (2012), 3 – 20.

[BR01] A. Basu and J. Riecke, *"Stability Issues in OSPF Routing"*, ACM Special Interest Group on Data Communication (SIGCOMM), vol. 31, 2001, pp. 225–236.

[BS04] D. Bertsimas and M. Sim, *"The Price of Robustness"*, Operations Research **52** (2004), 35 – 53.

[BTGN09] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *"Robust optimization"*, Princeton Series in Applied Mathematics, Princeton University Press, 2009.

[CCGS13] A. Capone, C. Cascone, L. G. Gianoli, and B. Sansò, *"OSPF Optimization via Dynamic Network Management for Green IP Networks"*, Sustainable Internet and ICT for Sustainability (SustainIT), 2013, pp. 1–9.

[CCRP13] L. Chiaraviglio, A. Cianfrani, E. L. Rouzic, and M. Polverini, *"Sleep Modes Effectiveness in Backbone Networks with Limited Configurations"*, Computer Networks **57** (2013), 2931–2948.

[CEL+10] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini, *"An Energy Saving Routing Algorithm for a Green OSPF Protocol"*, IEEE

INFOCOM Workshop, 2010, pp. 1–5.

[CEL+12] A. Cianfrani, V. Eramo, M. Listanti, M. Polverini, and A. V. Vasilakos, *"An OSPF-Integrated Routing Strategy for QoS-Aware Energy Saving in IP Backbone Networks"*, IEEE Transactions on Network and Service Management **9** (2012), 254 – 267.

[CELP11] A. Cianfrani, V. Eramo, M. Listanti, and M. Polverini, *"An OSPF Enhancement for Energy Saving in IP Networks"*, IEEE INFOCOM Workshop, 2011, pp. 325–330.

[Cisco05] *Cisco: Ospf design guide*, www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/7039-1.html, 2005.

[CKPT13] D. Coudert, A. Koster, T. K. Phan, and M. Tieves, *"Robust Redundancy Elimination for Energy-aware Routing"*, IEEE International Conference on Green Computing and Communications (GreenCom), 2013, pp. 179–186.

[CMN11] L. Chiaraviglio, M. Mellia, and F. Neri, *"Minimizing ISP Network Energy Cost: Formulation and Solutions"*, IEEE/ACM Transaction in Networking **20** (2011), 463 – 476.

[FT00] B. Fortz and M. Thorup, *"Internet Traffic Engineering by Optimizing OSPF Weights"*, Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 2, 2000, pp. 519–528.

[FT02] B. Fortz and M. Thorup, *"Optimizing OSPF/IS-IS Weights in a Changing World"*, IEEE Journal on Selected Areas in Communications **20** (2002), 756–767.

[FWMG13] F. Francois, N. Wang, K. Moessner, and S. Georgoulas, *"Green IGP Link Weights for Energy-efficiency and Load-balancing in IP Backbone Networks"*, International Conferences on Networking (IFIP NETWORKING), 2013, pp. 1–9.

[GMMO10] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy, *"Minimizing Routing Energy Consumption: from Theoretical to Practical Results"*, IEEE/ACM Green Computing and Communications (GreenCom), 2010, pp. 252–259.

[GMPR12] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, *"Minimization of Network Power Consumption with Redundancy Elimination"*, International Conferences on Networking (IFIP NETWORKING), 2012, pp. 247–258.

[IBM] IBM ILOG, *CPLEX Optimization Studio 12.4*.

[KKR11] A. M. C. A. Koster, M. Kutschka, and C. Raack, *"On the Robustness of Optimal Network Designs"*, IEEE International Conference on Communications (ICC), 2011, pp. 1 – 5.

[LTC12] S. S. W. Lee, P. Tseng, and A. Chen, *"Link Weight Assignment and Loop-free Routing Table Update for Link State Routing Protocols in Energy-aware Internet"*, Future Generation Computer Systems **28** (2012), 437–445.

[OWPT10] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, *SNDlib 1.0 - survivable network design library*, Networks **55** (2010), no. 3, 276–286.

[RKOW11] C. Raack, A. M. C. A. Koster, S. Orlowski, and R. Wessäly, *"On Cut-based Inequalities for Capacitated Network Design Polyhedra"*, Networks **57** (2011), 141 – 156.

[SLX+12] M. Shen, H. Liu, K. Xu, N. Wang, and Y. Zhong, *"Routing On Demand: Toward the Energy-Aware Traffic Engineering with OSPF"*, International Conferences on Networking (IFIP NETWORKING), 2012, pp. 232–246.

[ZWLW14] K. Zheng, X. Wang, L. Li, and X. Wang, *"Joint Power Optimization of Data Center Network and Servers with Correlation Analysis"*, IEEE INFOCOM, 2014, pp. 2598–2606.

Chapter 6

Energy-aware Routing with Software-Defined Networks

Contents

In this chapter, we focus on using Software-Defined Network (SDN) for energy-aware routing (EAR). Since traffic load has a small influence on power consumption of routers, EAR allows to put unused links into sleep mode to save energy. SDN can collect traffic matrix and then computes routing solutions satisfying QoS while being minimal in energy consumption. However, prior works on EAR have assumed that the table of OpenFlow switch can hold an infinite number of rules. In practice, this assumption does not hold since the flow table is implemented with Ternary Content Addressable Memory (TCAM) which is expensive and power-hungry. In this work, we propose an optimization method to minimize energy consumption for a backbone network while respecting capacity constraints on links and rule space constraints on routers. In details, we present an exact formulation using Integer Linear Program (ILP) and introduce efficient greedy heuristic algorithm. Based on computations, we show that using this smart rule space allocation, it is possible to save almost as much power consumption as the classical EAR approach.

6.1 Publications

This chapter corresponds to *Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing* by F. Giroire, J. Moulierac, and T. K. Phan which has been accepted for publication in the proceeding of IEEE Global Communications Conference (GlobeCom), 2014.

6.2 Introduction

Software-defined networking (SDN) in general, and OpenFlow in particular [MAB$^+$08], has been attracting a growing attention in the networking research community in recent years. In traditional networks, network devices such as routers and switches act as "closed" systems. They work as "black boxes" with applications implemented on them. Users can only control them via limited and vendor-specific control interfaces. Moreover, since the data plane (forwarding function) and control plane are integrated, it is difficult for current network infrastructure to evolve (e.g. to deploy new network protocols). SDN is a new networking paradigm that decouples the control plane from the data plane. It provides a flexibility to develop and test new network protocols and policies in real networks. Over past few years, many applications have been built using the OpenFlow API [MAB$^+$08].

In this chapter, we focus on one application of the OpenFlow, that is to use OpenFlow to minimize power consumption for an Internet service provider (ISP). As shown in literature, many existing works have used OpenFlow as a traffic engineering approach to deploy EAR in a network [HSM$^+$10][WYW$^+$12]. In these works, the flow table of each switch is assumed to hold an infinite number of rules. In practice, however, this assumption does not hold, and rule space becomes a significant bottleneck for large-scale SDN networks. It is because the flow table is implemented using Ternary Content Addressable Memory (TCAM) which is expensive and power hungry. Therefore, commodity switches typically support just from few hundreds to few thousands of entries [KLRW13][KHK13][SCF$^+$12]. Taking this limitation into account, we show that the rule space constraints are very important in EAR. An inefficient rule allocation can lead to an unexpected routing solution, causing network congestion and affecting QoS. In summary, we make the following contributions:

- To our best knowledge, this is the first work that defines and formulates the optimizing rule space problem in OpenFlow for EAR using ILP.
- As EAR is known to be NP-hard [GMMO10], we propose heuristic algorithm that is effective for large network topologies. By evaluation, we show that the heuristic algorithm achieves close-to-optimal solutions obtained by the ILP.
- Using real-life data traffic traces from SNDlib [OWPT10], we quantify energy savings achieved by our approaches. Moreover, we also present other QoS aspects such as routing length of EAR solutions.

The rest of this chapter is structured as follows. We summarize related work in Section 6.3. We present the ILP and heuristic algorithm in Section 6.4. Computation results are shown in Section 6.5. Finally, we conclude and propose future work in Section 6.6 and 6.7.

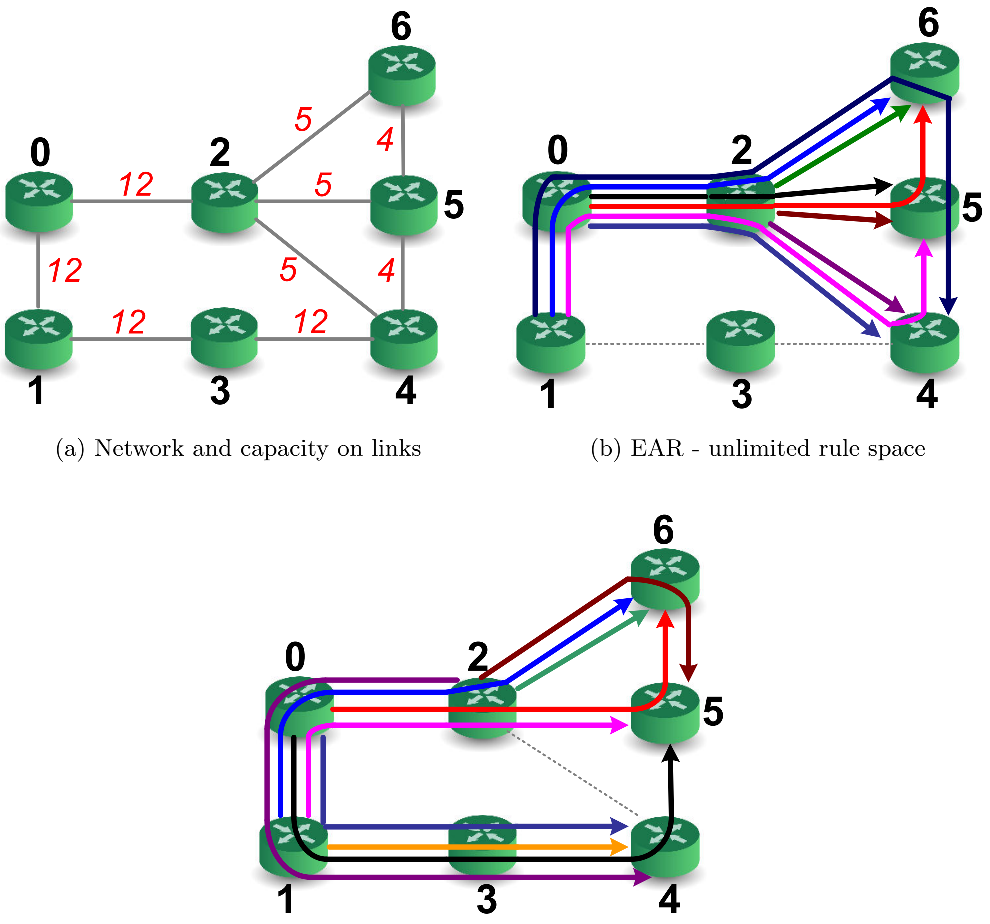
6.3 Related Work

6.3.1 Limited Rule Space in OpenFlow Switches

To support a vast range of network applications, OpenFlow rules are more complex than forwarding rules in traditional IP routers. For instance, access-control requires matching on source - destination IP addresses, port numbers and protocol [CFP+09] whereas a load balancer may match only on source and destination IP prefixes [WBR11]. These complicated matching can be well supported using TCAM since all rules can be read in parallel to identify the matching entries for each packet. However, as TCAM is expensive and extremely power-hungry, the on-chip TCAM size is typically limited. Many existing works in literature have tried to address this limited rule space problem. For instance, the authors in [MLT12] have proposed algorithms to reduce the number of rules needed to realize policies on a single switch. However, to the best of our knowledge, no work in literature solves the rule-placement problem for EAR. The closest papers to our work are [KLRW13] and [KHK13]. These works present efficient heuristic rule-placement algorithms that distribute forwarding policies while managing rule-space constraints at each switch. However, they do not rely on the exact meaning of the rules and the rules should not determine the routing of the packet. For instance, the work in [KHK13] focusing on access control, i.e. when a router receives a packet, it should decide whether to forward or to drop the packet. The goal of their work is to express all the access control rules with the constraint on routing table sizes at routers.

6.3.2 Energy Savings with OpenFlow

Starting from the pioneering work of Gupta [GS03], the idea of power proportionality has gained a growing attention in networking research area [BCRR12][CMN11]. Since power consumption of router is independent from traffic load, people suggested putting network components to sleep in order to save energy. OpenFlow is a promising method to implement EAR in a network. Without setting entries manually, OpenFlow can collect traffic matrix, performs routing calculation and then installs new routing rules on routers. For instance, the authors in [HSM+10] have implemented and analyzed ElasticTree on a prototype testbed built with production OpenFlow switches. The idea is to use OpenFlow to control traffic flows so that it minimizes the number of used network elements to save energy. Similarly, the authors in [WYW+12] have set up a small testbed using OpenFlow switches to evaluate energy savings for their model. OpenFlow switches have also been mentioned in existing work as an example of the traffic engineering method to implement the EAR



(a) Network and capacity on links

(b) EAR - unlimited rule space

(c) EAR - limited rule space

Figure 6.1: Example of EAR with and without rule space constraints

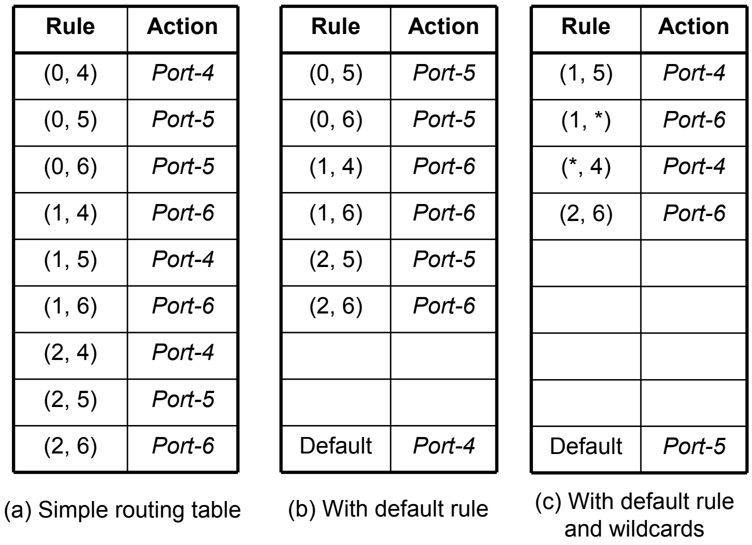
idea [CMTY11]. However, as we can see, the testbed setups with real OpenFlow switches are quite small. For instance, in [HSM+10], 45 virtual switches onto two 144-port 5406 chassis switches are used; or in [WYW+12], there is a testbed with 10 virtual switches on a 48-port Pronto 3240 OpenFlow-enabled switch. We argue that when deploying EAR in real network topologies, much more real OpenFlow switches should be used and they have to handle a large amount of traffic flows. In this situation, limited rule space in switches becomes a serious problem since we can not route traffic as expected. Therefore, we present in next Section a novel optimization method to overcome the rule placement problem of OpenFlow for EAR.

