ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

MINIMIZING SIGNALING COST IN GREEN ROUTING FOR SOFTWARE DEFINED NETWORKS

M.Sc. THESIS

Hadi ALIZADEH

Department of Computer Engineering

Computer Engineering Programme

MAY 2015

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

MINIMIZING SIGNALING COST IN GREEN ROUTING FOR SOFTWARE DEFINED NETWORKS

M.Sc. THESIS

Hadi ALIZADEH
(504121523)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Ayşegül YAYIMLI

Co-advisor: Assist. Prof. Dr. Didem GÖZÜPEK

MAY 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM TANIMLI AĞLAR İÇİN YEŞİL YÖNLENDİRMEDE SİNYALLEŞME MALİYETİNİ MİNİMİZE ETME

YÜKSEK LİSANS TEZİ

Hadi ALIZADEH
(504121523)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Assoc. Prof. Dr. Ayşegül YAYIMLI

Eş Danışman: Assist. Prof. Dr. Didem GÖZÜPEK

MAYIS 2015

Hadi ALIZADEH, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504121523 successfully defended the thesis entitled **"MINIMIZING SIGNALING COST IN GREEN ROUTING FOR SOFTWARE DEFINED NETWORKS"**, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Ayşegül YAYIMLI** Istanbul Technical University	....
Co-advisor :	**Assist. Prof. Dr. Didem GÖZÜPEK** Gebze Technical University	....
Jury Members :	**Prof. Dr. Sema FATMA OKTUĞ** Istanbul Technical University	....
	Assist. Prof. Dr. Berk CANBERK Istanbul Technical University	....
	Assist. Prof. Dr. Fatma CORUT ERGİN Marmara University	....

Date of Submission : 4 May 2015
Date of Defense : 29 May 2015

FOREWORD

I would like to express my sincere gratitude and appreciation to those who have helped me in the successful completion of this thesis. Firstly, I would like to thank my advisor, Assoc. Prof. Ayşegül YAYIMLI for her guidance throughout the whole process of this thesis.
Deepest gratitude is due to Assist. Prof. Didem GÖZÜPEK, who provided valuable suggestions for my master thesis. I am grateful for her constant support and help as my co-advisor.
This thesis has been supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under grant no.113E567.
Last but not least, I owe my thanks to my family for their support during my study.

May 2015 Hadi ALIZADEH

TABLE OF CONTENTS

Page

FOREWORD........ **vii**
TABLE OF CONTENTS........ **ix**
ABBREVIATIONS........ **xi**
LIST OF TABLES........ **xiii**
LIST OF FIGURES........ **xv**
SUMMARY........ **xvii**
ÖZET........ **xix**
1. INTRODUCTION........ **1**
1.1 Contributions........ 5
1.2 Network Model........ 5
1.3 Thesis outline........ 6
2. PROBLEM FORMULATION........ **7**
2.1 Signaling Overhead-Aware Green Routing (SOAGR)........ 7
2.2 Extended Signaling Overhead-Aware Green Routing (E-SOAGR)........ 11
3. HEURISTIC ALGORITHMS........ **17**
3.1 Heuristic Algorithms for SOAGR........ 17
3.2 Heuristic Algorithms for E-SOAGR........ 21
3.2.1 Path selection phase in E-SGH........ 24
3.3 Time Complexity Analysis of Our Proposed Heuristics........ 25
3.3.1 Time complexity analysis of FGH........ 25
3.3.2 Time complexity analysis of SGH........ 26
3.3.3 Time complexity analysis of E-SGH........ 27
4. NUMERICAL EVALUATION........ **29**
4.1 Numerical Evaluation of Experiments for SOAGR Problem........ 29
4.1.1 Simulation setup of experiments for SOAGR problem........ 29
4.1.2 Impact of network energy threshold........ 30
4.1.3 Impact of network topology density........ 33
4.2 Numerical Evaluation of Experiments for E-SOAGR Problem........ 34
4.2.1 Simulation setup of experiments for E-SOAGR problem........ 34
4.2.1.1 Topology 1: GEANT........ 35
4.2.1.2 Topology 2: JANOS-US........ 37
4.2.2 Impact of network energy threshold on the signaling overhead........ 37
4.2.3 Impact of traffic load on the blocking rate of demands........ 37
4.2.4 Impact of multipath routing on the blocking rate of demands........ 40
5. CONCLUSIONS AND FUTURE WORK........ **43**
REFERENCES........ **45**

CURRICULUM VITAE **49**

ABBREVIATIONS

OSPF : Open Shortest Path First
IBGP : Internal Border Gateway Protocol
SDN : Software Defined Networking
ILP : Integer Linear Programming
SOAGR : Signaling Overhead-Aware Green Routing
E-SOAGR : Extended Signaling Overhead-Aware Green Routing
ICT : Information and Communication Technologies
CDN : Content Delivery Network
MPLS : Multi-Protocol Label Switching
FGH : Fully Greedy Heuristic
SGH : Stepwise Greedy Heuristic
E-SGH : Extended Stepwise Greedy Heuristic

LIST OF TABLES

Page

Table 2.1 : Input parameters in SOAGR. 9
Table 2.2 : Decision variables in SOAGR. 9
Table 2.3 : Input parameters in E-SOAGR. 12
Table 2.4 : Decision variables in E-SOAGR. 13
Table 3.1 : Input parameters of FGH and SGH 18
Table 3.2 : Input parameters for E-SGH. 22

LIST OF FIGURES

Page

Figure 1.1 : The SDN architecture with decoupled control [1]............................ 6
Figure 3.1 : Fully Greedy Heuristic (FGH) for SOAGR problem 19
Figure 3.2 : Stepwise Greedy Heuristic (SGH) for SOAGR problem 20
Figure 3.3 : Extended Stepwise Greedy Heuristic (E-SGH) for E- SOAGR problem 23
Figure 3.4 : Path selection function for E-SGH 24
Figure 4.1 : Impact of W_T on the amount of control messages in SOAGR........... 31
Figure 4.2 : Impact of W_T on the number of feasible solutions in SOAGR. 32
Figure 4.3 : Amount of control messages with respect to different node degrees for $W_T = 60$.. 34
Figure 4.4 : Real topologies used in experiments for E-SOAGR.......................... 36
Figure 4.5 : Impact of W_T on the amount of signaling overhead in E-SOAGR 38
Figure 4.6 : Impact of traffic load on the amount of blocking rate in E-SOAGR . 39
Figure 4.7 : Impact of multipath routing on the blocking rate of demands in E-SOAGR 41

MINIMIZING SIGNALING COST IN GREEN ROUTING FOR SOFTWARE DEFINED NETWORKS

SUMMARY

Currently, with the development of next generation networks, there is an urgent request for reducing energy consumption in telecommunication networks due to its environmental impact and potential economic benefits. Some research studies show that energy consumption in communication networks is mainly related to active network elements such as communication links. Based on this approach, several energy management techniques, generally known as green techniques, have been proposed. Their main goal is to minimize energy consumption by routing network traffic through a set of network resources and powering off the remaining unused resources.

The current green techniques impose a signaling overhead on the routing system due to selectively powering off/on the network resources and require a centralized network management paradigm to determine which devices to switch off/on. However, the existing networking paradigms including OSPF or IBGP networks do not provide such a centralized management.

Recently, the idea of software defined networking (SDN) has been put forward. SDN is a networking paradigm in which the network control is (logically) centralized in the controller and network devices are simple forwarding devices that can be programmed by the controller. The centralized management of the network makes SDN a proper paradigm for developing green routing techniques.

In this thesis, we address the trade-off between energy efficiency and signaling overhead in a software defined network (SDN). We formulate two green routing problems which minimize signaling overhead in an SDN network with a single domain while taking the network energy consumption into account. We formulate these optimization problems using Integer Linear Programming (ILP). Our proposed ILP formulations are Signaling Overhead-Aware Green Routing (SOAGR) and Extended Signaling Overhead-Aware Green Routing (E-SOAGR). The ILP formulation of E-SOAGR, which is an extension of the ILP formulation of SOAGR, presents a more general model by introducing additional features such as multipath routing, taking into account routing table sizes of the forwarding hardware and considering different bandwidth requirements for demands.

We implement the two green routing models in the IBM ILOG CPLEX Optimization Studio and find optimal solutions for both SOAGR and E-SOAGR models. However, SOAGR and E-SOAGR are computationally very difficult problems. Therefore, we develop polynomial-time heuristic algorithms to find near-optimal solutions for both models.

We compare the performance of our proposed green routing models and heuristics by conducting experiments on both random and realistic network topologies. The obtained results demonstrate the tradeoff between energy efficiency and signaling overhead.

YAZILIM TANIMLI AĞLAR İÇİN YEŞİL YÖNLENDİRMEDE SİNYALLEŞME MALİYETİNİ MİNİMİZE ETME

ÖZET

Yeni nesil ağların kurulumuyla birlikte, çevresel etkileri ve potansiyel ekonomik faydaları nedeniyle telekomünikasyon ağlarında enerji tüketimini azaltmaya acil bir ihtiyaç vardır. Bu nedenle ulusal ve uluslararası kurum ve kuruluşlar maliyetleri düşürmek ve çevreyi korumak için enerji tasarrufu yönünde çalışmalar yapmaktadır. Örneğin, son yıllarda dünya enerji tüketiminin yaklaşık %10'unu oluşturması sebebiyle iletişim teknolojileri, enerji duyarlı çalışmaların ilgi odaklarından biri olmuştur.

Bazı araştırma çalışmaları, iletişim ağlarındaki enerji tüketiminin iletişim linkleri gibi aktif ağ elemanlarıyla ilişkili olduğunu göstermektedir. İletişim ağlarının düşük trafik yoğunluklarında bile çalışmalarına devam etmeleri enerji tüketimini artırmaktadır. Bu bağlamda, genel olarak yeşil teknikler olarak bilinen pek çok enerji yönetim tekniği önerilmiştir. Bu yöntemlerin temel amacı, ağ trafiğini belli bir ağ kaynakları kümesi aracılığıyla yönlendirerek ve kullanılmayan kaynaklarını kapatarak enerji tüketimini minimize etmektir.

Bazı araştırmacılar, var olan yeşil yöntemleri ağ temelli ve yerel (cihaz seviyesinde) yaklaşımlar olarak iki gruba ayırmıştır. Cihaz temelli yaklaşımlar ileri teknolojili donanımlar sayesinde ağ cihazlarının enerji verimliliğini iyileştirmeyi amaçlamaktadırlar. Bu yaklaşımlarda ağ cihazı yerel şekilde alınan kararlar sonucunda uyku moduna veya düşüş enerji tüketen bir moda alınmaktadır.

Yönlendirme perspektifinden bakarak ağdaki toplam enerji tüketimini azaltmayı amaçlayan ağ temelli yaklaşımlar ise kablolu ağlarda enerji tasarrufu için umut verici yaklaşımlar olarak değerlendirilmektedir. Ağ temelli yaklaşımlar yeşil yönlendirme adıyla da geçmektedir; amaçları ise ağ kaynaklarının belli bir alt kümesi üzerinde talepleri toplamak ve kullanılmayan diğer kaynakları kapatmak veya uyku moduna almaktır.

Şu anda var olan yeşil teknikler, seçmeli olarak ağ kaynaklarını kapatıp açmaları nedeniyle yönlendirme sistemi üzerinde bir sinyalleşme maliyeti oluşturmaktadırlar ve hangi cihazların kapatılıp açılacağını belirlemek için merkezi bir ağ yönetim paradigmasını gerektirmektedirler. Ancak, şu anda var olan OSPF veya IBGP gibi ağ paradigmaları böyle bir merkezi yönetimi sağlamamaktadır. OSPF veya IBGP komşuluk bilgilerine dayanan bir yönlendirme modeli üzerine kurulmuşlardır. Her yönlendiricinin komşu bağlantılar ile arasındaki bilgilere göre karar veriyor olması, komşuluk bilgilerinin güncellemesi ağın büyümesi ile zaman alan bir işlem olmakta, güncelleme yapılana kadar ağ bilgileri değişmektedir.

Yakın zamanda, yazılım tanımlı ağ (software defined networks, SDN) fikri öne atılmıştır. SDN modelinde, ağ elemanları merkezi bir kontrol birimine

doğrudan bağlanmakta ve kontrol katmanları anılan merkezi birimde sanal olarak toplanmaktadır. Merkezi kontrol biriminin ağ elemanlarına doğrudan bağlantısı olması, topolojinin anlık ve genel görünümünü sağlamasıyla kontrol biriminin trafiği etkin bir şekilde yönetmesini ve ağda meydana gelecek değişikliklere karşı hızlı düzenleme yapmasını sağlamaktadır. Ağın merkezi olarak yönetimi SDN'i yeşil ağ yönlendirmesi tekniklerini geliştirmek açısından uygun bir teknoloji kılmaktadır.

