2015

Green traffic engineering techniques for current and next generation networks

Alejandro Ruiz Rivera
University of Wollongong

Green Traffic Engineering Techniques for Current and Next Generation Networks

A thesis submitted in partial fulfilment of the requirements for the award of the degree

Doctor of Philosophy

from

UNIVERSITY OF WOLLONGONG

by

Alejandro Ruiz Rivera

Bachelor of Electronics Engineering (Telecommunications)

Master of Engineering (Telecommunications)

School of Electrical, Computer and Telecommunications Engineering

July 2015

Statement of Originality

I, Alejandro Ruiz Rivera, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institutions.

Signed

Alejandro Ruiz Rivera

July 1, 2015

Abstract

The Information and Communication Technology (ICT) industry is responsible for a non-negligible proportion of the world's global energy consumption. Unfortunately, the energy usage of the Internet will continue to rise due to the rapid growth in the number of connected devices and bandwidth intensive applications. In addition, networks are nowadays over-provisioned to provide redundancy and to preserve Quality of Service (QoS) during peak periods. This means current networks are designed to handle the "worst-case" scenario in terms of failures and traffic demands. In other words, they are not designed to be energy efficient. However, many works have shown that traffic exhibits diurnal patterns that correspond to business hours and weekends. This observation and the dire need to conserve energy have spurred intense research efforts into green approaches that consolidate traffic onto the minimal number of links/switches/routers during off-peak periods. In particular, researchers have designed many green or energy-aware traffic engineering (TE) techniques that jointly optimize energy savings and QoS constraints such as maximum link utilization, and end-to-end delays.

This thesis contains a number of novel green TE techniques. First, it studies green TE techniques in Multi-Protocol Label Switching (MPLS) networks. Approaches in this area aim to establish as many arriving Label Switched Path (LSP) requests as possible while utilizing the minimum number of links/routers. However,

no one has quantified these LSP establishment approaches in terms of the number of accepted LSP requests and the resulting energy savings. Therefore, this thesis studies six heuristics and proposes a novel metric that considers both energy savings and acceptance rates. In addition, it proposes a simple heuristic that selects paths that contain already established links and use the fewest number of new links.

Second, this thesis studies the controller placement problem in software defined networks (SDNs). Unlike past works, the aim is energy efficiency. Specifically, current solutions have only considered the placement of controllers and Switch-to-Controller (S2C) association such that the delay between a switch and its controller is within a given bound and controllers have a similar load. However, existing works have not jointly optimized S2C association and energy consumption. This thesis fills this gap and presents the first green TE method for SDNs. In particular, this thesis contains an algorithm that selects S2C paths that result in the lowest latency whilst using only the minimal number of links. Links that do not belong to any selected paths are then powered off. The proposed solution also guarantees that controllers have a similar number of associated switches.

Thirdly, this thesis studies the impact of green TE methods on the Border Gateway Protocol (BGP) and proposes two novel BGP-aware TE techniques. This is a significant investigation because BGP is the “glue” used by Autonomous Systems (ASes) to advertise reachability information. Past works have shown that hot-potato routing changes have many negative effects on BGP. Hence, it is important that existing green TE methods do not trigger these changes. To this end, this thesis presents the first study that quantifies the effects that green TE approaches have on the operation of BGP. Experimental results show that green TE techniques cause an increase in the percentage of hot-potato routing changes and the proportion of rerouted traffic. Motivated by these results, this thesis proposes two novel BGP-aware approaches that can be used to reduce energy consumption while minimizing any impacts on the operation of BGP.

Lastly, this thesis considers random traffic demands when shutting down links. This fills a key gap in the literature whereby past green TE approaches assume traffic demands are deterministic. Furthermore, in order to ensure robustness, they allocate resources according to peak demands, meaning a significant number of network elements may be idle during off-peak periods. Henceforth, this thesis presents the first green TE technique that considers traffic demands characterized by a polyhedral set. This set allows an operator to describe all possible demands between ingress and egress routers via a set of inequalities. A solution called Green-PolyH is then proposed to switch off as many links as possible whilst ensuring the residual network, where some links are switched off, remains robust to all traffic variations defined by the given polyhedral set. In other words, Green-PolyH guarantees that the maximum utilization of all links is bounded for all demands in the given polyhedral set.

Acknowledgments

I would like to thank my supervisor Associate Professor Kwan-Wu Chin. His incredible support, patience, prompt suggestions and feedback, and constant encouragement have made it possible for me to walk and finish this extremely difficult, challenging but enlightening PhD degree.

Special thanks to my co-supervisors, Dr Raad Raad and Dr Sieteng Soh, for their constructive suggestions and comments throughout my PhD studies.

Thanks to all the anonymous reviewers of conferences and journals for their time, efforts and feedback. Their suggestions have certainly raised the quality of my publications.

Sincere gratitude to the company that I work for, Catholic Education network (CEnet), and its staff. Notable colleagues include Ian and Chris; my former and current manager, respectively, who gave me the opportunity to start and finish this incredible endeavor. To my superiors, Greg (CEO) and Vicki (CIO), and co-workers, who have been there for me in times of difficulty, listening patiently and caring about my research.

Special thanks to all my friends in Australia, who have patiently listened to me for more times I can count about my research and difficulties. Their support and advise were incredibly valuable to me. In particular, Philip and Beatriz for their valuable support and encouraging advise.

All the thanks go to my parents, Alejandro and Bertha, for their incredible moral support and constant encouragement. They have made me the person I am today. I could not have achieved a single thing in my life without their incredible advise, dedication and sound principles they have infused in me.

Thanks to my brother, Julian, who always believes in my abilities, giving me the motivation to start and finish a number of projects. His wise advise and support have been critical during my PhD.

Heartfelt thanks to my beautiful and lovely wife, Claudia, for her support and patience during my studies. She was always there for me, listening, and providing valuable advise. For all of this I am indebted to her for the rest of my life.

Finally, to Jesus Christ my lord and saviour. He is the one who works in me to will and to act in order to fulfil his good purpose. He has always been, is, and will be, the constant friend who is always there, and the source of my perseverance and tenacity.

Agradecimientos

Me gustaría agradecer a mi supervisor Profesor Asociado Kwan-Wu Chin. Su increíble apoyo, paciencia, y sugerencias han hecho posible para mí el caminar y finalizar este camino tan desafiante y difícil.

Agradecimientos especiales a mis Co-Supervisores, Dr Raad Raad y Dr. Sieteng Soh, por sus constructivas sugerencias y comentarios a lo largo de todo mi PhD.

Gracias a todos los revisores anónimos en las conferencias y journals por su tiempo y esfuerzo invertido. Sus valiosas sugerencias han definitivamente mejorado la calidad de mis publicaciones.

Gracias a Catholic Education network (CEnet), la compañia para la que trabajo, y a todos sus empleados. En especial, Ian, mi supervisor cuando ingresé a CEnet y a Chris, mi supervisor actual, quienes me brindaron la oportunidad de empezar y terminar esta increíble labor. Gracias a mis jefes, Greg (CEO) y Vicki (CIO), y a todos mis colegas, quienes han estado ahí para mí cuando los he necesitado. Todos ellos me han escuchado pacientemente cuando les he compartido acerca de mi investigación y dificultades.

Gracias a todos mis amigos en Australia, quienes pacientemente me han escuchado durante los momentos más difíciles. Su apoyo me ha servido demasiado durante todos estos años. Particularmente Philip y Beatriz for sus valiosisimos consejos.

Muchas gracias a mis padres, Alejandro y Bertha, por su increíble apoyo moral y ánimo. Ellos han hecho de mi la persona que soy hoy en día. Sin sus consejos y enseñanzas no hubiera podido lograr nada de lo que he alcanzado en mi vida.

Muchas gracias a mi hermano Julián, quien siempre ha creído en mí, dándome la motivación para empezar y terminar muchos de mis proyectos. Sus consejos y apoyo han sido determinantes durante todo mi PhD.

Gracias desde el fondo de mi corazón a mi esposa Claudia, por todo su apoyo y paciencia durante mis estudios. Ella siempre ha estado ahí para mí, escuchándome y dándome consejos muy valiosos. Por todas estas cosas, estoy eternamente en deuda con ella.

Finalmente, gracias a Jesucristo, mi señor y salvador. Él es el quien ha puesto en mí el querer como el hacer para su honra y gloria. Él es el amigo que siempre ha estado, esta y estará constantemente a mi lado. Él es la fuente de mi perseverancia y tenacidad.

Contents

List of Figures

List of Tables

Abbreviations

AS	Autonomous System
ALU	Average Link Utilization
BGP	Border Gateway Protocol
BGP-LWO	BGP-aware Link Weight Optimizer
C2C	Controller-to-Controller
CSPF	Constraint Shortest Path First
CPU	Central Processing Unit
DFS	Depth-First Search
eLSR	Egress label Switch Router
ER	Egress Router
ECMP	Equal-Cost Multipath
FRR	Fast Reroute
FEC	Forwarding Equivalence Class
HotPLUZ	Hot Potato Low Utilization
HotPLEC	Hot-potato Low Energy Consumption
i-BGP	Internal BGP
IGP	Interior Gateway Protocol
IGP-WO	IGP Weight Optimizer
iLSR	Ingress Label Switch Router

ILP	Integer Linear Program
IoT	Internet of Things
IR	Ingress Router
ISU	IR Saving Unit
LAR	LSP Acceptance Requests
LSR	Label Switch Router
LSP	Label Switched Path
LSA	Link State Advertisement
LWO	Link Weight Optimizers
M2M	Machine-to-Machine
MALU	Maximum Average Link Utilization
MCMF	Multi-Commodity Minimum-cost Flow
MCF	Multiple Commodity Flow
MCSPF	Multiple Constraint-based Shortest Path First
MIP	Mixed Integer Programming
MLU	Maximum Link Utilization
MPLS	Multi-Protocol Label Switching
PSL	Percentage of Shut down Links
QoS	Quality of Service
ROD	Routing On Demand
S2C	Switch-to-Controller
S2S	Switch-to-Switch
SDN	Software Defined Network
SLA	Service Level Agreement
SPT	Shorter Path Tree
TE	Traffic Engineering
TE-LSA	Traffic Engineering Link State Advertisements
TM	Traffic Matrix

USA	United States of America
VPN	Virtual Private Network

Chapter 1

Introduction

The *SMART 2020* report by the Climate Group Organization [1] indicates that the Information Communications and Technology (ICT) industry is beginning to have a non-negligible impact on the environment. Its high energy usage has contributed up to 2% of global carbon gas emissions. This situation is expected to get worse as the traffic volume and number of devices are predicted to grow significantly due to technologies such as the Internet Protocol TV (IPTV) [2], Voice over IP (VoIP) [3], cloud computing [4] and more recently Internet of Things (IoT) [5]. Indeed, the global IP traffic is estimated to have a compound growth rate of 21% from 2013 to 2018 [6], and will surpass the zettabyte (1000 exabytes) threshold in 2016. It is also estimated that there will be 11.5 billion mobile-connected devices by 2019, including Machine-to-Machine (M2M) modules. Given that current ICT infrastructure is designed to handle the "worst-case" scenario in terms of failures and traffic demands [7], analysis such as [8] forecasts that the global carbon footprint of telecommunication network devices will grow by 5% each year between 2002 and 2020 due to the steady rise in electricity demand. The authors of [9] indicate that the United States of America (USA) alone uses 24TWh per year, costing around $24 billion annually.

The aforementioned statistics have thus motivated a flurry of "green" approaches. This started with the seminal work of Gupta et al. in [10], where the authors motivate research into methods that reduce the energy consumption rate of routers or switches. Since 2003, Gupta et al.'s work has spawned a number of research areas. Examples include (i) *sleeping* [11] [7], which aims to place sub-components of devices or devices themselves to sleep, (ii) *link adaptation* [12] [13], which scales the energy consumption according to varying link utilization, (iii) *proxying* [14] [9], which reduces network chatters by way of a proxy, and (iv) *traffic engineering (TE)* [15] [16] [17][18][19], whereby traffic is routed across the minimal number of links and routers, or spread across as many links as possible to minimize link load; this is assuming the energy consumption rate is proportional to utilization [7].

This thesis focuses on state-of-the-art green TE techniques that are designed for wired networks. Briefly, the main aim of TE is to optimize network performance and traffic delivery. For example, the routing of commodities or flows can be optimized to improve network capacity and satisfy QoS requirements [20]. In the case of green TE, the focus of past works is on energy efficient label switched path (LSP) establishment methods [21] [22] [23] [24] [25], and modifications to Interior Gateway Protocols (IGP), in order to route traffic demands over the minimum set of network devices [17] [26] [27] [28] [29]; see Chapter 2 for more details. Similarly, this thesis also aims to reduce the number of active links/routers/switches. Specifically, it covers four areas: (i) green LSPs establishment methods in Multi-Protocol Label Switching (MPLS) networks [30], (ii) reconfiguring the association of switches and controllers in green Software Defined Networks (SDNs) [31], (iii) minimizing the impact on the Border Gateway Protocol (BGP) [32] when switching off links/routers/switches, and (iv) robust TE, where the goal is to switch off as many network elements as possible subject to the constraint that the remaining active links must have sufficient capacity to handle varying traffic demands as defined by a polyhedral set [33]. In the sequel, these areas/topics will be elaborated further using specific examples.

Green MPLS Networks

MPLS is a popular TE tool used by network operators. Hence, it is critical that we develop energy-aware LSP establishment methods that are applicable in future green networks. To illustrate the problem at hand, consider Figure 1.1. There are seven Label Switch Routers (LSRs) interconnected by 16 directional links. Given a set of LSP requests, each with a source (s) and destination (d) address, and bandwidth demand (bw), the goal is to establish these LSPs in a manner that reduces the network's overall energy consumption.

Assume links have a capacity of 100 Mb/s and currently have zero utilization. Consider two LSP requests, each denoted as $<s, d, bw>$ arriving at *R1*, *R2*, and *R5* in the following order: LSP_1 $<R1, R2, 20>$ LSP_2 $<R1, R4, 60>$. When LSP_1 arrives, a green TE solution may first assign it to link/path $R1 - R2$ given that this is the shortest possible path and the link can accommodate the requested demand. After establishing LSP_1, the utilization of link $R1 - R2$ is 20%. After that, LSP_2 arrives. In this case, there are multiple paths to choose from; e.g., $[R1, R2, R4]$, and $[R1, R3, R4]$. The solution then proceeds to check whether any of these paths are able to accommodate the 60 Mb/s demand requested by LSP_2. If path $[R1, R2, R4]$ is selected, the final utilization of links $R1 - R2$ and $R2 - R4$ will be 80% and 60% respectively. If the path $[R1, R3, R4]$ is selected, the utilization of links $R1 - R3$ and $R3 - R4$ will be 60%. Therefore, both paths will be able to accommodate LSP_2. If path $[R1, R3, R4]$ is selected, routers *R1*, *R2*, *R3*, and *R4* need to be active, and will need a total of three active links, i.e., $R1 - R2$, $R1 - R3$, and $R3 - R4$, to serve LSP_1 and LSP_2's requests. However, if path $[R1, R2, R4]$ is selected, link $R1 - R2$ can be reused given that it is currently used to serve LSP_1. This means both LSP requests will only be served by a total of three active routers, i.e., *R1*, *R2* and *R4*, and use two active links; i.e., $R1 - R2$, and $R2 - R4$. Selection of path $[R1, R2, R4]$ for LSP_2 will therefore accommodate the requested demand while minimizing network resource usage.

In summary, given a set of LSPs and their respective bandwidth requirement, the aim is to determine the minimum number of links that can be used to support all LSPs. It is worth pointing out that research into LSPs establishment is divided into two areas: (i) *offline* case, meaning that the complete set of LSP requests are known in advance, and (ii) *online* approaches,where only the current and past LPS requests are known. Consequently, selecting a particular path may be sub-optimal given that future LSPs are unknown. Hence, an effective online policy that yields the same result as the offline case is highly desirable.

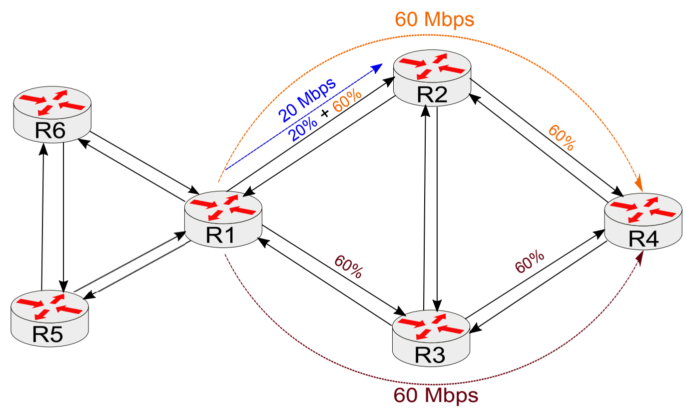
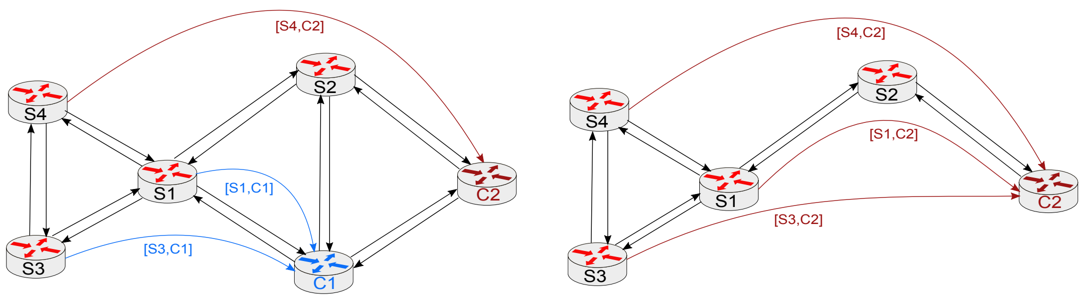


Figure 1.1: Illustration of a green LSP establishment problem

Green Controllers-Switches Associations

SDNs are becoming popular in both academia and industry. The new networking paradigm facilitated by SDNs promises to accelerate deployment of new services as well as simplifies network management [34]. In particular, controllers are now responsible for affecting the behavior of switches. To this end, the second problem addressed in this thesis is in the context of SDNs. Specifically, the controller placement problem. Consider Figure 1.2. It depicts a SDN before and after the operation of a green TE approach. First consider Figure 1.2a. Here, node *C1* is the controller for switches *S1* and *S3*, and *C2* is the controller for switch *S4*. Figure 1.2a also shows the respective Switch-to-Controllers (S2C) paths for *S1*, *S3* and *S4*, $[S1, C1]$, $[S3, C1]$, and $[S4, C2]$, respectively. Assume that a green TE technique decides to



(a) **(b)**

Figure 1.2: Switches and their assigned controllers. a) Before the shut down of controller *C1*, and b) after the shut down of controller *C1*.

shut down controller *C1* in order to reduce energy consumption. However, controller *C1* has switches *S1* and *S3* associated to it. Therefore, before shutting down *C1*, a green technique needs to associate *S1* and *S3* to a new controller, i.e, *C2*; see Figure 1.2b. The selection of a new controller for switches *S1* and *S3* will have an impact on the operation of the SDN. For example, the new S2C paths for *S1* and *S3*, $[S1, S2, C2]$ and $[S3, S1, S2, C2]$ respectively, are now longer, and therefore, these switches may experience increased delays. Note that Switch-to-switch (S2S) paths may also be longer, which may introduce additional delays in switch to switch communications. In addition, controller *C2* will have a larger number of associated switches, which will cause an increase in load. Furthermore, links $S1 - S2$ and $S2 - C2$ will observe an increase in link utilization.

In summary, the problem is to re-associate switches such that they use a smaller number of controllers and also the path used by these switches traverse the fewest number of links and switches. The number of switches associated to a controller plays a critical role in the performance of a SDN. Consequently, it is important that controllers have similar number of associated switches. Moreover, new S2C and S2S paths will need to meet a given propagation delay threshold [35] [36]. Lastly, the paths used to communicate with a controller must involve the minimum number of active links/switches.

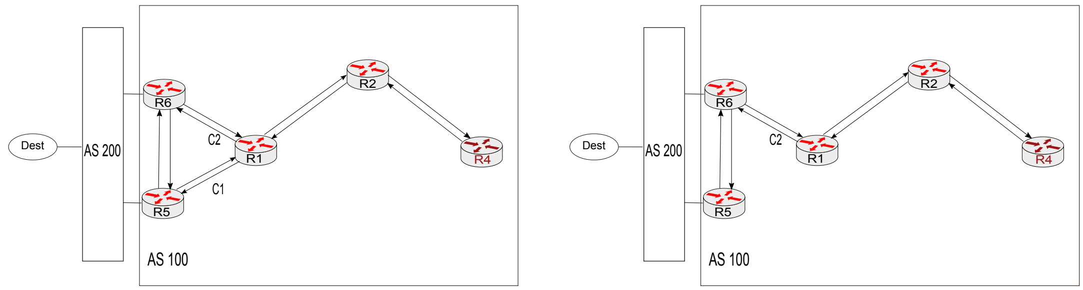
BGP Awareness

BGP is the "glue" that interconnects all Autonomous Systems (ASes) together. Without BGP, the different ASes will not be able to advertise reachability information, and thus the Internet will be disconnected. Given BGP's importance, this thesis studies the following question: does green TE impact BGP negatively? The observation is that green IGP approaches modify a network's routing, and consequently, may induce the well-known effects of hot-potato routing. Specifically, in [37] and [38], the authors establish three main impacts IGP routing has on BGP: (i) transient packet delays and loss while routers re-compute their forwarding tables, (ii) BGP routing changes that affect the BGP operation of peer ASes, and (iii) shift in traffic that may cause congestion on new paths to other ASes.

Problems (i) and (ii) cause a large number of BGP update messages when a router changes its egress point for a number of prefixes. Consequently, a burst of BGP updates can disrupt the forwarding plane by temporarily overloading the Central Processing Unit (CPU) of routers. With respect to problem (iii), routing changes can lead to a sudden increase in traffic at new egress points along downstream paths. This increase will affect popular destination prefixes, leading to even larger shifts in traffic than expected.

Figure 1.3 presents the topology before and after implementing a green TE technique. The network depicted in Figure 1.3 belongs to AS 100, which is connected to the Internet via AS 200 using BGP. The ingress router (IR), *R1*, is presented with two equally good routes via the egress routers (ERs) *R5* and *R6*, with IGP cost $C1$ and $C2$, respectively. Assuming $C1 < C2$, as noted in [37] [38], network operators employ *early-exit* or *hot-potato routing*, meaning routers prefer paths with the lowest IGP cost. Hence, *R1* will select router *R5* in order to reach *Dest*.

As mentioned, *R5* has been selected as the exit point for AS 100 due to *hot-potato* routing. Now, consider the case where *R5* has traffic for *R4*. There are two paths that can be used to route this traffic: $[R5, R1, R2, R4]$ and $[R5, R6, R1, R2, R4]$.



(a) **(b)**

Figure 1.3: Exit points for AS 100. a) Before a green TE approach performs any IGP cost change, b) after changes to IGP cost.

In order to reduce the number of active network elements, the green TE technique within AS 100 decides that *R5* can reach *R4* by using the path $[R5, R6, R1, R2, R4]$, and consequently, the return traffic *R4* to *R5* can be routed through the path $[R4, R2, R1, R6, R5]$. This means links $R5 - R1$ and $R1 - R5$ can be shut down and the egress point for AS 100 may no longer be *R5*. Before shutting down links $R5 - R1$ and $R1 - R5$, a green TE technique will need to divert traffic traversing those links onto link $R1 - R6$ and $R6 - R1$, respectively. This will cause changes in IGP cost within the AS. Such changes, as established in [37] and [38], have well known ramifications that may jeapordize the operation of the Internet. To date, past works have not quantified the impacts that green TE techniques have on BGP. In addition, green TE techniques must not negatively impact BGP.