6.4 Optimizing Rule Placement

Routing decision of an OpenFlow switch is based on flow tables implemented with TCAM. Each entry in the flow table defines a matching rule and is associated with an action. Upon receiving a packet, a switch identifies the highest-priority rule with a matching predicate, and performs the corresponding action. A packet that matches no rule is processed using the *default rule*, which has the lowest-priority. Depending on applications of OpenFlow, the default rule can be "drop packets" or "forward packets to the controller" over the OpenFlow channel. In this work, to avoid delay communication between routers and the centralized controller, we consider that the *default rule* is "forward packets to a default port" (without contacting the controller), and each switch has exactly one default port [NMN+13].

Table 6.1: Traffic demands and routing solutions

Traffic demand	Volume	Routing solution (Fig. 6.1b)	Routing solution (Fig. 6.1c)
(0, 4)	1	0 - 2 - 4	0 - 1 - 3 - 4
(0, 5)	2	0 - 2 - 5	0 - 1 - 3 - 4 - 5
(0, 6)	2	0 - 2 - 5 - 6	0 - 2 - 5 - 6
(1, 4)	1	1 - 0 - 2 - 6 - 5 - 4	1 - 3 - 4
(1, 5)	3	1 - 0 - 2 - 4 - 5	1 - 0 - 2 - 5
(1, 6)	3	1 - 0 - 2 - 6	1 - 0 - 2 - 6
(2, 4)	1	2 - 4	2 - 0 - 1 - 3 - 4
(2, 5)	1	2 - 5	2 - 6 - 5
(2, 6)	1	2 - 6	2 - 6



Rule	Action
(0, 4)	*Port-4*
(0, 5)	*Port-5*
(0, 6)	*Port-5*
(1, 4)	*Port-6*
(1, 5)	*Port-4*
(1, 6)	*Port-6*
(2, 4)	*Port-4*
(2, 5)	*Port-5*
(2, 6)	*Port-6*

(a) Simple routing table

Rule	Action
(0, 5)	*Port-5*
(0, 6)	*Port-5*
(1, 4)	*Port-6*
(1, 6)	*Port-6*
(2, 5)	*Port-5*
(2, 6)	*Port-6*
Default	*Port-4*

(b) With default rule

Rule	Action
(1, 5)	*Port-4*
(1, *)	*Port-6*
(*, 4)	*Port-4*
(2, 6)	*Port-6*
Default	*Port-5*

(c) With default rule and wildcards

Figure 6.2: Routing table at router 2 for routing of Fig. 6.1b

We show in Fig. 6.1 how the limited rule space impacts EAR solution. Assume that there are 9 traffic demands with volumes as shown in Table 6.1. The network topology and capacity on links are shown in Fig. 6.1a. For ease of reading, Table 6.1 also shows the routing of each traffic flow in Fig. 6.1b and Fig. 6.1c. These routing solutions are found by using the ILP in Section 6.4.1. As the classical EAR approach, Fig. 6.1b shows an optimal solution since it satisfies capacity constraints and uses a minimum number of active links (7 links). It is noted that, as the objective of EAR is to minimize the number of used links, some traffic flows may be routed via long paths. For instance, the flow (1, 4) is routed via 5 hops while its shortest path is only 2 hops. One possible way to avoid this is to have some constraints that limit the stretch of the path of each flow.

Assume that the routing table of router contains rules which are the mapping of [(src, dest) : port-to-foward]. As the routing in Fig. 6.1b, the router 2 needs to forward 9 flows, hence a simple routing table can be as Fig. 6.2a. However, we can reduce the size of the routing table by using a default rule (Fig. 6.2b), or combining default rule and wildcards (Fig. 6.2c). Note that the rules [(0, 5): port-5] and [(0, 6): port-5] of Fig. 6.2b can not be combined as [(0, *): port-5]. Indeed, in this case, the flow $(0, 4)$ will go to port-5 when it should go to port-4. In Fig. 6.2c, as the rule $(1, 5)$ has higher priority than the rule (1, *), the flow $(1, 5)$ is forwarded to port-4 while the flows $(1, 4)$ and $(1, 6)$ are forwarded to port-6 with the rule (1, *). Assume that we implement EAR on SDN network where each router can install at most 4 rules. As a result, the router 2 can install only 3 distinct rules and 1 default rule. However, as we have shown, the minimum routing table contains 5 rules (Fig. 6.2c). Therefore, in this situation, some flows need to be routed using the default port. For instance, if the flow $(2, 6)$ in Fig. 6.2c goes to the default port-5, then the link $(2, 5)$ will be overloaded. It is also easy to check that, when the rule capacity is equal to 4 and with a set of active links as in Fig. 6.1b, it is not possible to find a routing solution that satisfies both capacity constraints on links and rule space constraints on routers. However, if we consider the rule space constraints as inputs of the problem, we can find a feasible solution as Fig. 6.1c. Actually, since we add more constraints, the EAR with rule space is able to save less energy with respect to the classical EAR. For instance, there is only 1 inactive link in Fig. 6.1c while we can turn off 2 links and save more energy in Fig. 6.1b.

As we have shown in this example, the limited rule space is very important in EAR. Inefficient rule placement can cause unexpected routing solution, and hence result in network congestion. To overcome this problem, we present in this section a precise formulation (Integer Linear Program) and heuristic approach for large networks. However, we note that the current algorithms can find optimal solution of energy consumption (or close-to-optimal if it is heuristic) if we consider the default rule but not the wildcard. If the rule space is scarce, we can apply the work of “compressing policy on a single switch” [MLT12] as a post-processing step to further reduce the routing table size.

6.4.1 Integer Linear Program

We consider a backbone network as an undirected graph $G = (V, E)$. The nodes in V describe routers and the edges in E present connections between those routers. We denote by D^{st} the demand of traffic flow from node s to node t such that $D^{st} \geq 0,\ s, t \in V, s \neq t$. We assume that the capacity of links and the rule space at routers are constant. The objective is to find a feasible routing for all traffic flows, respecting the capacity and the rule space constraints and being minimal in energy consumption.

We first define the following notations and then formulate the problem as Integer Linear Program:

- $\mathcal{D}$: a set of all traffic demands to be routed.
- $D^{st} \in \mathcal{D}$: demand of the traffic flow from s to t.
- C_{uv}: capacity of a link (u, v).
- $\mu \in (0, 1]$: maximum link utilization that can be tolerated. It is normally set to a small value, e.g. $\mu = 0.5$.
- C_u: maximum number of rules can be installed at router u.
- $N(u)$: the set of neighbors of u in the graph G.
- x_{uv}: binary variable to indicate if the link (u, v) is active or not.
- f^{st}_{uv}: a flow (s, t) that is routed on the link (u, v) by a distinct rule. We call f^{st}_{uv} as normal flow.
- g^{st}_{uv}: a flow (s, t) that is routed on the link (u, v) by a default rule. g^{st}_{uv} is called default flow to distinguish from the normal flow f^{st}_{uv} .
- k_{uv}: binary variable to indicate if the default port of the router u is to go to v or not.

$$\min \sum_{(u,v)\in E} x_{uv} \tag{6.1}$$

$$\text{s.t.} \sum_{v\in N(u)} \left(f^{st}_{vu} + g^{st}_{vu} - g^{st}_{uv} - f^{st}_{uv}\right) = \begin{cases} -1 & \text{if } u = s, \\ 1 & \text{if } u = t, \\ 0 & \text{else} \end{cases}$$

$$\forall u \in V, (s, t) \in \mathcal{D} \tag{6.2}$$

$$f_{uv}^{st} + f_{vu}^{st} + g_{uv}^{st} + g_{vu}^{st} \leq 1 \ \forall (u,v) \in E, (s,t) \in \mathcal{D} \tag{6.3}$$

$$\sum_{(s,t)\in\mathcal{D}} D^{st}(f_{uv}^{st} + f_{vu}^{st} + g_{uv}^{st} + g_{vu}^{st}) \quad \leq \mu C_{uv} x_{uv}$$

$$\forall (u,v) \in E \tag{6.4}$$

$$\sum_{(s,t)\in\mathcal{D}} \sum_{v\in N(u)} f_{uv}^{st} \leq C_u - 1 \qquad \forall u \in V \tag{6.5}$$

$$\sum_{v\in N(u)} k_{uv} \leq 1 \qquad \forall u \in V \tag{6.6}$$

$$g_{uv}^{st} \leq k_{uv} \qquad \forall (u,v) \in E, (s,t) \in \mathcal{D} \tag{6.7}$$

$$x_{uv}, f_{uv}^{st}, g_{uv}^{st}, k_{uv} \in \{0,1\} \quad \forall (u,v) \in E, (s,t) \in \mathcal{D} \tag{6.8}$$

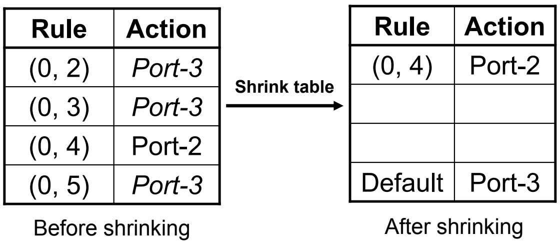
The objective function (6.1) minimizes the power consumption of the active links. The flow conservation constraints (6.2) express that the total flows entering and leaving a router are equal (except the source and the destination nodes). It is noted that a normal flow entering a router can become a default flow on outgoing link and vice versa. Constraints (6.3) ensure that a flow (s,t) on a link (u,v) cannot be both normal (f_{uv}^{st}) and default flow (g_{uv}^{st}) at the same time. Constraints (6.4) are capacity constraints. We consider an undirected link capacity model [RKOW11] in which the capacity of a link is shared between the traffic in both directions. Constraints (6.5) denote rule capacity constraints where we reserve one rule at each router to be the default rule. Constraints (6.6) and (6.7) are used to fix only one default port for each router.

6.4.2 Heuristic Algorithm

Since energy-aware routing problem is known to be NP-Hard [GMMO10], it is very challenging to find an exact solution. Therefore, we present in this section an efficient greedy heuristic for large networks. In summary, the heuristic algorithm works through two steps:

- Step 1: starting from the whole network, we compute a feasible routing which respects the capacity and the rule space constraints as described in Algorithm 5, 6 and 7. For each router $u \in V$, we keep the two sets $\mathcal{F}_u$ and $\mathcal{G}_u$ containing normal and default flows, respectively. At the beginning of the algorithm, we are freely to assign distinct rules for flows (line 10 - *Algorithm 5*) until the routing table is full ($|\mathcal{F}_u| = C_u$). Then, we try to shrink it (line 9 - *Algorithm 5*) as Fig. 6.3 by setting the port that carries most of traffic flows as the default port. As a result, the number of installed rules is reduced and there is some more space to install new rules.

- Step 2: remove in priority links that are less loaded (*Algorithm 8*). The aim of this step is to turn off the low loaded links and to accommodate their traffic on other links in order to reduce the total number of active links.



Rule	Action
(0, 2)	*Port-3*
(0, 3)	*Port-3*
(0, 4)	Port-2
(0, 5)	*Port-3*

Before shrinking

Shrink table

Rule	Action
(0, 4)	Port-2
Default	Port-3

After shrinking

Figure 6.3: Routing table at a router

An example of computing link weights (*Algorithm 6*) is shown in Fig. 6.4. The link weight is used to perform rule-balancing between routers. In line 3 of the algorithm 6, we set low weight for links connecting a node which has high available space in routing table. For instance, in Fig. 6.4, $w_{AC} < w_{AB}$ since node C has more available space in routing table than node B. For this reason, if the next demand is (A, D), then the routing solution using shortest path is (A, C, D) which is better than (A, B, D) since node C still has much more available rule space.

Algorithm 5: Finding a feasible routing

Input: An undirected graph $G = (V, E)$, link capacity C_e $\forall e \in E$, rule space capacity C_u $\forall u \in V$ and a set of demands $\mathcal{D}$.

Ouput: routing solution on graph G.

```
Residual capacity $R_e = C_e \ \forall e \in E$;
Initially, $\mathcal{F}_u = \emptyset$ and $\mathcal{G}_u = \emptyset \ \forall u \in V$;
Creating directed graph $G' = (V, E')$ from $G$ where $\forall (u,v) \in E$, we add both
directions $(u,v)$ and $(v,u)$ to $E'$ (Fig. 6.4). Initial weight on link
$w_e = 1 \ \forall e \in E'$;
while $D^{st} \in \mathcal{D}$ has no assigned route do
    find the shortest path $P^{st}$ on $G'$ such that $R_e \geq D^{st} \ \forall e \in P^{st}$;
    assign the routing $P^{st}$ to the demand $D^{st}$;
    update $R_e := R_e - D^{st} \ \forall e \in P^{st}$;
    update link weights proportionally to the size $|\mathcal{F}_u|$ as Algorithm 6;
    if $|\mathcal{F}_u| == C_u$, shrink table at $u \ \forall u \in P^{st}$;
    update $\mathcal{F}_u$ and $\mathcal{G}_u \ \forall u \in P^{st}$ using Algorithm 7;
end
return the routing (if it exists) assigned for $\mathcal{D}$
```

6.5 Computation Results

We solved the ILP model with IBM CPLEX 12.4 [IBM]. All computations were carried out on a computer equipped with 2.7 Ghz Intel Core i7 and 8 GB RAM. We consider real-life traffic traces collected from SNDlib [OWPT10]. To compare

Algorithm 6: Updating link weight

Input: An undirected graph $G = (V, E)$, a set of normal flows $\mathcal{F}_u$ $\forall u \in V$ and a maximum value of rule space capacity $C_{max} = max(C_u)$ $\forall u \in V$.

Ouput: weight setting on links of G'.

```
Create a digraph $G' = (V, E')$ as Fig. 6.4.
for $(u, v)$ in $E'$ do
    compute rule utilization at $v$: $U_v = C_{max} \times |\mathcal{F}_v|/C_v$;
    update $w_{uv} = max(U_v, 1)$;
end
```

Algorithm 7: Updating $\mathcal{F}_u$ and $\mathcal{G}_u$

Input: An undirected graph $G = (V, E)$, the shortest path P^{st} found in Algorithm 5, the set $\mathcal{F}_u$ and $\mathcal{G}_u$ $\forall u \in P^{st}$ and the default port of each router $d(u)$ $\forall u \in P^{st}$.