Bu tezde ele aldığımız SDN ağında, kontrol ünitesi iki tipte kontrol mesajı üretmektedir: yönlendirme güncellemesi (routing update messages) ve uyku mesajları (sleep messages). Yönlendirme güncellemesi mesajları yeni akış tablosu girdilerini içermektedirler ve hesaplanan patikalar üzerindeki tüm düğümlere gönderilmektedirler. Uyku mesajları ise uyuma kararı verilen tüm linklerin her iki ucundaki düğümlerinin portlarına birden gönderilmektedirler.

Bu tezde, tek bir kontrol alanından oluşan yazılım tanımlı bir ağda enerji verimliliği ve sinyalleşme maliyeti arasındaki ödünleşimi adresliyoruz. Sözü geçen SDN ağında ağın enerji tüketimini dikkate alarak sinyalleşme maliyetini minimize eden iki adet yeşil yönlendirme problemi formüle ediyoruz. Bu optimizasyon problemlerini tamsayı doğrusal programlama (Integer Linear Programming, ILP) problemleri olarak formüle ediyoruz. Önerdiğimiz ILP formülasyonları Sinyalleşme Maliyeti Farkındalığı olan Yeşil Yönlendirme (Signaling Overhead-Aware Green Routing, SOAGR) ve Genişletilmiş Sinyalleşme Maliyeti Farkındalığı olan Yeşil Yönlendirme (Extended Signaling Overhead-Aware Green Routing, E-SOAGR) olarak isimlendirilmiştir.

E-SOAGR'ın ILP formülasyonu, SOAGR'ın ILP formülasyonunun genişletilmiş halidir ve aşağıdaki ilave özellikleri içermektedir:

- çoklu patikalı yönlendirme
- yönlendirme tablosunun büyüklüğünü dikkate alma
- farklı trafik talepleri için farklı bantgenişliği gerekliliklerini dikkate alma

Her iki yeşil yönlendirme modelini de IBM ILOG CPLEX optimizasyon yazılımı aracılığıyla uyguluyor ve optimal çözümleri buluyoruz. Ancak SOAGR ve E-SOAGR hesaplama karmaşıklığı açısından çok zor problemlerdir. Dolayısıyla, her iki model için de optimale yakın sonuçlar bulmak amacıyla polinom zamanlı buluşsal algoritmalar öneriyoruz.

Tez kapsamında, ILP formülasyonlarımızın ve buluşsal algoritmalarımızın performanslarını karşılaştırmak için hem rastlantısal hem de gerçek ağ topolojilerinde simülasyon çalışmaları yaptık. Bu çalışmalar, önerdiğimiz modellerin sinyalleşme maliyeti ve enerji verimliliği arasındaki ödünleşimi adreslediğini ve buluşsal algoritmalarımızın ürettiği çözümlerin kalitesinin optimal çözümlere oldukça yakın olduğunu göstermektedir.

ILP'lerin ve heuristic'lerin performanslarını sinyalleşme maliyeti, enerji verimliliği ve bloke olma oranı açısından inceledik. SOAGR problemi için olan deneylerde, ağın enerji eşik değerinin ve ağ yoğunluğunun sinyalleşme maliyeti üzerindeki etkisini rastlantısal olarak üretilmiş topolojileri kullanarak inceledik. E-SOAGR ILP formülasyonu için ise gerçekçi topolojiler üzerinde bir takım deneyler yaptık. Bu amaçla, GEANT ve JANOS-US isminde iki topolojiyi seçtik. Yaptığımız deneylerde ağ enerji eşik değerinin sinyalleşme maliyeti üzerindeki etkisini inceledik. Ayrıca, trafik yükünün ve çoklu yönlendirmenin taleplerin bloke olma oranı üzerindeki etkilerini de inceledik.

1. INTRODUCTION

Energy conservation is an important issue for countries due to its contribution to the economy as well as its environmental influences. Among economy sectors, the information and communication technologies (ICT) sector has paid much more attention to energy efficiency over the past years. According to [2], the annual electricity consumption of networking devices in the U.S. is 6.06 Terra watt hours, which is equal to nearly 1 billion US dollars per year. The same study shows that transmitting data through Internet takes more energy (in bits per Joule) than transmitting data through wireless networks. The increasing energy consumption has an environmental impact. For instance, in 2007, ICT industry [3] accounted for 2% of the global CO_2 emissions, which is same amount as global air travel.

Energy management has received attention in wired networks very recently although it has been a critical issue in wireless communications for a long time [4]. Some studies indicate that traffic load of the routers has little impact on their energy consumption. They consider active network elements such as routers and links as the major causes of energy consumption [5]. Network elements are usually powered on even in their idle mode, during which they consume a large amount of energy. Thus, researchers have hitherto proposed many techniques to bring energy-awareness into network elements and processes. These techniques are generally called green techniques [6]. Fisher et al. [7] categorized the existing techniques into network-wide and local (device-level) approaches.

Device-level approaches aim to improve the energy efficiency of individual network devices by advanced technological hardware. In these approaches, the networking device is put into sleep or a lower energy mode based on local decisions. [5] presented a power measurement study of a variety of networking devices such as hubs, edge switches, core switches, routers and wireless access points. The authors proposed a link rate adaptation technique for low traffic periods and sleeping for no traffic periods. Mahadevan et al. [8] estimated the power consumption of a switch using a linear model.

In this model, the energy saving is obtained by adapting the rate of a port, disabling a port as well as completely switching off a device.

Network-wide schemes which aim to reduce the total network power consumption from routing perspective, have been put forward as promising approaches for energy conservation in wired networks. Network-wide schemes are concerned with *green routing*, whose objective is to aggregate demands over a subset of network resources and allow other resources to be switched off [9].

Many works have hitherto approached green routing as an optimization problem. Amongst these works, Chabarek et al. [10] formulated a green routing problem as a multiple commodity network flow problem and used mixed-integer programming techniques to investigate power consumption and network performance in both static and dynamic network design. [6] formulated a minimum energy routing problem as an optimization problem, analyzed a green routing algorithm, and solved it numerically considering a core-network scenario.

It is shown in [10] that the energy consumption of network devices is largely independent of their load. Thus, the power consumption in networks is mainly due to active devices. In line with this fact, the authors in [11] aimed to find a routing that minimizes the number of active links. For this purpose, they modeled the routing problem as an integer linear program (ILP) and then proposed a heuristic algorithm.

The work in [12] proposed an energy-aware routing model and compared it with an energy-agnostic routing model. The objective in [12] is to minimize the total power consumption of the network taking into consideration a user delay constraint.

Chiaraviglio et al. [13] proposed a green approach guaranteeing full connectivity and maximum link utilization. They also developed some heuristic algorithms to solve it. The works in [14, 15] presented network-wide power management techniques for data center networks.

A network-level strategy based on a modification of current link-state routing protocols, such as OSPF, is proposed in [16]. According to this strategy, IP routers are able to power off some network links during low traffic periods.

Authors in [17] studied the impact of using in-network caches and content delivery network (CDN) cooperation in energy-efficient routing. They formulated an energy efficient content distribution problem and proposed an ILP to reduce energy consumption by using caches and properly choosing content provider servers for each demand.

Since switching from one path to another requires time and additional signaling overhead, an energy-aware mechanism should in general avoid rerouting too often. [18] proposed an ILP formulation for traffic engineering including constraints on the changes of the device states and routing paths to limit the impact on quality of service and the signaling overhead.

The work in [19] proposed an algorithm called GRiDA to selectively switch off links in an Internet Service Provider IP-based network to reduce the system energy consumption. Their approach is completely distributed among the nodes and leverages link state protocol like OSPF to limit the amount of shared information and to reduce the algorithm complexity.

However, in current IP networks, turning off/on links would trigger link state routing protocols to reconverge to a new topology. Since the reconvergence process takes tens of seconds, routing table inconsistencies among routers may result in network disconnection [20]. Thus, making network-wide decisions for green routing is a problematic task in the current OSPF or IBGP networks as there is a need for a centralized decision making mechanism to determine which devices to switch off/on in green routing. In the current protocols, routers are responsible for establishing routes for demands. Thus, a green routing approach for the current networks has to collect link state data from the entire network, create a database and distribute it to all routers [2].

Recently, the idea of software defined networking (SDN) [1] has attracted considerable attention. SDN is a new networking paradigm which brings the separation of data and control planes via a uniform vendor-agnostic interface called OpenFlow [21]. In this paradigm, the network intelligence is (logically) centralized in the controller and network devices are simple forwarding devices that can be programmed by the controller. SDN provides a centralized management as well as a fast reshaping of the traffic through modifications of the flow tables in the forwarding devices. The aforementioned properties make SDN a proper environment for developing green routing techniques.

Using SDN technology, [22] explored a new solution to energy-aware flow scheduling in data centers. Authors in [23] proposed an energy-aware routing and resource management model for Multi-Protocol Label Switching (MPLS) networks using SDN-based approach. [24] addressed global power management at network level in SDN-based networks. An ILP is formulated to minimize the network power consumption under constraints of link utilization and packet delay. Moreover, two heuristic algorithms are proposed to solve this problem. Another work on SDN networks [25] proposed an optimization method to minimize energy consumption for a backbone network while respecting capacity constraints on links and rule space constraints on routers.

An SDN controller periodically collects port, flow, and table level statistics from switches in its domain using control messages [26]. Implementing energy-aware algorithms in an SDN controller increases the number of these control messages as well. Considering that a controller usually has a limited resource capacity, the energy efficiency and control overhead are conflicting parameters in a green routing scheme.

The research studies about routing in green networks [6, 10–20] and resource management in SDN [22–26] mainly focus on minimizing total energy consumption and ignore the impact of control overhead such as routing update and sleep messages. To the best of our knowledge, this thesis is the first study tackling the trade-off between energy efficiency of green routing and the control overhead it generates in an SDN domain with a single controller.

In this thesis, we formulate a green routing problem as a multiple commodity network flow problem. In particular, we propose an ILP problem whose objective function is to minimize the control overhead subject to an energy constraint that limits the total energy consumption of the network. We then propose polynomial-time heuristic algorithms to find near-optimal solutions for the problem. We evaluate the performance of our heuristic algorithms by comparison with the results obtained from our ILP formulation using optimization software CPLEX [27].

1.1 Contributions

In this thesis we propose two green routing problems which minimize signaling overhead while taking the network energy consumption into account. Our main contribution is to address the tradeoff between signaling overhead and energy efficiency. We formulate ILP problems for these two green routing problems in an SDN network with a single domain. We name these formulations as Signaling Overhead-Aware Green Routing (SOAGR) and Extended Signaling Overhead-Aware Green Routing (E-SOAGR). The ILP formulation of E-SOAGR which is an extension of the ILP formulation of SOAGR, presents a more general model by introducing additional features.

We have implemented the two green routing problems in the IBM ILOG CPLEX Optimization Studio and developed polynomial-time heuristic algorithms to find near-optimal solutions for the aforementioned problems.

By conducting experiments on random and realistic network topologies, we compare the performance of our proposed green routing problems and heuristics. The results show that our proposed methods achieve substantial energy savings while minimizing the signaling overhead.

1.2 Network Model

We consider an SDN network with a single domain for our green routing. As visualized in Fig. 1.1, the considered SDN architecture decouples control logic from the forwarding hardware. The solid lines define the data-plane links and the dashed lines the control-plane links.

In the considered system, controller is responsible for performing green routing. Since an SDN controller periodically collects port, flow, and table level statistics from the forwarding hardware in its domain, we assume that the nodes cannot be put into sleep mode. Therefore, the energy conservation in the network is obtained by putting links into sleep mode.

The controller generates two types of control messages: *routing update* and *sleep* messages. Routing update messages, which contain new flow table entries, are sent

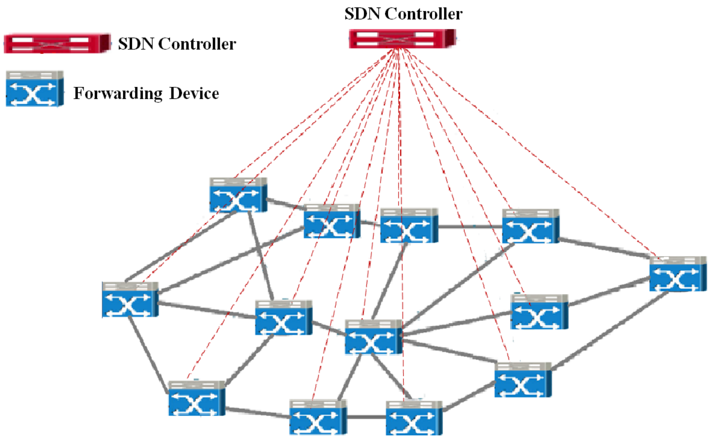


Figure 1.1: The SDN architecture with decoupled control [1]

to all nodes along calculated paths and sleep messages are sent to the nodes at both ends of the links for which a sleep decision has been made.