Uncertain Traffic

The last problem addressed in this thesis is to relax a key assumption of past green TE approaches. Specifically, existing TE approaches are idealistic in that they assume traffic is deterministic. Consequently, there is a need for more pratical and realistic green TE approaches. A key challenge, however, is that obtaining the required traffic matrices (TMs) is difficult and challenging [39]. To this end, a key innovation that simplifies the representation of random traffic demands is a polyhedra set [33]; see Chapter 6 for details. Briefly, the set or polytope describes

all possible demands that can be taken by source and destination pairs, and are described in terms of inequalities. Also, any convex combinations of the extreme points of the polytope are also valid TMs. Its simplicity coupled with the fact that it can be readily incorporated into a Linear Program (LP) have spurred many researchers to address network dimensioning problem over a given polyhedra set.

Consider the topology in Figure 1.4a. Assume all links are undirected and have unit capacity. Also, let the required link utilization be no more than $\gamma = 0.8$. In addition, router *R1* has outgoing demands d_{15} and d_{16} that are constrained by the inequality $d_{15} + d_{16} \leq 0.8$. The total incoming traffic to both routers $R5$ and $R6$ must be less than 0.8; i.e., $d_{15} \leq 0.8$ and $d_{16} \leq 0.8$. Given these inequalities, we thus have a polyhedron with extreme points $(0, 0)$, $(0, 0.8)$ or $(0.8, 0)$. The aim is to switch off as many links as possible whilst supporting the demands from router *R1*.

One possible solution is to switch off links $R2 - R5$ and $R5 - R6$ to yield the topology in Figure 1.4b; a saving of 28.6%; two links out of seven have been switched off. Notice that the active links are able to support the said extreme points as well as any convex combinations of these points. The optimal solution, as depicted in Figure 1.4c, is to switch off three links, i.e., $R1 - R3$, $R3 - R5$ and $R5 - R6$; a saving of 42.9%.

To date, no works have considered green TE over a polyhedra set. This is a critical gap because current green TE solutions, see [40], are not robust against random TMs. Consequently, they are unable to adapt to varying demands. Indeed, a new set of routers/links may need to be activated or deactivated every time demands change.

1.1 Contributions

This thesis presents a number of contributions and adds to the state-of-art by: i) introducing a study that evaluates the performance of online and offline approaches

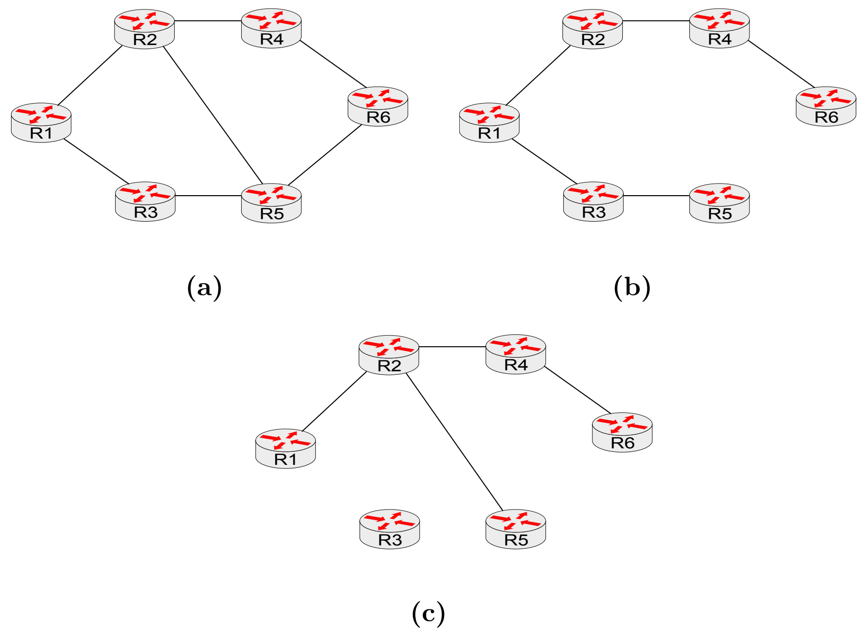


Figure 1.4: (a) Original topology, (b) a possible solution with two switched off links, and (c) the optimal solution that supports the polyhedral set defined by the inequalities $d_{15} + d_{16} \leq 0.8$, $d_{15} \leq 0.8$ and $d_{16} \leq 0.8$.

and presenting a novel metric to evaluate green approaches in terms of achieved energy savings and LSP acceptance rates, ii) proposing an energy aware controller association algorithm for SDNs, iii) quantifying the negative effect green TE approaches have on BGP and introducing a BGP-aware green TE technique, and iv) reporting the first robust green TE solution.

The following sections summarize each of these contributions.

1.1.1 Online and Offline LSP Establishments

This thesis presents a study that compares online and offline green LSP establishment methods using a novel metric that takes into account both energy savings and acceptance rates. Furthermore, it presents a new heuristic that minimizes energy usage by routing source-destination demands over paths that traverse established links and involve the fewest number of new links. Extensive experiments involving well known topologies such as Abilene and AT&T using varying traffic loads confirm that LSP acceptance rates above 90% are feasible with 20% of links shut down.

1.1.2 Controllers Association in SDNs

As mentioned earlier, in a SDN, S2C paths are selected according to their delay, link load, and the number of switches associated to a controller. However, a green TE technique may increase propagation delays, link utilization, and cause some controllers to have a disproportionate number of switches or load. In order to overcome such issues, this thesis presents a green centralized controller association algorithm, aka GreCo, that reduces the energy consumption of a SDN while ensuring that link load and S2C latencies are below a given threshold, and controllers have a similar number of associated switches.

GreCo determines the best path for each S2C pair by establishing the path with the lowest latency. Links that do not belong to the selected paths are shut down. Results over four well known topologies show that GreCo achieves energy savings of up to 55% during off-peak times. Moreover, in experiments over Abilene, GreCo uses no more than 20% additional links as compared to the optimal solution.

1.1.3 Protecting BGP

This thesis answers the following question in the affirmative: do green TE techniques have an impact on BGP operation? This thesis shows conclusively that green TE techniques cause a significant amount of hot-potato routing changes and reroute a large proportion of traffic that may overload neighboring ASes. Specifically, the results indicate that green TE approaches cause more route changes than when running a non-green TE technique. In particular, they show that the increase in egress/border router selection changes is 29% over non-green approaches. The results also show that the proportion of traffic shift is above 25% for a topology running a green networking technique. In particular, for Interior Gateway Protocol Weight Optimizers (IGP-WO) [41], the shift in traffic is directly related to the observed percentage of egress/border router changes. These results therefore, confirm green TE techniques have a negative impact on BGP.

In order to mitigate such effects, this thesis proposes two novel BGP aware approaches. The first one is called Hot Potato Low Utilization (HotPLUZ). The algorithm reroutes traffic from lowly utilized links and aggregate said traffic onto highly utilized links whilst minimizing any changes to the corresponding ER of a given destination. In addition, HotPLUZ considers link utilization in order to prevent packet loss and high latencies. Experimental results over four well known topologies, indicate an overall saving of up to 21% under low network load.

The second BGP aware TE approach is a Hot-potato Low Energy Consumption (HotPLEC) framework that allows IRs to work collaboratively to determine the set of links that should be carrying their respective traffic to ERs. IRs exchange information about their established paths with each other, and each IR determines whether shifting its existing paths to ERs improve energy savings. Each IR repeats the process until convergence; i.e., rerouting no longer yields energy savings. Unused links are then shut down, and thereby, lowering the energy consumption rate of a network. Experimental results over five topologies show that green TE techniques can use as little as 25% more active links than the optimal solution.

1.1.4 Uncertain Traffic Demands

Current green TE techniques have not considered random traffic demands. In fact, green TE techniques allocate resources, i.e., active network elements, according to peak demands. The disadvantage of such an approach is that it may yield many unnecessary network elements that contribute to the overall energy expenditure. Conversely, too few resources may cause congestion, high delays and/or packet loss.

Henceforth, this thesis: i) formalizes a novel problem that calls for the minimum resources to be used for a given polyhedra set, and ii) proposes the first green TE solution that ensures all demands described by a polyhedra set are supported with the key constraint that no link utilization exceeds a given threshold. Advantageously, as the resulting solution is robust for all demands within the polyhedra set, a network

operator does not need to recompute a new solution whenever the TM changes. Experiments over well-known topologies show that savings above 80% are achievable whilst remaining robust to traffic changes.

1.2 Publications

The work in this thesis has resulted in the following papers:

1. **A. Ruiz-Rivera**, K-W. Chin, R. Raad , and S. Soh. *HotPLUZ: a BGP-aware green traffic engineering approach*, IEEE International Conference on Communications (ICC), Sydney, Australia, June 2014.

2. **A. Ruiz-Rivera**, K-W. Chin, and S. Soh. *A novel framework to mitigate the negative impacts of green techniques on BGP*, Elsevier Journal of Network and Computer Applications, vol. 48, no. 2, pp. 22-34, February, 2014.

3. **A. Ruiz-Rivera**, K-W. Chin, and S. Soh. *GreCo: An Energy Aware Controller Association Algorithm for Software Defined Networks*, IEEE communications Letters, 19(4), p541-544, April, 2014.

4. **A. Ruiz-Rivera**, K-W. Chin, S. Soh and R. Raad. *On the Performance of Online and Offline Green Path Establishment Techniques*, EURASIP Journal on Wireless Communications and Networking, October, 2015.

5. **A. Ruiz-Rivera**, K-W. Chin, and S. Soh. *Green-PolyH: A Green Traffic Engineering Solution Over Uncertain Demands*, IEEE International Telecommunication Networks and Applications (ITNAC) Conference, Sydney, November, 2015.

1.3 Thesis Structure

1. *Chapter 2.* This chapter presents a literature review of existing green TE approaches.
2. *Chapter 3.* This chapter presents a study on the relationship between energy savings and LSP acceptance rates on green TE LSP establishment methods.
3. *Chapter 4.* This chapter introduces the challenges that green TE approaches face in SDN environments and outlines the GreCo algorithm.
4. *Chapter 5.* This chapter is the first of two chapters that studies BGP and proposes a novel BGP-aware solution called HotPLUZ.
5. *Chapter 6.* This chapter contains another BGP-aware TE solution. In particular, it presents a novel framework that allows ingress-egress routers to collaboratively determine the best paths for transit traffic that maximize energy savings.
6. *Chapter 7.* This chapter presents Green-PolyH, a green TE algorithm that is robust to traffic changes and considers random traffic demands.
7. *Chapter 8.* This chapter concludes the thesis, and provides a summary of research outcomes and future research directions.

Chapter 2

Literature Review

This chapter is divided into two parts. The first part explores works that deal with the problems related to this thesis. As it will become evident later, these works do not consider energy efficiency. The second part surveys green TE approaches.

2.1 Non-Green Approaches

This section presents non-energy aware works, and aim to place the problems and contributions outlined in Chapter 3, 4, 5 and 6 into context. In particular, the following sub-sections will cover the following areas: i) *LSP establishment*, and ii) *controller placement*.

2.1.1 LSP Establishment Approaches

Multi-Protocol Label Switching (MPLS) networks are conceived from the necessity of having better scalability and faster packet forwarding, especially in the Internet backbone [42]. One of the main advantages of MPLS is its capacity to perform TE [20]. Specifically, MPLS-TE is capable of explicit routing and arbitrary splitting of traffic across established Label Switched Paths (LSPs). Consequently, MPLS is a popular TE tool relied upon by network operators in today's Internet and private

networks. For instance, reports such as [43] indicate that the service market revenue for MPLS Virtual Private Network (VPN) services nearly doubled from \$5.5 to \$10.7 billions during the 2009-2014 period. Moreover, the same report also indicates that customers continue to migrate their voice, data, and video applications to MPLS networks. In addition, the adoption of cloud-based business applications, and the need to securely connect distributed users and data centers will continue to fuel market demand for MPLS services [44].

Figure 2.1 depicts an example MPLS network/domain. In a typical MPLS domain, an ingress Label Switch Router (iLSR) classifies incoming IP packets flows into Forwarding Equivalence Classes (FECs). Packets belonging to a given FEC are forwarded on the same LSP, which is nothing more than a sequence of LSRs whose final hop or destination is known as the egress Label Switch Router (eLSR). The iLSR receives LSP requests, each denoted as $<s, d, bw>$ from a customer equipment (CE) device. These LSP requests are then routed on a path that meets one or more objectives. In this regard, an important objective is to maximize LSP acceptance rates, i.e., minimize LSP interference. The idea here is to serve arriving LSP requests in a manner that maximizes the acceptance rate of future LSP requests; i.e., there is sufficient bandwidth to accept a high number of LSP requests. For instance, in Figure 2.1, the LSPs that serve LSP_1 and LSP_2 should be chosen carefully so they do not interfere too much with the bandwidth requested by LSP_3 and LSP_4; if after selecting LSPs for LSP_1 and LSP_2 there is no more available bandwidth, LSP_3 and LSP_4 will be rejected; this equates to an acceptance rate of 50%.

In the sequel, a number of works that deal with the problem of establishing LSPs in MPLS networks will be reviewed. Critically, these studies do not aim to conserve energy but aim to address the problem of establishing LSPs in order to satisfy one or more pre-established QoS constraints.

In [45], Hong et al. propose Multiple Constraint-based Shortest Path First (MC-SPF), which is based on the widely used Constraint Shortest Path First (CSPF)

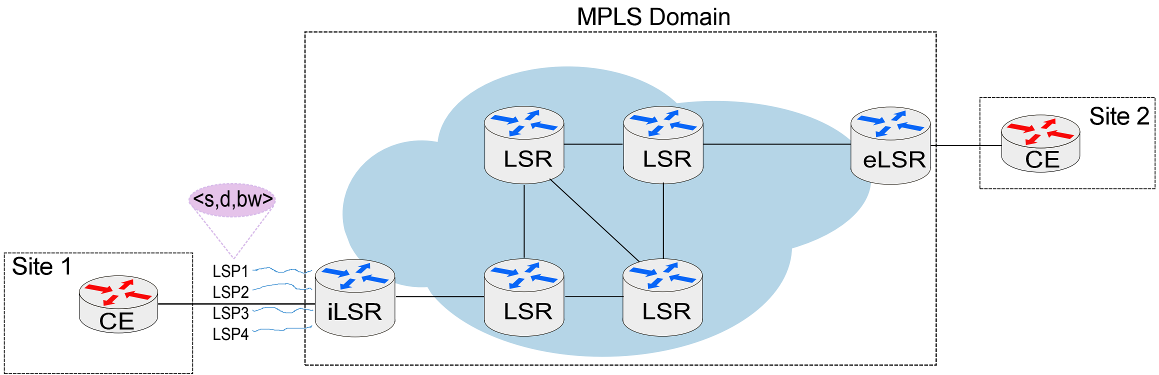


Figure 2.1: A typical MPLS network

[46] [47] [48] algorithm. Briefly, CSPF computes the shortest path between a given $<s,d>$ pair that satisfies a set of QoS requirements. Specifically, CSPF removes all infeasible links, i.e., links that are unable to support pre-defined QoS constraints, and picks the shortest path that traverse the remaining links. MCSPF have three stages: i) *Link pruning*. It removes links that do not meet the QoS requirement of a request, ii) *Weight assignment*. For each request, it assigns a weight value to each of the remaining links by taking into account current traffic load and link interference, iii) *QoS routing*. A routing decision is made based on the weight assigned to the remaining links and also the bandwidth and end-to-end delay requirements of an LSP setup request. A key advantage of MCSPF is its ability to increase LSP acceptance rates because it considers current traffic load and link interference. However, its running time scales proportionally to the number of nodes.

QoS aware LSP establishment approaches are discussed in [49] and [50]; the authors outline Widest-Shortest Path (WSP), Shortest-Widest Path (SWP) and Shortest-Distance Path (SDP) algorithms. These algorithms establish an LSP between a given $<s,d>$ pair by selecting paths in which the bottleneck bandwidth is greater than or equal to the requested bandwidth. In the case of WSP, the shortest path is considered as the feasible path. If there are multiple feasible paths, WSP selects the path with the maximum bandwidth. As for SWP, it selects the path that contains the maximum available bandwidth. If there are multiple paths, SWP

selects the shortest one. Finally, SDP selects a path with the least cost, where cost is defined as the reciprocal of the available path bandwidth. The main advantage of WSP, SWP and SDP is their capacity to maximize bandwidth usage while considering QoS constraints such as end-to-end delays. The main disadvantage of these algorithms is that they do not consider future LSP setup requests and therefore, network utilization and LSP acceptance rates are not maximized.

A number of works aim to reduce LSP interference; this is equivalent to maximizing the LSP acceptance rate. For instance, the authors of [51] aim to minimize LSP interference while fulfilling load balancing /packing QoS constraints. The proposed algorithm, called Multi-Objective Path Selection (MOPS), is a derivative of the Minimum Interference Routing Algorithm (MIRA) [52] [53]. Load balancing is traditionally achieved by avoiding highly utilized links. However, this approach has been known to cause fragmentation of available bandwidth, which reduces the acceptance rate of LSP requests [54]. Henceforth, the authors implement load packing, which groups LSP setup requests and serves them by selecting paths that do not contain critical links, i.e., highly loaded links. The algorithm operates in two phases: i) offline. This phase finds a set of paths that do not contain critical links by taking into account load balancing, load packing and resource conservation, and ii) online. Among the paths found during the offline phase, the algorithm selects the path that causes the minimum interference to future requests. The main advantage of MOPS is the reduction in blocking probability due to the minimization of LSP interference. However, the selection of less critical paths does not take into account current load conditions, and thereby, decreases LSP acceptance rates.

The work in [47] also aims to maximize LSP acceptance rates. The authors propose a modification to the Wang-Crowcroft (WC) algorithm [55], called the Wang-Crowcroft with Sorting (WCS). It aims to increase LSP acceptance rates by taking into account the arrival order of LSP requests. Briefly, the WC algorithm selects a path that satisfies multiple QoS constraints by taking into account the minimum

available bandwidth and delay. The new approach, called WCS, receives as inputs the minimum bandwidth and maximum delay requirements and it is comprised of the following phases: i) *WC*. In this phase, the WC algorithm is run without any modifications, and ii) *Sorting*. If some LSP requests are rejected, they are first re-ordered according to their requested bandwidth, and then re-ordered based on their delay requirement. After each reordering, the WC algorithm is run and the best solution, which is the one that yields the biggest LSP acceptance rate is chosen. The key advantage of WCS is how LSP requests are reordered according to their bandwidth and delay requirements. This reordering facilitates the classification and prioritization of traffic. Its main disadvantage is that future LSP requests are not taken into account when selecting the best path, which will lead to an increase in LSP interference.

Similarly, the work reported in [56] also aims to maximize LSP acceptance rate. The proposed algorithm, called Stochastic Performance Comparison Routing Algorithm (SPeCRA), adaptively selects the best LSP establishment method amongst a given set of candidate methods; e.g., K shortest path. The selection of the best method is performed by computing the blocking probability for each of the candidate methods for a given number of LSP setup requests that arrive during a time interval. Once the blocking probability for each of the LSP establishment schemes is known, SPeCRA selects one with the minimum blocking probability. The main advantage of SPeCRA is its versatility to evaluate different LSP establishment methods and its capacity to select the scheme that exhibits the maximum the LSP acceptance rate for a specific type of traffic. However, the process of selecting a candidate method increases convergence time, which has an impact on delay-sensitive applications.

The allocation of available resources, e.g., bandwidth, has an impact on LSP acceptance rates. To this end, Kuribayashi et al. in [57] propose Key-Direction algorithm (KDA), which aims to improve how resources are used by selected LSPs. Critically, KDA considers the bi-directionality of LSPs; i.e., the bandwidth require-

ment from their source to the destination, and vice-versa. This is in contrast to previous works, e.g., [58] [59], where bandwidth is assigned based on unidirectional LSPs. Consequently, KDA is ideal for real time applications such as voice and video, where traffic between nodes is delay-sensitive and symmetric. Three LSP selection methods are presented: i) *conventional method*. This method evaluates all LSP pairs in a round-robin fashion until an LSP pair with sufficient bandwidth is found, ii) *direction-dependent method*. This method selects the LSP with the least available bandwidth after identifying the unidirectional LSP that requires the largest proportion of bandwidth; i.e., the ratio between the required and maximum bandwidth for each unidirectional LSP, and iii) *network delay method*. The LSP pair with delay that is less than a given threshold will be selected.

An approach to reduce congestion is by load balancing traffic across multiple LSPs. The solution, called MPLS Adaptive Traffic Engineering (MATE) algorithm [60], routes traffic using pre-established LSPs according to characteristics such as packet delay, packet loss or network utilization. Ingress and egress nodes running MATE exchange probe packets, which provide one way, packet delay and packet loss statistics. These statistics are then used to determine the level of congestion, i.e., link utilization, along a LSP. The statistics are also used to determine the monitoring phase, where congestion is monitored continuously. They also indicate it is time to move to the load balancing phase. If the link utilization goes beyond a given threshold during a pre-defined amount of time, meaning there is persistent congestion, MATE moves into the load balancing phase, which helps improves resilient against failures due to congestion. Its main disadvantage is its high signaling overheads due to the use of probe packets.

The number of point-to-point LSPs (P2P LSPs) grows proportionally to the square of the number of nodes Therefore, the calculation of P2P LSPs, and their corresponding backup P2P LSPs, for large topologies, becomes intractable with increasing number of nodes. This problem is addressed by Saito et al. [61] with the

introduction of Multipoint-to-Point LSPs. Their main goal is to reduce the number of required P2P and backup P2P LSPs. A multipoint-to-point LSP represents paths from multiple ingress nodes to a single egress node. As only one LSP per egress node is needed, the total number of LSPs will be equal to the number of egress nodes. Setting up such LSPs require solving two sub-problems: i) *multipoint-to-Point LSP selection*. Primary and backup multipoint-to-point LSPs are established by taking into account the shortest path between source and destination nodes, ii) *route selection*. The objective is to reduce congestion by minimizing the overall network load. To this end, ingress nodes use traffic flow statistics such as link utilization in order to determine which LSPs will be used to route traffic demands.

The aforementioned works can be classified based on the time when LSP computation is performed: i) *Online*, such as [52] [53] [60], or ii) *Offline*, such as [47] [57]. The main difference between these two categories is the availability of a TM and an a priori knowledge of LSP setup requests [20] [47]. Many works tend to optimize the establishment of LSPs based on different QoS constraints, such as bandwidth, packet loss and end-to-end delay [45] [51]. The order of LSP setup requests is also important when establishing paths, as discussed in [47], whereas minimizing interference is critical in order to increase LSP acceptance rates [52] [53]. A key gap is that these methods have not taken energy consumption into account when selecting the optimum LSPs. Indeed, existing works assume the existence of redundant paths and nodes [60]. Green LSP establishment methods, see Section 2.2.2, however, have an opposite aim. They seek the minimal number of nodes or links.

2.1.2 Controller Placement Approaches

Traditional IP networks are complex and difficult to manage. Current networks are not sufficiently flexible to allow automatic reconfiguration when dealing with faults or to adapt to load changes. In addition, current networks are said to be "vertically integrated". That is, the control and data plane are bundled inside network devices.

The former implements network policies and performs routing decisions whilst the latter performs data forwarding. Unfortunately, the current network architecture is not flexible and slows innovation [31]. To this end, SDNs have become popular and touted as a promising solution that would allow operators to utilize and develop network programming languages and applications that simplify network management [62] [63].

Figure 2.2 presents an overview of a SDN architecture. Its key design is the separation of the data and control planes. Consequently, network switches become simple forwarding elements that are controlled by one or more controllers. On the other hand, the centralized controller platform consists of one or more controllers that maintain and implement global network policies. In particular, they control the fate of all packets. Each switch has a secure channel to a controller [64] and communicates using the OpenFlow protocol/standard [65]. Lastly, switches contain a simple flow table with entries populated by a controller.