Ouput: Updated sets of $\mathcal{F}_u$ and $\mathcal{G}_u$ $\forall u \in V$.

```
for $u \in$ $P^{st}$ do
    for $v \in G.neighbor(u)$ do
        if $(u, v) \in P^{st}$ and $v == d(u)$ then
            $\mathcal{G}_u = \mathcal{G}_u \cup g^{st}_{uv}$
        else if $(u, v) \in P^{st}$ and $v \neq d(u)$ then
            $\mathcal{F}_u = \mathcal{F}_u \cup f^{st}_{uv}$
        end
    end
end
```

Algorithm 8: Removing less loaded links

Input: An undirected graph $G = (V, E)$, link capacity C_e and residual capacity R_e $\forall e \in E$.

Ouput: routing solution on a set of active links.

```
while edges can be removed do
    remove the edge $e$ that has not been chosen and has smallest value $C_e/R_e$;
    compute a feasible routing with the Algorithm 5;
    if no feasible routing exists, put $e$ back to $G$;
end
return the feasible routing if it exists.
```

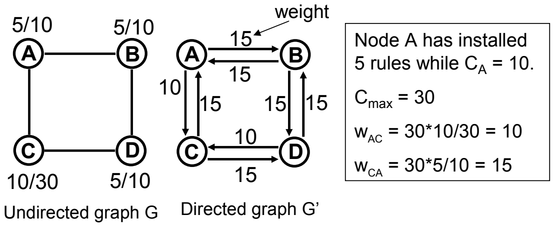


Figure 6.4: Example of updating link weight

the optimal and the heuristic solutions, we use a small network as Atlanta network ($|V| = 15, |E| = 22, |\mathcal{D}| = 210$). As mentioned in [KLRW13][KHK13], the routing table can support from 750 to few thousands of rules. To show that this is a realistic problem, we use three of the largest network topologies in SNDlib: ta2 (Telekom Austria: $|V| = 65, |E| = 108, |\mathcal{D}| = 4160$), zib54 (Zuse-Institut Berlin: $|V| = 54, |E| = 81, |\mathcal{D}| = 2862$) and germany50 ($|V| = 50, |E| = 88, |\mathcal{D}| = 2450$). In our test instances, five traffic matrices (D1 - D5) are used to represent daily traffic pattern (similar to Chapter 5). Since traffic load is low, the traffic matrix obtained from SNDlib is considered as $D1$. To achieve a network with high link utilization, we scale $D1$ with a factor of 1.5, 2.0, 2.5 and 3.0, and they form D2 - D5, respectively. Note that we can play with finer granularity traffic matrices by changing these factors.

6.5.1 Optimal vs. Heuristic Solutions

Assume that each router on the network has the same rule capacity represented by $C_u = (p \times |\mathcal{D}|)$ where $p \in (0, 1]$ and $|\mathcal{D}|$ is the total demands. The value of C_u is varied as we change the parameter p. Energy savings is computed as the number of links to sleep divided by the total number of links on the network ($|E|$).

Table 6.2: Atlanta network (optimal solution with default port)

Rule capacity (p - %)	energy savings (%)	computation time (s)
5%	9	2200
10%	22.7	9000
20%	22.7	445
30%	22.7	540
40%	22.7	300
100%	22.7	400

As shown in Table 6.2 and Table 6.3, the ILP model can find solution for very limited rule space ($p = 5\%$) while the heuristic algorithm needs $p = 16\%$ (if p is less than this value, no feasible solution can be found). Similarly, the heuristic algorithm is able to save less energy when p is small (e.g. $p \leq 16\%$). However, when the rule space capacity is large enough, e.g. $p = 30\%$, the gaps between the optimal

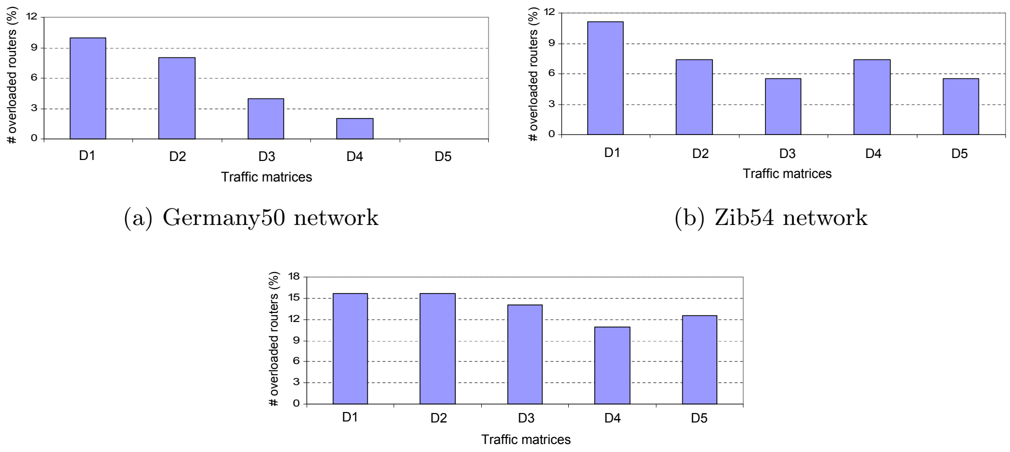
Table 6.3: Atlanta network (heuristic solution)

Rule capacity (p - %)	energy savings (%)	computation time (s)
16%	4.5	< 10
20%	13.6	< 10
30%	18.2	< 10
40%	18.2	< 10
100%	18.2	< 10

and the heuristic solutions are small. Moreover, the heuristic algorithm is better in computation time. For instance, the ILP model can take up to 9000 (s) to find solution while it is always less than 10 (s) for the heuristic algorithm.

6.5.2 Heuristic Solutions for Large Networks

6.5.2.1 Rule allocation at routers



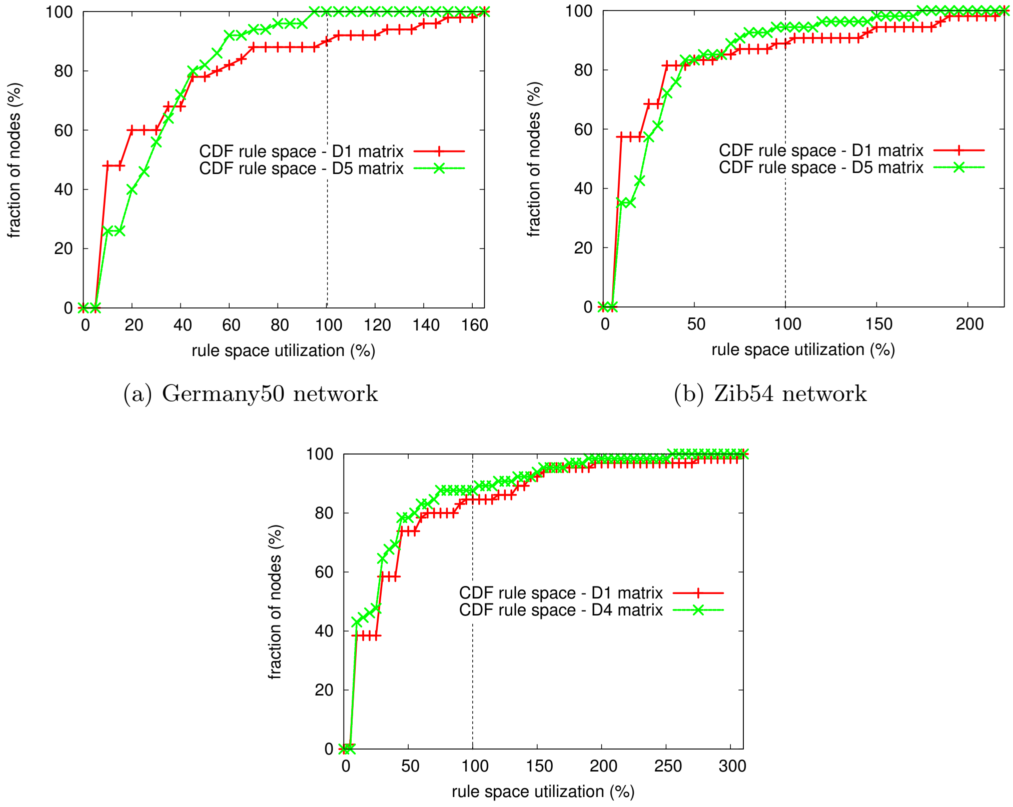
(a) Germany50 network

(b) Zib54 network

(c) Ta2 network

Figure 6.5: Number of overloaded routers in the three networks with unlimited rule-space algorithm

We assume that all the routers have the same rule space capacity $C_u = 750$ [KHK13]. As the EAR approach, we use the heuristic algorithm in [GMMO10] to represent the case where there is no limit of rule space. We run this algorithm on germany50, zib54 and ta2 networks to see if the rule space constraints are violated or not. As shown in Fig. 6.5a, Fig. 6.5b and Fig. 6.5c, most of the cases (except $D5$ of germany50 network), there are routers that use more than 750 rules in their routing tables. In zib54 network (resp. ta2 network), from 6% to 11% (resp. 11% to 16%) of routers exceed their rule space capacities. In germany50 network, with the



(a) Germany50 network

(b) Zib54 network

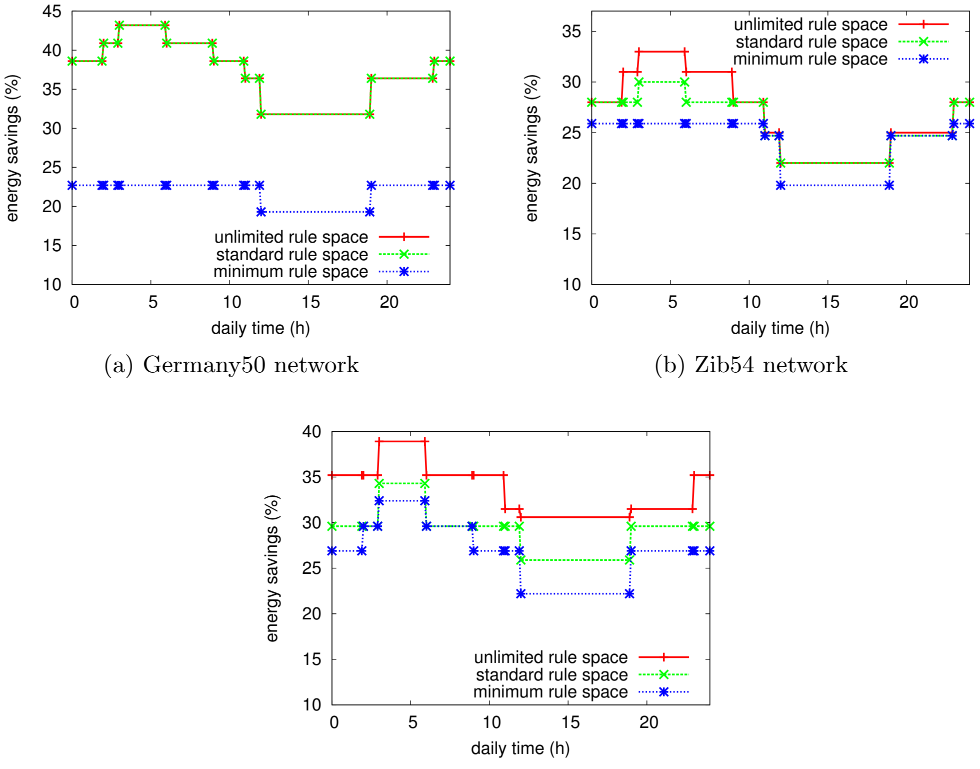
(c) Ta2 network

Figure 6.6: CDF rule space utilization in the three networks with unlimited rule-space algorithm

$D5$ traffic matrix, there is no router that uses more than 750 rules. For other traffic matrices, from 2% to 10% of the routers are overloaded in rule space (Fig. 6.5a). Therefore, the limited rule space is really a problem in real networks. This problem is extremely important since we cannot route traffic as expected. The number of routers overloaded in rule space depends on the traffic matrix. However, an accurate explanation is difficult (e.g. less overloaded routers when traffic load is high) since the algorithm in [GMMO10] does not care at all about the rule capacity.

To take a closer look at rule space allocation, we draw cumulative distribution function (CDF) for rule utilization at each router (Fig. 6.6a, Fig. 6.6b and Fig. 6.6c). When a rule space utilization is larger than 100%, it means that the router has used more than 750 rules. Based on Fig. 6.5a, Fig. 6.5b and Fig. 6.5c, we can find the traffic matrices that cause the maximum and the minimum number of routers violating their rule space constraints. We call these traffic matrices as the max-over-rule and the min-over-rule, respectively. For instance, in case of germany50 network, $D1$ is the max-over-rule and $D5$ is the min-over-rule. For each network, we draw two CDFs of rule utilization for the max-over-rule and the min-over-rule cases. As shown in Fig. 6.6a, Fig. 6.6b and Fig. 6.6c, the CDF of rule space allocation of the two cases are quite similar. However we can see in the max-over-rule case, more fractions of routers are overloaded. For instance, in Fig. 6.6b, only 89% of routers are less than 100% rule space utilization in the max-over-rule case (D1), meanwhile it is 94% of routers in the min-over-rule case (D5). In general, the larger the network is, the more rule space is needed at routers. For example, the maximum rule utilization of germany50, zib54 and ta2 networks are 165%, 220% and 310%, respectively.