1.3 Thesis outline

The objective of this thesis is to study and investigate the tradeoff between signaling overhead and energy efficiency. We focus on minimizing signaling overhead while considering the total energy consumption of the network. The remainder of this thesis is organized as follows:

Chapter 2 presents our ILP formulations for the two SOAGR and E-SOAGR problems. In this chapter, we describe the input parameters, decision variables, and the constraints in detail.

Chapter 3 introduces our heuristics for both SOAGR and E-SOAGR problems. We also present the complexity analysis of our heuristics in this chapter.

Chapter 4 presents numerical results of our simulations. We have conducted experiments on both random and realistic topologies to validate the performance of our proposed ILP formulations and heuristics. The performance of the ILPs and the heuristics is compared in terms of signaling overhead, energy efficiency, and blocking rate.

Chapter 5 concludes the thesis by summarizing the main contributions and putting forward some possible directions for future work.

2. PROBLEM FORMULATION

We consider an SDN network with a single domain for our green routing. In the considered system, controller is responsible for performing green routing. Since the forwarding hardware have to periodically send port, flow, and table level statistics to the SDN controller, we assume in our modeling that the nodes cannot be put into sleep mode. Therefore, the energy conservation in the network is obtained by putting links into sleep mode. In our modeling, the controller sends sleep messages to the ports at both ends of a link for which a sleep decision has been made.

We represent our considered SDN network as a graph $G(V,E)$, where the set of nodes V denotes interconnection devices and the set of edges E represents communication links. In this model, each link $(i, j) \in E$ between two nodes i and j has a capacity C_{ij}. A set of traffic demands P is given as an input to our formulation. Our ILP aims to establish a path for each demand $p \in P$.

Furthermore, an important aspect in our modeling is the temporal nature of the green routing problem. Arrival of new demands into the system or the release of a number of existing connections necessitates rerouting by which the system transits into a new state from the viewpoint of energy consumption. We will refer to these states as *configuration* in our modeling. In other words, a configuration is a network state in a particular time period, during which the set of traffic demands given as an input to our ILP is static. Results of the previous configuration are fed as input to the current configuration.

In the following, we describe the two ILP formulations SOAGR and E-SOAGR in detail.

2.1 Signaling Overhead-Aware Green Routing (SOAGR)

In our ILP formulation for SOAGR, the objective function is to minimize the signaling cost of green routing by accommodating a single path for each demand. The traffic demand between a pair of nodes is considered as a pair (s_p, t_p) where $s_p \in V$ is the source of the demand and $t_p \in V$ is the destination of the demand. It is assumed that the bandwidth requirement of all demands is the same.

Tables 2.1 and 2.2 outline the input parameters and the decision variables in the ILP formulation of SOAGR.

Objective Function:

The objective function consists of two terms: the first one is the total cost of routing update messages and the second one is the total cost of sleep messages generated by the SDN controller.

$$min \quad \sum_{i=1}^{|V|} c_i \cdot r_i + \sum_{i=1}^{|V|} d_i \cdot s_i \tag{2.1}$$

Constraints:

x_{ij} represents active (used) links in the new configuration. New configuration should belong to the given graph $G(V,E)$:

$$x_{ij} \leq g_{ij}; \qquad \forall i, j \in V \tag{2.2}$$

Considering the current configuration, a link (i, j) is active if some path p uses it:

$$x_{ij}^p \leq x_{ij}; \qquad \forall i, j \in V, \forall p \in P \tag{2.3}$$

The link (i, j) is inactive in the current configuration if no path uses it:

$$x_{ij} \leq \sum_{p=1}^{|P|} x_{ij}^p; \qquad \forall i, j \in V \tag{2.4}$$

A routing update message is sent to node *i* if link (i, j) is used by path *p* in the current configuration (x_{ij}^p=1) but not used by the same path *p* in the previous configuration (l_{ij}^p=0). The following constraint models this behavior, where r_i denotes the number of routing update messages sent to node *i*:

$$r_i = \sum_{j=1}^{|V|} \sum_{p=1}^{|P|} x_{ij}^p \cdot (1 - l_{ij}^p); \qquad \forall i \in V \tag{2.5}$$

Table 2.1: Input parameters in SOAGR.

Input Parameter	Explanation
g_{ij}	It represents the network topology. g_{ij} is equal to 1 if a link between node i and node j (link $(i,j) \in E$) exists in the topology.
l_{ij}	It is equal to 1 if link (i,j) is used by some path in the previous configuration.
l_{ij}^p	It is equal to 1 if link (i,j) is used by path p in the previous configuration.
c_i	Cost of a routing table update message sent to node i.
d_i	Cost of a sleep message sent to node i.
w_{ij}	Energy consumption of link (i,j).
W_T	The upper limit for total energy consumption of the network.
P	Set of demands
s_p	Source of demand $p \in P$.
t_p	Destination of demand $p \in P$.
C_{ij}	Capacity of link (i,j) in terms of the number of paths that link (i,j) can accommodate.

Table 2.2: Decision variables in SOAGR.

Decision variable	Explanation
x_{ij}	A binary variable that is equal to 1 if any path uses link (i,j) in the direction from i to j in the current configuration.
x_{ij}^p	A binary variable that is equal to 1 if link (i,j) is used by path p in the direction from i to j in the current configuration.
r_i	An integer variable that denotes the number of routing update messages sent to node i.
s_i	An integer variable that denotes the number of sleep messages sent to node i.

A sleep message is sent to node i if link (i, j) is active in the previous configuration ($l_{ij} = 1$), but not active in the current configuration ($x_{ij} = 0$). This situation implies that the link has to be put into sleep mode, which can be accomplished when nodes at both ends of the link put their respective ports into sleep. The following constraint models this behavior, where s_i denotes the number of sleep messages sent to node i:

$$s_i = \sum_{j=1}^{|V|} \Big(l_{ij} \cdot (1 - x_{ij}) + l_{ji} \cdot (1 - x_{ji}) \Big); \quad \forall i \in V \tag{2.6}$$

Flow conservation constraint:

The following constraints guarantee that each demand p is allocated a path from its source to its destination. $\forall i \in V, \forall p \in P$:

$$\sum_{j'=1}^{|V|} x^p_{ij'} - \sum_{j'=1}^{|V|} x^p_{j'i} = \begin{cases} 1; & \text{if } i = s_p \\ -1; & \text{if } i = t_p \\ 0; & \text{otherwise} \end{cases} \tag{2.7}$$

Energy consumption constraint:

Total energy consumption depends on the number of active links. The total energy consumption of all active links should not exceed the given threshold W_T:

$$\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} w_{ij} \cdot x_{ij} \leq W_T; \tag{2.8}$$

Link capacity constraint: The following constraint ensures that the maximum number of paths sharing link (i, j) does not exceed C_{ij}:

$$\sum_{p=1}^{|P|} x^p_{ij} \leq C_{ij}; \qquad \forall i, j \in V \tag{2.9}$$

Constraints on decision variables:

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in V \tag{2.10}$$

$$x^p_{ij} \in \{0, 1\} \qquad \forall i, j \in V, \quad \forall p \in P \tag{2.11}$$

$$r_i \in Z^+ \cup \{0\} \qquad \forall i \in V \tag{2.12}$$

$$s_i \in Z^+ \cup \{0\} \qquad \forall i \in V \tag{2.13}$$

2.2 Extended Signaling Overhead-Aware Green Routing (E-SOAGR)

In this section we present the ILP formulation of E-SOAGR which is an extension of the ILP formulation in Equations (2.1) to (2.13). E-SOAGR introduces new features as follows:

- providing multipath routing
- taking routing table sizes into account via an additional constraint
- considering different bandwidth requirements for traffic demands

E-SOAGR has the ability to either accommodate a single path for each demand or split each demand into multiple lower rate demands and accommodate a separate path for each of them if required. In this formulation, the traffic demand between a pair of nodes is considered as a triple (s_p, t_p, D_p) where $s_p \in V$ is the source of the demand, $t_p \in V$ is the destination of the demand and D_p is the amount of bandwidth requested by demand p.

Tables 2.3 and 2.4 outline the input parameters and the decision variables in the ILP formulation of E-SOAGR.

Objective Function:
Similar to SOAGR, the objective function of E-SOAGR consists of two terms: the first one is the total cost of routing update messages and the second one is the total cost of sleep messages.

$$min \quad \sum_{i=1}^{|V|} c_i.r_i + \sum_{i=1}^{|V|} d_i.s_i \qquad \textbf{(2.14)}$$

Constraints:
x_{ij} represents active links in the new configuration. New configuration should belong to the given graph $G(V,E)$:

$$x_{ij} \leq g_{ij}; \quad \forall i, j \in V \qquad \textbf{(2.15)}$$

Table 2.3: Input parameters in E-SOAGR.

Input Parameter	Explanation
g_{ij}	It represents the network topology. g_{ij} is equal to 1 if a link between node i and node j (link (i,j)) exists in the topology.
l_{ij}	It is equal to 1 if link (i,j) is used by some path in the previous configuration.
l_{ij}^{pk}	It is equal to 1 if link (i,j) is used by path k of pair p in the previous configuration.
c_i	Cost of a routing table update message for node i
d_i	Cost of a sleep message sent to node i from the controller
w_{ij}	Energy consumption of link (i,j)
P	Set of demands
s_p	Source of demand $p \in P$.
t_p	Destination of demand $p \in P$.
D_p	Bandwidth requested by source-destination pair p
W	The upper limit considered for total energy consumption
C_{ij}	Link capacity
K	Maximum number of paths for multipath routing
R	Upper limit for the routing table size

Table 2.4: Decision variables in E-SOAGR.

Decision Variable	Explanation
x_{ij}	A binary variable that is equal to 1 if some path uses link (i,j) in the direction from i to j in the current configuration
x_{ij}^{pk}	An integer variable that shows the amount of bandwidth used by path k of pair p in link (i,j) in the direction from i to j in the current configuration.
y_{ij}^{pk}	A binary variable that is equal to 1 if link (i,j) is used by path k of pair p in the direction from i to j in the current configuration.
B_{pk}	An integer variable that shows the amount of bandwidth allocated to path k of pair p.
r_i	An integer variable that shows the number of routing update messages sent to node i.
s_i	An integer variable that denotes the number of sleep messages sent to node i.

Considering the current configuration, a link (i,j) is active if any path k of any pair p uses it.

$$y_{ij}^{pk} \leq x_{ij}; \quad \forall i,j \in V,\ \forall p \in P,\ \forall k \in K \tag{2.16}$$

The link (i,j) is inactive if no path uses it:

$$x_{ij} \leq \sum_{k=1}^{|K|} \sum_{p=1}^{|P|} y_{ij}^{pk}; \quad \forall i,j \in V \tag{2.17}$$

A routing update message is sent to node i if link (i,j) is used by path k of pair p in new configuration (x_{ij}^{pk}=1) but not used by the same path k of pair p in the previous configuration (l_{ij}^{pk}=0). r_i shows the number of routing update messages sent to node i.

$$r_i = \sum_{j=1}^{|V|} \sum_{p=1}^{|P|} \sum_{k=1}^{|K|} y_{ij}^{pk}.(1-l_{ij}^{pk}); \quad \forall i \in V \tag{2.18}$$

The following constraint ensures that the number of routing update messages sent to each node does not exceed the given threshold R:

$$r_i \leq R; \quad \forall i \in V \tag{2.19}$$

A sleep message is sent to node i if link (i, j) is active in the previous configuration ($l_{ij} = 1$), but not active in the current configuration ($x_{ij} = 0$). This situation implies that the link (i, j) has to be put into sleep mode, which can be accomplished when nodes at both ends of the link put their respective ports into sleep. The following constraint models this behavior:

$$s_i = \sum_{j=1}^{|V|} \Big(l_{ij}.(1 - x_{ij}) + l_{ji}.(1 - x_{ji}) \Big); \quad \forall i \in V \tag{2.20}$$

Energy consumption constraint:

Total energy consumption depends on the number of active links, irrespective of the direction of data flow. Hence, the total energy consumption of all active links should not exceed the given threshold W_T:

$$\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} w_{ij}.x_{ij} \leq W_T \tag{2.21}$$

Flow conservation constraint:

The following constraints guarantee that each demand p is allocated at most K paths from its source to its destination. $\forall i \in V, \forall p \in P, \forall k \in K$:

$$\sum_{j'=1}^{|V|} x_{ij'}^{pk} - \sum_{j'=1}^{|V|} x_{j'i}^{pk} = \begin{cases} B_{pk}; & \text{if } i = s_p \\ -B_{pk}; & \text{if } i = t_p \\ 0; & \text{otherwise} \end{cases} \tag{2.22}$$