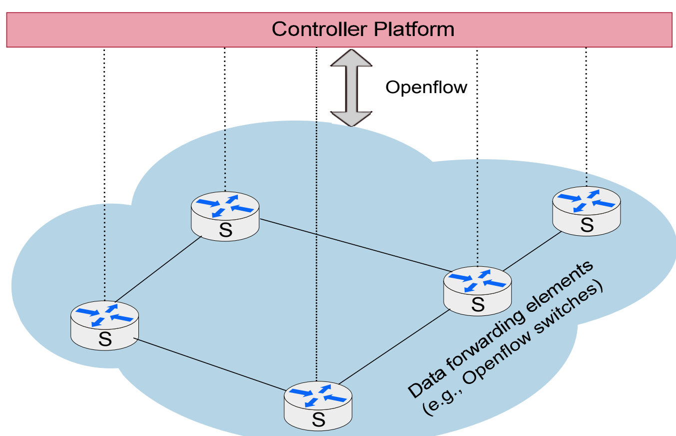


Figure 2.2: A high level view of a SDN architecture

In a SDN, there may be multiple controllers. This is due to the following reasons: i) *administrative*. Each controller is responsible for only a part of a network, which facilitates the grouping of devices, and thus allowing a network operator to simplify monitoring and management of network policies, ii) *geographical location*. In a SDN that spans a wide geographical area, controllers will have to be placed near switches

in order to reduce flow setup times or delay in which a new entry is added into a switch's flow table after detecting a new flow, iii) *switch-to-controller latency*. Large propagation delays slow network convergence, and impair a controller's ability to respond to network events, iv) fault tolerance. A single controller architecture may result in a single point of failure, and v) load balancing. This ensures all controllers have a similar load [66] [67] [68] [69].

A fundamental problem addressed in Chapter 4 is the Controllers Placement (CP) problem [35] [70]. It deals with two fundamental questions: i) where to place controllers, and ii) how many controllers are required. The first work that studies the CP problem is by Heller et al. [35]. They define a number of controller placement metrics that consider latency. Namely, *i)* average-case latency, where controllers are placed to minimize the average node-to-controller propagation delay, *ii)* worst-case latency, where the goal is to minimize the maximum node-to-controller propagation delay, and *iii)* latency bound, where controllers are placed in a manner that maximizes the number of nodes with latency below a given bound. The experimental results presented in [35] indicate that deriving a solution that is suitable for different topologies is not feasible, and therefore, the problem needs to be addressed according to the characteristics and conditions of each network.

References [70] [71] also introduce placement metrics that aim to maximize the reliability of a SDN. Specifically, they ensure switches have a reliable control path to their associated controller as well as ensure there is a path between controllers. In [70], the objective is to maximize the percentage of reliable control paths to protect against network failures. To this end, the authors argue that outages of control paths are the result of underlying physical failures that can be explicitly identified and are statistically independent. The authors extended their work in [71] where they quantify the number of impaired control paths due to network failures. Hence, the objective is to minimize the percentage of control path loss. In both [70] and [71], in order to simplify calculation, the authors assume the possibility of simulta-

neous failures involving multiple network components within a single administrative domain is very low [72]. Therefore, they only take into account scenarios where at most one network element fails. This assumption reduces the number of failure scenarios meaning it helps speed up the selection of the set of controllers and switches capable of meeting the reliability requirement of control paths.

Reliability and fault tolerance between controllers and nodes are also studied in [73], where the authors introduce the fault tolerant CP problem. The aim is to guarantee a control path with 99.999% reliability; this implies that the control path is only down for a total of five minutes in a given year. The authors develop a Binary Integer Programming (BIP) formulation that takes into account all possible states, i.e., up/down, of all network components, in order to compute the availability of control paths. Given that the number of possible states grows with the number of nodes and links, the fault tolerant CP problem is categorized as NP-hard. Therefore, the authors propose a heuristic that aim to minimize the overall number of controllers and switch-to-controllers connections adhering to the availability of control paths and fault tolerance constraints. Their heuristics reduce the search space by only considering subsets of facilities, i.e., locations where a controller can be deployed, that meet the desired control paths and fault tolerance requirements. These subsets are then ranked according to how they affect the expected reliability of control paths. Preference is given to facilities that allow the maximum number of nodes to achieve the required control paths availability.

The study in [73] indicates that the number of controllers cannot be fixed and needs to be calculated according to QoS and reliability requirements. To this end, reference [74] proposes the dynamic controller provisioning problem algorithm. Its goal is to adapt the number of controllers and their respective location according to network conditions. The presented algorithm aims to minimize flow setup times and reduce switch-to-controller communication overheads by considering a weighted sum of the following parameters: i) *statistic collection cost*. This cost is defined as

the number of messages, within a second, that is required by a controller to collect statistics from its associated switches, ii) *flow setup cost*. Defined as the total cost incurred for setting up the flow rules across end-to-end paths, iii) *synchronization cost*. This cost represents the number of messages that are exchanged between controllers in order to maintain a consistent global view of the network, and iv) *switch reassignment cost*. This is the cost of assigning a switch to a new controller. The weight of each of the aforementioned costs can be manipulated by network operators in order to reflect specific network characteristics.

A controller's load is loosely determined by the number of associated switches. To this end, the k-critical algorithm [67] aims to find, among k sets of controllers, the smallest set that satisfies delay, latency, or convergence time. The k-critical algorithm places controllers according to network requirements and physical network characteristics. In particular, when associating switches to a given controller, a function evaluates the switch-to-controller propagation delay between a set of candidate controllers and a set of unmanaged nodes. A controller is considered part of the candidate set if its location satisfies a given propagation delay. In addition, k-critical aims to minimize the loss of switch-to-controller communications as a result of network failures. This is achieved by guaranteeing robustness at the control layer. That is, before placing a controller, the algorithm evaluates the disruption to control paths resulting from a link or controller failure. Therefore, fault tolerance is ensured by not selecting controllers that result in significant loss of data if they fail.

Reference [75] considers the latency between controllers and switches. The authors introduce the Pareto-based optimal CP algorithm. The algorithm evaluates the maximum latency between nodes and controllers in order to identify the set of controllers and their associated switches that fulfill latency and resilience constraints. The algorithm also considers fault tolerance and load balancing. A key conclusion is that achieving QoS requirements, such as tolerance to failure and load balancing, at the same time is difficult. Therefore, trade-offs are necessary. To this

end, the Pareto-based optimal CP algorithm offers network operators the possibility of placing controllers according to network conditions. For instance, bounding the maximum latency might be more important than preventing controller overload during off-peak times. On the other hand, at peak times, the probability of overload increases, potentially causing the controller to become a bottleneck [76]. Hu et al. in [77] propose an approach called Balanceflow that partitions control traffic load among different controllers. All controllers maintain their own load information and publish this information periodically through a cross-controller communication system. BalanceFlow also introduces a super controller to detect traffic condition changes. It also load balances flow setups among controllers.

The work in [78] also aims to optimize the load of controllers. The authors introduce the concept of cascading failures. In current SDN designs, the load of a failed controller is redistributed to other controllers. However, the redistribution of load may cause other controllers to exceed their capacity, meaning they will also fail; hence, the term cascading failure. As a solution, in the event of a controller failure, their strategy redistributes the load among remaining controllers by taking into account the capacity of controllers. Similarly, the authors of [36] address load distribution among controllers. Specifically, they introduce a version of the CP problem that takes into account the load of controllers. The capacitated CP problem extends the k-center problem presented in [79], where controller loads are not considered. In the k-center problem, controllers are considered to have constant load, and therefore, their capacity is not a constraint. In contrast, the capacitated CP problem takes into account the load of each controller and its capacity when performing switch-to-controller associations. In addition, the problem also considers controller-to-switches latency constraints. The authors model the problem as a linear programming formulation that aims to minimize the latency/distance from each switch to its assigned controller. The formulation takes into account the load of each of the controllers and their respective capacity. In order to find the optimum

set of switch-to-controller assignments, the formulation evaluates all the possible placement and switch-to-controller assignment combinations. This, however, is intractable for large networks. The authors then propose an efficient algorithm that uses binary search to find the minimum latency/distance and only considers the distance between any pair of nodes instead of considering all possible distance values.

The aforementioned works deal with the CP problem by considering a number of QoS characteristics such as latency, reliability, fault tolerance, and controller load. An interesting model is presented in [69] where the authors present a multi-controller architecture. Specifically, idle controllers are placed into the dormant state with the goal of minimizing the total system operating cost. However, system operating cost does not refer to the energy consumption of a SDN but to the value of a cost function that is defined in terms of key performance metrics. Specifically, *i)* the number of active and dormant controllers, *ii)* the number of incoming messages that will wake up the dormant controllers, and *iii)* system capacity, which refers to the maximum number of messages the controller layer is capable of handling. In order to find the values for these metrics, the authors develop an expected total cost function and use a genetic algorithm; briefly, genetic algorithms are based on biological evolution behavior [80]. The value of the key performance metrics are used to calculate a fitness function cost. During each generation, all chromosomes are sorted in an ascending order according to their fitness. Chromosomes with a high fitness are eliminated. Otherwise, they are passed to the next generation. The final generation, which presumably is the fittest, represents the optimum value for the performance metrics.

On the other hand, an approach that does consider the energy consumption of a SDN is presented in [81]. Here, a single controller SDN architecture that employs traffic aggregation in order to increase energy savings is studied. In this architecture, the controller considers current traffic conditions and historical traffic statistics with the aim of aggregating traffic flows over the minimum number of links. Links that

are not carrying any traffic are put into a suspended state. To this end, the controller uses its knowledge of the network topology to calculate a path for a given traffic demand. Higher loaded links are preferred over lighter ones as long as link loads are within a predefined value.

2.2 Green TE Approaches

Green routing TE approaches are motivated by the fact that current networks are over-provisioned to provide redundancy and to preserve QoS during peak periods. This means current IP networks are designed to handle the "worst-case" scenario in terms of failures and traffic demands [7]. Interestingly, many works have shown traffic to exhibit diurnal patterns, corresponding to business hours and during weekends [82] [83]. Moreover, it has been reported in [9] that the utilization of backbone networks can be less than 30%. Green routing techniques, therefore, play an important role in regulating the energy consumption of a network as they can be designed or configured to maximize the number of network elements, such as links and routers, that are put into sleep mode during off-peak times.

This section groups existing green TE works into three categories: i) *energy-aware network protocols*. These green TE techniques rely on existing network protocols such as OSPF [84], ii) *energy aware LSP establishment methods*. These approaches establish LSPs by considering energy usage, and iii) *Miscellaneous*, which will include techniques that make use of strategies such as traffic aggregation and failover mechanisms. However, they are not tied to a particular routing protocol.

It is worth noting that green TE works can also be classified according to where routing decisions are made [40]. Specifically, *i)* centralized. Decisions are taken by a single node or central controller that has complete knowledge of the topology. Note that centralized approaches suffer from a single point of failure, and *ii)* distributed, which includes approaches that make use of information local to a node.

2.2.1 Energy-Aware Network Protocols

The techniques in this category run on top of or modify traditional network protocols in order to minimize the number of network elements. The key difference between them is the local or global information required to perform routing decisions. A centralized approach will require information about a specific set of nodes and links. On the other hand, in a distributed approach, each node makes its decisions using only local information.

The authors of [27] present a centralized approach called Energy Saving in the Internet based on Occurrence of Links (ESOL). A central entity quantifies the the number of times in which nodes and links appear in paths. It then marks those with a low occurrence as candidates to be powered off. After the removal of one or more links, ESOL verifies that the network remains connected; i.e., at least a path between every pair of nodes exists. If not, ESOL ignores the marked candidate links and selects a new set of links to be shut down. The authors present four versions of ESOL. Each has a different execution time and energy saving. These versions are as follows: i) basic-ESOL (b-ESOL). This algorithm sorts unidirectional links in decreasing order of their occurrence in paths. Links with the lowest occurence are shut down first assuming they do not disconnect the network. This approach has a slow convergence time given that in each iteration a low number of links is switched off, ii) fast-ESOL (f-ESOL). This approach aims to switch off the largest number of links without disconnecting the network. Links with a low occurrence are shut down, iii) (f+b)-ESOL. The fast and basic ESOL algorithms are run consecutively. The idea is to first shut down a large proportion of links quickly using f-ESOL. After that, other links are slowly powered off using b-ESOL, and iv) (fx2)-ESOL. f-ESOL is run twice. The second f-ESOL run receives as input the set of remaining active links after running f-ESOL for the first time. Their results demonstrate that the most effective algorithm in terms of energy savings, i.e., highest percentage of shut down links, is also the algorithm that has the slowest convergence time. On the

other hand, the fastest convergence algorithm is the one that is able to shut down the fewest number of links.

Zhang et al. in [15] propose GreenTE, an intra-domain, centralized TE mechanism that finds a set of links that can be turned off under a given traffic load or matrix. Here, the links can belong to paths computed by OSPF or are part of LSPs. The problem at hand is formulated as a Mixed Integer Program (MIP) where the objective is to shut down the most number of ports or line cards. The authors then present GreenTE, a heuristic to solve the MIP problem. It takes advantage of redundant links and those with a low utilization. In particular, when the network load is low, and assuming multiple paths between a given $<s, d>$ pair exist, the authors investigate mechanisms for rerouting traffic onto a subset of links, which in turn allow GreenTE to shut down idle links. GreenTE also limits the selection of candidate paths to the k-shortest paths for any given $<s, d>$ pair. These k-shortest paths remain constant and do not change under different traffic matrices. The authors also consider a number of constraints. The first constraint ensures candidate paths do not exceed the network diameter. The second constraint is end-to-end delay, which limits the length of a candidate path to less than twice that of the shortest path. Given these two constraints, the authors explore three different methods to select candidate paths. i) Basic. This method aims to minimize the Maximum Link Utilization (MLU) metric. ii) basic+network diameter, where MLU, and the network diameter are considered, and iii) basic+end-to-end. In addition to the MLU constraint, this method introduces an end-to-end delay constraint for each $<s, d>$ pair. The advantages of GreenTE include reduction in path lengths and computation times given that it only considers k-shortest paths. The main disadvantage is its dependency on green hardware.

In [85], the authors introduce an OSPF-TE [86] based Routing On Demand (ROD) mechanism that considers both QoS requirements and energy savings. A centralized controller is responsible for computing link weights and periodically con-

figures all routers in order to find the minimum number of links that is able to support a given traffic demand. In particular, ROD uses Traffic Engineering Link State Advertisement (TE-LSA) to report link load information. Using non-linear optimization techniques, the researchers combine OSPF-TE and energy-aware TE techniques in order to maximize energy savings and network performance while ensuring all links do not exceed their maximum link utilization threshold. The aim is to minimize the risk of network congestion. Specifically, ROD determines the minimum set of links that can be used to support a given traffic demand. Therefore, their problem is an energy-aware TE problem. The authors then formulate a problem that aims to to minimize both the MLU and energy savings of a network subject to capacity and flow conservation constraints. The authors then consider different scenarios; each of them with a particular trade-off between MLU and energy savings. For each of these scenarios they construct a different objective function. The idea is, for each objective function, they aim to find the set of optimal routes for all $<s, d>$ pairs. The authors argue that according to a property of optimal routing, the set of optimal routes between every $<s, d>$ pair in a network corresponds to the set of shortest paths connecting every source s with every destination d. They then prove mathematically, that for each set of shortest paths, a set of link weights exist. These link weights are the solution of their approach.

The authors of [28] propose two centralized link weight optimizer approaches that modify how link weights are computed in OSPF in order to save energy: Greedy Algorithm for Energy Saving (GAES) and Two-stage Algorithm (TAES). The former first calculates link weights using the Interior Gateway Protocol Weight Optimizer (IGP-WO); see [87]. Links are then sorted in a non-decreasing order according to traffic intensity and link weight. The lowest ranked links become candidates to be switched off. GAES recalculates OSPF link weights without considering the links and nodes previously identified as candidates. The algorithm then determines if the given traffic matrix is supported without exceeding the maximum utilization of

links. In TAES, the algorithm first selects the set of network elements that could be switched off. This is carried out using an Integer Linear Program (ILP) that minimizes power consumption by shutting down idle nodes and links. After that, the resulting topology is passed to IGP-WO, which then calculates new link weights that route current traffic demands whilst ensuring no links exceed their maximum link utilization.

Bianzino et al. in [88] introduce the Green Distributed Algorithm (GrIDA) for backbone networks. GrIDA switches off links according to local information such as current load, and power consumption. Every node calculates a utility function that takes into account its energy usage, which is computed as the sum of its own power consumption as well as that of its active ports. It also calculates a penalty value based on the status of its incident links. Here, a possible link status is inactive, active but non-congested, or active and congested. Low utility links are marked as good candidates to be powered off. GrIDA also ensures the network is always connected. For instance, if a node running GrIDA determines that shutting down a candidate link will cause the network to become disconnected, the node will not proceed. Subsequently, the node will increase its calculated penalty and utility. Otherwise, the candidate link will be shut down and the node will decrease its penalty value. GrIDA also controls the level of congestion by monitoring the traffic statistics reported by LSAs. In particular, if shutting down a link leads to a highly congested network, GrIDA reverts to the previous state, i.e., powers on recently shut down links. In addition, GrIDA also avoids congestion by not shutting down links during peak times.

Cianfrani et al. [17] [26] propose the distributed Energy-Aware Routing (d-EAR) algorithm. Routers are divided into the following sets: exporter, importer, and neutral. Each exporter router is associated with a number of importer routers. However, each importer router is only associated with a single exporter router. Both exporter and neutral routers calculate their Shortest Path Tree (SPT) as per the

Dijkstra algorithm. Importer routers compute a modified path tree by running the Dijkstra algorithm with their associated exporter router as the root node. As a result, the resulting modified path tree has fewer links and thus all links that do not appear in an exporter router's modified path tree can be switched off. The authors indicate that the number of exporter routers plays a significant role in the algorithm's performance. However, as the paper only focuses on how to select such exporter routers instead of defining their optimum number, they assume this optimum number to be fixed a priori. Exporter routers with the highest number of neighbors are selected; this is carried out to ensure each exporter router is able to export its SPT to a large number of importer routers, and thereby, maximize the number of links that can be powered off. D-EAR has a number of advantages. First, d-EAR is compliant with OSPF and hence deployable in current networks. Second, EAR is able to uniformly distribute traffic load. However, in practice, there are a number of issues to be resolved. For example, in regards to exporter routers, it is necessary to select their optimal number to ensure the load on active links do not exceed a given load threshold. In addition, d-EAR can only be used when the traffic load is low. In a subsequent work, Cianfrani et al. [26] present an enhancement to d-EAR that considers positive and negative effects of exporting SPT. To this end, the term "move" is introduced to denote a set of strategies that places a target link into sleep mode. In particular, they introduce the concept of a set of compatible "moves" that will minimize energy consumption; note, "moves" are evaluated according to their consequences on energy savings and network performance, i.e., all the positive and negative effects of an SPT exportation. Therefore, d-EAR is able to have control over the set of links that will be powered off, and by applying the concept of "compatibility" among moves, d-EAR monitors network performance and facilitates the implementation of QoS strategies.

The Green Distributed Routing Protocol for Sleep Coordination (GDRP-PS) approach is presented in [89]. Here, core routers are divided into two sets: *i)* tradi-

tional, i.e., those that are not allowed to sleep, and *ii)* power saving, where a router is allowed to enter sleep mode when the traffic load is low. GDRP-PS has the following stages: *i)* initialization. A coordinator is randomly selected among all power saving routers. The coordinator, which is not allowed to sleep, will remain in that role for a specific period of time until a new election is performed, *ii)* sleeping. If a power saving router finds the traffic load is low, it verifies that network connectivity will not be affected without its presence and sends a message to the coordinator requesting permission to go into sleep mode. If the request is approved, the power saving router will inform its neighbors about its disconnection via OSPF HELLO messages, *iii)* working. After a certain sleep time, the power saving router contacts its coordinator which then decides whether the power saving router is needed to support the current load. If not the router re-enters sleep mode. The main advantage of GDRP-PS is its non-dependency on green hardware. This facilitates deployment in current networks. The main disadvantage of GDRP-PS is that decisions are made based on the link utilization reported by each power saving router. This decreases opportunities to minimize a network's total energy consumption.

A distributed link weight optimizer approach, called ECO friendly Routing Protocol (ECO-RP), is presented by Arai et al. [29]. Here, a subset of routers is selected to gather and disseminate traffic information periodically using LSAs. Each router then modifies OSPF link weights dynamically according to traffic conditions. A subset of network elements, including nodes and links, is computed based on new link weights. When the network load is low, routers will route traffic to these subset of routers and place unused elements to sleep.

All the techniques discussed thus far are either deployed in a centralized or a distributed fashion. The next example work, however, presents a technique that can be run centrally or in a distributed manner. The authors of [90] present Energy Profile Aware Routing (EPAR). It exploits network equipment energy profiles and builds on the Energy Aware Routing (EAR) concept proposed in [10]. EPAR can

be implemented in a centralized fashion, e.g., a network management system or as a distributed approach by utilizing the control plane of existing protocols and their corresponding TE extensions; e.g., OSPF-TE [86]. The authors consider the following energy profiles: i) *linear*, whereby energy consumption depends linearly on traffic load, as observed in certain switch architectures [91], ii) *on-off*. The most common energy profile in use today, where the total energy consumed is independent of traffic load, iii) *Log10*. This profile is exhibited by devices that try to send data as fast as possible in order to return to a low power state. Log10 is now part of the IEEE 802.3az Task Force [92], iv) *Log100*. This profile lies between Log10 and on-off. It matches current energy usage, and v) *cubic*. An atypical profile exhibited by devices that use dynamic voltage scaling or dynamic frequency scaling [93] in order to save energy. The main idea is to combine energy profiles and traffic load to make energy aware TE decisions such as selecting energy paths that use network elements that consume the least amount of energy. The main advantage of EPAR is its ability to make energy saving decisions based on different power requirements and energy consumption characteristics of components. However, the challenge is applying the correct energy profiles in networks where devices are of different models and built by different manufacturers.

All the aforementioned techniques are based on link state routing protocols. However, researchers have also investigated distance vector protocols, which tend to yield simpler protocols. A distributed path-based traffic control method called path-based Energy Cost saving Overlay Routing (path-based ECO-R) is proposed in [94]. The Path-based ECO-R algorithm is a distance vector routing algorithm, similar to the Routing Information Protocol (RIP) [95], that utilizes pre-established paths. Briefly, ECO-R is inspired by how ants find the shortest path between their nest and food; i.e., they use trails of pheromones left earlier by other ants [96]. ECO-R has the following characteristics: *i)* there are a number of candidate paths for each pair of source and destination nodes, *ii)* control packets are sent from egress

to ingress nodes to simulate pheromone being carried over each path that connects them. Pheromone is just an attribute that is decreased when a packet traverses a node that is forwarding a small amount of traffic, or when a packet traverses a link that has a low load. Traffic is aggregated onto paths with a high pheromone level. Each ingress node determines the usage probability by using control packets. A low usage probability indicates that the volume of the traffic is low and therefore, traffic between a given $<s, d>$ pair is aggregated onto one single path. This thus allows nodes on redundant paths to go to sleep. On the other hand, a high usage probability indicates that the traffic volume is large and requires multiple paths. The main advantage of ECO-R is its capacity to route current traffic load using the minimum number of links. However, it generates a significant number of signaling messages when making routing decisions. To this end, it is unclear how these messages affect the overall energy savings.

Table 2.1 presents a summary of the aforementioned energy-aware TE network protocols. It provides a brief description of each approach, and their classification according to the entities making routing decisions.

2.2.2 Green LSP Establishment Methods

As mentioned in Chapter 1, MPLS-TE is a powerful tool that allows ISPs to route traffic according to a given set of requirements. In this respect, MPLS presents an opportunity to reduce energy consumption by establishing the most energy efficient paths. In addition to the centralized versus distributed classification discussed in Section 2.2.1, these green LPS establishment methods can be further categorized as: i) *offline*, where the complete set of LSP requests is known, including future LSP requests, and ii) *online*, where only past and present LSP requests are known.