6.5.2.2 Energy savings



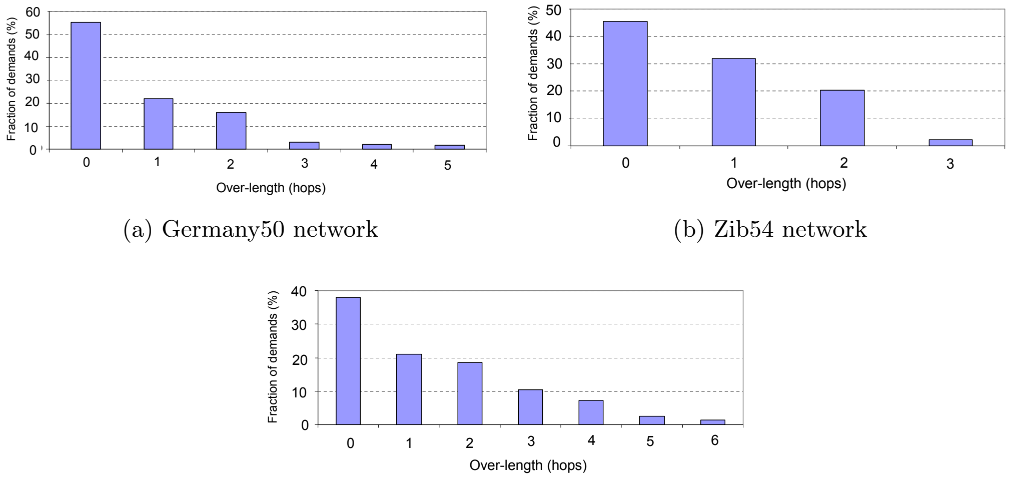
(a) Germany50 network

(b) Zib54 network

(c) Ta2 network

Figure 6.7: Energy savings in daily time in the three networks

We collect energy savings for each network in three cases: the minimum, the standard and the unlimited rule spaces. In the standard case, each router can install at most 750 rules ($C_u = 750$). To find the minimum rule space, we reduce the value of C_u until we get a minimum value of C_u for which it is possible to find a feasible solution. The minimum values of C_u for germany50, zib54 and ta2 networks are 227, 670 and 695, respectively. The unlimited rule space case is equivalent to the classical EAR model in which we do not consider at all rule space constraints at routers. In general, as shown in Fig. 6.7a, Fig. 6.7b and Fig. 6.7c, the larger the rule space at routers is, the more flexible routing solutions we have and more energy



(a) Germany50 network

(b) Zib54 network

(c) Ta2 network

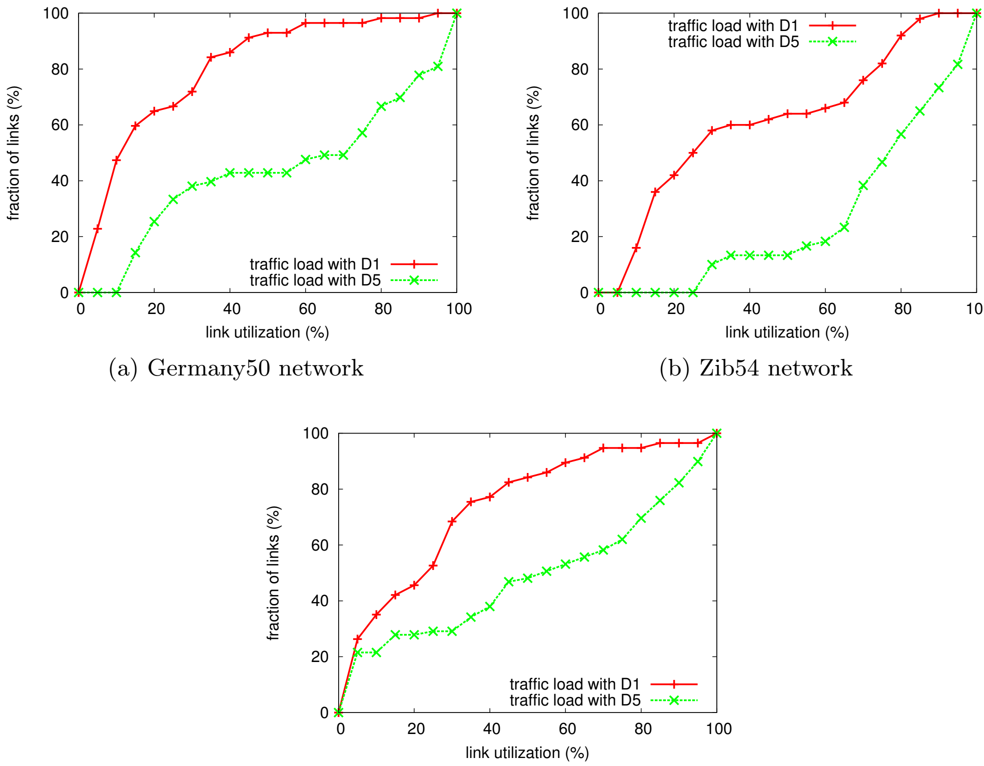
Figure 6.8: Additional stretch compared with shortest paths.

can be saved for the network. The energy savings gap between the standard and the unlimited rule space cases is small. In some traffic matrices, both cases offer the same amount of energy savings. For instance in germany50 network, the standard and unlimited cases offer the same amount of energy savings. It means that if we use a smart rule allocation, it is possible to achieve equivalent energy savings as the classical EAR approach. The maximum energy savings gaps of the standard and the unlimited cases are 3% and 5.6% for zib54 and ta2 networks, respectively. As expected, less energy is saved in the minimum rule space case as we do not have enough installed rules to route traffic in a better way. As shown in Fig. 6.7a, there is a big gap of the energy savings between the minimum and the standard cases. This can be explained as the difference between the minimum and the standard values of C_u in germany50 network is large.

6.5.2.3 Route length

Intuitively, EAR would affect the length of routing flows as we redirect traffic flows to minimize the number of active links. In Fig. 6.8a, Fig. 6.8b and Fig. 6.8c, we evaluate the impact of EAR on routing length with respect to the shortest path routing. For each traffic demand, we collect the length of its routing flow and the corresponding shortest path. We use the notation *over-length* to denote the difference (in number of hops) between the length of the routing solution and the shortest path. When *over-length* is equal to 0, it means that the routing solution is exactly the shortest path. As shown in Fig. 6.8a, Fig. 6.8b and Fig. 6.8c, a large fraction of the demands follow their shortest paths. Indeed, in the heuristic algorithm, we use the shortest path to find routing solution. In germany50 and zib54, the maximum number of additional hops for a demand is 3 and 5 hops, respectively. The ta2 network is larger, up to 6 hops can be added to a demand, but it happens only for 1.4% of the demands. However, if latency is important, especially for sensitive delay applications such as voice or video streaming, we can add constraints to limit the route length so that it will not excess a predefined threshold value.

6.5.2.4 Link load



(a) Germany50 network

(b) Zib54 network

(c) Ta2 network

Figure 6.9: CDF of link load in the three networks

In this computation, for simplicity, we set the maximum link utilization $\mu = 100\%$. Intuitively, EAR would affect the utilization of links as fewer links are used to carry traffic. In this subsection, we evaluate the impact of EAR on link utilization. We draw the CDFs of link load of germany50, zib54 and ta2 networks in Fig. 6.9a, Fig. 6.9b and Fig. 6.9c. For each network, we collect traffic load of the lowest ($D1$) and the highest ($D5$) traffic matrices. As shown in Fig. 6.9a, Fig. 6.9b and Fig. 6.9c, the solutions with $D5$ have heavier link load than in $D1$. It means for low traffic matrix, fewer links are used but the link load does not increase too much. High link utilization (e.g. with $D5$ traffic matrix) can affect QoS as it causes packet drop and long queuing delay. One possible way to overcome this problem is to set the maximum link utilization to a small value, e.g. $\mu = 50\%$.

6.6 Conclusions

To our best knowledge, this is the first work considering rule space constraints of OpenFlow switch in energy-aware routing. We argue that, in addition to capacity constraint, the rule space is also important as it can change the routing solution and affects QoS. Based on computations with real traffic traces, we show that our smart rule allocation can achieve high energy efficiency for a backbone network while respecting both the capacity and the rule space constraints.

6.7 Future work

In this chapter, to deal with the traffic variation, daily time periods are characterized by different traffic levels and in each period, a single traffic matrix is assumed to be accurately collected. As traffic matrices are considered independently, different sets of rules are installed on routers for each traffic matrix. However, from the view point of traffic engineering, frequent changes in the routing configuration can cause network disruption [CCRP13, FT02]. Therefore, we argue that the future work should take into account the stability of rule setting for the energy-aware traffic engineering problem. Rather than computing a new rule placement from scratch for each traffic matrix, we must allow to incrementally update the rule setting to minimize the computation time and avoid service deteriorations for end users.

Although the power savings is worthwhile, the performance effects must be minimal. The further work should consider the trade-offs between energy efficiency, performance and robustness. For instance, one approach is to perform load-balancing on top of the EAR techniques [FWMG13]. This helps to achieve energy efficient network but without sacrificing the traditional traffic engineering. In addition, other performance effects such as network fault-tolerance, packet latency should be also considered.

As prior works in literature, OpenFlow has been used to evaluate energy savings in data center network [HSM$^+$10, WYW$^+$12]. However, as the limited rule space is ignored, the evaluation may not be as expected, especially in large scale data center networks. We propose as a future work to re-evaluate these models while taking into account rule space constraints. This will provide a more accurate evaluation of energy efficiency for the network.

6.8 Bibliography

[BCRR12] A. P. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, *"A Survey of Green Networking Research"*, IEEE Communication Surveys and Tutorials **14** (2012), 3 – 20.

[CCRP13] L. Chiaraviglio, A. Cianfrani, E. L. Rouzic, and M. Polverini, *"Sleep Modes Effectiveness in Backbone Networks with Limited Configurations"*, Computer Networks **57** (2013), 2931–2948.

[CFP+09] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. Gude, N. McKeown, and S. Shenker, *“Rethinking Enterprise Network Control”*, IEEE/ACM Transaction in Networking **17** (2009), 1270 – 1283.

[CMN11] L. Chiaraviglio, M. Mellia, and F. Neri, *“Minimizing ISP Network Energy Cost: Formulation and Solutions”*, IEEE/ACM Transaction in Networking **20** (2011), 463 – 476.

[CMTY11] A. R. Curtis, J. C. Mogul, J. Tourrilhes, and P. Yalagandula, *“DevoFlow: Scaling Flow Management for High-Performance Networks”*, ACM Special Interest Group on Data Communication (SIGCOMM), vol. 41, 2011, pp. 254 – 265.

[FT02] B. Fortz and M. Thorup, *“Optimizing OSPF/IS-IS Weights in a Changing World”*, IEEE Journal on Selected Areas in Communications **20** (2002), 756–767.

[FWMG13] F. Francois, N. Wang, K. Moessner, and S. Georgoulas, *“Green IGP Link Weights for Energy-efficiency and Load-balancing in IP Backbone Networks”*, International Conferences on Networking (IFIP NETWORKING), 2013, pp. 1–9.

[GMMO10] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy, *“Minimizing Routing Energy Consumption: from Theoretical to Practical Results”*, IEEE/ACM Green Computing and Communications (GreenCom), 2010, pp. 252–259.

[GS03] M. Gupta and S. Singh, *“Greening of the Internet”*, ACM Special Interest Group on Data Communication (SIGCOMM), 2003, pp. 19–26.

[HSM+10] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, *“ElasticTree: Saving Energy in Data Center Networks”*, USENIX conference on Networked systems design and implementation (NSDI), 2010.

[IBM] IBM ILOG, *CPLEX Optimization Studio 12.4*.

[KHK13] Y. Kanizo, D. Hay, and I. Keslassy, *“Palette: Distributing Tables in Software-defined Networks”*, IEEE INFOCOM Mini-conference, 2013.

[KLRW13] N. Kang, Z. Liu, J. Rexford, and D. Walker, *“Optimizing the “One Big Switch” Abstraction in Software-Defined Networks”*, ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT), 2013.

[MAB+08] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, *“Openflow: Enabling Innovation in Campus Networks”*, ACM Computer Communication Review **38** (2008), 69 – 74.

[MLT12] C. R. Meiners, A. X. Liu, and E. Torng, *“Bit Weaving: A Non-prefix Approach to Compressing Packet Classifiers in TCAMs”*, IEEE/ACM Transaction in Networking **20** (2012), 488 – 500.

[NMN+13] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, *“A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks”*, IEEE Communications Surveys and Tutorials (2013), 1 – 18.

[OWPT10] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, *SNDlib 1.0 - survivable network design library*, Networks **55** (2010), no. 3, 276–286.

[RKOW11] C. Raack, A. M. C. A. Koster, S. Orlowski, and R. Wessäly, *“On Cut-based Inequalities for Capacitated Network Design Polyhedra”*, Networks **57** (2011), 141 – 156.

[SCF+12] B. Stephens, A. Cox, W. Felter, C. Dixon, and J. Carter, *“PAST: Scalable Ethernet for Data Centers”*, ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT), 2012.

[WBR11] R. Wang, D. Butnariu, and J. Rexford, *“OpenFlow-based Server Load Balancing Gone Wild”*, USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE), 2011.

[WYW+12] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, *“CARPO: Correlation-Aware Power Optimization in Data Center Networks”*, Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2012.

CHAPTER 7

Conclusions and further research

In this thesis, focusing on green networking, we proposed several methods to increase energy efficiency for backbone networks. We here briefly describe the obtained results and many open questions for further research.

In Chapter 3, we presented a new energy-aware routing model called GreenRE. We showed that traffic redundancy elimination, which was initially developed for reducing traffic load on the Internet, is also interesting to save energy for backbone networks. However, combining energy-aware routing and redundancy elimination, namely the GreenRE model, is a non-trivial task. We formulated the GreenRE problem as Mixed Integer Linear Program and then proposed greedy heuristic algorithms that can be used for large-scale networks. By simulation with several real network topologies, we showed that the GreenRE model can gain further 37% of energy savings compared to the *classical EAR* model.

In Chapter 4, we extended the GreenRE model by proposing a robust model in which fluctuation of demand volumes and redundancy elimination rates are considered. Using this extra knowledge on the dynamics of the traffic pattern, we are able to significantly increase energy efficiency for backbone networks. We formally defined the problem and modeled it as Mixed Integer Linear Program (MILP). We then proposed an efficient heuristic algorithm that is suitable for large networks. Simulation results with real traffic traces showed that our approach allows for $16-28\%$ extra energy savings with respect to the existing EAR models.

Beyond the scope of GreenRE model, in this thesis, we also studied the impacts of energy-aware routing on network protocols. In particular, we considered real problems when deploying EAR with Open Shortest Path First (OSPF) and Software-defined network (SDN). In Chapter 5, we considered to save energy using Open Shortest Path First protocol. From the perspective of traffic engineering, we argue that stability in routing configuration also plays an important role in QoS. Therefore, we proposed a novel optimization method to compute link weights so as to limit the changes in network configurations in multi-period traffic matrices. We formally defined the problem and model it as Mixed Integer Linear Program (MILP) and then proposed efficient heuristic algorithm. We carried simulations with real traffic traces on different networks. The results showed that our approach achieves high energy savings and less pain for QoS (in term of less changes in network configuration).

Finally, in Chapter 6, we focused on using Software-Defined Network for energy-aware routing. Prior works on EAR assumed that the table of OpenFlow switch can hold an infinite number of rules. In practice, this assumption does not hold since the flow table is implemented with Ternary Content Addressable Memory (TCAM) which is expensive and power-hungry. To address this problem, we proposed an optimization method to minimize energy consumption for a backbone network while respecting capacity constraints on links and rule space constraints on routers. In details, we presented an exact formulation using Integer Linear Program (ILP) and introduced efficient greedy heuristic algorithm. Based on simulations, we showed that using this smart rule space allocation, it is possible to save almost as much power consumption as the classical EAR approach.