Link capacity constraint:

The following constraint ensures that the overall bandwidth allocated to the paths sharing link (i, j) does not exceed C_{ij}:

$$\sum_{p=1}^{|P|} \sum_{k=1}^{|K|} x_{ij}^{pk} \leq C_{ij}; \quad \forall i, j \in V \tag{2.23}$$

The total bandwidth allocated to at most K sub-demands of demand p must be equal to the amount of bandwidth requested by demand p. We refer by sub-demands to the portions of a demand which are routed over separate paths. The following constraint models the splitting of demand p into multiple smaller sub-demands:

$$D_p = \sum_{k=1}^{|K|} B_{pk}; \quad \forall p \in P \tag{2.24}$$

The following constraints model the relationship between the decision variables x_{ij}^{pk} and y_{ij}^{pk} where M is a sufficiently large number:

$$y_{ij}^{pk} \leq x_{ij}^{pk}; \quad \forall i,j \in V,\ \forall p \in P,\ \forall k \in K \tag{2.25}$$

$$x_{ij}^{pk} \leq M.y_{ij}^{pk}; \quad \forall i,j \in V,\ \forall p \in P,\ \forall k \in K \tag{2.26}$$

Constraints on decision variables:

$$x_{ij} \in \{0,1\} \quad \forall i,j \in V \tag{2.27}$$

$$y_{ij}^{pk} \in \{0,1\} \quad \forall i,j \in V,\ \forall p \in P,\ \forall k \in K \tag{2.28}$$

$$r_i \in Z^+ \cup \{0\} \qquad \forall i \in V \tag{2.29}$$

$$s_i \in Z^+ \cup \{0\} \qquad \forall i \in V \tag{2.30}$$

$$x_{ij}^{pk} \in Z^+ \cup \{0\} \qquad \forall i \in V \tag{2.31}$$

$$B_{pk} \in Z^+ \cup \{0\} \qquad \forall i \in V \tag{2.32}$$

3. HEURISTIC ALGORITHMS

Our proposed green routing problems, SOAGR and E-SOAGR, formulated in Equations (2.1) to (2.13) and Equations (2.14) to (2.32) are integer programming problems, which are in general computationally very difficult problems. We implement these two ILP formulations in optimization software CPLEX and find optimal solutions for the problems. However, finding an optimal solution may take too long time especially for large scale problems. In addition, all computers have a limited amount of memory resources, which may pose an obstacle to the execution of CPLEX for large scale networks. Therefore, designing efficient heuristic algorithms for SOAGR and E-SOAGR problems is vital. To this end, we present in this chapter polynomial-time heuristic algorithms for both SOAGR and E-SOAGR problems.

3.1 Heuristic Algorithms for SOAGR

In this section we explain our two proposed heuristic algorithms for SOAGR problem. We name these heuristics *Fully Greedy Heuristic* (FGH) and *Stepwise Greedy Heuristic* (SGH).

Table 3.1 shows the input parameters of our heuristic algorithms, whereas, Fig. 3.1 and Fig. 3.2 present the pseudocodes of FGH and SGH, respectively.

Before elaborating on the details of each heuristic algorithm, we explain the commonalities between the two heuristics. At each configuration, a number of demands may arrive into the system or a number of existing connections may need to be released. We refer to these demands as *arriving* (F_a) and *terminated* demands (F_r). At each configuration, both heuristics first deal with the terminated demands and release the resources allocated to these demands. These resources can further be used in accommodating the arriving demands.

At the beginning of the first configuration, the value of current energy consumption of the network (ε) is set to zero because it is assumed that there is no demand in

Table 3.1: Input parameters of FGH and SGH

Input Parameter	Explanation
G	Network topology
F_a	Set of arriving demands
F_r	Set of terminated demands
C	Capacity of each link
W	Energy consumption of each link
W_T	Energy threshold of the network
W_{temp}	Varying energy threshold
ε	Current energy consumption of the network
k	A parameter for Yen's algorithm
δ	Step value for varying threshold

the system and all network links are inactive. The value of ε may be increased by accommodating arriving demands and decreased by releasing the terminated demands.

Recall that routing update messages are sent to all nodes along a calculated path. Thus, the number of total routing update messages generated by the SDN controller for a calculated path is equal to the length of that path in terms of the number of hops. Hence, it is necessary to allocate the possible shortest path for each demand to minimize the number of routing update messages required to establish a connection for that demand. In both heuristics, Yen's K-shortest path algorithm [28] is used to find a set containing k shortest paths for each demand.

Yen's algorithm computes K shortest paths from a source to a destination for a graph with non-negative edge costs. Yen's algorithm can use any efficient shortest path algorithm to determine the shortest path from a source to a destination. For this purpose, we use Dijkstra's shortest path algorithm [29] in implementation of our heuristic algorithms.

```
Require: G, F_r, F_t, C, W, k, W_T
for each demand f ∈ F_r do
    Release demand f;
    Update C and ε;
end
for each demand f ∈ F_a do
    S ← Yen_K_ShortestPath(G,f,k);
    p ← Select_Best_Path(S);
    if (p ≠ ∅) then
        Accommodate path p for demand f;
        Update C and ε;
    else
        return InfeasibleSolution;
    end
end
return TotalControlMessages;
```

Figure 3.1: Fully Greedy Heuristic (FGH) for SOAGR problem

The output of Yen's algorithm is a set containing k paths which are sorted in increasing order according to their lengths. The function *Select_Best_Path()* scans this set and chooses the first path which satisfies both energy threshold and network capacity constraints. The path selected by *Select_Best_Path()* is assigned to the demand f.

If all demands in a configuration are accommodated, the heuristics calculate routing update and sleep messages for that configuration. However, failure of the algorithms in finding a path for even one of the demands in a configuration yields an infeasible solution for that configuration and for the experiment, which is performed for a certain value of the energy threshold of the network (W_T).

The heuristics differ only in the way they accommodate the demands. FGH greedily activates new links and assigns shortest paths for the demands as long as ε is below W_T. If $\varepsilon = W_T$, since activating new links violates the energy threshold, FGH checks whether the previously activated links can be used to find a path for the following demands.

SGH is also a greedy algorithm; however, it is more flexible than FGH. In addition to W_T, SGH employs a varying threshold W_{temp}, which dynamically changes according to the arrival of new demands and release of the existing demands. W_{temp} varies in a range between an initialized value and W_T by using an input parameter δ. W_{temp} is initially

```
Require: G, F_r, F_t, C, W, W_T, k, δ
for each demand f ∈ F_r do
    Release demand f;
    Update C and ε;
    if (ε ≤ W_temp − δ) then
        W_temp ← W_temp − δ;
    end
end
for each demand f ∈ F_a do
    S ← Yen_K_ShortestPath(G,f,k);
    flag ← true;
    while (flag= true) do
        p ← Select_Best_Path(S);
        if (p ≠ ∅) then
            Accommodate path p for demand f;
            Update C and ε;
            flag ← false;
        else
            if (W_temp < W_T) do
                W_temp ← W_temp + δ ;
            else
                return InfeasibleSolution;
            end
        end
    end
end
return TotalControlMessages;
```

Figure 3.2: Stepwise Greedy Heuristic (SGH) for SOAGR problem

set to the value of δ at the beginning of the first configuration. SGH then increases or decreases the value of W_{temp} by δ depending on the situation.

Taking into consideration the value of W_{temp} as the upper limit for energy consumption at each time, SGH starts assigning shortest paths to the demands in a greedy manner. By accommodating the demands, the value of ε reaches the value of W_{temp}. In this situation, SGH checks whether it can find a path for the demands through the previously activated links. The paths found in this situation are not necessarily the possible shortest paths for the demands. However, if accommodation of new demands with the current value of W_{temp} is not possible and if increasing the value of W_{temp} by δ does not cause it to exceed the value of W_T, SGH increases the value of W_{temp} by δ and checks whether the demands can be accommodated in the new situation. This process is repeated until W_{temp} reaches W_T. At this point, if finding a path for a demand is not possible, SGH returns infeasible solution.

3.2 Heuristic Algorithms for E-SOAGR

In this section, we explain our proposed heuristic algorithm for E-SOAGR Problem. We name this heuristic *Extended Stepwise Greedy Heuristic* (E-SGH). E-SGH is similar to SGH explained in the previous section; however, there exist some differences between them. Among these differences, providing multipath routing is the most important one. In other words, E-SGH can split each demand into multiple lower rate demands and accommodate a separate path for each of them. It is worth noting that E-SGH prefers multipath routing when accommodation of a demand is not possible by single path routing or when multipath routing generates less amount of signaling traffic than that of single path routing.

Table 3.2 shows the input parameters of E-SGH, while Fig. 3.3 presents the pseudocode of E-SGH. In lines (1-7), E-SGH deals with the terminated demands and releases the resources allocated to these demands. After releasing each terminated connection, E-SGH updates network capacity (C), routing table of the forwarding hardware (R) and the current energy consumption of the network (ε). Releasing a number of existing connections may reduce the value of ε. This in turn causes E-SGH to check whether it can reduce the value of W_{temp} by δ.

Starting from line 8, E-SGH deals with arriving demands. Yen's K-shortest path algorithm is performed for each arriving demand f in line 9. The output of Yen's algorithm is later used as an input in path selection phase.

Path selection phase takes the graph G, the demand f, M and S as input. If path selection phase can find a path, E-SGH accommodates demand f and updates C,R, and ε. In line 16, E-SGH terminates the while loop by setting the value of flag to false and then deals with the next demand in F_a. The detailed explanation of path selection phase as well as its pseudocode is given in the following section. There exist two cases upon the failure of path selection phase. In the case where the value of W_{temp} is smaller than W_T, E-SGH increases the value of W_{temp} by δ and reexecutes

Table 3.2: Input parameters for E-SGH.

Input Parameter	Explanation
G	Network topology
F_a	Set of arriving demands
F_r	Set of terminated demands
C	Capacity of each link
R	Routing table of each node
M	Maximum number of paths in multipath routing
W	Energy consumption of each link
W_T	Energy threshold of the network
W_{temp}	Varying energy threshold
ε	Current energy consumption of the network
k	A parameter for Yen's algorithm
δ	Step value for varying threshold

Require: $G, F_r, F_t, C, W, W_T, k, \delta, M$

1: **for** *each demand* $f \in F_r$ **do**
2: Release demand *f*;
3: Update C, R, ε;
4: **if** ($\varepsilon \leq W_{temp} - \delta$) **then**
5: $W_{temp} \leftarrow W_{temp} - \delta$;
6: **end**
7: **end**
8: **for** *each demand* $f \in F_a$ **do**
9: $S \leftarrow$ Yen_K_ShortestPath(G,f,k);
10: *flag* $\leftarrow$ *true*;
11: **while** (*flag*= true) **do**
12: $p \leftarrow$ Path_Select(S, M); Described in Figure 3.4
13: **if** ($p \neq \emptyset$) **then**
14: Accommodate solution p for demand f;
15: Update C, R and ε;
16: $flag \leftarrow false$;
17: **else**
18: **if** ($W_{temp} < W_T$) **do**
19: $W_{temp} \leftarrow W_{temp} + \delta$;
20: **else**
21: Block demand f;
22: Increment the number of blocked demands;
23: $flag \leftarrow false$;
24: **end**
25: **end**
26: **end**
27: **end**
28: **return** ($TotalControlMessages$, $NumberofBlockedDemands$);

Figure 3.3: Extended Stepwise Greedy Heuristic (E-SGH) for E- SOAGR problem

path selection process by considering the new value of W_{temp} (lines 18-19). In the second case where the value of W_{temp} is equal to W_T, E-SGH blocks the demand and increment the number of blocked demands by one. Here, E-SGH terminates the while loop immediately by setting the flag value to false (line 23) and proceeds with the next demand.

At the end of the configuration, E-SGH calculates the amount of control messages for the accommodated demands. It returns the amount of total control messages and the number of blocked demands as output.

3.2.1 Path selection phase in E-SGH

Path selection function takes M and S as input. We categorize the solutions produced by this function into three types: single path, double path, and multiple path (between 3 and M paths) ones. Fig. 3.4 gives the outline of Path_Select function.

```
function Path_Select(S,M)
1:    p1 ← Select_Single_Path(S);
2:    p2 ← Select_Double_Path(S);
3:    if (p1 ≠ ∅ OR p2 ≠ ∅) then
4:        return the best solution between p1 and p2
5:    else
6:        for i = 3 to M
7:            pi ← Select_Multiple_Path(S,i);
8:            if (pi ≠ ∅) then
9:                return pi;
10:           endif
11:       endfor
12:   endif
13:   return InfeasibleSolution;
```

Figure 3.4: Path selection function for E-SGH

In Line 1, single path selection is executed. Single path selection checks whether it can find a feasible single path from the set S by taking into account the following three constraints: network capacity, the routing tables of the network nodes, and the energy threshold of the network. The output of this function (p_1) is considered as the single path candidate solution for the path selection phase.