The centralized and offline Greedy Green MPLS Traffic Engineering Scheme (GGMTES) is introduced in [25]. GGMTES identifies the set of network elements that can be put to sleep during off-peak times. GGMTES receives as inputs the

Table 2.1: A summary of energy-aware network protocols TE techniques

Green TE techniques	**Characteristics**	**Centralized /Distributed**
ESOL [27]	Shutting down network elements based on their occurrences in shortest paths	Centralized
GreenTE [15]	Usage of the k-shortest paths in order to reduce energy consumption	Centralized
ROD [85]	Combination of conventional and green TE approaches. Energy savings are achieved under MLU constraints	Centralized
GAES [28]	Calculation of OSPF link weights in order to minimize the overall link utilization. Links and nodes are sorted and switched off according to different criteria	Centralized
TAES [28]	The algorithm receives as inputs the network topology and traffic matrix and solves a mixed ILP with an objective function similar to GAES	Centralized
GrIDA [88]	Reduces energy consumption based on the load and power consumption of nodes	Distributed
d-EAR [17] [26]	Distributes traffic uniformly to reduce energy usage	Distributed
GDRP-PS [89]	Energy consumption is reduced without affecting network connectivity	Distributed
ECO-RP [29]	LSAs are used in order to disseminate traffic flow information. Traffic is routed on a subset or all nodes according to load	Distributed
EPAR [90]	Reducing energy consumption based on energy profiles	Centralized /Distributed
ECO-R [94]	Distance vector algorithm. Maximizes taffic aggregation based on link usage probability	Distributed

predicted non-peak traffic demands and routing information. It establishes the least cost path that uses the fewest links in order to minimize the number of active links without affecting the performance of the network in terms of MLU. In addition, it limits the number of LSPs in order to reduce configuration overheads. GGMTES operates as follows. It starts by sorting core routers in ascending order according to the number of LSPs traversing them with the aim of identifying the most unpopular router, i.e., the core router with the smallest number of LSPs. GGMTES then identifies the set of traffic demands routed over the LSPs traversing the unpopular router and removes them from the network. For all the links that are not part of the removed LSPs , link costs are recalculated. The cost of a given link is calculated by considering its current load, the amount of rerouted bandwidth and its capacity. Least cost paths are then computed for all the traffic demands that traverse the unpopular router. They are only established if the resulting link utilization stays below a pre-defined threshold. GGMTES repeats the aforementioned procedure until it considers all core routers.

References [97] and [98] aim to reduce energy consumption by introducing an offline and centralized energy model that allows a network operator to selectively shut down devices and links according to traffic load. The authors present an ILP with the objective to minimize the number of active links/routers during a given period of time while satisfying all traffic demands. The ILP's objective function considers the following factors: *i)* energy consumed by an active chassis, *ii)* energy consumed by line cards when powered on, and *iii)* the energy consumed by a chassis when transitioning from the off to the on state. The decision variable is the minimum number of network components required without impacting QoS and also minimizes signaling overhead.

Coiro et al. in [23] and [24] present the online algorithm Distributed and Adaptive Interface Switch-off for Internet Energy Saving (DAISIES). The algorithm dynamically reroutes traffic demands according to link load or number of active line

cards. A full-meshed topology is considered. That is, an LSP is configured between each pair of nodes. Each transmitting node computes a path to a given destination by considering the state of each link, i.e., on/off. All nodes aim to use the fewest links as possible. To this end, two cost functions are proposed in order to calculate link weights: *i)* on-off. This cost function only considers the number of active line cards that are necessary to support a given request. Here, the link weight increases according to the number of active line cards, and *ii)* V-like. Unlike on-off, this cost function considers the actual link load when computing costs. That is, a lowly utilized link will see an increase in its cost. This deters nodes from routing their traffic over the link, meaning it can be shut down once it is idle. The first hop node of a set of LSPs continuously monitors the traffic traversing these LSPs. If the amount of traffic traversing one of these LSPs goes below or above a predefined threshold, the node triggers the re-computation of link weights using on-off or V-like. After that, the node finds the shortest path to route a given $<s, d>$ demand. If the on-off cost function is used, the final set of paths is the one that yields the minimum number of active line cards. On the other hand, if the V-cost function is used, the final set of paths will be the one that results in the maximum number of idle links. These idle links can then be powered off.

In [21], the authors propose the online and centralized Energy Efficient Multi-Constrained Routing Algorithm (E^2-MCRA), which aims to maximize LSP acceptance rates and minimize the number of active nodes and links while also considering additive QoS constraints. Here, a flow request is defined by a triple $<s, d, \vec{Q}>$ where $\vec{Q}$ is a vector representing QoS requirements such as requested bandwidth, delay, jitter, and packet loss. E^2-MCRA is built on the following key concepts: i) it includes a path length function, which allows traffic demands to be routed over paths with the minimum number of hops while minimizing the number of blocked LSP requests. This path length function guarantees that for each traffic demand, the path used is the one with bandwidth, delay, jitter, and packet loss characteristics

that meet specified QoS requirements, and ii) when selecting a path for a given source-destination pair, instead of exploring all paths exhaustively, the algorithm searches only sub-graphs induced by active routers and links. The best path for each $<s, d>$ pair is then found using a look-ahead and a depth-first search strategy. E^2-MCRA returns, among all the feasible shortest paths, the path that employs the least number of active links, hence, reducing energy wastage.

In [22], the authors introduce the online, distributed Green Backup Paths (GBP) algorithm that takes advantage of MPLS's failure protection mechanisms; i.e., using backup paths to protect failed links in primary LSPs [99]. Energy saving is achieved by re-routing traffic from protected links onto backup paths. Protected links are then shut down. Routers running GBP are aware of their local traffic conditions and cooperatively determine how to reroute traffic with the goal of shutting down as many links as possible. When diverting traffic from primary to backup paths, QoS constraints such as packet delays are considered by avoiding long backup paths. GBP is designed using two components: i) *offline*, which identifies in advance eligible backup paths according to their length, and selects one that has a low propagation delay. GBP concurrently and independently offloads traffic from logical links, i.e., with a bundle of physical links, by identifying the set of logical links that are affected, in terms of link utilization, after rerouting traffic that traverses a given link, and ii) *online*. This component periodically reroutes traffic from links on primary LSPs onto backup paths. The algorithm then proceeds to shut down idle links. In order to know where to divert traffic, routers need to have an updated and consistent view of the state of the network, which is achieved by collecting link state information from other routers.

The aforementioned works aim to minimize the number of network components used when routing arriving LSP requests. In addition, they consider QoS constraints such as maximum link utilization, packet loss, delay, and jitter. However, they do not consider energy savings and LSP acceptance rates jointly. Hence, Chapter 3

adds to the existing literature by introducing a novel metric in order to compare their attained energy savings versus LSP acceptance rates. Table 2.2 shows a summary of the reviewed green LSP establishment approaches. It contains their main characteristics and classifies them according to: i) the entities making routing decisions, centralized/distributed, and ii) knowledge of LSP requests, either online or offline.

Table 2.2: Summary of green LSP establishment methods

Green TE techniques	**Characteristics**	**Centralized /Distributed**	**Online /Offline**
GGMTES [25] [24]	Performs offline calculations and uses a least cost path method to determine the number of active elements	Centralized	Offline
Addis et al. energy model [97] [98]	Introduction of an energy model that allows a network operator to selectively shut down devices and links according to traffic load	Distributed	Offline
DAISIES [23] [24]	Dynamically reroutes traffic demands according to link load	Centralized	Online
(E^2-MCRA) [21]	LSPs are established based on a path length function. Employs Look-Ahead and DFS strategy	Distributed	Online
GBP [22]	Takes advantage of a MPLS failure protection mechanism and establishes LSPs in the resulting backup paths	Distributed	Online

2.2.3 Miscellaneous

This section presents green TE techniques that employ strategies such as traffic aggregation, failover methods, and power consumption models that take into account traffic conditions in order to reduce energy usage. Except for reference [81], these TE techniques are not tied to any particular protocol or network, which make them amenable to implementation on any networks. The authors of [100] propose the distributed Energy Aware TE (EATe) algorithm, whereby energy consumption is reduced by spreading traffic among multiple paths. Each intermediate router monitors the link utilization of its incident links and reports this utilization to edge

routers, which use this information to determine the distribution of traffic across multiple paths. EATe defines three different approaches to conserve energy: *i)* rate adaptation. Energy consumption is directly proportional to link rates. The goal is to reroute and aggregate traffic. This ensures links on selected paths have minimal utilization, *ii)* putting links to sleep. This approach aims to aggregate traffic onto as few links as possible and thereby allowing other links to sleep. To this end, it defines two energy levels: low and high. By setting the energy level to a low value, there will be more active links. These active links therefore will have enough capacity to absorb any sudden increase in traffic. On the other hand, if the energy level is high, traffic demand will be routed over a smaller number of links. This thus helps maximize the number of links that are powered off, and iii) putting routers to sleep. This third approach considers placing routers to sleep. Here, every router calculates its total link utilization and reports this value to every node that is sending traffic to it. A node then uses this information to determine the routers with lowly utilized incident links. Traffic over these links is then rerouted, and idle links are then placed into sleep mode. The main advantages of EATe include the ability to aggregate traffic onto a small number of links and the possibility of waking up sleeping links when necessary. In addition, it allows traffic to be rerouted onto alternative paths when a link failure is detected. The main disadvantage of EATe is its dependency on hardware that supports sleeping and rate-adaptation.

Athanasiou et al. [18] propose Energy-Aware Traffic Engineering (ETE), a distributed and offline algorithm that load-balances traffic while minimizing energy consumption. ETE aims to minimize the maximum link utilization and consists of the following algorithms: i) load balancing. Each source node finds the amount of traffic destined to a given destination node and determines how this traffic needs to be load balanced over its incident links in order to maximize link utilization, and ii) energy saving. Each ingress node takes the traffic information found by the load balancing algorithm and uses it to calculate the minimum number of links that can

carry that traffic. The main advantage of ETE is that it allows each ingress node to adaptively load balance traffic directed to a particular egress node. A key limitation is its inability to exploit varying traffic conditions.

Cuomo et al. [101] study robustness again link failures when reducing the energy consumption of a network. Here, the authors suggest a topology aware algorithm called Energy Saving based on Algebraic CONnectivity (ESACON) that relies on the algebraic connectivity of a network. Briefly, algebraic connectivity is defined as the second smallest eigenvalue of the Laplacian matrix of a given graph. The Laplacian matrix is obtained from the difference between the degree matrix, which is a diagonal matrix containing the number of edges attached to each vertex, and the adjacency matrix, which describes the connection between nodes. Algebraic connectivity is known to be a good indication of a graph's robustness to node and link failures [102]. Links are first sorted based on their impact on the network's algebraic connectivity and ESACON removes those with little impact. The main advantage of ESACON is its low complexity and fast execution time, which facilitates its implementation in both centralized and distributed scenarios. It, however, is not traffic-aware. In other words, it only uses the topological information when disabling links.

Reference [103] investigates existing failover methods in order to reduce energy wastage. The authors present a green TE scheme that aims to eliminate packet loss due to route instability when traffic is rerouted from links that have been selectively put into sleeping mode. They focus on IP networks that implement the Fast Reroute (FRR) NotVia technique [104]. NotVia is a path delegation approach that diverts traffic from failed links onto pre-computed shortest paths, and thereby, minimizes adverse effects associated with traffic re-routing. The authors propose an offline algorithm that determines the links to be included in the Scheduled Sleeping Link Group (SSLG) set. In particular, the traffic traversing links in the SSLG set will be rerouted onto alternate shortest paths during off-peak periods. The offline algorithm constructs the SSLG set as follows. It identifies and creates a list of candidate links.

Candidates links are lowly utilized links containing traffic that can be rerouted onto alternative shortest paths; aka NotVia paths. The algorithm then checks: i) that the utilization of the links on alternate paths are within a given threshold, and ii) the resulting topology is connected. If the foregone conditions are met, a candidate link is included in the SSLG set. Otherwise, the next link is analyzed. The process repeats until all links are explored. The main advantage of NotVia is its ability to automatically reroute traffic onto alternate paths in case of link failures. This is because NotVia will treat sleeping links as failed links. However, the authors have not considered failures during off-peak times. That is, non-SSLG links are not protected by an alternative path. These off-peak failures are considered by the authors as a future research topic.

Power consumption models are considered in references [105] and [106]. The authors formulate the problem of minimizing the total energy consumption of a network as a variant of the well-known, NP-hard Capacitated Multi-Commodity Minimum Cost Flow (MCMF) problem [107]. To address this problem, they assume a power consumption model whereby routers consume one unit of power, and links have negligible energy expenditure. The authors also propose a heuristic that operates as follows. Routers are sorted in ascending order based on one of the following criteria: randomly, according to the number of incident links, as per the number of flows traversing them, or in terms of the number of edge-nodes associated to a given source router that can be powered off. An edge-node is a router that acts as a sink for several source routers. Note that each source router should have at least one active edge-node. Once sorted, the heuristic selects the highest ranked router to be shut down. Traffic that traverses the chosen router is then rerouted onto alternative shortest paths for a given $<s, d>$ pair. This ensures only the minimal number of routers carries the rerouted traffic load, and hence maximizes energy savings. In [105] the authors improved the heuristic in [106] by sorting routers according to their power usage so the one with the highest power consumption will be shut down

first. Another improvement over [106] is using power values reported by a large Italian ISP. In both [105] and [106], the main disadvantage is their dependency of hardware and network protocols that support shutting down links or/and nodes.

Researchers have also considered different power consumption models. For example, the authors of [108] propose a centralized algorithm that employs a Path Computation Element (PCE) to continuously monitor traffic. The PCE dynamically decides which links are to be powered on/off. The objective is, therefore, to find the network configuration that consumes the least amount of energy within a search space that contains all possible combinations of nodes and active/sleeping links. Given the computational complexity of searching all combinations, the authors present a heuristic to reduce the search space by generating all patterns for a given topology. A pattern is defined as a version of the original topology but with sleeping links; the authors refer to the number of sleeping links as the depth of a network. The heuristic employs an algorithm that receives as initial input the original topology and provides as output a local solution, representing the pattern that satisfies a given set of QoS constraints. At each iteration, the algorithm takes as input the previous output, and iterates until none of the solutions satisfy the set of QoS constraints. Once the algorithm has converged, the PCE reconfigures the network according to the pattern in the final output. Note that the value of depth increases for large topologies.

Table 2.3 summarizes the aforementioned miscellaneous approaches. The table provides a brief description for each of the approaches, and their classification according to: i) the entity(s) making the routing decisions, and ii) the strategy they use in order to reduce energy consumption. The presented approaches are not tied to any particular technology or network. In addition, works such as [101] and [103] take advantage of existing link failure prevention mechanisms, such as [104], to reduce energy consumption and minimize negative effects such as link overload and packet loss due to rerouting of traffic. Lastly, power consumption models are presented in

[105], [106], and [108]. However, the main disadvantage of these approaches is their dependency on hardware that is capable of supporting sleeping and rate adaptation.

Table 2.3: Summary of miscellaneous TE techniques

Green TE Techniques	Characteristics	Centralized /Distributed	Strategy
EATe [100]	Energy usage is reduced by achieving a uniform distribution of traffic load	Centralized	Traffic aggregation
ETE [18]	Energy savings are achieved via load balancing	Distributed	Traffic aggregation
ESACON [101]	The concept of algebraic connectivity is used to decrease energy wastage	Centralized /Distributed	Failover mechanism
Green TE with FRR NotVia [103]	This approach minimizes negative effects such as link overload and packet loss due to rerouting of traffic	Distributed	Failover mechanism
Chiaraviglio et al. algorithm [106] [105]	Traffic demands are routed using the minimum number of network components	Centralized	Power consumption model
PCE [108]	A central entity decides which links remain active. The search space is reduced by only considering versions of the original network with varying number of sleeping links	Centralized	Power consumption model

2.3 Summary

This chapter has presented a review of TE techniques. In particular, it has discussed: i) *non-green approaches*. These are TE techniques that aim to satisfy one or more pre-established QoS constraints such as end-to-end delay, maximum link utilization, and packet loss, and ii) *green approaches*. These TE techniques are motivated by the high energy consumption rate of current networks. Moreover, the energy expenditure of the Internet is projected to continue to grow significantly with increasing volume of traffic and number of devices in the coming years. Hence, green approaches will be critical, especially those that consolidate traffic demands onto the minimal number of active links/routers/switches, which in turn help place unused links/routers/switches into sleep mode.

Current methods, however, have a number of weaknesses. First, in MPLS networks, LSP establishment methods focus on routing traffic demands over the minimum number of network components. Unfortunately, existing works neglect LSP acceptance rates. Furthermore, there is little or no work that provides a thorough comparison of online and offline LSP establishment methods. Second, controller placement works do not consider energy efficiency when associating switches to a controller. In general, there is little energy aware works in the context of SDNs. Third, existing works do not consider the impact on BGP. This is critical as shown in Chapter 5 where any links/routers that are shut down will cause changes in IGP cost, and subsequently, trigger the negative impacts reported in [37] and [38]. Hence, there is a critical need for green techniques that are cognizant of BGP behaviors. Fourth, green TE techniques have yet to consider uncertainty in traffic demands. In fact, existing TE approaches allocate resources according to peak demands. This constitutes a significant gap in the literature as traffic demands are random in practice and thus any links/routers/switches that are shut down must ensure the remaining network elements are capable of absorbing any fluctuation in demands.

Chapter 3

Online and Offline Green Path Establishment Techniques

From Chapter 2, we see that significant effort has gone into designing green TE techniques that consolidate traffic onto the minimal number of links/switches/routers during off-peak periods. However, little works exist that aim to save energy in MPLS networks. Critically, to date, there is only one preliminary work on the performance of green LSPs establishment methods in terms of energy savings and acceptance rates. Henceforth, this chapter presents a comprehensive study of two offline and four online LSP establishment methods. Online methods rely only on past and current LSP requests whilst offline ones act as a theoretical benchmark whereby they also have available to them future LSP requests. This chapter also introduces a novel metric that takes into account both energy savings and acceptance rates. Lastly, it identifies a new and simpler heuristic that minimizes energy usage by routing source-destination demands over links with other paths and also requiring

the fewest number of new links. In a nutshell, this chapter makes the following contributions:

1. It compares the LSP acceptance rates of existing online and offline energy aware algorithms for establishing LSPs. It studies (i) *offline* approaches, where the complete set of LSP setup requests are known in advance, and (ii) *online* approaches, where only the current and past LSP setup requests are known. In fact, this is the first study that compares all these approaches over the same topologies. Moreover, it proposes an Integer Linear Program (ILP) formulation for the offline version of the problem.

2. It presents and studies the performance of two offline and four online heuristics: i) Offline Most Overlapped (*Offline-MO*) [22], a technique that aims to use paths that share the most links with past or/and future LSP requests, ii) Offline with Ratio (*Offline-R*), which is similar to *Offline-MO* but favors paths that require fewer number of new links, iii) Online Most Overlapped (*Online-MO*), which is similar to its offline counterpart, is an algorithm that uses paths that share the most links with already established links, iv) Online with Ratio (*Online-R*) , which is similar to *Online-MO* but prefers paths that involve the minimal number of new links, v) Online Minimum Hops (*Online-MinH*) [21], an approach that gives priority to paths with a small number of hops, and vi) Online Random LSP (*Online-R-LSP*) [109] which selects paths randomly.

3. It introduces and uses a ratio, called ρ, of the *percentage of shut down links* (PSL) and *LSP acceptance requests* (LAR) to quantify how well a method performs in terms of energy savings and its ability to accept new LSP requests. This metric ρ allows us to evaluate whether a solution that is able to accept a larger number of LSP requests also has significant energy savings. Extensive experiments involving well known topologies such as Abilene and AT&T using

varying traffic loads confirms that LSP acceptance rates above 90% are feasible with 20% of links shut down.

3.1 Problem Description

The network is modelled as a directed graph $G(V, E)$, with V being the set of $|V|$ nodes, and E representing the set containing $|E|$ edges. The link between node i and j is denoted as e_{ij} or (i, j). Each link has capacity c_{ij} and utilization u_{ij}. Let Q be the set of LSP establishment requests that arrive at the set of ingress routers $I \subset V$. Each LSP establishment request $q \in Q$ is a tuple $<s, d, bw>$ where s and d denote the source and destination of a request, and $bw > 0$ is the requested bandwidth. Let $B(q)$ be a function that returns the bandwidth of request q. Let P_q be the set of all simple paths that can be used to serve LSP request $q \in Q$. Specifically, $P_q = \{p_q^1, p_q^2, \ldots p_q^{|P_q|}\}$ is a set of candidate paths for q sorted in increasing path length order. Each path p_q^k in P_q contains a set of $|p_q^k|$ links, meaning $p_q^k \subseteq E$. The set of paths that use link (i, j) is defined as $P_{ij} = \{p_q^k \mid e_{ij} \in p_q^k\}$, for all $q \in Q$ and $k = 1, 2, \ldots, |P_q|$. Hence, the total traffic over a given link (i, j) is $B_{ij} = \sum_{p \in P_{ij}} B(p)$ and its link utilization is $u_{ij} = B_{ij}/c_{ij}$.

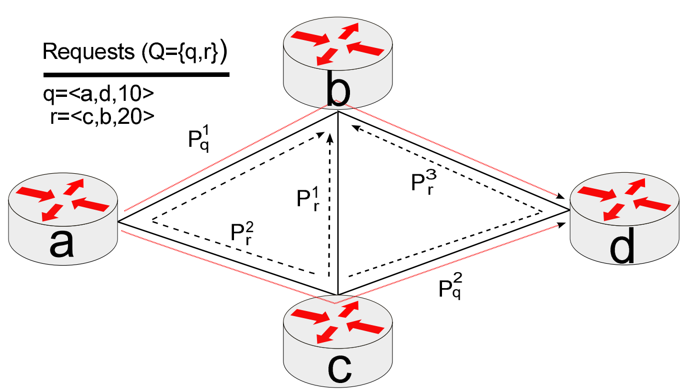


Figure 3.1: Example network. A few paths have been omitted in order to avoid cluttering the figure and the explanation.

The problem at hand is as follows: given i) a MPLS network consisting of Label Switch Routers (LSRs) and directional links with fixed capacity, ii) traffic demands

described as a set of tuples $<s, d, bw>$ which may be given a priori, i.e., offline, or in a real time manner, i.e., online, the problem at hand is to minimize the overall energy consumption of the MPLS network by finding a set of LSPs that satisfy the given traffic demand of each request using the minimal number of links/routers.

It is worth noting that in the online version of the described problem, the establishment of current LSPs affects the utilization of links/routers and hence may affect the acceptance of future LSP requests. The challenge is therefore to assign LSPs such that energy usage is reduced, while accommodating future traffic demands. The said problem can be modeled as an ILP.

In the sequel, a mathematical model of the offline version of the problem is presented. Let $X_q^k \in \{0, 1\}$ be a binary variable that represents whether path k of request q, i.e., p_q^k, is selected. As an example, consider Figure 3.1 with the request $q = <a, d, 10>$ and two alternative paths: p_q^1 and p_q^2. Note that only one of the two paths can be selected, and thus, either ($X_q^1 = 1$ and $X_q^2 = 0$) or ($X_2^q = 0$ and $X_q^2 = 1$) is set. In general, we have the following constraint

$$\sum_{k=1}^{|P_q|} X_q^k = 1, \quad \forall q \in Q \tag{3.1}$$

The next constraint ensures the load is within link capacity. Let $\bar{T_{ij}}$ be the set of decision variables X_q^k that represent the paths that are using link e_{ij}; i.e., $\bar{T_{ij}} = \{X_q^k | e_{ij} \in p_q^k\}$, where k is an index corresponding to path k of request q that uses edge e_{ij}. For example, assuming request $q = <a, d, 10>$ and $r = <c, b, 20>$ link (a, b) of Figure 3.1 will have $\bar{T_{ab}} = \{X_q^1, X_r^2\}$. Therefore, we have the following link capacity constraint,

$$\sum_{\chi \in \bar{T_{ij}}} \chi B(\chi) \leq c_{ij}, \quad \forall e_{ij} \in E \tag{3.2}$$

where the function $B(\chi)$ returns the bw value associated with the path/request corresponding to decision variable χ. As an example, consider Figure 3.1 with

demands $q =<a, d, 10>$ and $r =<c, b, 20>$. The capacity constraint for link (a, b) is therefore $X_q^1 \times 10 + X_r^2 \times 20 \leq 100$, with $c_{ab} = 100$.