Although the obtained results have shown a great opportunity for saving energy using our proposals, there are a number of open problems that arise for further research. For instance, the model in Chapter 3 and 4 only capture the intra-flow redundancy elimination. Indeed, further traffic load (and power consumption) can be reduced by aggregating similar traffic flows, which have high potential of inter-redundant traffic, on the same links to achieve inter-flows RE. As a consequence, we plan to extend the GreenRE model with inter-flow RE to increase energy efficiency for the networks. On the other hand, the research topic of saving energy with SDN is quite new. We suggest that the future work should take into account the side effects when deploying EAR using SDN. Although the power savings is worthwhile, the performance effects must be minimal. For instance, rather than computing a new rule placement from scratch for each traffic matrix, we should allow to incrementally update the rule setting to minimize the computation time and avoid service deteriorations for end users. Moreover, we should propose methods to protect energy-aware routing solution from network failures. In addition, the limited rule space is one of the strict constraints in OpenFlow switches, we plan to propose methods to efficiently reduce the number of required rules for energy-aware routing problem.

Appendix A

Xcast6 Treemap Islands

Contents

Due to the complexity and poor scalability, IP Multicast has not been used on the Internet. Recently, Xcast6 - a complementary protocol of IP Multicast has been proposed. However, the key limitation of Xcast6 is that it only supports small multicast sessions. To overcome this, we propose Xcast6 Treemap islands (X6Ti) - a hybrid model of Overlay Multicast and Xcast6. In summary, X6Ti has many advantages: support large multicast groups, simple and easy to deploy on the Internet, no router configuration, no restriction on the number of groups, no multicast routing protocol and no group management protocol. Based on simulation, we compare X6Ti with IP Multicast and NICE protocols to show the benefits of our new model.

A.1 Publications

This chapter corresponds to *Xcast6 Treemap Islands - Revisiting Multicast Model* by T. K. Phan, J. Moulierac, N. C. Tran, and N. Thoai which has been accepted for publication in ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT) (Student Workshop), 2012.

A.2 Introduction

As mentioned in [BFI+07], IP Multicast encounters a major obstacle in the routing table size in routers when supporting multiple groups at the same time. For this reason, Xcast6 protocol - RFC 5058 [BFI+07] has been proposed. Because of its simplicity, Xcast6 has been successfully tested in real networks (e.g. Xcast Communication from a Flying Airplane in Japan [X6Demo]). Recently, Xcast6 Treemap (X6T) [PTM+10b] has been proposed which works well even without Xcast-aware router on the network. However, there are two disadvantages hindering the deployment of Xcast6 (and X6T): first, the lack of IPv6 world-wide deployment and second, only small multicast sessions can be supported. In this paper, we develop Xcast6 Treemap islands (X6Ti) - a hierarchical architecture Overlay Multicast with the core is X6T. In other words, X6Ti allows interchange (without any configurations) between Overlay Multicast mode (end-hosts duplicate and forward data) and X6T mode (Xcast-aware routers duplicate and forward data like IP Multicast). The example in the next section explains how X6Ti works in the two modes.

A.3 Xcast6 Treemap islands

A.3.1 Xcast6 Treemap in an island

We keep the format of the X6Ti packet same as X6T [PTM+10b] but change the forwarding algorithm in X6Ti end-hosts [X6T11]. In summary, each X6Ti packet header includes a list of destination IP addresses, a list of bitmap (bitmap = 1 is to mark the end-hosts which have not received the packet yet) and a treemap (which encodes an overlay tree of end-hosts). It is clear that the packet header contains all necessary information about the multicast group and also the overlay tree, hence the routers do not need to store any additional multicast information. This explains why X6Ti has no restriction on the number of multicast groups. Moreover, following IPv6 standard, X6Ti packets can be handled by both Xcast-aware and normal IPv6 routers on the network. When receiving an X6Ti packet, Xcast router uses its unicast routing table to look up all IP addresses in the list of destinations, then duplicates, changes the bitmap and forwards the packet to appropriate network interfaces (Fig. A.1). Therefore, there is no need to develop a new multicast routing protocol. On the other hand, X6Ti packet also set the first unsent IP address in the list of destinations as its destination like a unicast packet. Thus, normal routers can route an X6Ti packet like a normal unicast packet. In addition, when an end-host receives an X6Ti packet, based on the bitmap and the treemap, it knows which end-hosts have not received the packet yet and then forwards the packet as the overlay tree.

As shown in Fig. A.1, in parts of the network which has Xcast routers (X1 and X4), the overlay tree is not used and packets are duplicated by routers, otherwise, data are multicasted by X6Ti end-hosts based on the treemap.

A.3.2 Xcast6 Treemap islands

Assuming the MTU is 1500 bytes, then the payload length of an X6Ti packet is equal to $(1348-16N)$ (N is the number of IP addresses in the packet header) [X6T11]. In the current implementation [X6T11], at most 64 IPv6 addresses can be embedded

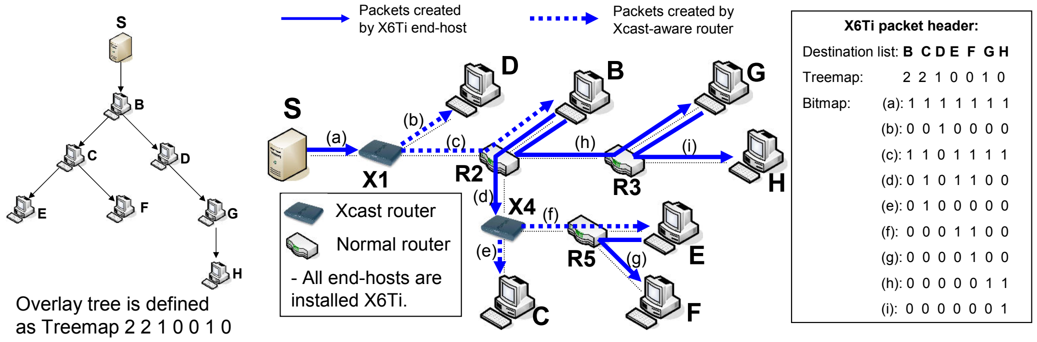


Figure A.1: Source S sends data to a group of recipients

in each packet header and it is also the maximum number of members in each island (an overlay tree with one branch from the root as shown in Fig. A.2). With hierarchical design (2-tiers), many islands are connected using special hosts (X6Ti sub-roots) to form a larger Overlay Network. In fact, this model can be extended to n-tiers in which X6Ti clients can work as X6Ti sub-roots to connect islands. There are several criteria to select an X6Ti sub-root such as out-going bandwidth, stability of end-host, etc. (more details can be found in [X6T11]). Note that, in the traditional Overlay Multicast, only end-hosts duplicate and forward data, thus there is still much traffic redundancy on the network. For X6Ti, it works like the traditional Overlay Multicast when there is no Xcast router, otherwise Xcast routers automatically duplicate and forward data like IP Multicast.

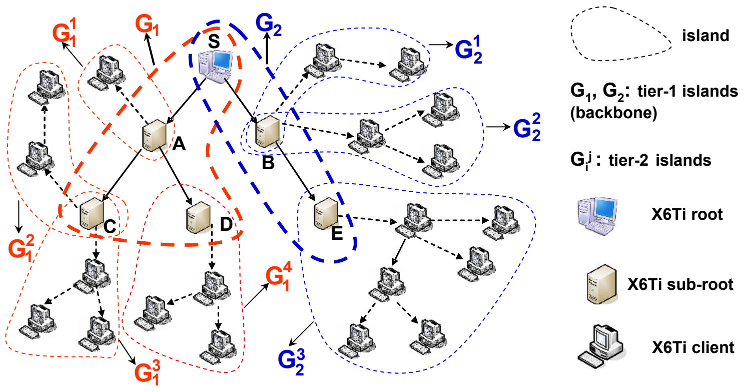


Figure A.2: 2-tiers X6Ti model

A.4 Simulation and Discussion

In this section, we compare X6Ti with NICE [BBK02] and IP Multicast. We use the network topology of France [OWPT10] with 25 routers in the backbone. There is a single source and 24 X6Ti sub-roots, each of them connect to a backbone router.

Each sub-root manages several tier-2 islands in which X6Ti clients connect to the nearest sub-root. We increase the number of members in each island uniformly so that the total number of hosts in the multicast session is from 200 to 2600. For each test, we select at random 0% (X6Ti-0), 30% (X6Ti-30), 70% (X6Ti-70) and 100% (X6Ti-100) of the routers to be Xcast-aware routers. We collect end-to-end delay (total amount of time a packet is transmitted from the source to a receiver) and link stress (the number of duplicate packets that a physical link has to carry).

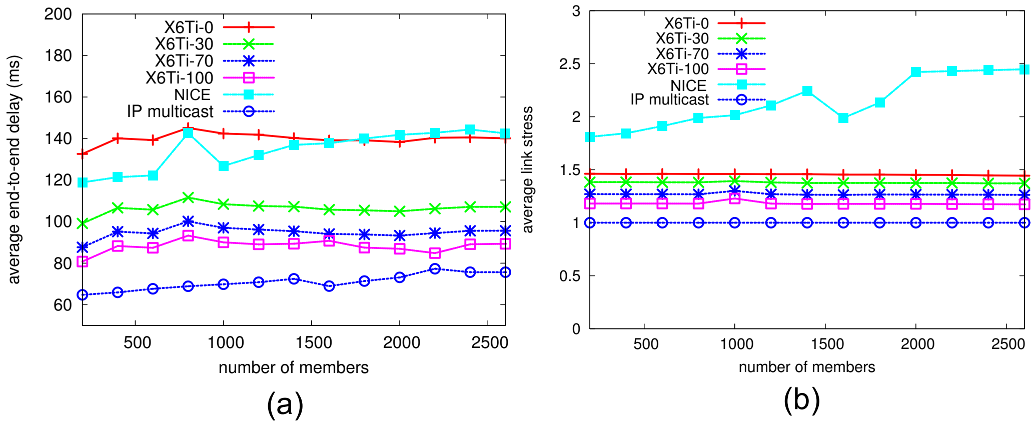


Figure A.3: (a) End-to-end delay and (b) Link stress

IP Multicast transmit traffic optimally with low end-to-end delay and no traffic redundancy (link stress = 1) (Fig. A.3). X6Ti works as pure Overlay Multicast when there is no Xcast-aware router (X6Ti-0 has less link stress than NICE since we choose a good overlay tree as described previously). And obviously, performance of X6Ti is improving when more Xcast-aware routers are deployed on the network. Table A.1

Table A.1: A comparison table

	IP multicast	Overlay Multicast	X6Ti
Efficiency in bw/delay	High	Low-Medium	Medium
Ease of deployment	Low	High	High
Fast route adaptation	High	Low	Medium

shows some properties in comparison between IP multicast, Overlay Multicast and X6Ti. X6Ti is easy to deploy and no need any router configurations. In addition, with the world-wide IPv6 deployment [IPv6Day], X6Ti can be deployed at end-hosts first and works like Overlay Multicast. Then, the network operators will deploy Xcast-aware routers when they see it is useful. In fact, X6Ti is even better than Overlay Multicast in term of “Fast route adaptation”. When the overlay tree is changed (e.g. hosts join/leave), the X6Ti root or sub-root simply modifies the treemap, there is no need for other hosts to update or store new overlay tree in local like Overlay Multicast. More details and more features on the comparisons can be found in [X6T11].

A.5 Conclusion

We believe that the X6Ti model overcomes most shortcomings of IP Multicast, Overlay Multicast and Xcast6. In summary, X6Ti solves the problem of difficulty in deployment and state scalability at routers (IP Multicast); multicast group size (Xcast6); traffic redundancy and instability when hosts join/leave (Overlay Multicast). For future work, real applications using X6Ti should be deployed on the Internet to test the feasibility of the protocol.

A.6 Bibliography

[BBK02] Suman Banerjee, Bobby Bhattacharjee, and Christopher Kommareddy, *"Scalable Application Layer Multicast"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2002, pp. 205–217.

[BFI+07] R. Boivie, N. Feldman, Y. Imai, W. Livens, and D. Ooms, *Explicit Multicast (Xcast) Concepts and Options*, RFC 5058, 2007.

[IPv6Day] http://www.worldipv6launch.org/.

[OWPT10] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, *SNDlib 1.0 - survivable network design library*, Networks **55** (2010), no. 3, 276–286.

[PTM+10] T. Khoa Phan, N. Thoai, E. Muramoto, K. K. Ettikan, B. P. Lim, and P. Y. Tan, *"Treemap - The Fast Routing Convergence Method for Application Layer Multicast"*, IEEE Consumer Communications and Networking Conference (CCNC), 2010.

[X6T11] *Research report*, hal.inria.fr/docs/00/63/76/56/PDF/RR-7784.pdf, 2011.

[X6Demo] www.cogma.org/press/video/xcast_e_1000k.wmv.

Appendix B

MaxNet TCP Congestion Control

Contents

Congestion control is a distributed algorithm to share network bandwidth among competing users on the Internet. In the common case, quick response time for mice traffic (HTTP traffic) is desired when mixed with elephant traffic (FTP traffic). The current approach using loss-based with Additive Increase, Multiplicative Decrease (AIMD) is too greedy and eventually, most of the network bandwidth would be consumed by elephant traffic. As a result, it causes longer response time for mice traffic because there is no room left at the routers. MaxNet is a new TCP congestion control architecture using an explicit signal to control transmission rate at the source node. In this chapter, we show that MaxNet can control well the queue length at routers and therefore the response time to HTTP traffic is several times faster than with TCP Reno/RED.

B.1 Publications

This chapter corresponds to *MaxNet and TCP Reno/RED on Mice Traffic* by T. K. Phan, T. T. Tran, D. D. Nguyen, and N. Thoai which has been accepted for publication in Modeling, Simulation and Optimization of Complex Processes (Springer), 2012.

B.2 Introduction

TCP Reno [Jac90] uses AIMD mechanism [Jac88] in which the sending rate is increased until packet loss happens. To avoid buffer overflows at router, AQM RED (Active Queue Management Random Early Detection) [FJ93] can be used in conjunction with TCP Reno. The weakness of RED is that it does not take into account the number of incoming flows arrived at a bottleneck link to perform appropriate treatments to lighten down the heavy load. When there are a large number of sharing flows at a bottleneck link, the offered load will not be decreased, also the queue length at router does not change. This results in longer response time for the mice traffic.

MaxNet [SAW$^+$08][SWW05][WAZ03] is a new congestion control mechanism using multi-bit signal instead of packet loss to control the sending rate. Besides, MaxNet router can control the magnitude of transient queues well regardless the number of new arrival flows. In other words, MaxNet can always keep a free space at routers for mice traffic to fly though. As a result, the response time to HTTP requests is much shorter with MaxNet than with TCP Reno/RED.