In Line 2, double path selection is executed. This function tests all 2-element subsets of the set S in order to find a double path solution considering the three constraints, namely, the network capacity, the routing tables of the network nodes, and the energy threshold of the network. The output of this function (p_2) is considered as the double path candidate solution for the path selection phase.

In lines (3-4) the function checks if there exist a single or a double path solution and selects the solution which generates less amount of signaling overhead as the final solution for the path selection phase. However, if both single path and double path selection functions fail in finding a path for the demand, multiple path solution is tested.

In lines (5-12) we handle multiple path solutions starting from checking three-path solutions. At each iteration i, Select_Multiple_Path function checks whether it can generate a solution containing i paths. The path selection phase tests all K paths in set S for single path solutions and all 2-element subsets of the paths in set S for double path solutions. Nevertheless, due to complexity reason, we prefer to use a sliding window of size i (the number of paths in the solution) for multiple path solutions instead of testing all possible i-element subsets. We apply the sliding window from the first element in set S and move it by one position to the right at each time to check the new set of i paths inside the window. When a feasible solution is found in iteration i, we terminate the For loop and return that solution (p_i). The path selection phase returns an empty solution if $Select\_Multiple\_Path$ function fails to find a solution.

3.3 Time Complexity Analysis of Our Proposed Heuristics

In this section, we present the time complexity analysis of our proposed heuristics FGH, SGH, and E-SGH. We analyze the worst case complexity of our heuristics. We assume that the number of vertices is $|V|$, the number of edges is $|E|$, and the number of demands at each configuration is $|P|$.

3.3.1 Time complexity analysis of FGH

The basic task in FGH is dealing with the arriving demands. Since other operations including releasing terminated demands have constant time complexity, we do not include them in the calculations. For each arriving demand, two basic operations are executed. Firstly, Yen's k-shortest algorithm is executed for each demand. The worst case complexity of Yen's k-shortest path is $O\big(k \cdot |V| \cdot (|E| + |V| \cdot log(|V|))\big)$ according to [30].

Secondly, the output of Yen's algorithm, which contains k paths, is given as an input to the path selection function. In the worst case, the selection function checks all of these k paths. Furthermore, the selection function checks the feasibility of each path by considering two constraints: link capacities and network energy consumption. Both constraints are checked for all links of a path. The worst case occurs when a path has the maximum possible length, namely ($|V|$-1). Thus, the complexity of the path selection for FGH is $O(k \cdot |V|)$.

We denote the complexity of dealing with a single demand by C_f. In FGH, C_f is as follows:

$C_f = O\big(k \cdot |V| \cdot (|E| + |V| \cdot log(|V|))\big) + O(k \cdot |V|) = O\big(k \cdot |V| \cdot (|E| + |V| \cdot log(|V|))\big)$

Since there exist $|P|$ arriving demands at each configuration, the time complexity of FGH is:

$$C_{FGH} = O(|P| \cdot C_f)$$
$$= O\big(|P| \cdot k \cdot |V| \cdot (|E| + |V| \cdot log(|V|))\big)$$

3.3.2 Time complexity analysis of SGH

The algorithm of SGH is very similar to that of FGH. Yen's k-shortest algorithm is executed once for each demand. However, the value of W_T is divided into multiple steps (W_{temp}) in SGH. The feasibility of finding a path from the output of Yen's algorithm is checked for the values of W_{temp}. If accommodating a demand for a certain value of W_{temp} is not possible, the selection function is reexecuted for the next value of W_{temp}. The number of the values that W_{temp} can take is obtained from (W_T/δ). The worst case occurs when δ=1. Thus, the worst case complexity of selection function in SGH becomes $O(W_T \cdot k \cdot |V|)$. The complexity of dealing with a single demand in SGH is:

$$C_f = O\big(k \cdot |V| \cdot (|E| + |V| \cdot log(|V|))\big) + O(W_T \cdot k \cdot |V|)$$
$$= O\big(k \cdot |V| \cdot (|E| + |V| \cdot log(|V|) + W_T)\big)$$

Since there exist $|P|$ arriving demands at each configuration, the time complexity of SGH is:

$$C_{SGH} = O(|P| \cdot C_F)$$
$$= O\big(|P| \cdot k \cdot |V| \cdot (|E| + |V| \cdot log(|V|) + W_T)\big).$$

3.3.3 Time complexity analysis of E-SGH

The algorithm of E-SGH is similar to that of SGH. In E-SGH, Yen's k-shortest algorithm is executed once for each demand. Like SGH, the value of W_T is divided into multiple steps and the feasibility of finding a path from the output of Yen's algorithm is checked for each W_{temp}. However, the only difference lies in the complexity of path selection function of E-SGH. Thus, if we denote the complexity of the path selection function of E-SGH by C_{PSF}, the worst case complexity of dealing with a single demand in E-SGH becomes:

$$C_f = O\big(k \cdot |V| \cdot (|E| + |V| \cdot log(|V|)) + C_{PSF}\big) \tag{3.1}$$

The path selection function for E-SGH takes the output of Yen's algorithm, which contains k paths, as well as the maximum number of paths in multipath routing (M) as input. The path selection function first checks whether it can generate a single path solution (p_1) or a double path solution (p_2). If the path selection function fails in finding a single or a double path solution, E-SGH checks solutions with i paths where i ranges from 3 to M.

The path selection function checks all k paths to find a single path solution. Thus, the worst case complexity of checking single path solutions is $O(k \cdot |V|)$. In the case of double path solution, the path selection function checks all 2-element subsets of k paths in the output of Yen's algorithm. Thus, the worst case complexity for double path solutions is $O(k^2 \cdot |V|)$.

For the solutions containing more than two paths, we apply a sliding window on the output of Yen's algorithm. At each iteration $i \in [3,M]$, the size of the window is i and the path selection function checks whether the set of i paths inside the window generate a feasible solution. The path selection function then moves the sliding window by one position on the output of Yen's algorithm to check the next set of paths inside the window. It stops as soon as it finds a feasible solution. The complexity of checking whether the set of i paths inside the sliding window generates a feasible solution is $O(|V|)$. In the worst case, the sliding window traverses all $(k-i+1)$ positions in the output of Yen's algorithm. Hence, the worst case complexity for finding solutions at iteration i is $O(k \cdot |V|)$. Since $i \in [3,M]$, the total number of iterations is $(M-2)$ and the worst case complexity of checking solutions with more than two paths is $O(M \cdot k \cdot |V|)$. Therefore, the worst case complexity of path selection function of E-SGH is:

$$C_{PSF} = O(k \cdot |V|) + O(k^2 \cdot |V|) + O(M \cdot k \cdot |V|)$$
$$= O\big(k \cdot |V| \cdot (k+M)\big)$$

By plugging in the worst case complexity of the path selection function in Equation (3.1), the worst case complexity of dealing with a single demand in E-SGH becomes:

$$C_f = O\big(k \cdot |V| \cdot (|E| + |V| \cdot log(|V|)) + O(k \cdot |V| \cdot (k+M))\big)$$
$$= O\big(k \cdot |V| \cdot (|E| + |V| \cdot log(|V|) + k + M)\big)$$

Since there exist $|P|$ arriving demands at each configuration, the worst case time complexity of E-SGH is:

$$C_{E-SGH} = O\big(|P| \cdot k \cdot |V| \cdot (|E| + |V| \cdot log(|V|) + k + M)\big)$$

4. NUMERICAL EVALUATION

In this chapter, we present the performance evaluation of our proposed ILP formulations and heuristics. For this purpose, we have conducted a number of experiments for both SOAGR and E-SOAGR problems on both random and realistic topologies. In the following, we explain these experiments in detail.

4.1 Numerical Evaluation of Experiments for SOAGR Problem

In this section we present the numerical results of our proposed ILP formulation for SOAGR and our proposed heuristics, FGH and SGH. In our experiments, we investigate the impact of energy threshold of the network and network density on the amount of control overhead. We evaluate the performance of our heuristics by comparing them with the results obtained from our ILP formulation.

4.1.1 Simulation setup of experiments for SOAGR problem

We use randomly generated topologies in our experiments for SOAGR. For the first test, we consider a network with 30 nodes where each node has degree 4. We generate 50 different instances of this network. In the second experiment, we consider a network with 30 nodes and vary the node degrees from 3 to 7. For each value of node degree, we generate 50 different instances.

For our simulations, a network model that has high variation in its traffic demands is required. The available realistic traffic models are not applicable in our work as they are mainly intended for backbone networks, which have demands with low variations. For this purpose, we generate two traffic loads with 30 and 60 demands. The sources and destinations of each demand are randomly chosen. The arrival rate and the duration of the demands follow Poisson distribution with average inter-arrival time of 500 *ms* and exponential distribution with average duration of 10 *s*. With these parameters, we calculate the total amount of control messages in 20 configurations, which corresponds to a total simulation time of 20s. The number of demands on each configuration depends on the Poisson and exponential distributions.

As in [11,17,20], we assume that all links have the same value of energy consumption; therefore, W_T corresponds to the maximum number of links that can be active in routing the demands. Hence, the unit of W_T in figs. 4.1 and 4.2 is the number of active links. We also assume that the links have the same capacity and each link can accommodate at most 25 demands.

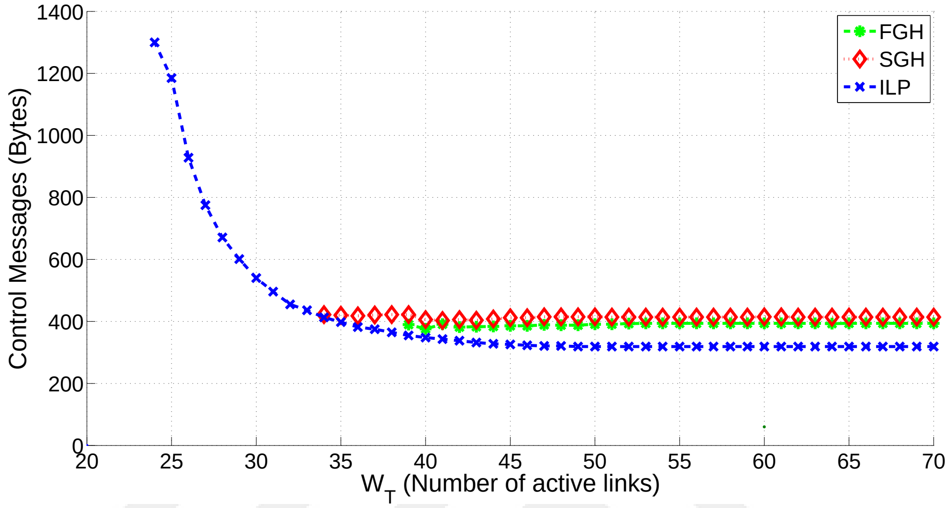
The cost of control overhead is stated in bytes. In our green routing mechanism, sleep messages contain the identity of the interface to be switched off on a specific node. We thus consider a sleep message to have a size of one byte. We also consider three bytes for a routing update message, which contains a routing table entry consisting of the source and destination of the demand as well as the next hop. In all experiments, the value of the parameter of Yen's K-shortest path algorithm (k) is set to 15 and the step value for varying threshold in SGH (δ) is set to be $0.1 \times W_T$.

4.1.2 Impact of network energy threshold

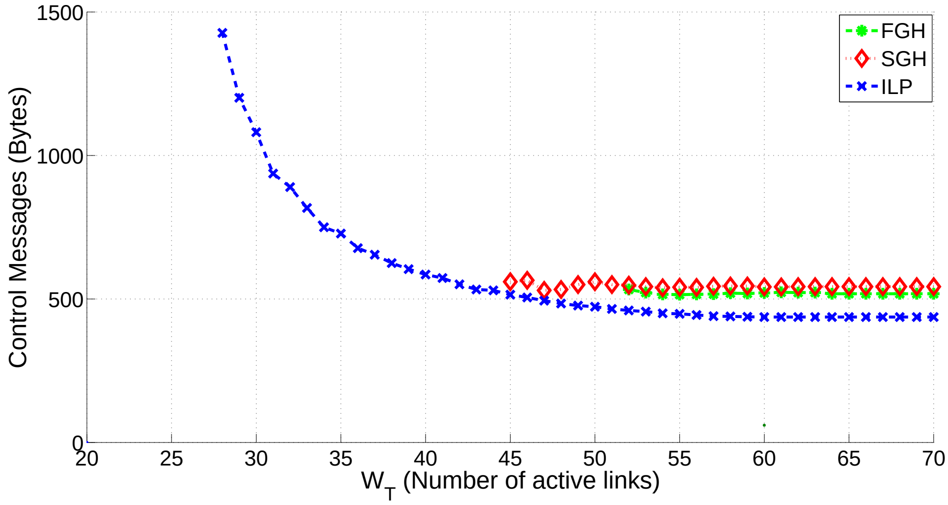
We first investigate the impact of W_T on the amount of control messages considering two traffic loads with 30 and 60 demands. Fig. 4.1 shows the results of this experiment. The network instances used for this test have 120 links. We have started the experiment by setting W_T to 120, which means that the algorithm is allowed to use all the links in routing. We have then repeated the test for the smaller values of W_T.