The final set of constraints ensure that a link or router remains active only if it is used by a path. Let X_{ij} and Y_v be binary variables that denote whether a link $(i, j) \in E$ and router $v \in V$ is active or inactive, respectively. In addition, the function $F(\bar{v})$ returns the X_{ij} of incident links on router v. To ensure that a link or router is active only if there is a path that uses it, we have the following constraint

$$X_{ij} \succeq \bar{T_{ij}}, Y_v \succeq F(\bar{v}) \tag{3.3}$$

where $\succeq$ represents the $\geq$ operator executed component wise on the set $\bar{T_{ij}}$ or $F(\bar{v})$; referring to Figure 3.1, link e_{ab} with decision variable X_{ab} will have the following constraints: $X_{ab} \geq X_q^1$ and $X_{ab} \geq X_r^2$. That is, the decision variable X_{ab} is set to one only if X_q^1 or X_r^2 or both are set to one.

With constraints (3.1), (3.2) and (3.3) in hand, the objective to optimize is as follows,

$$\text{MIN} \sum_{e_{ij} \in E} X_{ij} + \sum_{v \in V} Y_v \tag{3.4}$$

In other words, the aim is to minimize the number of active links and routers. The objective function can also be adapted to include the specific power consumption of a NIC. This is left as a future work.

The aforementioned offline version of the problem is solvable only for small networks due to the number of binary variables that grow exponentially with network size and demands. In particular, there could be an exponential number of paths that can be used for a given demand q. In fact, the offline version of the problem corresponds to the well-known Multi-Commodity Minimum-Cost Flow (MCMF) problem and is therefore NP-complete; please refer to [110] for details. Henceforth, the next section presents different heuristics to address both online and offline version of the formulated problem.

3.2 Heuristics

In the discussion to follow, Figure 1.1 is used to describe the offline and online heuristics of interest. Moreover, the LSP requests that arrive over time are shown in Table 3.1. The table also shows their respective k shortest paths. These paths can be calculated using Yen's k-Shortest paths algorithm [111].

Table 3.1: LSP setup requests and k paths shared by all implemented algorithms

LSP_q	s	d	bw	k **Shortest Paths**		
1	R3	R1	14	$[R3, R1]$	$[R3, R2, R1]$	$[R3, R4, R2, R1]$
2	R4	R2	41	$[R4, R2]$	$[R4, R3, R2]$	$[R4, R3, R1, R2]$
3	R2	R4	40	$[R2, R4]$	$[R2, R3, R4]$	$[R2, R1, R3, R4]$

In the following sections, the following heuristics are described in detail:

Offline

- Offline Most Overlapped (*Offline-MO*): aims to use paths that share the most links with past or future LSP requests.

- Offline with Ratio (*Offline-R*): same as *Offline-MO* but favoring paths that require fewest number of new links.

Online

- Online Most Overlapped (*Online-MO*): aims to use paths that share the most links with already established LSPs.

- Online with Ratio (*Online-R*): same as *Online-MO* but prefers paths that involve the minimal number of new links.

- Online Minimum Hops (*Online-MinH*): gives priority to paths with a small number of hops.

- Online Random LSP (*Online-R-LSP*): selects paths randomly.

Note that *Online-MinH* and *Online-R-LSP* have been considered in [21] and [109], respectively. However, all other heuristics are new. Moreover, as noted in Section 2.1.1, no works have compared all these heuristics comprehensively. As Section 3.4 reports, *Online-R* has the best performance in terms of energy saved and LSP acceptance rate.

3.2.1 Offline Approaches

Algorithm 1 presents a general overview of how offline heuristics are applied to each LSP request $q \in Q$. By definition, all these heuristics know in advance all LSP setup requests in Q, and their respective k shortest paths P_q. This means they can determine the best links to use or avoid by looking at past, current and future LSP requests. Hence, the results obtained via offline heuristics constitute the best possible performance for any online heuristics. For the reader's convenience, Algorithm 1 also defines the variables used by the different $Heuristic(.)$ functions. For each arriving LSP request, $q = <s, d, bw>$ the set of all shortest (s, d) paths is generated. $Heuristic(.)$ then processes all the (s, d) paths and returns a candidate path to serve the LSP request. If $Heuristic(.)$ returns multiple paths, the algorithm selects the one with the fewest number of hops; if there is a tie, the first path is selected. If all the links on the selected candidate path are able to meet the required bandwidth demand, the path is assigned to (s, d). The algorithm then subtracts the requested demand from the available bandwidth, see line-9, of each link on the established LSP and each of these links are marked as active permanently. On the contrary, if the selected candidate path is not able to serve the requested demand, $Heuristic(.)$ evaluates the remaining paths of q. If no paths with sufficient bandwidth is found, it rejects LSP request q and moves to the next one.

Algorithm 1: Pseudocode for offline heuristics

Var:
cand_path: a variable that stores a candidate LSP
links_used: an array that stores used links
a: number of accepted requests
r: number of rejected requests

$a = r = 0$, $links\_used = \emptyset$
Generate set Q
for each q =<s,d,bw> $\in Q$ **do**
  Generate P_q
  while $P_q \neq \{\}$ **do**
    $cand\_path \leftarrow Heuristic(P_q)$
    if all links in $cand\_path$ satisfy $c_{ij} \geq bw$ **then**
      $LSP \leftarrow cand\_path$
      $UpdateBandwidth(LSP, bw)$
      $links\_used \cup \{e_{ij} \mid e_{ij} \in cand\_path\}$
      `a++`
    else
      $P_q - \{cand\_path\}$
      if $P_q = \{\}$ **then**
        `r++`
      end if
    end if
  end while
end for

3.2.1.1 Offline-MO

The goal is to select paths that share the most links. *Offline-MO* compares each p_q^k of a given request q with the candidate paths of other requests in Q. For each p_q^k, where $k = 1, 2, .., |P_q|$, the function $Heuristic(.)$ finds and stores the number of matching links in a variable *score* ≥ 0 that gives the total number of its links that are in common with paths for other requests. *Offline-MO* selects p_q^k that has the maximum *score* value. The links within the chosen p_q^k are then added into *links_used*. If a given p_q^k has insufficient bandwidth, which depends on the MLU of the different links composing that path, it is removed from P_q.

An example is presented in Table 3.2. Here, it is only shown how *Offline-MO* selects the candidate LSP for request 1; i.e., LSP_1 (<R3, R1, 14Mb>). Other LSP

requests are processed in the same way. From Table 3.1, the generated paths in P_q for LSP_1 are: $p_1^1 = [R3, R1]$, $p_1^2 = [R3, R2, R1]$, $p_1^3 = [R3, R4, R2, R1]$. Each cell of Table 3.2 contains the links of candidate paths that belong to other requests; e.g., the second candidate path of LSP_2, p_2^2 is $[R4, R3, R2]$ and its links are $R4 - R3$ and $R3 - R2$. Next to each link is a label that indicates whether it is used by a path belonging to LSP_1. For example, the path $[R4, R3, R2]$ belonging to LSP_2 has the link $R3 - R2$ in common with the second candidate path, i.e., p_1^2 of LSP_1, i.e., $[R3, R2, R1]$. This explains the "p_1^2" below link $R3 - R2$. Note that links can appear in multiples paths of a given request as it is the case for $R2 - R1$, which is shared by p_1^2 and p_1^3.

Table 3.2: *Offline-MO* example. LSP_1 has a request <R3, R1, 14Mb>, and three candidate paths: $[R3, R1], [R3, R2, R1], [R3, R4, R2, R1]$.

LSP_q	p_q^1 **links**	p_q^2 **links**		p_q^3 **links**		
2	$R4 - R2$	$R4 - R3$	$R3 - R2$	$R4 - R3$	$R3 - R1$	$R1 - R2$
	p_1^3		p_1^2		p_1^1	
3	$R2 - R4$	$R2 - R3$	$R3 - R4$	$R2 - R1$	$R1 - R3$	$R3 - R4$
			p_1^3	p_1^2		p_1^3
				p_1^3		

Table 3.2 presents the *score* for each of the P_q paths for LSP_1. For example, for p_1^1, i.e., $[R3, R1]$, its score is 1, given that p_1^1 appears one time. The score for $[R3, R2, R1]$ is 2 and the score for $[R3, R4, R2, R1]$ is 4. This means the $Heuristic(.)$ function for LSP_1 returns $[R3, R4, R2, R1]$ as the path that has the highest overlap, and therefore, is chosen by *Offline-MO*.

3.2.1.2 Offline-R

Similar to *Offline-MO*, *Offline-R* aims to use paths that have as many common links as possible to other paths and additionally, gives preference to the ones that require the fewest number of new links to be set up. Note, "new links" are defined as those that are not carrying any traffic, i.e., not in *links_used*. In order to do this, this heuristic reuses the *score* variable from the *Offline-MO* algorithm and introduces

the variable $Ratio_{\text{off}}$ for each of the p_q^k paths of a given request q. The said variable is defined as *score/new_links_number*, where *new_links_number* stores the number of new links that would have to be established if path p_q^k is selected. The function $Heuristic(.)$ calculates the $Ratio_{\text{off}}$ for each of the p_q^k paths and selects one with the maximum $Ratio_{\text{off}}$ value. In the special case when *new_links_number* is equal to zero, i.e., all the links in a given p_q^k path already exist, the variable is set to one. This is to avoid division by zero.

Table 3.3 presents an example with two LSP requests: LSP_1, <R3, R1, 14Mb>, and LSP_2, <R4, R2, 41Mb>. Each of them has three candidate paths $[R3, R1]$, $[R3, R2, R1]$, $[R3, R4, R2, R1]$ and $[R4, R2]$, $[R4, R3, R2]$, $[R4, R3, R1, R2]$ respectively. The *score* variable is calculated similarly to the *Offline-MO* heuristic. For example, consider the first arriving LSP request, LSP_1, with path $[R3, R4, R2, R1]$, and *score* value of 4; see Table 3.2. Given that there is no established link, the variable *new_links_number* will be set to 3 as links $R3 - R4$, $R4 - R2$ and $R2 - R1$ need to be set up. Therefore, the value of $Ratio_{\text{off}}$ is $4/3 = 1.33$. The other two paths of LSP_1, i.e., $[R3, R1]$, and $[R3, R2, R1]$, will also need new links to be established, and their $Ratio_{\text{off}}$ value is 1/1=1 and 2/2=1. $Heuristic(.)$ will then return path $[R3, R4, R2, R1]$ as the candidate path given that its $Ratio_{\text{off}}$ is the maximum among these three paths. For LSP_2 and subsequent LSP requests, paths needing more new links to be set up will have less chance of being selected as LSP candidates.

Table 3.3: *Offline-R* example. LSP_1 requests <R3, R1, 14Mb>, and LSP_2 requests <R4, R2, 41Mb>. There are the following candidate paths: $[R3, R1]$, $[R3, R2, R1]$, $[R3, R4, R2, R1]$ and $[R4, R2]$, $[R4, R3, R2]$, $[R4, R3, R1, R2]$, respectively

LSP_q	***links_used***	k **paths**	*score*	***new_links_number***	$Ratio_{\text{off}}$	**Selected path** k
1	{}	$[R3, R1]$	1	1	1	$[R3, R4, R2, R1]$
		$[R3, R2, R1]$	2	2	1	
		$[R3, R4, R2, R1]$	4	3	1.33	
2	$R3 - R4$	$[R4, R2]$	1	0	1	$[R4, R2]$
	$R4 - R2$	$[R4, R3, R2]$	0	2	0	
	$R2 - R1$	$[R4, R3, R1, R2]$	1	3	0.33	

3.2.2 Online Approaches

Algorithm 2 presents the pseudocode for the online heuristics. Note that this pseudocode is similar to the pseudocode presented in Algorithm 1. The difference is that by definition, online approaches only have knowledge of the current and past LSP requests.

Algorithm 2: Pseudocode for online heuristics

```
Var:
cand_path: a variable that stores the candidate LSP
links_used: a table that stores the links used
a: number of accepted requests
r: number of rejected requests
Request q =<s, d, bw>arrives
Generate P_q
while P_q ≠ {} do
   cand_path ← Heuristic(P_q)
   if all links in cand_path satisfy c_ij ≥ bw then
      LSP ← cand_path
      UpdateBandwidth(LSP, bw)
      links_used ∪ {e_ij | e_ij ∈ cand_path}
      a++
   else
      P_q − {cand_path}
      if P_q = {} then
         r++
      end if
   end if
end while
```

3.2.2.1 Online-MO

Online-MO is similar to its offline version. It selects a path p_q^k with links that overlap the most with existing links. $Heuristics(.)$ calculates for each p_q^k the number of links in common with already established links and stores this in the *num_used_link* variable. The p_q^k path with the maximum number of links in common is selected as the candidate LSP. The variable *num_used_link* indicates path p_q^k contains at least one link that is already established and therefore, can be reused.

Table 3.4 shows an example for the first two arriving LSP requests. For LSP_1, *num_used_link* is zero initially for paths $[R3, R1]$, $[R3, R2, R1]$ and $[R3, R4, R2, R1]$. Here, $Heuristic(.)$ will break the tie by returning the shortest path $[R3, R1]$ as the candidate path. This is not the case for LSP_2 given that $Heuristic(.)$ will consider that link $R3 - R1$ has been setup. The paths for LSP_2 are $[R4, R2]$, $[R4, R3, R2]$, and $[R4, R3, R1, R2]$. Their corresponding *num_used_link* value is 0, 0 and 1 respectively, given that path $[R4, R3, R1, R2]$ is the only one that can reuse link $R3-R1$. Therefore, $Heuristic(.)$ will return as candidate path $[R4, R3, R1, R2]$.

Table 3.4: *Online-MO* example. LSP_1 <R3, R1, 14Mb> and LSP_2 <R4, R2, 41Mb>

LSP_q	***links_used***	k **paths**	***num_used_link***	**Selected path** k
1	{}	$[R3, R1]$	0	$[R3, R1]$
		$[R3, R2, R1]$	0	
		$[R3, R4, R2, R1]$	0	
2	$R3 - R1$	$[R4, R2]$	0	$[R4, R3, R1, R2]$
		$[R4, R3, R2]$	0	
		$[R4, R3, R1, R2]$	1	

3.2.2.2 Online-R

This heuristics is similar to its offline counterpart; i.e., *Offline-R*. The objective here is to reduce energy consumption by utilizing established links and additionally, favoring paths that require fewest new links. Note that this approach is similar to that of [22]. Specifically, for the routing of a given (s, d) pair demand, the technique in [22] uses an existing shortest backup path. It aims to minimize the establishment of new links. In contrast, *Online-R* does not only consider backup paths of a given (s, d) pair paths but considers the shortest paths used to route demands for other (s, d) pairs. This helps reduce the need to establish new links. The heuristic reuses the term *num_used_links* from the *Online-MO* algorithm and *new_links_number* from *Offline-R* and introduces the term $Ratio_{\text{on}}$ as the ratio *num_used_links/new_links_number*. For each p_q^k path of the current request q, $Heuristic(.)$ calculates its $Ratio_{\text{on}}$ and then selects as the candidate LSP p_q^k whose

$Ratio_{\text{on}}$ is maximum. In the special case when *new_links_number* is equal to zero, i.e., all the links in a given p_q^k paths have already been established, the variable is set to one. This is to avoid division by zero.

Table 3.5 describes how *Online-R* calculates the candidate LSP for the first two LSP requests. Notice that paths that require more new links to be setup have a lower probability of being selected as a candidate LSP, whilst those that need fewer new links are preferred. As an example, any of the paths in LSP_1, i.e., $[R3, R1]$, $[R3, R2, R1]$ and $[R3, R4, R2, R1]$, will require all their links to be setup. Therefore, their *num_used_links* value will be zero, and their ratio will also be zero. $Heuristic(.)$ will break the tie by selecting the shortest path $[R3, R1]$. LSP_2 is now considered. Its paths, i.e., $[R4, R2]$, $[R4, R3, R2]$, and $[R4, R3, R1, R2]$, will have a *num_used_links* value of 0, 0, and 1, respectively. The value of *new_links_number* for each of candidate path of LSP_2 can be found by counting the links that are not included in the **"Links used"** column. Specifically, the corresponding *new_links_number* value for $[R4, R2]$, $[R4, R3, R2]$, and $[R4, R3, R1, R2]$ is 1, 2, and 2, respectively. Given the value of *num_used_links* and *new_links_number*, $Ratio_{\text{on}}$ can be calculated and $Heuristic(.)$ returns the candidate path with highest value. In this case, path $[R4, R3, R1, R2]$ is selected.

Table 3.5: *Online-R* example. LSP_1 has request <R3, R1, 14Mb> and LSP_2 has request <R4, R2, 41Mb>

LSP_q	***links_used***	k **paths**	***num_used_links***	***new_links_number***	$Ratio_{\text{on}}$	**Selected path** k
1	{}	$[R3, R1]$	0	1	0	$[R3, R1]$
		$[R3, R2, R1]$	0	2	0	
		$[R3, R4, R2, R1]$	0	3	0	
2	$R3 - R1$	$[R4, R2]$	0	1	0	$[R4, R3, R1, R2]$
		$[R4, R3, R2]$	0	2	0	
		$[R4, R3, R1, R2]$	1	2	0.5	

3.2.2.3 Online-MinH

This heuristic, which is also reported in [21], chooses the p_q^k path of the current request q with the minimum number of hops.

3.2.2.4 Online-R-LSP

The $Heuristic(.)$ for *Online-R-LSP* randomly selects one of the p_q^k paths in the set P_q for the current request q. Note that random path selection is essentially similar to Equal-Cost Multipath (ECMP), as used by CSPF [109].

3.3 Evaluation

The performance of the aforementioned heuristics is evaluated using two popular topologies: Abilene and AT&T North America [112] [113]. The Abilene network consists of 11 nodes and 28 directional links, whereas the AT&T network consists of 25 nodes and 112 directional links.

The simulations are conducted in MATLAB [114]. The three components of a LSP request $<s, d, bw>$ are generated randomly as follows: i) s and d are set to an integer from the range $[1, |V|]$, where $s \neq d$, ii) bw is a value in $[1, BW_{\text{Max}}]$. LSP requests are generated in advance in both online and offline scenarios. It is assumed that when all links are active, all these LSP requests can be admitted.

Algorithm 3 describes the procedure used for all simulations. Please note Steps 4 and 7. These steps show the calculations performed by $Heuristic(.)$ for a given set of LSP requests Q. In particular, for all requests q in Q, $Heuristic(.)$ needs to explore $|P_q|$ shortest paths; each of them with a maximum length of $|V|$ hops. Therefore, the presented algorithms have a running time complexity of $O(|P_q||Q||V|$.

In order to measure the goodness of a solution, a new metric is defined, $\rho = PSL/(100 - LAR)$. Recall that PSL is the percentage of shut down links and LAR is the LSP acceptance rate. Consider two green LSP methods: LSP_A and LSP_B.

Assume both can shut down the same number of links. For instance, $PSL_A = PSL_B = 40\%$. However, let's assume they have an LSP acceptance rate of $LAR_A = 90\%$ and $LAR_B = 80\%$, respectively. Therefore, $\rho_A = 4$ and $\rho_B = 2$. Given that $\rho_A > \rho_B$, it can be concluded that LSP_A is better than LSP_B. Consider a second example. Let's assume that $PSL_A = 30\%$ and $PSL_B = 40\%$, and both have the same LAR, say $LAR_A = LAR_B = 70\%$. Therefore, $\rho_A = 1$ and $\rho_B = 1.3$, and $\rho_B > \rho_A$. In this case, LSP_B is better because it is able to shut down a larger number of links while keeping the same LSP acceptance rate. Please note that when a green approach attains LAR=100%, its ρ will go to infinity. In this case, ρ is set to PSL.

This study conducted 30 simulation runs for each of the heuristics discussed in Section 3.2 using the following number of arriving LSP requests ($|Q|$): 50, 300, 500, 700, 1000, and 2000. In order to simulate different network loads, for each $|Q|$ value, BW_{Max} is set to 50, 200, 400, 600 and 1000 Mb/s. The results are within 95% confidence interval. Finally, ρ is computed for each of the evaluated approaches.

Algorithm 3: Pseudocode for different simulation.

```
Generate Q
if heuristic is offline then
    Call offline Heuristic(.) with Q as its input.
    Assign or reject the request q according to offline Heuristic(.), and compute
    the accepted/rejected LSP requests.
else
    Call online Heuristic(.) with current request q as its input
    Assign or reject the request q according to online Heuristic(.), and compute
    the accepted/rejected LSP requests.
end if
for each link(i, j) do
    Calculate u_ij
end for
Calculate the LSP acceptance rate
Calculate the number of active links in the final topology
Compute ρ
```

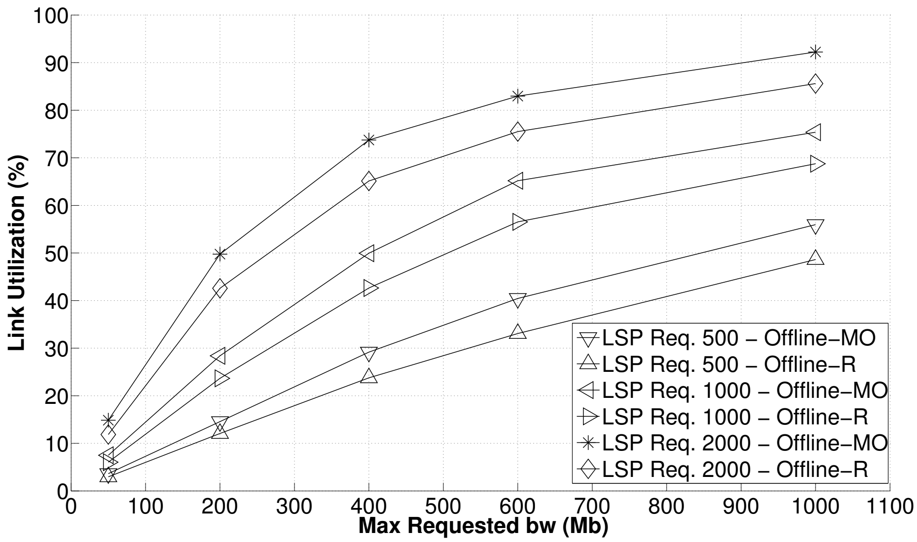
3.4 Results

In the following sections, note that "low" network load refers to scenarios with no more than 300 LSP requests and their max requested bandwidth is less than or equal to 200Mb. Conversely, the term "high" network load refers to scenarios where the number of LSP requests is at least 1000 and their max requested bandwidth is greater than or equal to 600Mb.