The rest of this chapter is structured as follows. Section B.3 is the theoretical analysis of queueing delay of RED and MaxNet routers. Section B.4 shows the efficiency of MaxNet's quick control of the transfer rate of mice flows. We have some experiments and evaluations in Section B.5. Finally, we present the conclusions and future work in Section B.6.

B.3 Equilibrium queueing delay at RED and MaxNet router

B.3.1 Queueing delay at RED router

RED routers calculate and compare the average queue length based on the two parameters: maximum threshold and minimum threshold. Base on this comparison, RED router operates in three modes [FJ93]:

- "No dropped": when the average queue length is less than the value of the minimum threshold, router assumes that its link is under-utilized. As a result, all packets are allowed to go through without marking or dropping.

- "Probabilistic dropped": when the average queue length is between the minimum and maximum thresholds, router assumes that the network can be saturated, it then marks/drops packets with a probability corresponding to the traffic load.

- "Forced dropped": when the average queue length is greater than the maximum

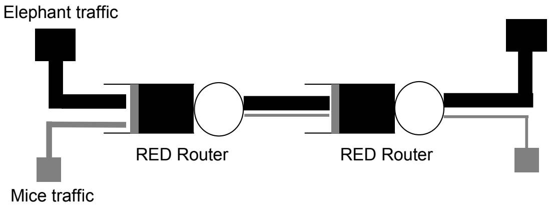


Figure B.1: Queueing delay of RED routers

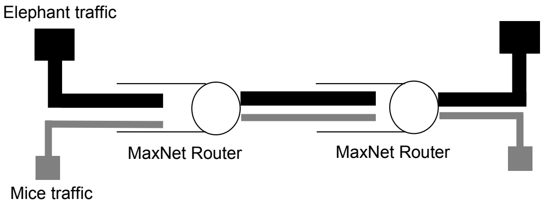


Figure B.2: Queueing delay of MaxNet routers

threshold, all packets that go through the router will be marked/dropped to reduce heavy load on the link.

From the flow-level model of AIMD, the window size of TCP Reno is updated using the following equation [WJLH06]:

$$w_i^{'}(t) = \frac{1}{T_i(t)} - \frac{2}{3}x_i(t)q_i(t)w_i(t) \tag{B.1}$$

where $T_i(t)$ is the round-trip-time; $x_i(t) = w_i(t)/T_i(t)$ (packets/s); $w_i(t)$ is the current window size and $q_i(t)$ is the end-to-end loss probability. At the equilibrium point, window size adjustment $w_i^{'}(t) = 0$, then from B.1, the end-to-end equilibrium mark/drop probability feedback to source i is derived as following:

$$q_i^* = \frac{3}{2.w_i^{*2}} > 0 \tag{B.2}$$

Equation B.2 implies that, at the equilibrium point, end-to-end mark/drop probability of source must be greater than zero. As a result, from the marking scheme of RED, it can be asserted that each router on the end-to-end always keeps a backlog. This consequently causes inevitable queueing delay for mice traffic such as HTTP requests to fly through. Figure B.1 illustrates two RED routers which always maintain backlog at equilibrium point.

B.3.2 Queueing delay at MaxNet router

The marking mechanism of MaxNet router uses an explicit multi-bit signal instead of marking/dropping packet as RED routers. Congestion price p_l at MaxNet router is defined in [SAW$^+$08]:

$$p_l(t+dt) = p_l(t) + dt\frac{y_l(t) - \mu_l C_l}{C_l} \tag{B.3}$$

where $y_l(t)$ is the aggregated rate at link l; C_l is the link capacity and μ_l is the target link utilization. In MaxNet router, at the equilibrium point, the link price adjustment B.3 tries to match the aggregated input rate $y_i(t)$ with $\mu_l C_l$, leaving spare $(1-\mu_l)C_l$ capacity to absorb mice traffic and reduce the queueing delay.

Figure B.2 illustrates the queueing delay of two MaxNet bottleneck links at the equilibrium point. In contrast to RED routers, there is no backlog in both two MaxNet routers when the target link utilization is set to μ_l where $0 < \mu_l < 1$, hence, mice traffic can fly through links without being blocked.

B.4 Magnitude of transient queue of RED and MaxNet routers

As pointed out in [FKSS99][LAJS03], the weakness of RED is that it does not take into account the number of flows arriving a bottleneck link to have proper treatments to avoid heavy load. Assuming there are n flows sharing a bottleneck link. If a packet is marked or dropped then the offered load is reduced by a factor of $(1 - 0.5n^{-1})$. When n is large then $(1 - 0.5n^{-1}) \rightarrow 1$, which means the offered load will not be decreased and the queue length doesn't change, either. That means if RED is not configured aggressively then marking a single packet could result in simple "droptail" packet drops. The packet loss then severely declines the throughput of Reno sources due to their AIMD mechanism.

MaxNet router well controls the magnitude transient queue regardless the number of new arrival flows. At the equilibrium point, when the number of new arrival flows is small, transient queues exist, but the magnitude of these queues decreases rapidly as the number of flows increases [SAW$^+$08]. This can be explained with the following simple case: assuming that there are N flows sharing a bottleneck link. At the equilibrium point, each flow transmits at the rate of $\frac{\mu_i C_i}{N}$. Thus, when a new flow joins, its advertised rate is at most $\frac{\mu_i C_i}{N}$. The aggregated arrivals at router are at most $\mu_l C_l + \frac{\mu_l C_l}{N}$. Thus, this causes the overload:

$$0 \leq overload \leq ((1 + \frac{1}{N})\mu_l - 1)C_l \tag{B.4}$$

Obviously, the larger the N is, the smaller the magnitude of transient queue becomes and eventually when $N > \frac{\mu}{1-\mu}$, the transient queue size drops to zero.

As mice traffic is short-lived flows, an effective congestion control should quickly controls the rate of such flows to avoid uncontrolled overshoot and transient queue.

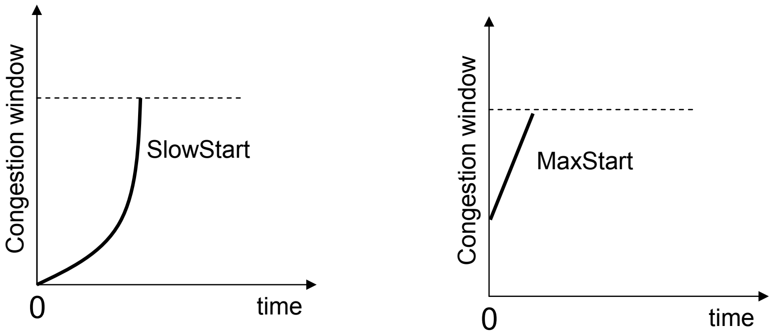


Figure B.3: SlowStart and MaxStart

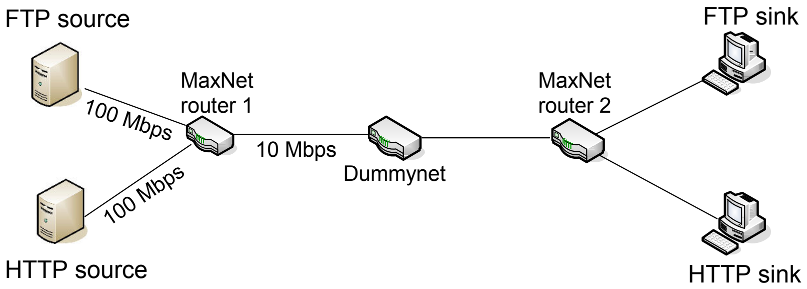


Figure B.4: MaxNet networks testbed

Unlike SlowStart mechanism of Reno [Jac88], MaxNet source employs MaxStart mechanism [SAW$^+$08] to seek for the target rate at initial state (Fig. B.3).
By adopting the multi-bit explicit signaling mechanism, MaxStart enables source to seek for its target rate within a significant short duration. New MaxNet flow is initiated at the minimum spare capacity of all links on its end-to-end path and then ramped up linearly to the advertised rate over two Round Trip Time [SAW$^+$08]. Therefore MaxNet source converges to the target rate more quickly than Reno source.

B.5 Experiment and Evaluation

B.5.1 Testbed Layout

In this testbed, Pentium IV PCs (CPU 1.8GHz, 256MB RAM, 100Mbps network cards) are used. The MaxNet Router1 is configured with the output capacity 10Mbps to make sure it is the bottleneck link. The target utilization of both MaxNet routers (μ_l) is set to 0.95. Dummynet [Dummynet] is configured with 20ms RTT delay.

The testbed of Reno/RED is same as MaxNet testbed where MaxNet routers are changed to RED routers. All of RED routers are configured with the RED

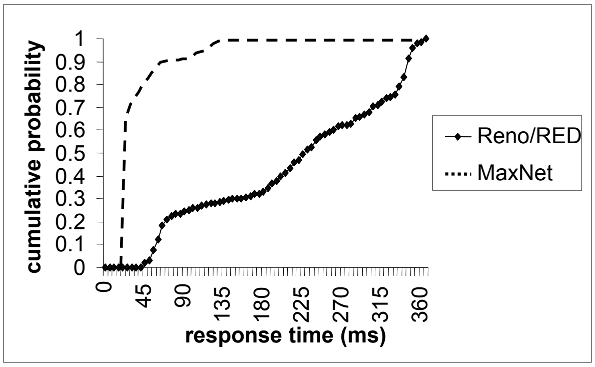


Figure B.5: Response time of TCP Reno vs MaxNet with 200 HTTP connections

parameters for web traffic [FKSS99] as following:
- $w_q = 0.002$: weighting factor for computing average queue size as suggested in [CKH04] for web traffic.
- $q_{avg} = (1 - w_q).q_{avg} + w_q.q$ with q is instantaneous queue size.
- $min_{th} = 30$: average queue length threshold for triggering probabilistic drops/marks.
- $max_{th} = 90$: average queue length threshold for triggering forced drops/marks.
- $max_p = 0.1$: maximum mark/drop probability.

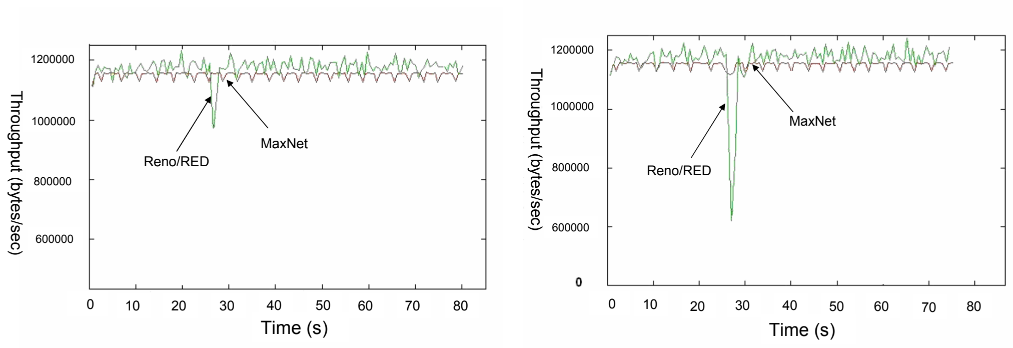
We simulated elephant traffic and mice traffic with iperf [Iperf] and httperf [Httperf] tools. For the both networks of Reno/RED and MaxNet, one long live FTP connection is generated at the "FTP source" in approximately of 60s. Then 20s after, HTTP connections are generated on the "HTTP source". Each HTTP connection sends one request of 62 bytes and HTTP response size is 4KB. The HTTP response time is computed at the application layer by the duration from the first byte being sent out to the time when the first byte of response received.

B.5.2 Response time of HTTP connections

We adopted the cumulative probability for statistical analysis of the response time of HTTP requests. In Figure B.5, the response time of HTTP requests in MaxNet is significantly less than in TCP Reno/RED. Particularly, in the experiment with 200 HTTP requests spawn, 100% of MaxNet HTTP requests receive the reply at most at 135ms while in TCP Reno/RED, only 30% of the total HTTP requests receive the first reply less than 135ms.

B.5.3 Throughput of elephant flow

In Fig. B.6, packets drop occur at the RED bottleneck link, thus the throughput of Reno/RED is decreased. The greater the number of connections is, the more severity the drop becomes. In contrast, the throughput of elephant flow in MaxNet



(a) 50 HTTP flows join

(b) 200 HTTP flows join

Figure B.6: Throughput of elephant traffic when HTTP flows join

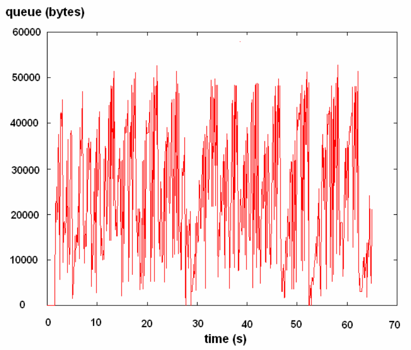


Figure B.7: Backlog at RED router

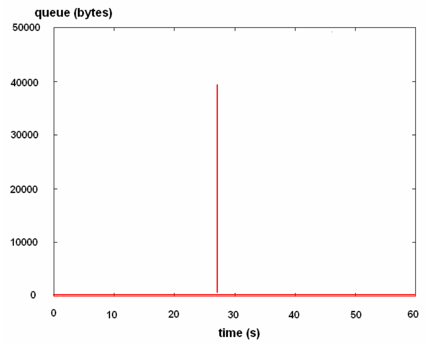


Figure B.8: Backlog at MaxNet router

networks is not impacted regardless the number of arrival flows.

B.5.4 Transient queue

In this section, we analyse the transient queue size (or the backlog at router) in comparison between RED and MaxNet routers (Figs. B.7 and B.8).

In this experiment, we configure two same 100Mbps links and keep the other configurations and parameters of MaxNet and RED router as the same as the above experiments to compare transient queue. Under MaxStart mechanism, the transient queue happens within short duration when HTTP connections join, meanwhile RED router always keep a backlog all the time.

B.6 Conclusions and Future work

At the equilibrium point, MaxNet can clear the buffer while Reno/RED always keeps a backlog in routers. Therefore, when elephant traffic is mixed with mice traffic, MaxNet has a shorter response time for mice traffic than TCP Reno/RED. If the number of arrival mice flows is large, Reno without proper treatment can cause packet loss which in turn eventually degrades the throughput of elephant traffic. In addition, MaxStart mechanism of MaxNet (using multi-bit signaling) can control mice flows to the target rate more quickly than Reno sources. By the experiments, we showed that the performance of mice and elephant traffic when using with MaxNet is better than with TCP Reno/RED's in terms of response time and network utilization. For future work, more experiments should be conducted with realistic web workload and other performance properties such as fairness, TCP friendly should be evaluated.