We observe that ILP and both heuristics have a similar behavior when W_T decreases from 120 to 45 in Fig. 4.1(a). The same behavior is observed when the value of W_T decreases from 120 to 55 in Fig. 4.1(b). In these ranges of W_T, the solutions found by our heuristics are close to those of ILP. The heuristics generate nearly 30 and 50 bytes more control messages than ILP in both traffic loads.

In the case of ILP, the number of control messages increases dramatically as W_T decreases below 45 in Fig. 4.1(a) and below 55 in Fig. 4.1(b). Our ILP formulation provides a lower bound for the amount of control messages. The heuristics generate higher amounts of control messages than this bound when they reach a feasible solution. This behavior shows that there is a trade-off between energy efficiency and

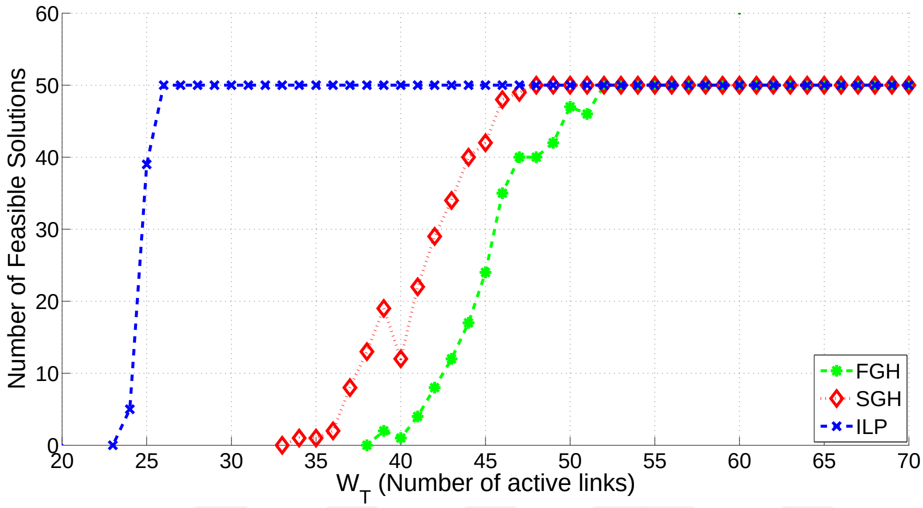


(a) Control message with respect to W_T for traffic load with 30 demands.

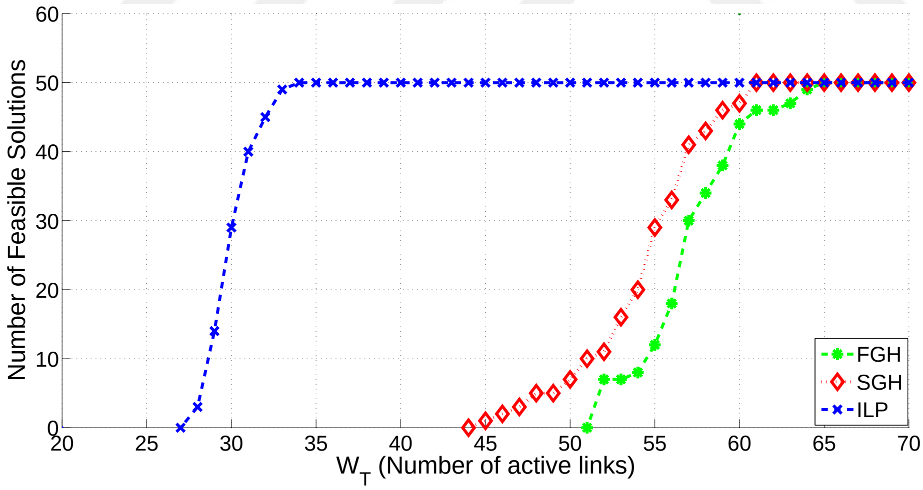


(b) Control message with respect to W_T for traffic load with 60 demands.

Figure 4.1: Impact of W_T on the amount of control messages.



(a) Number of feasible solutions with respect to W_T for traffic load with 30 demands.



(b) Number of feasible solutions with respect to W_T for traffic load with 60 demands.

Figure 4.2: Impact of W_T on the number of feasible solutions.

control overhead for values of W_T smaller than a certain value. In other words, in order to obtain more energy efficiency, more control messages have to be generated.

We also observe that the value of W_T at which the trade-off starts depends on the traffic load. As the number of demands increases from 30 to 60, this value also increases from 45 to 55. As to the performance of our heuristics, FGH outperforms SGH with a small difference of about 9 bytes in Fig. 4.1(a) and 15 bytes in Fig. 4.1(b) for the ranges of W_T where both heuristics reach a feasible solution.

Fig. 4.2 shows the performance of the algorithms in finding feasible solutions with respect to the values of W_T for two traffic loads with 30 and 60 demands. According to Fig. 4.2(a). FGH can find feasible solutions in all 50 instances for values of W_T greater than 59. However, SGH outperforms FGH and can find feasible solutions for W_T ranging from 54 to 120. From link sparing point of view, the results denote that FGH and SGH can guarantee reducing link usage in routing up to 50% and 55%, respectively, in the case of the traffic load with 30 demands. However, ILP is able to find feasible routing for the values of W_T equal to or greater than 26; hence, it guarantees 78% reduction in link usage. In the case of the traffic load with 60 demands, Fig. 4.2(b) shows that the guaranteed link sparing drops to 45%, 49%, and 75% for FGH, SGH, and ILP, respectively.

4.1.3 Impact of network topology density

In the second set of simulations, we investigate the impact of network density on the amount of control overhead with respect to a fixed value of W_T. The considered network instances for this experiment have 30 nodes and their node degrees range from 3 to 7. We have used the traffic pattern with 30 demands in this experiment. We set the value of W_T to be 60, at which point all three methods yield feasible solutions for all values of node degrees in all network instances. Fig. 4.3 shows the amount of control messages with respect to different node degree values. The value of each point in the figure is the average of 50 different network instances.

The results show that there is a considerable difference between the performance of our heuristics and that of our ILP when node degree is 3. For this value of node degree, both FGH and SGH generate nearly 200 bytes more than the optimal amount achieved by ILP. The behavior, however, improves as the node degree increases. As

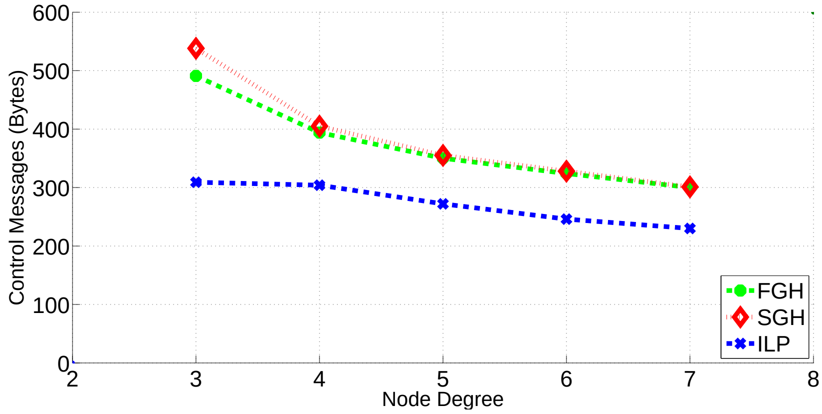


Figure 4.3: Amount of control messages with respect to different node degrees for $W_T = 60$

the density of the instances increases, the performance of the heuristics approaches the one of ILP. The difference in the amount of overhead generated by the heuristics and the ILP method reaches only 50 bytes when node degree is 7. Although FGH outperforms SGH when node degree is 3, both heuristics show a similar performance as node degree increases.

4.2 Numerical Evaluation of Experiments for E-SOAGR Problem

In this section, we present the numerical results of our proposed ILP formulation for E-SOAGR and our proposed heuristic E-SGH on two different network instances. In our experiments, we investigate the impact of energy threshold of the network on the amount of control overhead and the impact of traffic load and multipath routing on the blocking rate of demands.

4.2.1 Simulation setup of experiments for E-SOAGR problem

In order to evaluate E-SOAGR problem, we perform a number of experiments on realistic topologies. For this purpose, we select two referenced topologies from SNDlib [31], which is a library of test instances for survivable fixed telecommunication network design. Our selected topologies are *GEANT* and *JANOS-US*. We explain these topologies and our assumptions on the capacity and energy consumption of the network links in the following.

For the experiments of SOAGR problem, we assumed that all the network links have the same value of energy consumption. However, we consider in this section a more precise energy modeling for the network links in the set of experiments for E-SOAGR. In the considered energy model, we assume that the energy consumption of each link (W_{ij}) is proportional to its capacity (C_{ij}) and length (d_{ij}). We calculate the length of links according to the values provided in SNDlib. We refer to the minimum values of link capacity and length as (C_{ij}^{min}) and (d_{ij}^{min}). We then consider the value $d_{ij}^{min} \times C_{ij}^{min}$ as one unit of energy and denote this unit by dc. We normalize the energy consumption of the links according to this value.

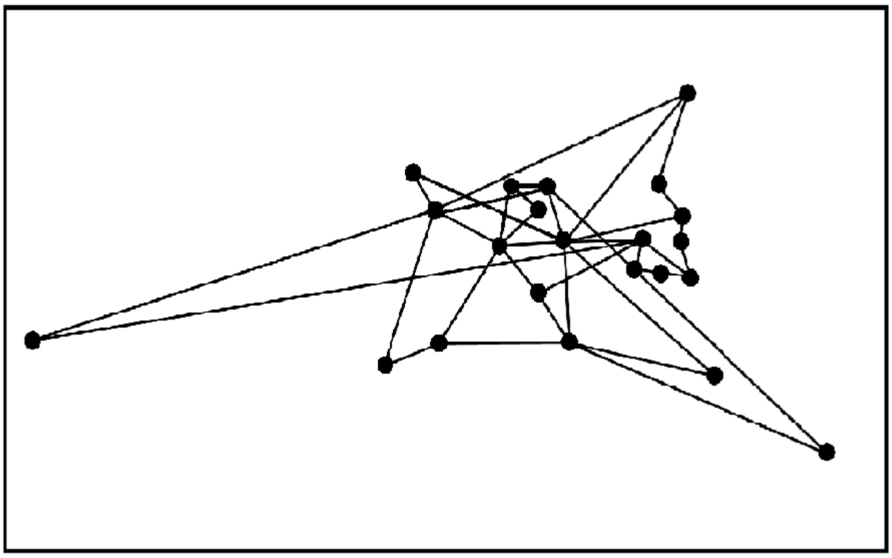
We consider a sleep message to have a size of one byte. We also consider three bytes for a routing update message. In all experiments, the value of the parameter of Yen's K-shortest path algorithm (k) is set to 15 and the step value for varying threshold in E-SGH (δ) is set to be $0.1 \times W_T$. We set the value of upper limit for the size of routing table (R) to 25.

For the traffic scenario, we consider an upper bound and a lower bound for the total number of demands at each configuration. At first configuration, the number of demands is equal to the average of the upper and lower bounds. The sources and destinations of these demands are selected randomly. For the next configuration, we randomly select a number of demands in the current configuration to be terminated and a number of demands (not present in the current configuration) to arrive. However, we control the number of arriving and terminated demands at each configuration such that the total number of demands in each configuration remains in a range between the upper and the lower bounds.