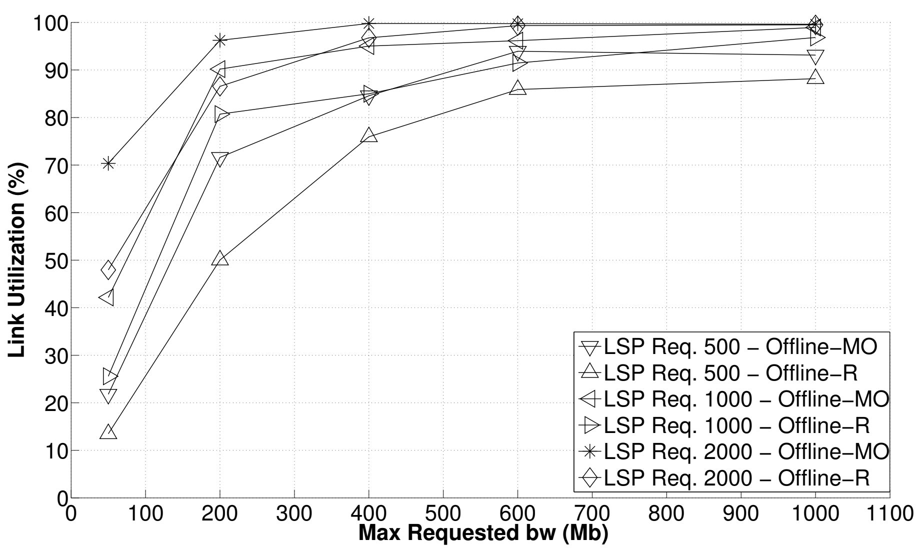
3.4.1 Offline Approaches

Figure 3.2 present the average link utilization of the two proposed offline heuristics for the AT&T and Abilene networks. As expected, under low network load, link utilization is similar for both approaches as few links are used and similar LSPs are selected. For both topologies, *Offline-R* presents the best performance with an average link utilization of 28.18% and 59.4% for AT&T and Abilene respectively. This is due to its ability to reuse established links, which allows it to obtain the lowest overall link utilization, whereas *Offline-MO* presents an average link utilization of 32.4% and 68.6% for AT&T and Abilene respectively. Recall that *Offline-R* and *Offline-MO* aim to use paths that share links as much as possible with other paths. However, *Offline-R* has the advantage of preferring paths that require the least number of new links to be established. Hence, it has the best overall performance. However, both approaches show a similar rate of increase in their average link utilization at higher network load. For AT&T, see Figure 3.2a, if the network load is low with $|Q| = 50$, *Offline-R* has an average link utilization of 2.9% against 3.23% for *Offline-MO*. In high network load scenarios, i.e., $|Q| = 2000$, the average link utilization of *Offline-R* reaches 56.13% versus 62.71% for *Offline-MO*; this indicates an increase of 1935.5% and 1941.4% respectively. These results are also consistent for Abilene, see Figure 3.2b. Under low network load, i.e., $|Q| = 50$, *Offline-R* has an average link utilization of 12.64% and *Offline-MO* 18.18%; when the network load

is increased to $|Q| = 2000$, the average link utilization rises to 86.01% and 93.15% respectively, which means a rate of increase of 680.45% for *Offline-R* and 512% for *Offline-MO*. The average link utilization of Abilene is consistently 2.1 times that of AT%T under the same network load. This is because Abilene has fewer links, or smaller network capacity. This difference in link utilization has a direct impact on the final number of active links.



(a)



(b)

Figure 3.2: Average link utilization for the offline heuristics under different values of LSP requests. a) AT&T and b) Abilene

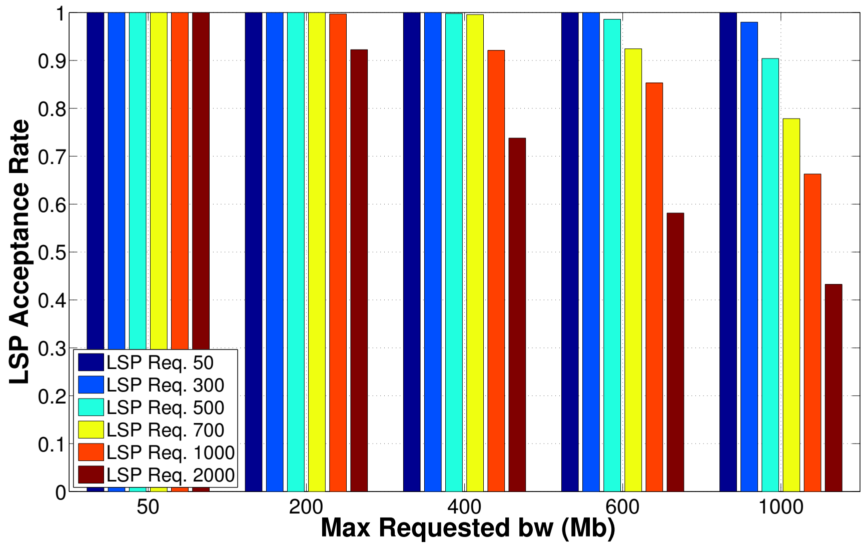
Table 3.6 presents the average LSP acceptance rate for AT&T and Abilene topologies. Figure 3.3 and 3.4 depict the data in Table 3.6 for AT&T and Abilene, respectively. For AT&T, both approaches show good performance. In particular, *Offline-R* exhibits a slightly better overall LSP acceptance rate of 93% as compared to 92% for *Offline-MO*. Figure 3.3 shows the LSP acceptance rate for this scenario is above 40%. This is in spite of the average link utilization being above 80% as observed in Figure 3.2a. With respect to Abilene, Figure 3.4 shows that both approaches present a similar performance with an average LSP acceptance rate of 73% for *Offline-R* and 67% for *Offline-MO*. However, given that the network utilization of Abilene increases more rapidly than AT&T, the observed LSP acceptance rate also decreases significantly; as an example, consider the case when the number of LSP requests is 2000 and the max requested bandwidth is 400; For AT&T, the LSP acceptance rate is above 70% for both approaches, whereas for Abilene, the acceptance rate is below 40%.

Table 3.6: Overall acceptance rate of offline heuristics over AT&T and Abilene

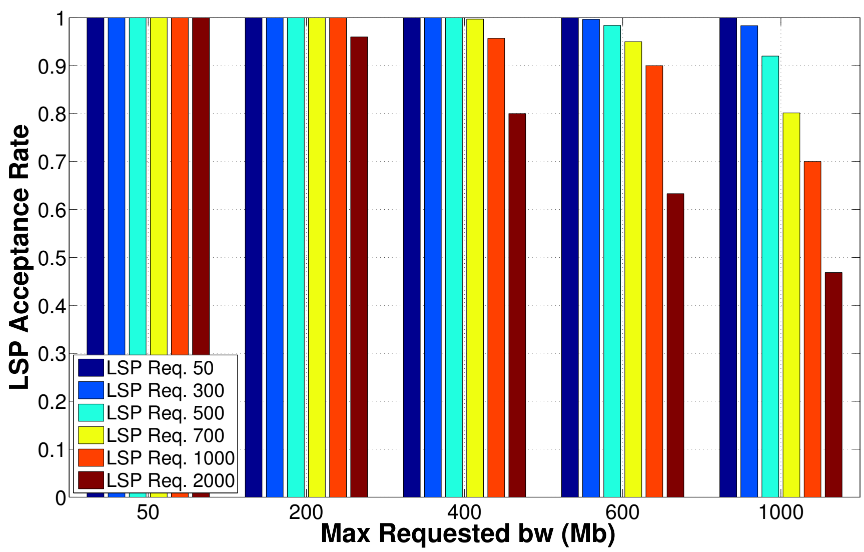
	AT&T		***Abilene***	
Max Req. bw (Mb)	***Offline-R***	***Offline-MO***	***Offline-R***	***Offline-MO***
50	1.0	1.0	1.0	0.99
200	0.99	0.99	0.88	0.81
400	0.96	0.94	0.71	0.62
600	0.91	0.89	0.59	0.52
1000	0.82	0.79	0.47	0.42
Avg.	**0.94**	**0.92**	**0.73**	**0.67**

3.4.2 Online Approaches

Figure 3.5 present the average link utilization for the different online heuristics for AT&T and Abilene. In the case of AT&T, see Figure 3.5a, the lowest average link utilization is observed for *Online-MinH*, at 27.32%, and the second lowest average link utilization of 31.19% belongs to *Online-R*, which utilizes established links and prefers paths that require the fewest number of new links to be set up. The overall

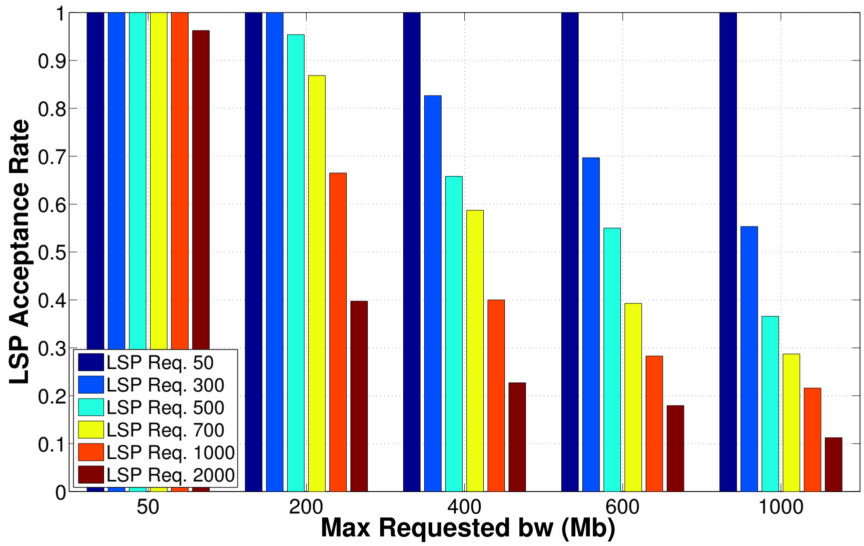


(a)

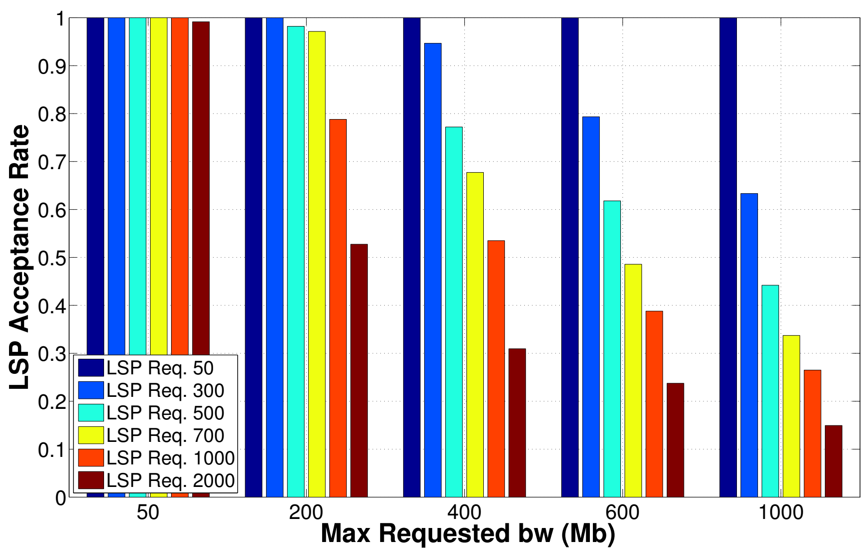


(b)

Figure 3.3: AT&T topology. LSP acceptance rates of the offline heuristics for different number of LSP requests. (a) Offline-MO and (b) Offline-R



(a)



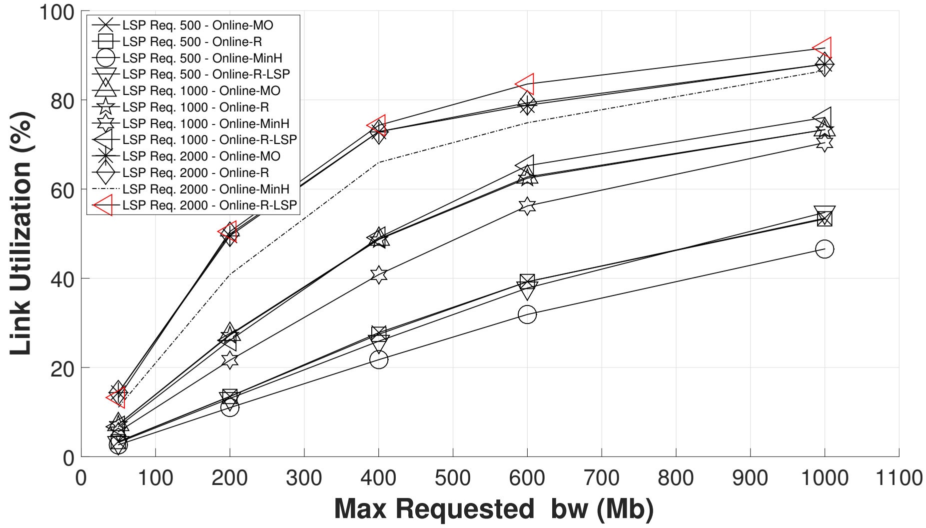
(b)

Figure 3.4: Abilene topology. LSP acceptance rates of the offline heuristics for different number of LSP requests. (a) Offline-MO and (b) Offline-R

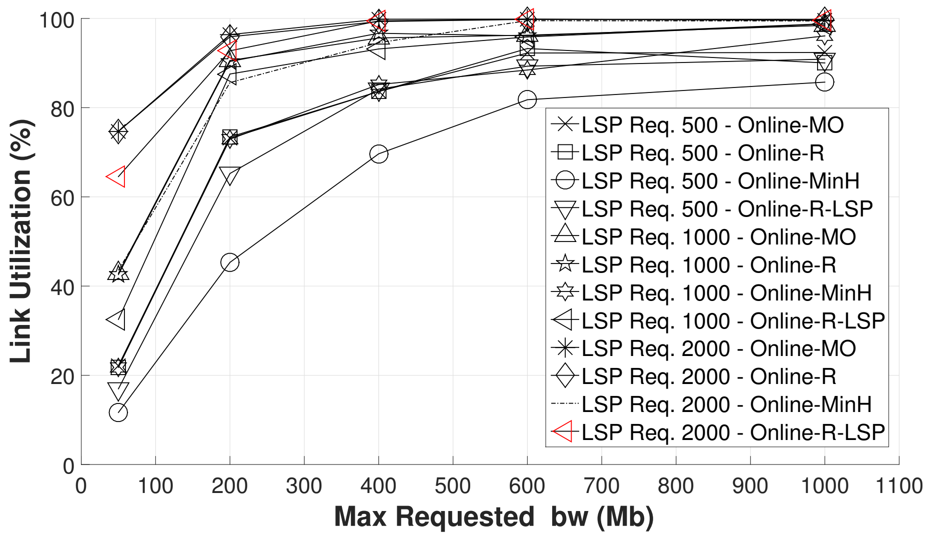
link utilization is low for both heuristics. This is because most LSPs are routed over fewer links and leaving many links little to no load. In the case of Abilene, see Figure 3.5b *Online-MinH* also produces the lowest average link utilization at 56.48%. Surprisingly, the second best performer at 65.2% is *Online-R-LSP* that selects LSPs randomly. However, the utilization of *Online-R-LSP* is very close to that of other approaches. As expected, when the network load increases, link utilization also increases. In particular, *Online-MinH* shows an increase of 53.44% and 74.68% for AT&T and Abilene respectively when going from the lowest to the highest possible network loads.

Table 3.7 shows the average LSP acceptance rate for the online heuristics that exhibit the highest LSP acceptance rates for AT&T and Abilene. Figure 3.6 and 3.7 are plots of Table 3.7. *Online-MinH* and *Online-R-LSP* exhibit the best performance for both topologies. Overall, *Online-MinH* has a slightly better performance than *Online-R-LSP*. Specifically, for AT&T, *Online-MinH* and *Online-R-LSP* present the same overall LSP acceptance rate of 94%. For Abilene, *Online-MinH*, shows an average LSP acceptance rate of 75% against 71% for *Online-R-LSP*. These LSP acceptance rates are due to *Online-MinH* attaining the lowest average link utilization for both topologies, see Figure 3.5a and 3.5b.

Note that the total average LSP acceptance rate for the online approaches, 83.5%, is larger than the total average LSP acceptance rate for the offline approaches, 81.5%. These total average LSP acceptance rates are obtained by computing the mean of the average values presented in Table 3.6 and 3.7, respectively. The main goal of offline approaches when establishing LSPs is to minimize the overall energy consumption of the network even if this implies a decrease in LSP acceptance rates. On the other hand, Online-MinH and Online-R-LSP do not consider energy savings as the main factor when establishing LSPs and their main objective is to accept as many future LSP requests as possible. Consequently, they have higher LSP acceptance rates. This trade-off between energy savings and LSP acceptance rates will be discussed



(a)



(b)

Figure 3.5: Average link utilization for the online heuristics under different values of LSP requests. (a) AT&T (b) Abilene

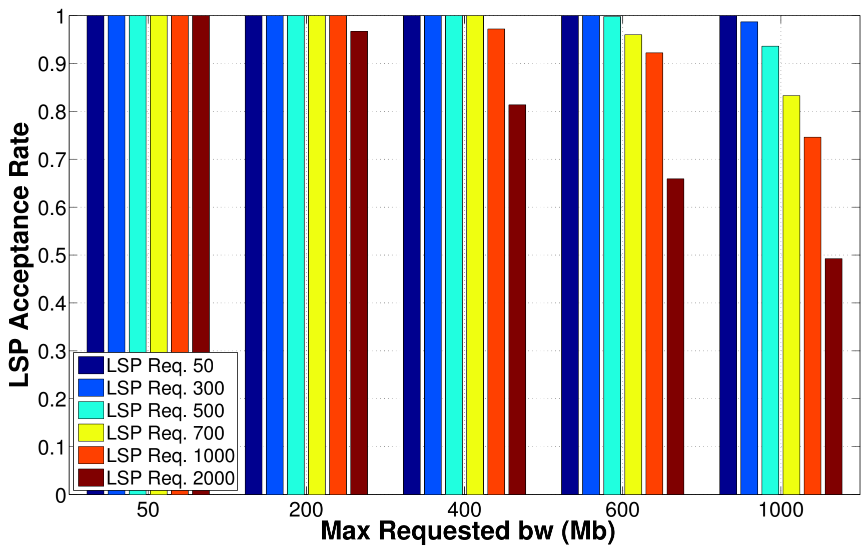
in more detail in Section 3.4.3 when the tested heuristics are compared according to their ρ ratio.

Table 3.7: Online heuristics that exhibit the largest LSP acceptance rates for AT&T and Abilene

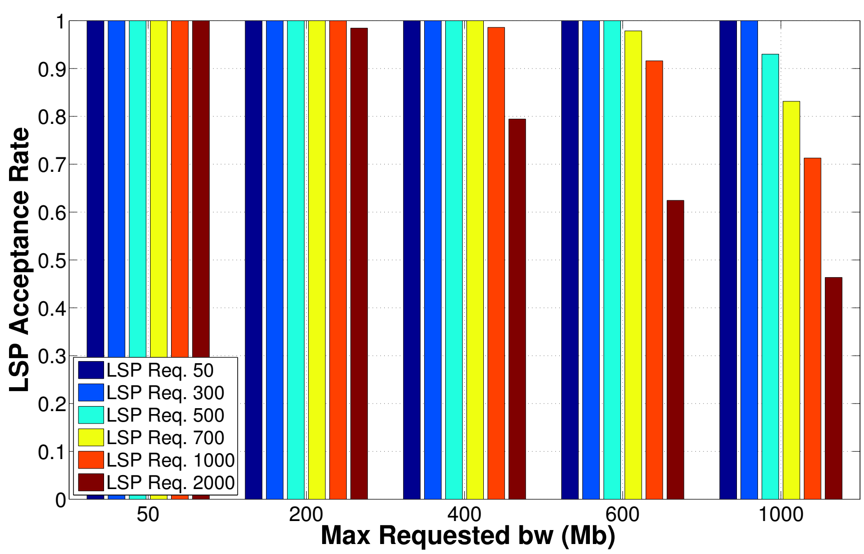
	AT&T		***Abilene***	
Max Req. bw (Mb)	***Online-MinH***	***Online-R-LSP***	***Online-MinH***	***Online-R-LSP***
50	1.0	1.0	1.0	1.0
200	0.99	1.0	0.89	0.86
400	0.96	0.96	0.74	0.67
600	0.92	0.92	0.63	0.55
1000	0.83	0.82	0.49	0.45
Avg.	**0.94**	**0.94**	**0.75**	**0.71**

3.4.3 Discussion

Table 3.8 and 3.9 show a comparison between the percentage of shut down links achieved by online approaches and the overall percentage of shut down links for the best offline approach; namely, *Offline-R*. *Offline-R* is able to shut down 22.1% and 8.6% of the links in AT&T and Abilene, respectively. Note that *Offline-MO* exhibits a much lower percentage of shut down links than *Offline-R*; Therefore, the results for *Offline-MO* are omitted. *Offline-MO* shuts 9.9% of the links for AT&T and less than 1% for Abilene. The tables also show the LSP acceptance rates for *Offline-R* and online approaches. The ρ ratio for each heuristic is also presented. The better performance exhibited by *Offline-R* is the result of its LSP selection policy that requires fewer new links to be established, which has a direct impact on the final number of active links. It is interesting to see that *Online-MinH* and *Online-R-LSP* are among the approaches with the worst performance in regards to the overall percentage of shut down links, with 17% and 5.7%, for AT&T, and 2.13% and 0.12%, for Abilene, respectively. At the same time, these two approaches have the highest LSP acceptance rate. *Online-MinH* has an overall LSP acceptance rate of 94% and 75% for AT&T and Abilene, respectively. The corresponding values for *Online-R-LSP* are 94% and 71% for AT&T and Abilene respectively. On the other

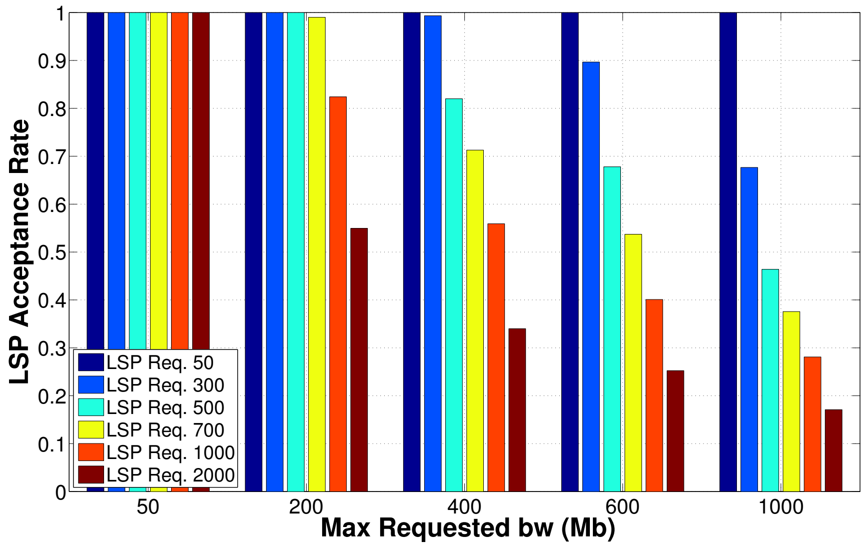


(a)

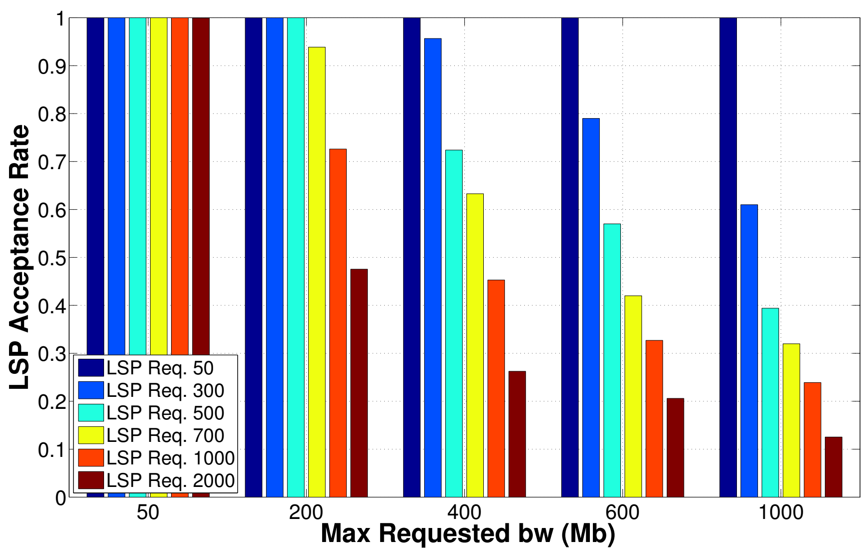


(b)

Figure 3.6: AT&T topology. Online heuristics that exhibit the best LSP acceptance rates for different number of LSP requests. (a) Online-MinH (b) Online-R-LSP



(a)



(b)

Figure 3.7: Abilene topology. Online heuristics that exhibit the best LSP acceptance rates for different number of LSP requests. (a) Online-MinH (b) Online-R-LSP

hand, *Online-MO* and *Online-R* show the lowest LSP acceptance rate; both recorded a percentage of 92% for AT&T and 67% for Abilene, respectively. However, these two approaches are the ones that were able to shut down the largest percentage of links. For AT&T, the percentage of shut down links when using these approaches is 21.6% and 19.3%, respectively. When tested over Abilene, *Online-R* exhibits a slightly better performance than *Online-MO*; i.e., 2.9% versus 2.1%, respectively. As expected, there is a clear trade-off between LSP acceptance rates and the number of active links. The larger the LSP acceptance rate the fewer the number of links a green technique is able to shut down. The good performance of *Online-MO* and *Online-R* is due to their low overall link utilization; see Figure 3.5a and 3.5b.