B.7 Bibliography

[CKH04] M. Claypool, R. Kinicki, and M. Hartling, *"Active Queue Management for Web traffic"*, IEEE International Performance, Computing, and Communications Conference (IPCCC), 2004.

[Dummynet] *Dummynet live network emulation tool*, http://info.iet.unipi.it/~luigi/dummynet/.

[FJ93] S. Floyd and V. Jacobson, *"Random Early Detection Gateways for Congestion Avoidance"*, ACM Transactions on Networking **1** (1993), 397 – 413.

[FKSS99] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, *"A Self-Configuring RED Gateway"*, Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 3, 1999, pp. 1320 – 1328.

[Httperf] http://www.hpl.hp.com/research/linux/httperf/.

[Iperf] iperf.fr.

[Jac88] V. Jacobson, *"Congestion Avoidance and Control"*, ACM Special Interest Group on Data Communication (SIGCOMM), 1988.

[Jac90] V. Jacobson, *"Berkeley TCP Evolution from 4.3-Tahoe to 4.3-Reno"*, 18th Internet Engineering Task Force, 1990.

[LAJS03] L. Le, J. Aikat, K. Jeffay, and F. D. Smith, *"The Effects of Active Queue Management on Web Performance"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2003.

[SAW+08] M. Suchara, L. Andrew, R. Witt, K. Jacobsson, B. Wydrowski, and S. Low, *"Implementation of Provably Stable MaxNet"*, International Conference on Broadband Communications, Networks and Systems (BROADNETS), 2008.

[SWW05] M. Suchara, R. Witt, and B. Wydrowski, *"TCP MaxNet - Implementation and Experiments on the WAN in Lab"*, IEEE International conference on Networks (ICON), 2005.

[WAZ03] B. Wydrowski, L. L. H. Andrew, and M. Zukerman, *"MaxNet: A Congestion Control Architecture for Scalable Networks"*, IEEE Communications Letters **7** (2003), 511 – 513.

[WJLH06] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, *"FAST TCP: Motivation, Architecture, Algorithms, Performance"*, ACM Transactions on Networking **14** (2006), 1246–1259.

Bibliography

[AABP07] A. Altin, E. Amaldi, P. Belotti, and M. C. Pinar, *"Provisioning Virtual Private Networks under Traffic Uncertainty"*, Networks **49** (2007), 100–115. (Cited in pages 29 and 56.)

[ACC$^+$13] B. Addis, A. Capone, G. Carello, L. G. Gianoli, and B. Sansò, *"A Robust Optimization Approach for Energy-aware Routing in MPLS Networks"*, International Conference on Computing, Networking and Communications (ICNC), 2013, pp. 567–572. (Cited in page 88.)

[ACC$^+$14] ———, *"Energy Management Through Optimized Routing and Device Powering for Greener Communication Networks"*, IEEE/ACM Transactions on Networking **22** (2014), 313 – 325. (Cited in page 84.)

[ACG13] E. Amaldi, A. Capone, and L. .G. Gianoli, *"Energy-aware IP Traffic Engineering with Shortest Path Routing"*, Computer Networks **57** (2013), 1503–1517. (Cited in pages 14, 84, 86, 87 and 91.)

[ACGM11] E. Amaldi, A. Capone, L. G. Gianoli, and L. Mascetti, *"Energy Management in IP Traffic Engineering with Shortest Path Routing"*, IEEE WoWMoM, 2011, pp. 1–6. (Cited in page 84.)

[AGA$^+$08] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker, *"Packet Caches on Routers: the Implications of Universal Redundant Traffic Elimination"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2008, pp. 219–230. (Cited in pages 4, 8, 9, 12, 13, 35, 37, 61 and 73.)

[AMAR09] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, *"Redundancy in Network Traffic: Findings and Implications"*, ACM Special Interest Group for the Computer Systems Performance Evaluation Community (SIGMETRICS), 2009, pp. 37–48. (Cited in pages 4, 8, 9, 10, 12, 13, 35, 36, 61 and 73.)

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *"Network Flows: Theory, Algorithms, and Applications"*, Prentice-Hall, Inc., 1993. (Cited in page 28.)

[ASA09] A. Anand, V. Sekar, and A. Akella, *"SmartRE: an Architecture for Coordinated Network-wide Redundancy Elimination"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2009, pp. 87–98. (Cited in pages 9, 12, 13, 35 and 36.)

[Ata02] A. Atamtürk, *"On Capacitated Network Design Cut-set Polyhedra"*, Mathematical Programming **60** (2002), 425–437. (Cited in page 39.)

[BBC$^+$14] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, and P. Lago, *"A Closed-Form Model for the IEEE 802.3az Network and Power Performance"*, IEEE Journal on Selected Areas in Communications **32** (2014), 16 – 27. (Cited in page 4.)

[BBDC11] 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 Communication Surveys and Tutorials **13** (2011), 223 – 244. (Cited in pages 1, 4 and 84.)

[BBK02] Suman Banerjee, Bobby Bhattacharjee, and Christopher Kommareddy, *"Scalable Application Layer Multicast"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2002, pp. 205–217. (Cited in page 129.)

[BCL$^+$10] A. P. Bianzino, C. Chaudet, F. Larroca, D. Rossi, and J. L. Rougier, *"Energy-aware Routing: A Reality Check"*, IEEE GLOBECOM Workshop, 2010. (Cited in page 6.)

[BCRR12] A. P. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, *"A Survey of Green Networking Research"*, IEEE Communication Surveys and Tutorials **14** (2012), 3 – 20. (Cited in pages 1, 4, 84 and 107.)

[BDB$^+$11] R. Bolla, F. Davoli, R. Bruschi, K. Christensen, F. Cucchietti, and S. Singh, *"The Potential Impact of Green Technologies in Next-generation Wireline Networks: Is There Room for Energy Saving Optimization?"*, IEEE Communications Magazine **49** (2011), 80 – 86. (Not cited.)

[BFI$^+$07] R. Boivie, N. Feldman, Y. Imai, W. Livens, and D. Ooms, *Explicit Multicast (Xcast) Concepts and Options*, RFC 5058, 2007. (Cited in page 127.)

[BlueCoat] *Bluecoat: Wan optimization*, `http://www.bluecoat.com/`. (Cited in pages 9, 34 and 35.)

[BM86] F. Barahona and A. Mahjoub, *"On the Cut Polytope"*, Mathematical Programming **36** (1986), 157–173. (Cited in page 41.)

[BR01] A. Basu and J. Riecke, *"Stability Issues in OSPF Routing"*, ACM Special Interest Group on Data Communication (SIGCOMM), vol. 31, 2001, pp. 225–236. (Cited in page 84.)

[BS03] D. Bertsimas and M. Sim, *"Robust Discrete Optimization and Network Flows"*, Mathematical Programming **98** (2003), 49 – 71. (Cited in pages 29, 56, 59, 60 and 65.)

[BS04] ______, *"The Price of Robustness"*, Operations Research **52** (2004), 35 – 53. (Cited in pages 29, 59, 88 and 93.)

[BTGN09] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *"Robust optimization"*, Princeton Series in Applied Mathematics, Princeton University Press, 2009. (Cited in pages 29 and 88.)

[BTN02] A. Ben-Tal and A. Nemirovski, *"Robust Optimization - Methodology and Application"*, Mathematical Programming **92** (2002), 453 – 480. (Cited in page 29.)

[CCGS13] A. Capone, C. Cascone, L. G. Gianoli, and B. Sansò, *"OSPF Optimization via Dynamic Network Management for Green IP Networks"*, Sustainable Internet and ICT for Sustainability (SustainIT), 2013, pp. 1–9. (Cited in pages 14 and 84.)

[CCRP13] L. Chiaraviglio, A. Cianfrani, E. L. Rouzic, and M. Polverini, *"Sleep Modes Effectiveness in Backbone Networks with Limited Configurations"*, Computer Networks **57** (2013), 2931–2948. (Cited in pages 4, 7, 84 and 121.)

[CEL$^+$10] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini, *"An Energy Saving Routing Algorithm for a Green OSPF Protocol"*, IEEE INFOCOM Workshop, 2010, pp. 1–5. (Cited in page 84.)

[CEL$^+$12] A. Cianfrani, V. Eramo, M. Listanti, M. Polverini, and A. V. Vasilakos, *"An OSPF-Integrated Routing Strategy for QoS-Aware Energy Saving in IP Backbone Networks"*, IEEE Transactions on Network and Service Management **9** (2012), 254 – 267. (Cited in page 84.)

[CELP11] A. Cianfrani, V. Eramo, M. Listanti, and M. Polverini, *"An OSPF Enhancement for Energy Saving in IP Networks"*, IEEE INFOCOM Workshop, 2011, pp. 325–330. (Cited in page 84.)

[CFP$^+$09] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. Gude, N. McKeown, and S. Shenker, *"Rethinking Enterprise Network Control"*, IEEE/ACM Transaction in Networking **17** (2009), 1270 – 1283. (Cited in page 107.)

[Chv83] Vašek Chvátal, *"Linear Programming"*, W. H. Freeman, 1983. (Cited in pages 27 and 28.)

[Cisco05] *Cisco: Ospf design guide*, `www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/7039-1.html`, 2005. (Cited in pages 14 and 90.)

[Cisco13] *Cisco visual networking index: Forecast and methodology, 2012 - 2017*, `http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html`, 2013. (Cited in page 2.)

[CKH04] M. Claypool, R. Kinicki, and M. Hartling, *"Active Queue Management for Web traffic"*, IEEE International Performance, Computing, and Communications Conference (IPCCC), 2004. (Cited in page 138.)

[CKP14a] D. Coudert, A. Kodjo, and T. K. Phan, *"Robust Energy-aware Routing with Redundancy Elimination"*, Computers and Operations Research (submitted), 2014. (Cited in pages 17 and 19.)

[CKP14b] ———, *"Robust Energy-aware Routing with Redundancy Elimination"*, `http://hal.inria.fr/hal-00936745`, 2014, INRIA Research Report. (Cited in pages 17 and 20.)

[CKP14c] ———, *"Robust Optimization for Energy-aware Routing with Redundancy Elimination"*, Algotel, 2014. (Cited in pages 17 and 19.)

[CKPT13] D. Coudert, A. Koster, T. K. Phan, and M. Tieves, *"Robust Redundancy Elimination for Energy-aware Routing"*, IEEE International Conference on Green Computing and Communications (GreenCom), 2013, pp. 179–186. (Cited in pages 17, 19 and 88.)

[CLRS09] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *"Introduction to Algorithms"*, MIT press, 2009. (Cited in page 30.)

[CMN09] L. Chiaraviglio, M. Mellia, and F. Neri, *"Energy-aware Backbone Networks: a Case Study"*, IEEE International Conference on Communications (ICC) Workshop, 2009. (Cited in page 4.)

[CMN11] ———, *"Minimizing ISP Network Energy Cost: Formulation and Solutions"*, IEEE/ACM Transaction in Networking **20** (2011), 463 – 476. (Cited in pages 35, 36, 46, 56, 57, 61, 72, 91 and 107.)

[CMTY11] A. R. Curtis, J. C. Mogul, J. Tourrilhes, and P. Yalagandula, *"DevoFlow: Scaling Flow Management for High-Performance Networks"*, ACM Special Interest Group on Data Communication (SIGCOMM), vol. 41, 2011, pp. 254 – 265. (Cited in pages 15 and 108.)

[CRN$^+$10] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. A. Maestro, *"IEEE 802.3az: The Road to Energy Efficient Ethernet"*, IEEE Communications Magazine **48** (2010), 50–56. (Cited in page 4.)

[CSB$^+$08] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, *"Power Awareness in Network Design and Routing"*, IEEE International Conference on Computer Communications (INFOCOM), 2008. (Cited in pages 4, 5, 7, 34 and 56.)

[DKK$^+$13] S. Duhovniko, A. M. C. A. Koster, M. Kutschka, F. Rambach, and D. Schupke, "T-*Robust Network Design for Mixed-Line-Rate-Planning* *of Optical Networks"*, National Fiber Optic Engineers Conference (OFC/NFOEC), 2013. (Cited in page 29.)

[Dummynet] *Dummynet live network emulation tool*, `http://info.iet.unipi.it/~luigi/dummynet/`. (Cited in page 137.)

[EUF14] *"European Union Funded Projects Information in Green Networks"*, `http://cordis.europa.eu/fp7/ict/sustainable-growth/`, 2014. (Cited in page 3.)

[FJ93] S. Floyd and V. Jacobson, *"Random Early Detection Gateways for Congestion Avoidance"*, ACM Transactions on Networking **1** (1993), 397 – 413. (Cited in page 134.)

[FKSS99] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, *"A Self-Configuring RED Gateway"*, Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 3, 1999, pp. 1320 – 1328. (Cited in pages 136 and 138.)

[FT00] B. Fortz and M. Thorup, *"Internet Traffic Engineering by Optimizing OSPF Weights"*, Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 2, 2000, pp. 519–528. (Cited in pages 14, 85, 87, 92 and 95.)

[FT02] ______, *"Optimizing OSPF/IS-IS Weights in a Changing World"*, IEEE Journal on Selected Areas in Communications **20** (2002), 756–767. (Cited in pages 84, 85, 91 and 121.)

[FWMG13] F. Francois, N. Wang, K. Moessner, and S. Georgoulas, *"Green IGP Link Weights for Energy-efficiency and Load-balancing in IP Backbone Networks"*, International Conferences on Networking (IFIP NETWORKING), 2013, pp. 1–9. (Cited in pages 14, 84, 87, 91 and 121.)

[GC07] T. Jr. Grevers and J. Christner, *"Application Acceleration and WAN Optimization Fundamentals"*, Cisco Press, 2007. (Cited in pages 9, 10, 11, 12, 34, 35, 36 and 45.)

[GGS13] R. G. Garroppo, S. Giordano, and G. Nencioniand M. G. Scutella, *"Mixed Integer Non-Linear Programming Models for Green Network Design"*, Computers and Operations Research **40** (2013), 273 – 281. (Cited in page 7.)

[Global07] *Global action plan*, `http://globalactionplan.org.uk`, 2007. (Cited in page 2.)

[GMMO10] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy, *"Minimizing Routing Energy Consumption: from Theoretical to Practical Results"*, IEEE/ACM Green Computing and Communications (Green-

Com), 2010, pp. 252–259. (Cited in pages 4, 8, 30, 35, 41, 47, 48, 49, 50, 72, 91, 106, 112, 116 and 117.)

[GMP14a] F. Giroire, J. Moulierac, and T. K. Phan, *"Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing"*, IEEE Global Communications Conference (GlobeCom) (submitted), 2014. (Cited in pages 18 and 19.)