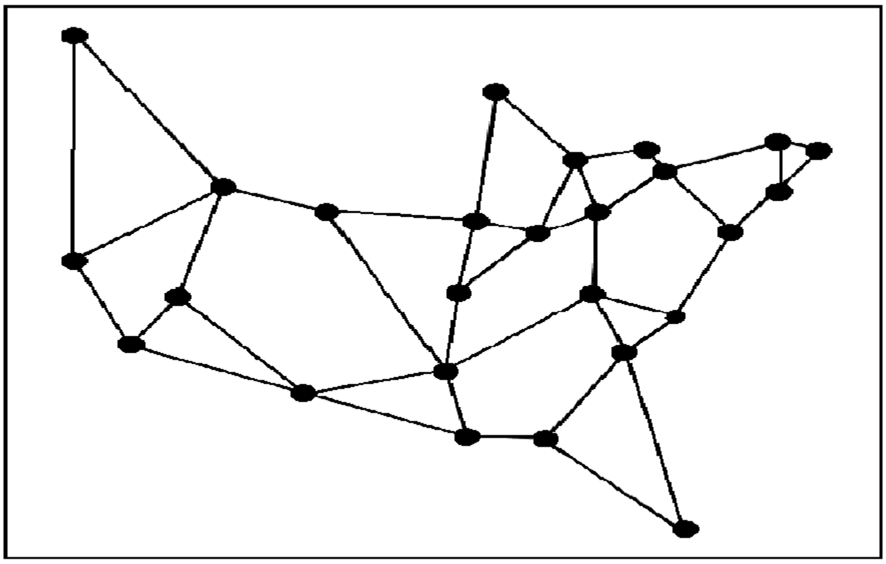
In the set of experiments for E-SOAGR problem, we consider two traffic loads referred to as T_1 and T_2. The number of demands at each configuration of T_1 varies between 15 and 25, while the number of demands at each configuration of T_2 is between 40 and 50. The total number of demands in both traffic loads is 5000. In addition, we randomly select values from a range between 100 Mbps and 1 Gbps for the bandwidth requirements of the demands.

4.2.1.1 Topology 1: GEANT

GEANT is a network with 22 nodes, 72 directional links and average node degree of 3.27. GEANT topology is depicted in Fig. 4.4(a). We set the values of link capacities



(a) GEANT topology [31] with 22 nodes, 72 links and average node degree of 3.27



(b) JANOS-US topology [31] with 26 nodes, 84 links and average node degree of 3.23

Figure 4.4: Real topologies used in experiments for E-SOAGR

of GEANT according to the values in [32] for our experiments. These values include the rates 10, 20, 30 and 100 Gbps.

4.2.1.2 Topology 2: JANOS-US

JANOS-US is a network with 26 nodes, 84 directional links and average node degree of 3.24. JANOS-US topology is depicted in Fig. 4.4(b). For the values of link capacity in JANOS-US network, we generate random rates among 10, 20, 30 and 100 Gbps.

4.2.2 Impact of network energy threshold on the signaling overhead

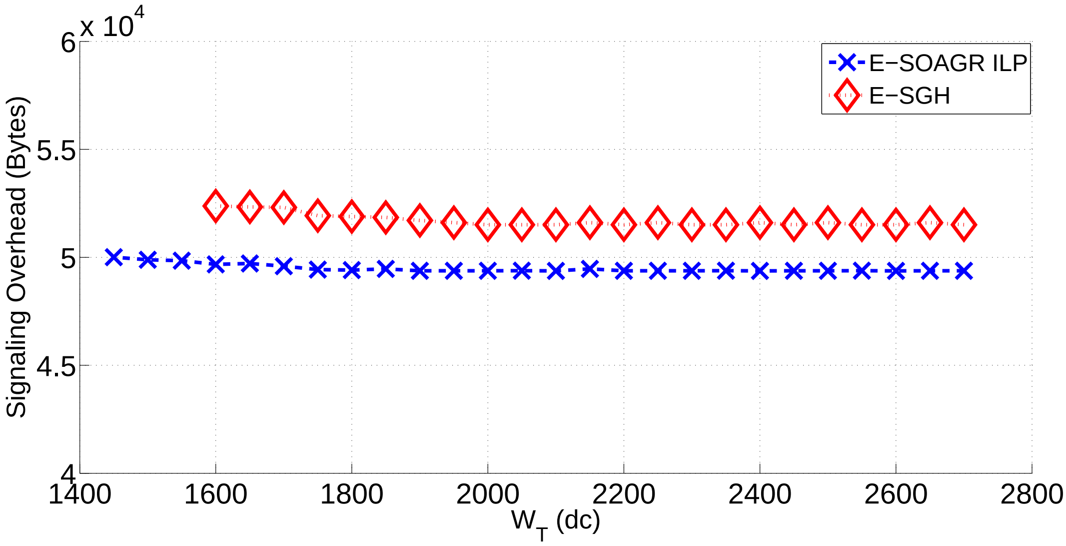
We first investigate the impact of W_T on the amount of signaling overhead considering the traffic load T_1. Fig. 4.5 shows the results of this experiment. For this experiment we set the value of the maximum paths in multipath routing (M) to 3. We have started the experiment by setting W_T to 2720 dc for GEANT and 1200 dc for JANOS-US. These values of W_T denote the situation where the algorithms are allowed to use all the links in routing. We have then repeated the experiment for the smaller values of W_T.

We observe that E-SOAGR ILP and E-SGH have a similar behavior when W_T decreases in both Fig. 4.5(a) and Fig. 4.5(a). In both topologies, the solutions found by E-SGH are close to those of ILP. E-SGH generates nearly 4000 bytes more than that of E-SOAGR ILP, namely 8% more signaling overhead, on GEANT topology. In the case of JANOS-US topology, the difference between E-SGH and E-SOAGR is about 5000 bytes, which is equal to 7.8% more signaling overhead. In both topologies, the number of control messages increases as W_T decreases for both algorithms. This behavior shows the trade-off between energy efficiency and signaling overhead.

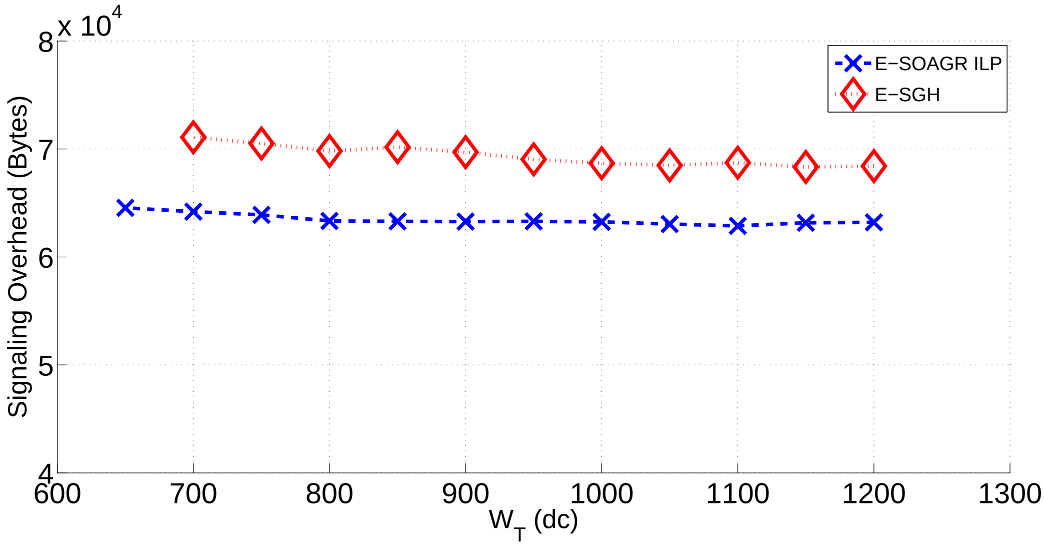
4.2.3 Impact of traffic load on the blocking rate of demands

In this experiment, we investigate the impact of traffic load on the blocking rate. We consider the ratio of the number of blocked demands to the total number of arriving demands as blocking rate in our experiment. Fig. 4.6 shows the performance of E-SGH from the perspective of blocking rate with respect to W_T considering two traffic loads $T1$ and $T2$.

In the case of GEANT topology (Fig. 4.6(a)), E-SGH can accommodate all arriving demands for the values of W_T ranging from 1500 to 2700 considering the traffic load

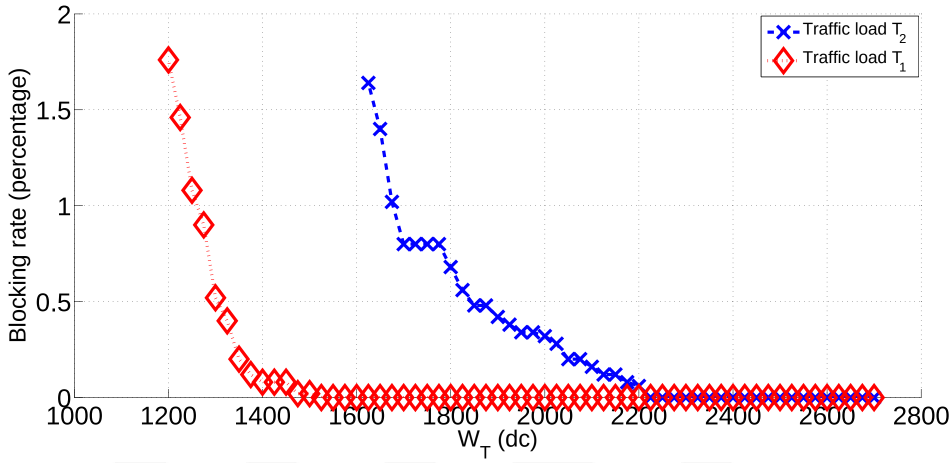


(a) Signaling overhead with respect to W_T for GEANT topology .

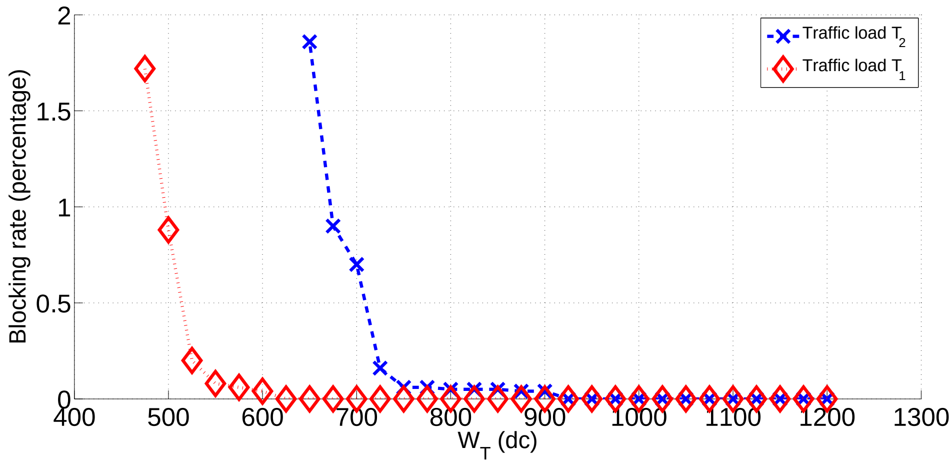


(b) Signaling overhead with respect to W_T for JANOS-US topology.

Figure 4.5: Impact of W_T on the amount of signaling overhead.



(a) Blocking rate with respect to W_T for GEANT topology for traffic loads $T1$ and $T2$.



(b) Blocking rate with respect to W_T for JANOS-US topology for traffic loads $T1$ and $T2$.

Figure 4.6: Impact of traffic load on the amount of blocking rate.

$T1$. From the perspective of energy conservation, the results show that E-SGH can guarantee 44% reduction in total energy consumption of the network. The reduction in energy consumption can even reach 55% when the blocking rate is 2%. However, it is clear that these numbers decrease as traffic load increases. In the case of the traffic load T_2, the energy conservation reaches 18% (W_T= 2200) when the blocking rate is zero; while it reaches 40% when the blocking rate is 2%.

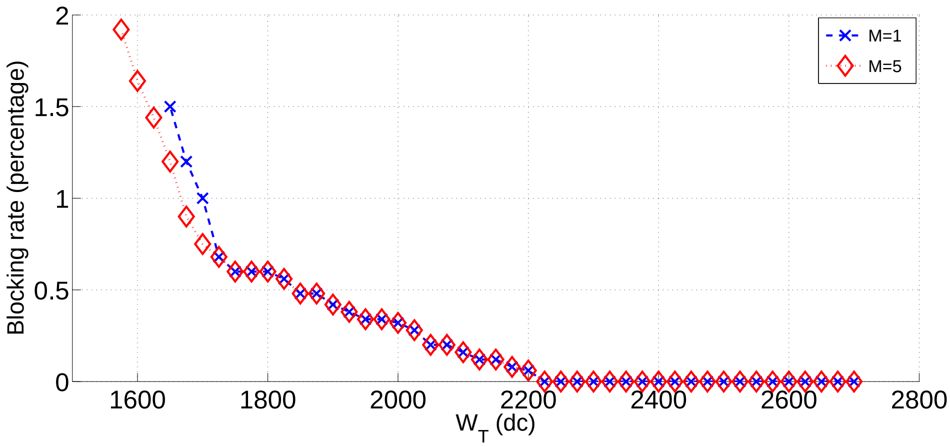
Fig. 4.6(b) denotes the result of this experiment for JANOS-Us topology. E-SGH accommodates all arriving demands for W_T ranging from 625 to 1200 in the case of $T1$ and for W_T ranging from 900 to 1200 in the case of $T2$. These results show that E-SGH can guarantee reduction in energy consumption of routing up to 50% and 25%, in the case of the traffic loads $T1$ and $T2$ when the blocking rate is zero. However, E-SGH is able to reach 60% and 45% reduction in energy consumption for the traffic loads $T1$ and $T2$ when the blocking rate is 2%.