Table 3.8 also shows that for AT&T, the percentage of shut down links for the best online approach, *Online-MO*, is around 97.7% of the percentage of shut down links observed for *Offline-R*. *Online-MO* selects paths that require the fewest number of new links, which decreases the overall percentage of active links. On the other hand, *Online-R-LSP* randomly selects paths without considering energy consumption. This results in *Online-R-LSP* exhibiting the smallest percentage of shut down links among the studied approaches, with only 12.2% of the recorded percentage of *Offline-R*. For Abilene, Table 3.9 indicates that *Online-R* and *Online-R-LSP* exhibit the best and worst performance, respectively. Specifically, *Online-R* is able to shut down 24.9% of the links shut down by *Offline-R*, whereas, *Online-R-LSP* only shuts down 1.4% of the links shut down by *Offline-R*.

Finally, ρ is studied; see Figure 3.8. The figure qualitatively compares the performance of all the studied approaches for the AT&T and Abilene topologies. *Online-MinH* is the best online approach with a ρ ratio of 3.7. This is an interesting result given that *Online-MO* and *Online-R* are the approaches that present the largest energy savings. Note that *Online-MinH* exhibits better LSP acceptance rates than *Online-MO* and *Online-R*. However, its percentages of shut down links are slightly smaller than the percentages exhibited by *Online-MO* and *Online-R*. This means

that in this case, it is better to sacrifice some energy in exchange for better LSP acceptance rates. Observe that *Online-MO* presents the third and fourth best performance for AT%T and Abilene, respectively. On the other hand, *Online-R* presents the fourth and third best performance for AT%T and Abilene, respectively. In this particular case, both approaches present the same LSP acceptance rate regardless of the topology. However, the approach with the higher percentage of shut down links is the one that exhibits the best overall performance.

Table 3.8: Comparison of the performance of online approaches and *Offline-R* according to their ρ ratio for AT&T.

Heuristic	**Shut down links (%)**	**Links shut down** by *Offline-R*	**Overal LSP acceptance rate (%)**	ρ
Offline-R	22.1	100	94	3.7
Online-MO	21.6	97.7	92	2.7
Online-R	19.3	87.3	92	2.4
Online-MinH	17.0	76.9	94	2.8
Online-R-LSP	5.7	12.2	94	0.95

Table 3.9: Comparison of the performance of online approaches and *Offline-R* according to their ρ ratio for Abilene

Heuristic	**Shut sown links (%)**	**Links shut down by *Offline-R***	**Overal LSP acceptance rate (%)**	ρ
Offline-R	8.6	100	73	0.31
Online-R	2.9	33.4	67	0.08
Online-MO	2.14	24.9	67	0.06
Online-MinH	2.13	24.8	75	0.09
Online-R-LSP	0.12	1.4	71	0.004

3.5 Conclusions

This chapter has studied the problem of reducing the energy consumption of an MPLS network using online and offline path establishment methods. This study is the first extensive work on green LSP establishment solutions. Specifically, the study considers six heuristics over the same topologies. Notably, online and offline heuristics are compared in terms of energy savings and LSP acceptance rates. On

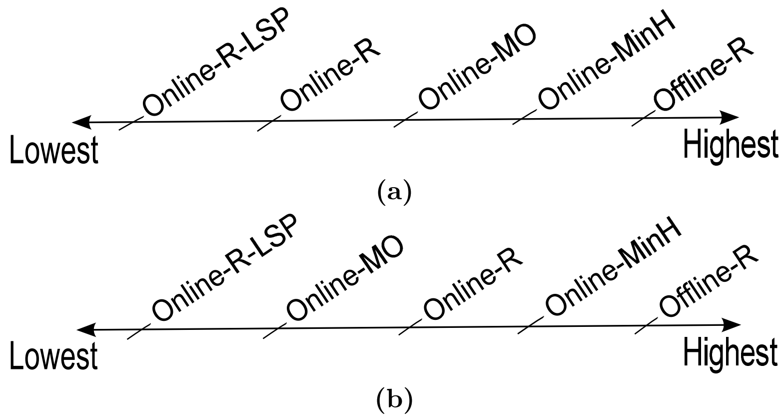


Figure 3.8: A qualitative comparison of online heuristics according to their ρ ratio for (a) AT&T, and (b) Abilene topologies

the Abilene and AT&T topologies, results indicate that during off-peak periods, LSP acceptance rates above 90% are possible with 20% of links shut down to conserve energy.

A key observation is that in MPLS networks, devices have an integrated data and control plane. This integration makes current networks complex and hard to manage, hence, reducing the capability to perform TE. To ease and facilitate the evolution of future networks, SDNs implement a new network paradigm whereby the control and data planes are separated. In addition, controllers are introduced to manage the control plane. The next chapter presents a green TE solution that aims to reduce the energy consumption of a SDN by assigning a switch to a controller in an energy efficient manner while considering delay, link utilization and the number of switches associated to a controller.

Chapter 4

GreCo: An Energy Aware Controller Association Algorithm for Software Defined Networks

SDNs separate the control and data planes of devices to expedite the deployment, configuration and evolution of networks. In this respect, controllers play a key role, see Section 2.1.2. In particular, they control the behavior of switches and must do so quickly. This, however, may be at odds with the goal of reducing the energy consumption of a SDN because powering down links may cause controllers to use paths with larger propagation delays. To date, existing works have not considered the controller placement problem and energy efficiency jointly.

To this end, this chapter addresses the problem of assigning switches to controllers with the aim of switching off the maximum number of links subject to the

following constraints: delay, link and the number of switches associated to a controller. In addition, the approach presented in this chapter considers the Maximum Link Utilization (MLU) of each link when routing demands. A Binary Integer Program (BIP) is also presented in order to derive the optimal solution for the considered problem. The investigation is conducted using five topologies: Abilene, AT&T, GEANT and SURFNet and 8N24L [115].

4.1 Network Model

A directed graph $G(V, E)$ is used to model a SDN, with the set V containing nodes and E is the set of edges. The set $S = \{s_1, s_2, \ldots s_{|S|}\} \subset V$ contains layer three switches and $C = \{c_1, c_2, \ldots c_{|C|}\} \subset V$ is the set of controllers. Each link $e = (i, j)$ between node i and j has capacity c_e. All links are assumed to consume the same amount of energy; see [15]. Let Θ be the set representing the communications between Switch-to-Controller (S2C), Controller-to-Controller (C2C), and Switch-to-Switch (S2S). In this respect, $\theta \subseteq \Theta$ is used to denote S2C communications. Each $(t, r) \in \Theta$ has demand d_{tr}, where $t, r \in V$ indicate the pair of nodes involved in S2C, C2C or S2S communications. The set P_{tr} contains the first K simple shortest paths for each (t, r) pair; each of which is indexed by k. Specifically, the k-th path is p_k^{tr}, and its associated propagation delay is ℓ_k^{tr}. Note, K is an upper bound as some (t, r) pairs may not have K paths. Let $\delta_{e,p}^{tr} \in \{0, 1\}$ be an indicator variable that is set to one if path $p \in P_{tr}$ uses link e to carry demand d_{tr}. The set of paths using a link e is $P_e = \{p \mid \delta_{e,p}^{tr} = 1, \forall p \in P_{tr}, \forall (t, r) \in \Theta\}$. Hence, the total traffic over a given link is $L_e = \sum_{p \in P_e} B(p)$, where the function $B(p)$ returns the demand transmitted on path p. Each link has utilization $u_e = L_e / c_e$, and its bounded by u_{max}. The average propagation delay for each $(t, r) \in \theta$ is

$$D_{tr} = \frac{\sum_{p_k^{tr} \in P_{tr}} \ell_k^{tr}}{|P_{tr}|} \tag{4.1}$$

As mentioned, the aim is to power down links. In the process, links within the shortest path between a switch and a controller may be shut down, meaning the controller will have to use an alternative path. In order to bound the increase in latency, a maximum deviation is specified, labeled z, from the average delay D_{tr}. Hence, the maximum delay tolerance or deviation from the mean is $T_{tr} = D_{tr} + z$. Lastly, $Q_{tr} \subseteq P_{tr}$ is written as the set containing paths for a demand $(t, r) \in \theta$ that satisfy $\ell_k^{tr} \leq T_{tr}$. For the purpose of load balancing, the path set $P_c = \{p \mid \forall(t, r) \in \theta, t \in S, r = c \in C, p \in P_{tr}\}$ is defined; i.e., all S2C paths that terminate at controller $c \in C$.

4.2 The Problem

Given a SDN $G(V, E)$ with $|C|$ controllers, the set of communicating pairs Θ with demand d_{tr}, the aim is to shut down as many links as possible during off-peak periods. Let $x_e \in \{0, 1\}$ indicate whether link e is active. With a slight abuse of notation, p_k^{tr} will be used to denote a binary decision variable that indicates whether the said path is chosen. Mathematically, we have the following BIP,

$$\min_{\forall e \in E} \sum x_e \tag{4.2}$$

Subject to,

$$\sum_{k=1}^{K} p_k^{tr} = 1 \quad \forall(t, r) \in \Theta \tag{4.3}$$

$$x_e \geq p_k^{tr} \quad \forall e \in E, \forall p_k^{tr} \in P_e \tag{4.4}$$

$$\sum_{(t,r)\in\Theta,\, p_k^{tr} \in P_e} d_{tr} p_k^{tr} \leq c_e u_{max} \quad \forall e \in E \tag{4.5}$$

$$\sum_{p_k^{tr} \in P_c} p_k^{tr} \leq \mathcal{L} \quad \forall c \in C \tag{4.6}$$

The objective is to minimize the number of active links that can be used to route all demands in Θ. The constraints include: only one path of each (t, r) pair can

be selected (4.3), a link is active only if there is a path that uses it (4.4), the total link utilization of each link is within the MLU (4.5), and the number of switches connected to each controller is less than a given value $\mathcal{L}$. It is also worth noting that each pair $(t, r) \in \theta$ consists of up to K paths, each of which may be connected to a different controller that belong to the set Q_{tr}; i.e., these K paths meet the delay bound T_{tr}. This thus allows us to consider the delay constraint imposed on S2C communication paths. Apart from that, powering down of switches is not considered because controllers may need to gain access to their associated switches quickly in order to respond to network events. In particular, the process of waking up one or more switches, either those on the path or the target switch, will take variable and non-negligible delays. Lastly, to determine $\mathcal{L}$, the BIP solver is repeatedly called until a feasible solution in the range $[\lceil \frac{|S|}{|C|} \rceil, |S|]$ is found.

Figure 4.1 will be used to show a simple instance of the presented formulation. Only one demand (A, D) is considered. The objective is thus to minimize $x_1 + x_2 + x_3 + x_4$ subject to the following constraints: (i) $p_1^{AD} + p_2^{AD} = 1$, (ii) $x_1 \geq p_1^{AD}$, $x_3 \geq p_1^{AD}$, $x_2 \geq p_2^{AD}$, $x_4 \geq p_2^{AD}$, and (iii) assuming the demand between node A and D has a rate of d_1 and $u_{max} = 1$, $d_1 p_1^{AD} \leq c_1$, $d_1 p_1^{AD} \leq c_3$, $d_1 p_2^{AD} \leq c_2$ and $d_1 p_2^{AD} \leq c_4$, (iv) $p_1^{AD} + p_2^{AD} \leq 1$, assuming $\mathcal{L} = 1$.

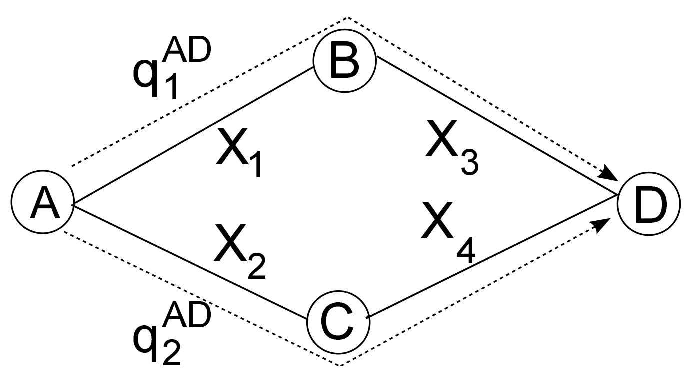


Figure 4.1: Example for BIP formulation

4.3 Heuristic

This section introduces GreCo, a green centralized controller association algorithm that addresses the problem in Section 4.2. It aims to reduce the energy consumption of a SDN while meeting link utilization, S2C latency and load balancing constraints. For each pair $(t, r) \in \Theta$, the algorithm determines the best p_k^{tr} path; i.e., it establishes the path with the lowest latency. Links that are not within the selected p_k^{tr} paths will be shut down. A demand is "rejected" if there is insufficient bandwidth to route it. Otherwise, it is "accepted". GreCo requires the set C and S. It is assumed that all controllers have the capability to also act as a switch.

Figure 4.2 depicts the flow chart of the introduced approach. The algorithm first computes K paths for each demand using Yen's algorithm [116]. After computing Q_{tr}, it associates each node with the closest controller. This is achieved by selecting from the set $Q_{tr} \subseteq P_{tr}$ the p_k^{tr} path with the lowest latency for a given S2C pair. Recall that the set Q_{tr} contains paths with latency less than T_{tr}. Moreover, GreCo aims to ensure all controllers have a similar number of switches. To this end, $OpCon$ is defined to be the "ideal controller number" or $\lceil \frac{|S|}{|C|} \rceil$. If a controller is assigned more than $OpCon$ switches, GreCo will check whether some of its associated switches can be moved to another controller. If the latency from a switch to another controller is less than or equal to T_{tr}, the node will be associated to a new controller that has the lowest latency. If there are no other controllers that meet the required latency bound, the switch will remain assigned to its initial controller. The links in the set p_k^{tr} that are used to route all demands in Θ are defined as the "surviving" links. Other links can be shut down. The algorithm then ascertains whether "surviving" links form a connected network. GreCo checks this by iterating over each node pair and determines if a path exists. If a path does not exist for a given (t, r) pair, GreCo picks a path from P_{tr} that traverses the maximum number of "surviving" links. If two or more p_k^{tr} paths have the same number of surviving links, the first of those K-paths will be chosen. In the last step, GreCo establishes a path for each demand

greedily and rejects a demand if it cannot find a route with sufficient bandwidth. That is, establishing the demand on any of its K paths mean the utilization of a link will exceed the MLU threshold.

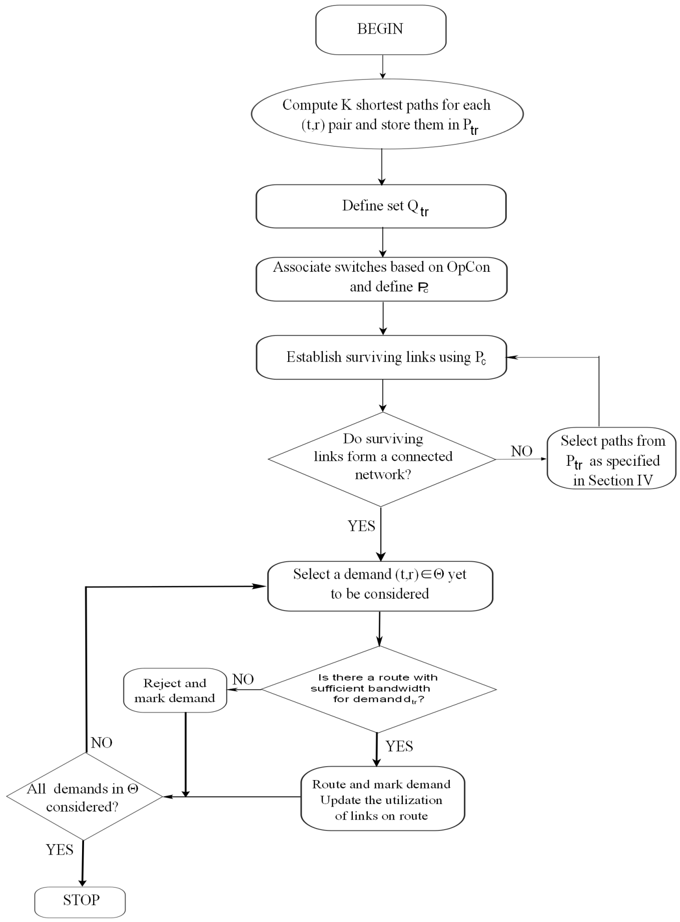


Figure 4.2: Flowchart for GreCo.

The topology, named *7N18L*, depicted in Figure 4.3, will be used to show how GreCo associates and load balances S2C communications. Let $S = \{1, 2, 3, 4, 6\}$ and $C = \{5, 7\}$ be respectively the set of switches and controllers. Therefore, the following S2C pairs are defined: $\theta = \{[1, 5], [1, 7], [2, 5], [2, 7], [3, 5], [3, 7], [4, 5], [4, 7], [6, 5], [6, 7]\}$. The value of $OpCon$ is $\lceil |S|/|C| \rceil = \lceil 5/2 \rceil = 3$. Referring to Figure 4.3, it is observed that all the nodes are closer to controller 5 than to controller 7. Consequently, they are initially assigned to controller 5. This assignment is not ideal because it is not load balanced. In order to correct this, the algorithm explores, in ascending order of node ID, whether a node can be re-associated to another controller. Therefore,

switch 1, 2, and 3 will be associated to controller 5. Up to this point, the number of switches associated to controller 5 is already equal to $OpCon$. Therefore, for switches 4 and 6, GreCo will verify that the latency to controller 7 is less than or equal to T_{tr}. Assume this to be the case. Hence, switches 4 and 6 will be associated to controller 7. Only links carrying S2C communications survive. These surviving links are the solid lines in Figure 4.3. Dotted lines represent links that can be shut down. Notice that the resulting topology is a connected network, and the achived energy savings are $6/18 = 33.33\%$. These surviving links can then be used to route C2C and S2S demands subject to the MLU constraint.

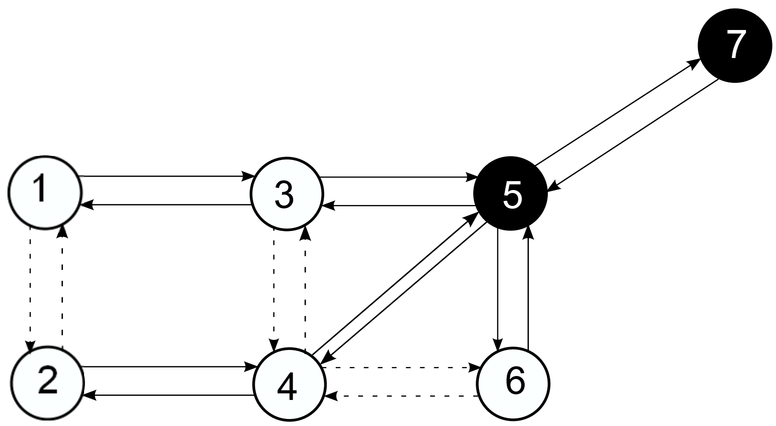


Figure 4.3: Example topology 7N18L.

This section concludes by outlining GreCo's time complexity. The most expensive step is using Yen's algorithm [116], which requires a time complexity of $\mathcal{O}(K|V|(|E| + |V|log|V|))$, to compute the K paths for each demand. As there are up to $\mathcal{O}(|V|^2$ demands and $\mathcal{O}(|V|^2)$ links, GreCo thus have time complexity $\mathcal{O}(K|V|^5)$.

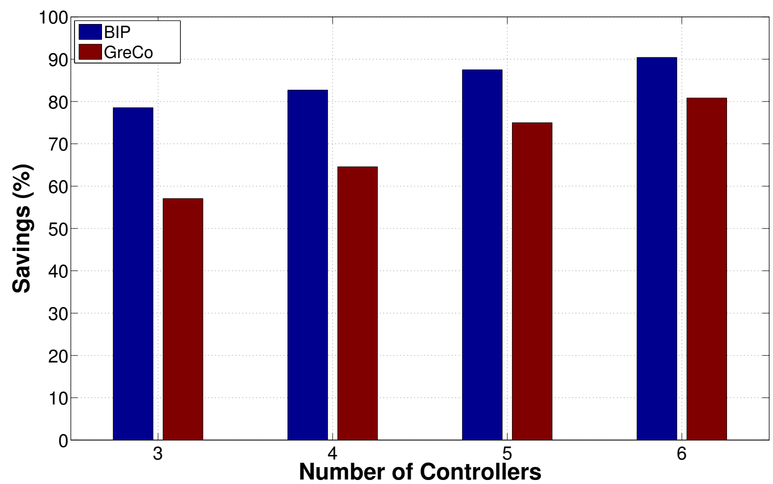
4.4 Evaluation

This section presents the evaluation of GreCo in networks with varying number of controllers; i.e., $|C|$. These networks include Abilene (11 nodes, 28 links), AT&T (25 nodes, 112 links), GEANT (40 nodes, 122 links), SURFnet (50 nodes, 138 links)

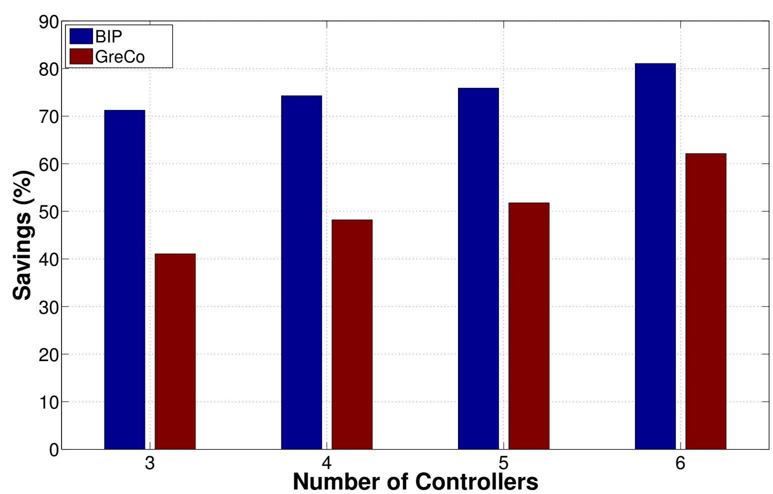
[112] [117] [113] and a synthetic topology 8N24L (8 nodes, 24 links) [115]. Traffic matrices (TMs) are obtained using the well known gravity model [118] [119]. As per [120], the MLU for each link is set to 80%. Simulations are conducted in MATLAB [114]. For each topology, a base TM is generated, called TM_{Base}, that represents the traffic load for a given (t, r) pair. Each node is randomly classified as a switch or a controller. One hundred simulation runs are conducted for different number of controllers; i.e., $|C| = 3, 4, 5, 6$. For a given (t, r) pair, the maximum deviation, i.e., z, is set randomly to a value in the range $[0, D_{tr}]$. Paths selected to transport S2C demands will be used as long as it does not violate the MLU, set to 80%, of surviving links. The presented results are within 95% confidence interval.