[GMP14b] ———, *"Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing"*, `http://hal.inria.fr/hal-00990038`, 2014, INRIA Research Report. (Cited in pages 18 and 20.)

[GMPR12] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, *"Minimization of Network Power Consumption with Redundancy Elimination"*, International Conferences on Networking (IFIP NETWORKING), **7289** (2012), 247–258. (Cited in pages 7, 13, 17, 20, 36, 41, 47, 57, 72 and 91.)

[GMPR14] ———, *"Minimization of Network Power Consumption with Redundancy Elimination"*, Computer Communications (submitted), 2014. (Cited in pages 17, 19 and 41.)

[GNTS13] R. Garroppo, G. Nencioni, L. Tavanti, and M. G. Scutella, *"Does Traffic Consolidation Always Lead to Network Energy Savings?"*, IEEE Communications Letters **17** (2013), 1852 – 1855. (Cited in page 7.)

[GreenTouch13] *Greentouch green meter research study: Reducing the net energy consumption in communications networks by up to 90% by 2020*, `http://www.greentouch.org/index.php?page=green-meter-research`. (Cited in page 2.)

[GS03] M. Gupta and S. Singh, *"Greening of the Internet"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2003, pp. 19–26. (Cited in pages 3, 56 and 107.)

[HSM$^+$10] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, *"ElasticTree: Saving Energy in Data Center Networks"*, USENIX conference on Networked systems design and implementation (NSDI), 2010. (Cited in pages 15, 106, 107, 108 and 121.)

[Httperf] `http://www.hpl.hp.com/research/linux/httperf/`. (Cited in page 138.)

[HWG12] E. Halepovic, C. Williamson, and M. Ghaderi, *"Enhancing redundant network traffic elimination"*, Computer networks, 2012, pp. 795–809. (Cited in page 12.)

[IBM] IBM ILOG, *CPLEX Optimization Studio 12.4*. (Cited in pages 47, 61, 73, 94 and 113.)

[Iperf] iperf.fr. (Cited in page 138.)

[IPv6Day] http://www.worldipv6launch.org/. (Cited in page 130.)

[Jac88] V. Jacobson, *"Congestion Avoidance and Control"*, ACM Special Interest Group on Data Communication (SIGCOMM), 1988. (Cited in pages 134 and 137.)

[Jac90] ———, *"Berkeley TCP Evolution from 4.3-Tahoe to 4.3-Reno"*, 18th Internet Engineering Task Force, 1990. (Cited in page 134.)

[JKM$^+$13] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Holzle, S. Stuart, and A. Vahdat, *"B4: Experience with a Globally-Deployed Software Defined WAN"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2013. (Cited in page 14.)

[Juniper] *Juniper networks: Application acceleration*, http://www.juniper.net/us/en/products-services/application-acceleration. (Cited in page 9.)

[KHK13] Y. Kanizo, D. Hay, and I. Keslassy, *"Palette: Distributing Tables in Software-defined Networks"*, IEEE INFOCOM Mini-conference, 2013. (Cited in pages 106, 107, 115 and 116.)

[KKR11] A. M. C. A. Koster, M. Kutschka, and C. Raack, *"On the Robustness of Optimal Network Designs"*, IEEE International Conference on Communications (ICC), 2011, pp. 1 – 5. (Cited in pages 29, 56, 65 and 88.)

[KKR13] ———, *"Robust Network Design: Formulation, Valid Inequalities, and Computations"*, Networks **61** (2013), 128 – 149. (Cited in pages 71, 76, 80 and 88.)

[KLRW13] N. Kang, Z. Liu, J. Rexford, and D. Walker, *"Optimizing the "One Big Switch" Abstraction in Software-Defined Networks"*, ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT), 2013. (Cited in pages 106, 107 and 115.)

[KPT13] A. Koster, T. K. Phan, and M. Tieves, *"Extended Cutset Inequalities for the Network Power Consumption Problem"*, International Network Optimization Conference (INOC), Electronic Notes in Discrete Mathematics, **41** 2013, 69 – 76. (Cited in pages 17, 19, 39, 41 and 56.)

[LAJS03] L. Le, J. Aikat, K. Jeffay, and F. D. Smith, *"The Effects of Active Queue Management on Web Performance"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2003. (Cited in page 136.)

[Lan09] C. Lange, *"Energy-related aspects in backbone networks"*, 35th European Conference on Optical Communication, 2009. (Not cited.)

[LHV$^+$12] S. Lambert, W. V. Heddeghem, W. Vereecken, B. Lannoo, D. Colle, and M. Pickavet, *"Worldwide Electricity Consumption of Communication Networks"*, Optics Express **20** (2012), 513–524. (Cited in page 2.)

[LKL$^+$09] B. P. Lim, E. K. Karrupiah, E. S. Lin, T. K. Phan, N. Thoai, E. Muramoto, and P. Y. Tan, *"Bandwidth fair application layer multicast for multi-party video conference application"*, IEEE Consumer Communications and Networking Conference (CCNC), 2009, pp. 1–5. (Cited in pages 17, 18 and 20.)

[LKWG11] C. Lange, D. Kosiankowski, R. Weidmann, and A. Gladisch, *"Energy Consumption of Telecommunication Networks and Related Improvement Options"*, IEEE Journal of Selected Topics in Quantum Electronics **17** (2011), 285 – 295. (Cited in page 2.)

[LL10] J. T. Linderoth and A. Lodi, *"MILP software"*, Wiley Encyclopedia of Operations Research and Management Science, 2010. (Cited in page 28.)

[LRCBS05] E. Lev-Ran, I. Cidon, and I. Z. Ben-Shaul, *"Method and Apparatus for Reducing Network Traffic over Low Bandwidth Links"*, Patent US 7636767, 2005. (Cited in page 9.)

[LTC12] S. S. W. Lee, P. Tseng, and A. Chen, *"Link Weight Assignment and Loop-free Routing Table Update for Link State Routing Protocols in Energy-aware Internet"*, Future Generation Computer Systems **28** (2012), 437–445. (Cited in page 84.)

[MAB$^+$08] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, *"Openflow: Enabling Innovation in Campus Networks"*, ACM Computer Communication Review **38** (2008), 69 – 74. (Cited in pages 14 and 106.)

[Report11] `www-sop.inria.fr/mascotte/rapports_stages/KhoaPhan_internship-2011.pdf`. (Cited in page 45.)

[Min06] Michel Minoux, *"Multicommodity network flow models and algorithms in telecommunications"*, Handbook of Optimization in Telecommunications, Springer, 2006, pp. 163–184. (Cited in page 28.)

[MLT12] C. R. Meiners, A. X. Liu, and E. Torng, *"Bit Weaving: A Non-prefix Approach to Compressing Packet Classifiers in TCAMs"*, IEEE/ACM Transaction in Networking **20** (2012), 488 – 500. (Cited in pages 107 and 110.)

[MMV93] T. Magnanti, P. Mirchandani, and R. Vachani, *"The Convex Hull of Two Core Capacitated Network Design Problems"*, Mathematical Programming **60** (1993), 233–250. (Cited in page 39.)

[MP14a] J. Moulierac and T. K. Phan, *"Optimizing IGP Link Weights for Energy-efficiency in a Changing World"*, Computer Communications (submitted), 2014. (Cited in pages 18 and 19.)

[MP14b] ______, *"Optimizing IGP Link Weights for Energy-efficiency in a Changing World"*, `http://hal.inria.fr/hal-00988882`, 2014, INRIA Research Report. (Cited in pages 18 and 20.)

[MPTT11] J. Moulierac, T. K. Phan, N. Thoai, and N. C. Tran, *"Xcast6 Treemap Islands - A Mixed Model of Application and Network Layer Multicast"*, `http://hal.inria.fr/inria-00637656/`, 2011, INRIA Research Report. (Cited in pages 18 and 20.)

[MSB09] P. Mahadevan, P. Sharma, and S. Banerjee, *"A Power Benchmarking Framework for Network Devices"*, International Conferences on Networking (IFIP NETWORKING), 2009, pp. 795–808. (Cited in pages 7 and 34.)

[NMN^{+}13] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, *"A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks"*, IEEE Communications Surveys and Tutorials (2013), 1 – 18. (Cited in page 109.)

[OWPT10] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, *SNDlib 1.0 - survivable network design library*, Networks **55** (2010), no. 3, 276–286. (Cited in pages 46, 47, 61, 73, 94, 106, 113 and 129.)

[PMTT12] T. K. Phan, J. Moulierac, N. C. Tran, and N. Thoai, *"Xcast6 Treemap Islands - Revisiting Multicast Model"*, ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT) (Student Workshop), 2012, pp. 33–34. (Cited in pages 17, 18 and 20.)

[Powerlib] *Powerlib*, `http://powerlib.intec.ugent.be/database/`. (Cited in page 7.)

[PTM^{+}10a] K. T. Phan, N. Thoai, E. Muramoto, K. K. Ettikan, B. P. Lim, and P. Y. Tan, *"Treemap - the Fast Routing Convergence Method for Application Layer Multicast"*, IEEE Consumer Communications and Networking Conference (CCNC), 2010, pp. 1–5. (Cited in pages 17, 18 and 20.)

[PTM^{+}10b] T. Khoa Phan, N. Thoai, E. Muramoto, K. K. Ettikan, B. P. Lim, and P. Y. Tan, *"Treemap - The Fast Routing Convergence Method for Application Layer Multicast"*, IEEE Consumer Communications and Networking Conference (CCNC), 2010. (Cited in page 128.)

[PTNT12] T. K. Phan, T. T. Tran, D. D. Nguyen, and N. Thoai, *"MaxNet and TCP Reno/RED on Mice Traffic"*, Modeling, Simulation and Optimization of Complex Processes, 2012, pp. 247–255. (Cited in pages 17, 19 and 20.)

[Riverbed] `http://www.riverbed.com/us/solutions/wan_optimization/`. (Cited in pages 9, 34 and 35.)

[RKOW11] C. Raack, A. M. C. A. Koster, S. Orlowski, and R. Wessäly, *"On Cut-based Inequalities for Capacitated Network Design Polyhedra"*, Networks **57** (2011), 141 – 156. (Cited in pages 38, 39, 87 and 112.)

[SAW$^+$08] M. Suchara, L. Andrew, R. Witt, K. Jacobsson, B. Wydrowski, and S. Low, *"Implementation of Provably Stable MaxNet"*, International Conference on Broadband Communications, Networks and Systems (BROADNETS), 2008. (Cited in pages 134, 136 and 137.)

[SCF$^+$12] B. Stephens, A. Cox, W. Felter, C. Dixon, and J. Carter, *"PAST: Scalable Ethernet for Data Centers"*, ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT), 2012. (Cited in page 106.)

[Sch98] A. Schrijver, *"Theory of Linear and Integer Programming"*, John Wiley & Sons, Inc., 1998. (Cited in pages 27 and 28.)

[SGG10] Y. Song, K. Guo, and L. Gao, *"Redundancy-aware Routing with Limited Resources"*, International Conference on Computer Communications and Networks (ICCCN), 2010, pp. 1–6. (Cited in pages 4, 9, 13, 35 and 36.)

[SLX$^+$12] M. Shen, H. Liu, K. Xu, N. Wang, and Y. Zhong, *"Routing On Demand: Toward the Energy-Aware Traffic Engineering with OSPF"*, International Conferences on Networking (IFIP NETWORKING), 2012, pp. 232–246. (Cited in pages 14 and 84.)

[Sma10] *"SMART 2020: Enabling the low carbon economy in the information age"*, `http://www.smart2020.org/_assets/files/02_Smart2020Report.pdf`. (Cited in pages 2 and 3.)

[SW00] N. T. Spring and D. Wetherall, *"A Protocol-Independent Technique for Eliminating Redundant Network Traffic"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2000. (Cited in pages 8, 9 and 35.)

[SWW05] M. Suchara, R. Witt, and B. Wydrowski, *"TCP MaxNet - Implementation and Experiments on the WAN in Lab"*, IEEE International conference on Networks (ICON), 2005. (Cited in page 134.)

[TSP] *"Approximation Algorithms for the Traveling Salesman Problem"*, http://people.orie.cornell.edu/dpw/orie6300/Recitations/rec11.pdf. (Cited in page 30.)

[VNS+11] N. Vasic, D. Novakovic, S. Shekhar, P. Bhurat, M. Canini, and D. Kostic, *"Identifying and Using Energy-Critical Paths"*, ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT), 2011. (Cited in page 15.)

[WAZ03] B. Wydrowski, L. L. H. Andrew, and M. Zukerman, *"MaxNet: A Congestion Control Architecture for Scalable Networks"*, IEEE Communications Letters **7** (2003), 511 – 513. (Cited in page 134.)

[WBR11] R. Wang, D. Butnariu, and J. Rexford, *"OpenFlow-based Server Load Balancing Gone Wild"*, USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE), 2011. (Cited in page 107.)

[WJLH06] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, *"FAST TCP: Motivation, Architecture, Algorithms, Performance"*, ACM Transactions on Networking **14** (2006), 1246–1259. (Cited in page 135.)

[WVS+99] A. Wolman, M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy, *"On the Scale and Performance of Cooperative Web Proxy Caching"*, ACM Symposium on Operating Systems Principles (1999), 16–31. (Cited in page 9.)

[WYW+12] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, *"CARPO: Correlation-Aware Power Optimization in Data Center Networks"*, Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2012. (Cited in pages 15, 88, 106, 107, 108 and 121.)

[X6T11] *Research report*, hal.inria.fr/docs/00/63/76/56/PDF/RR-7784.pdf, 2011. (Cited in pages 128, 129 and 131.)

[X6Demo] www.cogma.org/press/video/xcast_e_1000k.wmv. (Cited in page 128.)

[ZA13] Y. Zhang and N. Ansari, *"On Protocol-Independent Data Redundancy Elimination"*, IEEE Communications Surveys and Tutorials **16** (2013), 455 – 472. (Cited in pages 4, 9 and 12.)

[ZC11] E. Zohar and I. Cidon, *"The Power of Prediction: Cloud Bandwidth and Cost Reduction"*, ACM Special Interest Group on Data Communication (SIGCOMM), 2011. (Cited in page 4.)

[ZWLW14] K. Zheng, X. Wang, L. Li, and X. Wang, *"Joint Power Optimization of Data Center Network and Servers with Correlation Analysis"*, IEEE INFOCOM, 2014, pp. 2598–2606. (Cited in page 88.)

[ZYLZ10] M. Zhang, C. Yi, B. Liu, and B. Zhang, *"GreenTE: Power-aware Traffic Engineering"*, IEEE International Conference on Network Protocols (ICNP), 2010, pp. 21–30. (Cited in pages 3, 4 and 56.)