4.2.4 Impact of multipath routing on the blocking rate of demands

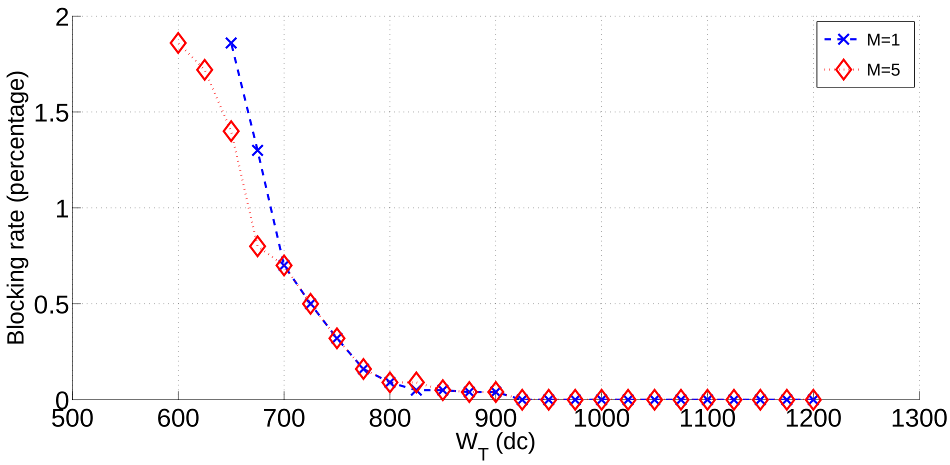
In this experiment, we investigate the impact of multipath routing on the amount of blocking rate. Fig. 4.7 shows the results of this experiment in which we use traffic load T_2 and perform E-SGH for two values: M=1 (single path accommodation) and M=5 (Multipath accommodation).

Fig. 4.7(a) denotes the result of this experiment for GEANT topology. In the case of GEANT, E-SGH with M=1 and M=5 follow the same behavior for the values of W_T ranging from 1750 to 2700. We observe the impact of multipath routing as we reduce the value of W_T below 1750. In this range, multipath E-SGH has a smaller blocking rate compared to single path E-SGH. In addition, for the values of W_t ranging from 1575 to 1625, E-SGH with M=5 generates solutions with blocking rates below 2% while E-SGH with M=1 fails in finding any feasible solutions.

Fig. 4.6(b) denotes the results for JANOS-US topology. The results obtained in the case of JANOS-Us are largely similar to those obtained for GEANT. Single path and multipath E-SGH follow the same behavior for the values of W_T ranging from 700 to 1200; whereas the impact of multipath becomes evident for the values of W_T below 700. In this range, multipath E-SGH succeeds in finding solutions with smaller blocking rates compared to single path E-SGH. Furthermore, multiple path E-SGH



(a) Blocking rate with respect to W_T for GEANT topology .



(b) Blocking rate with respect to W_T for JANOS-US topology.

Figure 4.7: Impact of multipath routing on the blocking rate of demands in E-SOAGR.

finds solutions with blocking rates below 2% for the two values: W_T=625 and W_T=600 while single path E-SGH fails to find any feasible solution.

5. CONCLUSIONS AND FUTURE WORK

Energy conservation is increasingly receiving attention in wired networks. Researchers have proposed many techniques to bring energy-awareness into wired networks. These techniques are generally called green techniques. However, the existing green techniques focus only on achieving energy reduction and ignore the signaling overhead generated by these techniques. To the best of our knowledge, this thesis is the first work in the literature that tackles the tradeoff between energy efficiency and signaling overhead in a software defined network (SDN).

In this thesis, we have formulated two green routing problems which minimize signaling overhead while taking the network energy consumption into account in an SDN network with a single domain. Our proposed ILP formulations are Signaling Overhead-Aware Green Routing (SOAGR) and Extended Signaling Overhead-Aware Green Routing (E-SOAGR). E-SOAGR, which is an extension of SOAGR, offers a more general model with additional features such as multipath routing, taking into account routing table sizes of the network nodes and considering different bandwidth requirements for traffic demands.

We have implemented the two ILP formulations in the IBM ILOG CPLEX Optimization Studio. In addition, we have developed polynomial-time heuristic algorithms to find near-optimal solutions for both ILP problems. We have conducted experiments on both random and realistic topologies to evaluate the performance of our proposed ILP formulations and heuristics. The obtained results demonstrate the tradeoff between energy efficiency and signaling overhead.

Possible future work can be in two directions. One is to formulate different ILP problems with additional constraints by taking into consideration the issues of security, survivability, and energy fairness as well as developing heuristic algorithms which produce solutions much closer to those of ILP. The other direction is to design approximation algorithms, which have theoretically provable performance guarantee, to address the signaling overhead in green routing.

REFERENCES

[1] **Nunes, B.A.A.**, **Mendonca, M.**, **Nguyen, X.N.**, **Obraczka, K. and Turletti, T.** (2014). A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks, *Communications Surveys Tutorials, IEEE*, ***16***(3), 1617–1634.

[2] **Gupta, M. and Singh, S.** (2003). Greening of the internet, *ACM SIGCOMM*, 19–26.

[3] **Fettweis, G. and Zimmermann, E.** (2008). ICT energy consumption-trends and challenges, *Proceedings of the 11th International Symposium on Wireless Personal Multimedia Communications*, volume 2, p. 6.

[4] **Wang, X.**, **Vasilakos, A.**, **Chen, M.**, **Liu, Y. and Kwon, T.** (2012). A survey of green mobile networks: opportunities and challenges, *Mobile Networks and Applications*, ***17***(1), 4–20.

[5] **Mahadevan, P.**, **Sharma, P.**, **Banerjee, S. and Ranganathan, P.** (2009). A Power Benchmarking Framework for Network Devices, *IFIP Networking*, pp.795–808.

[6] **Bianzino, A.**, **Chaudet, C.**, **Larroca, F.**, **Rossi, D. and Rougier, J.** (2010). Energy-aware routing: A reality check, *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pp.1422–1427.

[7] **Fisher, W.**, **Suchara, M. and Rexford, J.** (2010). Greening backbone networks: reducing energy consumption by shutting off cables in bundled links, *ACMSIGCOMM Workshop on Green networking*, pp.29–34.

[8] **Mahadevan, P.**, **Sharma, P.**, **Banerjee, S. and Ranganathan, P.** (2009). Energy Aware Network Operations, *INFOCOM Workshops, IEEE*.

[9] **Bianzino, A.**, **Chaudet, C.**, **Rossi, D. and Rougier, J.** (2012). A Survey of Green Networking Research, *Communications Surveys Tutorials, IEEE*, ***14***(1), 3–20.

[10] **Chabarek, J.**, **Sommers, J.**, **Barford, P.**, **Estan, C.**, **Tsiang, D. and Wright, S.** (2008). Power Awareness in Network Design and Routing, *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp.457–465.

[11] **Giroire, F.**, **Mazauric, D.**, **Moulierac, J. and Onfroy, B.** (2010). Minimizing Routing Energy Consumption: From Theoretical to Practical Results, *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pp.252–259.

[12] **Chiaraviglio, L. and Matta, I.** (2010). GreenCoop: Cooperative Green Routing with Energy-efficient Servers, pp.191–194.

[13] **Chiaraviglio, L.**, **Mellia, M. and Neri, F.** (2009). Reducing Power Consumption in Backbone Networks, *Communications, 2009. ICC '09. IEEE International Conference on*, pp.1–6.

[14] **Shang, Y.**, **Li, D. and Xu, M.** (2010). Energy-aware Routing in Data Center Network, *Proceedings of the First ACM SIGCOMM Workshop on Green Networking*, Green Networking '10, pp.1–8.

[15] **Heller, B.**, **Seetharaman, S.**, **Mahadevan, P.**, **Yiakoumis, Y.**, **Sharma, P.**, **Banerjee, S. and McKeown, N.** (2010). ElasticTree: Saving Energy in Data Center Networks, *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pp.17–17.

[16] **Cianfrani, A.**, **Eramo, V.**, **Listanti, M.**, **Marazza, M. and Vittorini, E.** (2010). An Energy Saving Routing Algorithm for a Green OSPF Protocol, *INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, pp.1–5.

[17] **Araujo, J.**, **Giroire, F.**, **Liu, Y.**, **Modrzejewski, R. and Moulierac, J.** (2013). Energy efficient content distribution, *Communications (ICC), 2013 IEEE International Conference on*, pp.4233–4238.

[18] **Addis, B.**, **Capone, A.**, **Carello, G.**, **Gianoli, L. and Sanso, B.** (2014). Energy Management Through Optimized Routing and Device Powering for Greener Communication Networks, *IEEE/ACM Transactions on Networking*, ***22***(1), 313–325.

[19] **Bianzino, A.P.**, **Chiaraviglio, L.**, **Mellia, M. and Rougier, J.L.** (2012). GRiDA: GReen Distributed Algorithm for Energy-efficient IP Backbone Networks, *Comput. Netw.*, ***56***(14), 3219–3232.

[20] **Lee, S.S.W.**, **Tseng, P.K. and Chen, A.** (2012). Link Weight Assignment and Loop-free Routing Table Update for Link State Routing Protocols in Energy-aware Internet, *Future Gener. Comput. Syst.*, ***28***(2), 437–445.

[21] **Bakshi, K.** (2013). Considerations for Software Defined Networking (SDN): Approaches and use cases, *Aerospace Conference, 2013 IEEE*, pp.1–9.

[22] **Li, D.**, **Shang, Y. and Chen, C.** (2014). Software defined green data center network with exclusive routing, *INFOCOM, 2014 Proceedings IEEE*, pp.1743–1751.

[23] **Celenlioglu, M.**, **Goger, S. and Mantar, H.** (2014). An SDN-based energy-aware routing model for intra-domain networks, *Software, Telecommunications and Computer Networks (SoftCOM), 2014 22nd International Conference on*, pp.61–66.

[24] **Wang, R.**, **Jiang, Z.**, **Gao, S.**, **Yang, W.**, **Xia, Y. and Zhu, M.** (2014). Energy-aware routing algorithms in Software-Defined Networks, *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on*, pp.1–6.

[25] **Giroire, F.**, **Moulierac, J. and Phan, T.K.** (2014). Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing, **Research Report RR-8537**, https://hal.inria.fr/hal-00996780.

[26] **Bari, M.**, **Roy, A.**, **Chowdhury, S.**, **Zhang, Q.**, **Zhani, M.**, **Ahmed, R. and Boutaba, R.** (2013). Dynamic Controller Provisioning in Software Defined Networks, ***18***, 18–25.

[27] **ILOG, Inc**, (2006), ILOG CPLEX: High-performance software for mathematical programming and optimization, see http://www.ilog.com/products/cplex/.

[28] **Yen, J.** (1971). Finding the k shortest loopless paths in a network, *management Science*, 712–716.

[29] **Dijkstra, E.W.** (1959). A Note on Two Problems in Connexion with Graphs, *NUMERISCHE MATHEMATIK*, ***1***(1), 269–271.

[30] **de Queirós Vieira Martins, E. and Pascoal, M.M.B.** (2003). A new implementation of Yen's ranking loopless paths algorithm., *4OR*, ***1***(2), 121–133.

[31] **Orlowski, S.**, **Pióro, M.**, **Tomaszewski, A. and Wessäly, R.** (2007). SNDlib 1.0–Survivable Network Design Library, *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium*, http://www.zib.de/orlowski/Paper/OrlowskiPioroTomaszewskiWessaely2007-SNDlib-INOC.pdf.gz, http://sndlib.zib.de, extended version accepted in Networks, 2009.

[32] GEANT connectivity map 2014, see http://www.geant.net/Resources/Media_Library/Documents/GEANT_Project_TopologyMAR14_Web.pdf.

CURRICULUM VITAE

Name Surname: Hadi ALIZADEH

Place and Date of Birth: Marand, Iran, 01.08.1987

Adress: Department of Computer Engineering, Istanbul Technical University, 34469 Maslak, Istanbul, Turkey

E-Mail: alizadehhadi@itu.edu.tr

B.Sc.: Computer Engineering, K.N.T University of Technology, Tehran

M.Sc.: Computer Engineering, Istanbul Technical University

PUBLICATIONS/PRESENTATIONS ON THE THESIS

- Alizadeh H., Gözüpek D., Buhari Seyed M., Yayımlı A., 2015: Minimizing Signaling Cost in Green Routing for Software Defined Networks. *accepted to IEEE-ISCC 2015*, July 06-09, 2015 Larnaca, Cyprus.