Figure 4.4 shows a comparison between the performance of BIP and GreCo when using $TM = TM_{Base}$ for 8N24L and Abilene respectively. GreCo uses only 9% and 19% more links than BIP when $|C| = 6$ for 8N24L and Abilene, respectively. Notice that any results for the larger topologies, e.g., AT&T, GEANT, SURFnet are not included because BIP is intractable on these topologies.

Figure 4.5a and 4.5b depict the percentage of shut down links for varying number of controllers when using $TM = TM_{Base}$ and $TM = 5{*}TM_{Base}$, respectively. GreCo has the best performance when operating over *AT&T* and *GEANT* with around 55% and 33% of shut down links, respectively. Note that the percentage of shut down links remains almost constant regardless of network load for all topologies. GreCo only allows links to remain active if the paths they belong to are used to carry S2C traffic and adhere to latency constraints. Results show that that S2C paths do not change when the network load for a given controller number is modified. Consequently, the surviving links remain the same. That is, for a given value of $|C|$, switches continue to select the same S2C paths as this selection is based on latency rather than link utilization. Hence, for a given (t, r) pair, its path p_k^{tr} will remain the same.

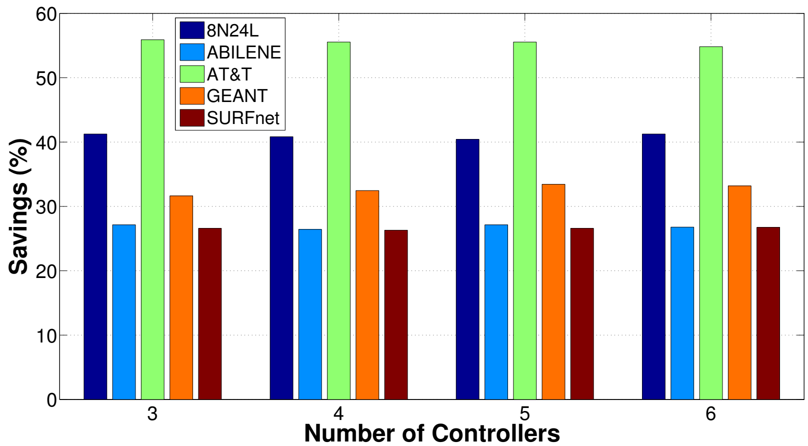


(a)

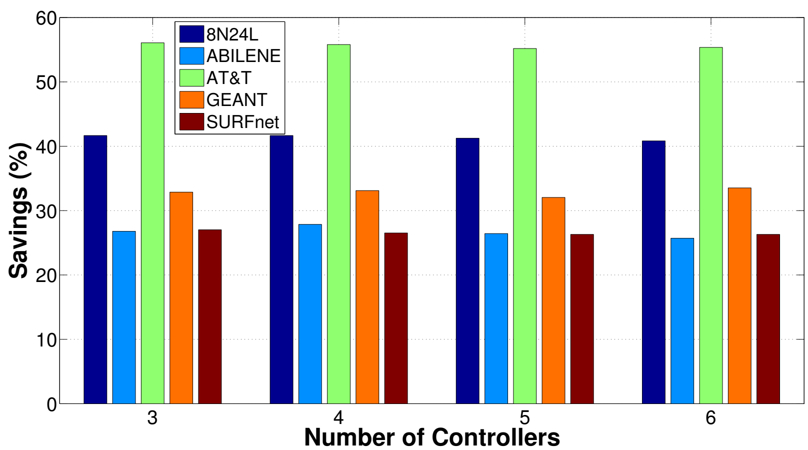


(b)

Figure 4.4: Percentage of shut down links for BIP and GreCo. a) 8N24L and b) Abilene



(a)



(b)

Figure 4.5: Percentage of shut down links for all tested topologies when using: a) TM_{Base} and b) $5 * TM_{Base}$

It is also observed that the number of (t, r) routed demands decreases when the network load increases. In fact, all the topologies observe a reduction in the number of accepted demands when the network load increases. This behavior is expected given that GreCo does not route traffic over links with utilization larger than the MLU; i.e., 80%.

4.5 Conclusion

Two key concerns of SDN operators are the energy consumption rate of switches and the delay between controllers and switches. To this end, this chapter has considered the problem of reducing the number of active links subject to the following constraints: delay between switches and controllers, link utilization and number switches assigned to a controller. This chapter also contains a BIP as well as a heuristic. The obtained results demonstrate savings up to 55% during off-peak hours and the proposed heuristic uses as few as 9% more links as compared to the optimal solution.

GreCo and the LSP establishment methods presented in Chapter 3 will cause the IGP to continuously re-compute paths in order to reduce the number of active network elements. This process, however, may have a negative impact on BGP. In particular, it is well known that changes in IGP will cause hot-potato routing changes that have been shown to negatively affect BGP. Unfortunately, existing green TE has neglected BGP. Therefore, the next chapter first adds to the state-of-the-art by quantifying how green TE techniques affect BGP. After that, based on the obtained results, it proposes a technique that reduces energy consumption while minimizing the negative effects on BGP.

Chapter 5

BGP Aware Traffic Engineering – Part I

As mentioned in Chapter 1, it is widely known that hot-potato routing causes numerous negative impacts on the operation of BGP. In [37] and [38], the authors establish three main impacts of IGP on BGP: (i) transient packet delay and loss while routers recompute their forwarding table, (ii) local BGP routing changes that cause BGP in peer Autonomous System (ASes) to continuously reconverge, and (iii) shift in traffic that may cause congestion on new paths to other ASes.

To date, there are no studies that measure the impact that green TE techniques have on the operation of BGP. Therefore, this chapter is the first work that quantifies the impacts due to the use of green IGP link weight optimizers and green OSPF; see Section 2.2.1. This chapter also introduces the first green BGP-aware approach namely Hot Potato Low UtiliZation (HoTPLUZ). Its key features include: (i) minimizing the impact on BGP whilst minimizing overall energy consumption using the least cost IR-ER paths, ii) diverting traffic away from lowly utilized links and aggregate them onto highly utilized links, and iii) consideration for link utilization to avoid packet loss and high latencies.

5.1 Hot-Potato Routing Effects

As mentioned, to date, there are no studies that determine whether green TE methods have any effects on BGP. To this end, this section aims to fill this gap. It first discusses the simulation methodology used to obtain the results presented in Section 5.1.1. In order to quantify a green TE method's affects on BGP, the following metrics are recorded: (i) proportion of egress router changes, and (ii) percentage of rerouted traffic within a single AS.

Two example green TE approaches are used in all experiments: i) Greedy Algorithm for Energy Saving (GAES) [28]. This technique shuts down the least utilized links while taking into account the average network load whilst maintaining network connectivity. In the experiments, initial link weights are calculated using IGP-WO [87] – available as part of the TOTEM toolbox [121]. The objective function of IGP-WO is the overall link utilization, which is directly related to network congestion. After the initial link weight calculation, links are sorted in non-decreasing order according to their utilization and the least loaded links are switched off. The algorithm then determines if the given traffic matrix is supported by the remaining links; i.e., no links exceed their maximum utilization, and ii) Energy Saving in the Internet based on Occurrence of Links (ESOL) [27]. This technique represents green approaches that only place links to sleep without modifying link weights. ESOL runs on top of OSPF and uses LSAs to determine the occurrence of nodes and links in all calculated shortest paths. Links with a low frequency of occurences are placed into sleep mode. ESOL also considers network connectivity and maximum link utilization when powering down links.

Experiments are conducted over the following topologies: SURFnet (50 nodes, 138 links), GEANT (40 nodes, 122 links), AT&T (25 nodes, 112 links), Abilene (11 nodes, 28 links) [112] [117] [113]. These networks are constructed in TOTEM [121]. In order to simulate a MPLS network, links are replaced with two unidirectional links with the same capacity. This guarantees that each link is assigned the same

TE link weight. Note, the MPLS network is assumed to run OSPF. This is to allow the deployment of green TE techniques [109]. TMs are then generated using the well known gravity model [118][119].

Once a given topology and its respective TM have been simulated in TOTEM, GAES and ESOL are run. In the case of GAES, it runs IGP-WO. This algorithm calculates the optimal link weights in order to route the TM. The output is a network that contains fewer links capable of supporting the TM. In the case of ESOL, OSPF is run without performing any modifications to link weights. Similar to GAES, it generates a topology with fewer links that can support the TM. In order to minimize IGP cost changes, both approaches only eliminate five links; the least loaded for GAES, and the ones with low occurrence in shortest paths for ESOL. In both approaches, the key constraints are that the overall link utilization is below 50% [122][123] and the resulting network remains connected. After the last iteration, GAES and ESOL generate as output a *final* topology. The remaining links are then recorded, and also the link weights in the case of GAES.

The following steps are run one hundred times:

1. A random number of routers is selected to be part of the following two groups: i) *Egress routers*, which act as exit points, and ii) *Ingress routers*, which use the said egress routers in order to reach a particular destination outside the local AS.

2. Run Dijkstra's algorithm on the original topologies; i.e., ones with no links shut down. For each ingress router, its distance to each egress router is calculated and the one that has the minimum distance is selected as the exit point. The traffic being sent to the selected exit point is also recorded.

3. Call GAES and ESOL with the original topologies. For each resulting topology generated by GAES or ESOL, the exit points are determined using the same process as Step 2.

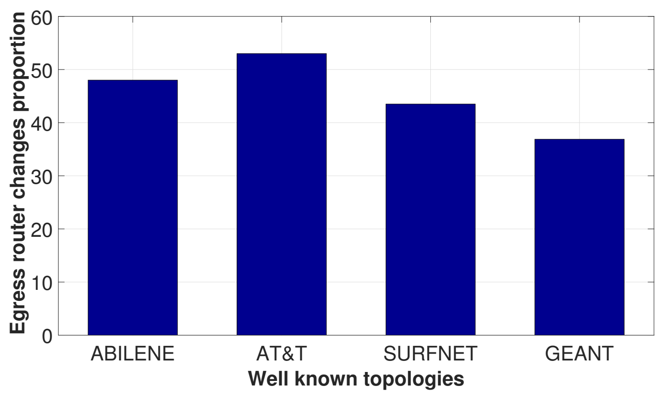
4. The results obtained in Steps 2 and 3 are compared and the proportion of ingress routers that have changed egress routers as well as percentage of rerouted traffic are recorded.

5.1.1 Hot-Potato Routing Results

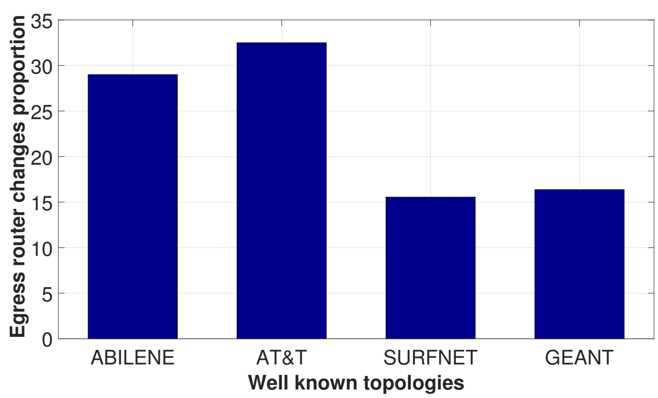
Figure 5.1 shows the percentage of changes in a path or cost to an existing egress router, aka, “hot-potato routing change”, for GAES and ESOL. The results indicate that the proportion of rerouted traffic is directly related to the percentage of changes in selected egress routers. This percentage along with the proportion of traffic that needs to be rerouted depends on the number of links that are switched off and continuous changes in link weights, causing IGP cost recalculation leading to long convergence delays.

Figure 5.1a and 5.1b show that for all simulated topologies, the percentage of hot-potato routing changes for GAES is 108% greater than when using ESOL. In particular, it is observed that the most affected topology is AT&T, where, 53% and 32.5% of ingress routers select a new egress router after the implementation of GAES and ESOL, respectively, which causes a corresponding 48.4% and 24.7% shift in traffic. This large shift in traffic is likely to cause undesired effects such as external BGP routing changes [124].

The proportion of rerouted traffic is depicted in Figure 5.2. These results show that the proportion of rerouted traffic for GAES is greater, i.e., 141%, than for ESOL. This discrepancy can be explained by the fact that ESOL does not modify link weights, and only shuts down links. Therefore, traffic does not get rerouted as much as in GAES. That is, ESOL causes fewer changes in IGP cost and hence shortest paths between nodes do not change often. For ESOL, Abilene has the highest proportion of rerouted traffic, with 25.3%. Abilene is the smallest among the four simulated topologies and therefore, the number of link changes in computed shortest paths due to the shut down of links are more severe than in other topologies.

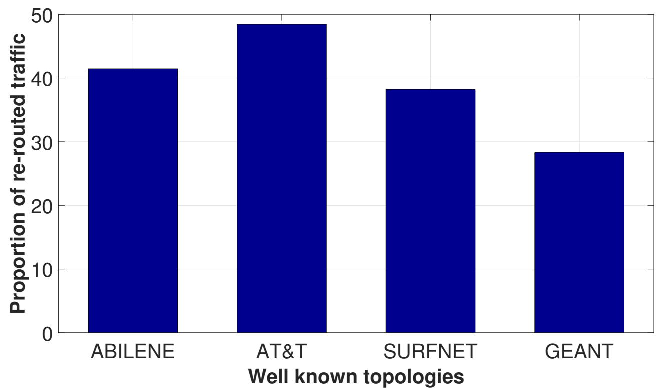


(a)

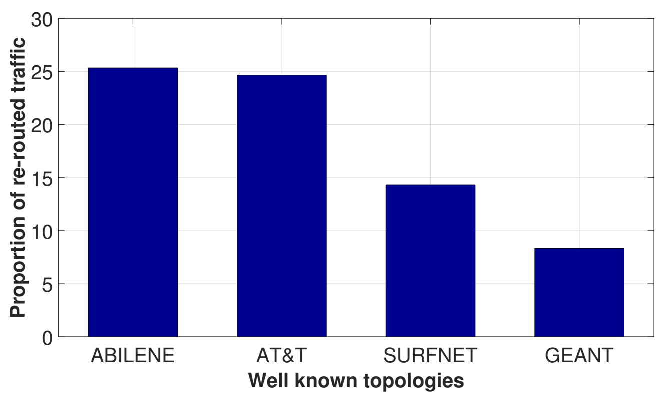


(b)

Figure 5.1: Percentage of hot-potato routing changes. a) GAES and b) ESOL.



(a)

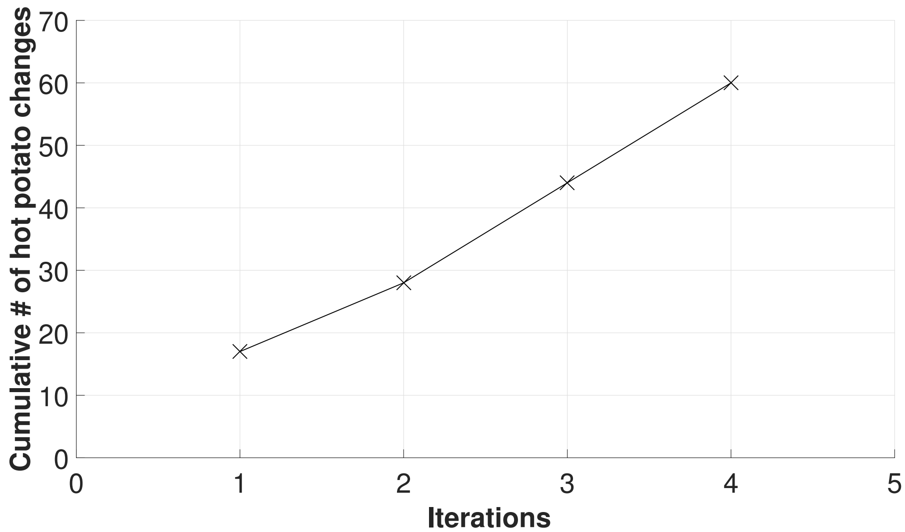


(b)

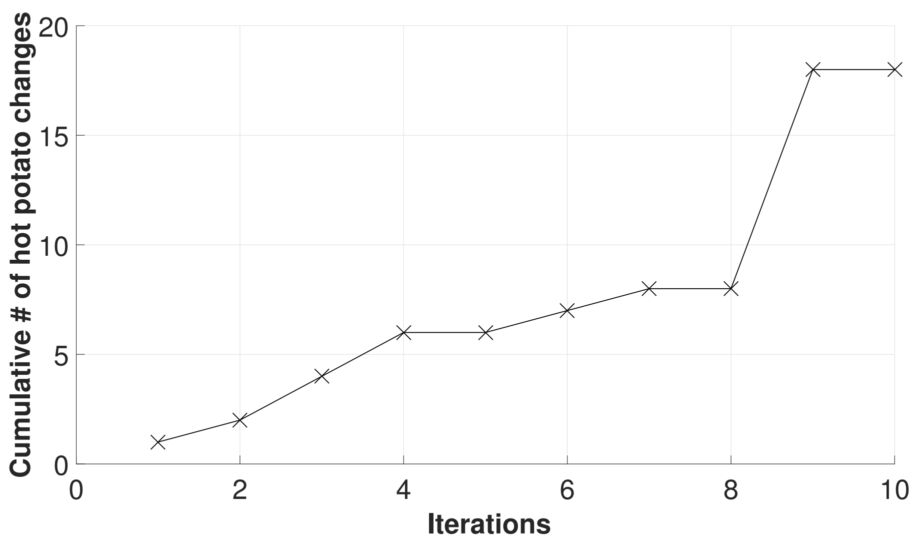
Figure 5.2: Proportion of rerouted traffic due to hot-potato routing changes. a) GAES and b) ESOL

Figures 5.3 shows the correlation between the number of switched off links and their impact on BGP as measured by the "cumulative number of hot-potato changes" for each of the iterations; this is defined as the number of BGP next hop changes. In both approaches, the network became disconnected after five and 11 iterations, respectivey. Hence, data for only four and 10 runs were collected for GAES and ESOL respectively. For GAES, Figure 5.3a shows that the number of BGP next-hop changes rises steadily with every iteration, reaching 60 in the final run. Each iteration increases the number of links being shut down and link weight changes. A particular interesting observation is that a significant number of hot-potato changes are the result of the algorithm deciding to shut down one of the links connecting the local AS 45768 with the remote AS 45953. The shut down of the said link has an impact on the first stages of GAES as it affects every route that uses this link. Once traffic shifts to other links, no more route changes are observed because links that have been shut down are not brought up again. For ESOL, Figure 5.3b shows that BGP next-hop changes do not increase at a constant rate in comparison to GAES. Moreover, the total hot-potato changes, which is 18, is less than the total value recorded for GAES, which is 60. This means the number of BGP next-hop changes when using GAES is 233% greater than ESOL. Another observation is that in iterations 4, 7, and 9, the number of BGP next-hop changes do not increase. This is because ESOL does not modify link weights but only shuts down links. Consequently, link weights change infrequently, which decreases SPF recalculations and the number of hot-potato routing changes.

The presented results indicate that changes in IGP have non-negligible effects in the performance of BGP. As mentioned in Chapter 2, the main goal of green routing approaches is to conserve power by switching off routers and links. However, this needs to be carried out without affecting the operation of BGP. Henceforth, the following list design considerations that a green technique should take into account in order to avoid negatively impacting BGP. Note, these considerations are



(a)



(b)

Figure 5.3: Hot-potato routing changes caused by: a) GAES and b) ESOL

similar conceptually to those put forth by Teixeira et al. [124], which only consider minimizing hot potato changes as opposed to energy consumption:

1. IRs must select their optimum egress points. In this case, optimum refers to the shortest IR-ER path, which accomplishes the objective of hot-potato routing.

2. IRs use only one path per ERs. This aims to reduce the size of BGP routing tables and the number of BGP update messages. This consideration also aims to reduce re-routing computation and hence, the load of routers.

3. Tunnels are established between IRs and their chosen ER. The purpose of these tunnels is to ensure IR-ER path stability; they help reduce IGP changes and increase BGP stability.

5.2 Hot Potato Low UtiliZation (HotPLUZ)

Given the aforementioned design considerations, this section proposes HotPLUZ, a distributed approach that allows a network operator to switch off links to conserve power usage. In addition, HotPLUZ ensures IRs are able to establish a connection to their respective ERs and the maximum link load is minimized. The latter is important as a high load will lead to increased delays and possibly packet loss, despite significant power savings.

Before outlining HotPLUZ, it is necessary to define a few terms and key concepts. The node set is denoted as V and the edges are recorded in the set E. Each link $e \in E$ has capacity c_e and L'_e is defined as the sum of all traffic traversing such link at any given moment, i.e., link utilization. An IR-ER pair is represented as an (r, s) pair with a demand of d_{rs}. Let P_{rs} be the set of all simple paths for pair (r, s). Note, an IR will have a different set of paths for each ER. For each IR r, the chosen path for ER s is denoted as p^*_{rs}. The definition of how link cost, w_e, is calculated is as

follows. Initially, every link $e \in E$ has an IGP cost of $1/c_e$. Subsequent IGP costs are recomputed based on link utilization using Equ (5.1), where $(c_e - L'_e)$ represents the available bandwidth of a given link e. The use of Equ (5.1) causes IRs to divert traffic from lowly utilized links and aggregate said traffic onto highly utilized links. This is carried out with the goal of switching off lowly utilized links.

$$w_e = \begin{cases} \frac{1}{1-(c_e - L'_e)}, & 0 < L'_e \leq MLU \\ \infty, & L'_e > MLU \end{cases} \tag{5.1}$$

$$ALU = \frac{\sum_{e=1}^{|E|} L'_e}{|E|} \tag{5.2}$$

To avoid congestion and packet loss on links, a Maximum Average Link Utilization (MALU) threshold is defined, where the Average Link Utilization (ALU) is calculated using Equ (5.2). HotPLUZ also restricts link utilization, L'_e, to a predefined Maximum Link Utilization (MLU) value; i.e., $L'_e \leq MLU$. This MLU value is a percentage of the total capacity of a given link e; i.e., γc_e, with $0 \leq \gamma \leq 1$. These two parameters, MALU and MLU, are predefined by the network operator. Also, the MLU can be set to a value that allows a link to absorb any sudden burst in traffic. In particular, as per [120], to keep delay low, the MLU is set to 80% and the MALU to 70% as per [125].

HotPLUZ is now ready to be described. It works in rounds, and consists of the following key ideas:

- In round zero, all IRs select their corresponding ERs using hot potato routing, whereby they select a BGP speaker or ER whose intra-domain distance or IGP cost is the smallest.

- In subsequent rounds, each IR determines the link cost as per Equ (5.1). After that, for each IR, it determines whether there exists a new least cost path to a corresponding ER. If there is, the IR changes to the new path.

The above process continues until no IR makes any changes. That is, all IRs converge to the least cost path for each corresponding ER. Upon completion, the network may have a number of links with zero utilization. The network operator then has the option to switch these links off.

Figure 5.4 presents the flow chart of HotPLUZ. Initially, HotPLUZ computes the shortest IR-ER paths based on hop-count. Every IR then sends an UPDATE message to other IRs to inform them its path selection and corresponding demands. Once an IR receives an UPDATE message, it uses said information to calculate new link weights as per Equ. (5.1) and computes possible new least cost paths. If one is found, the IR then establishes the new least cost path, p^*_{rs}. An UPDATE message is then sent to other IRs and the process repeats until any of the following three conditions occur: i) the maximum predefined link utilization, MALU, is reached, ii) none of the paths $p \in P_{rs}$ are able to carry the given d_{rs} demand, or iii) when there are no more changes in the selected (r, s) path.

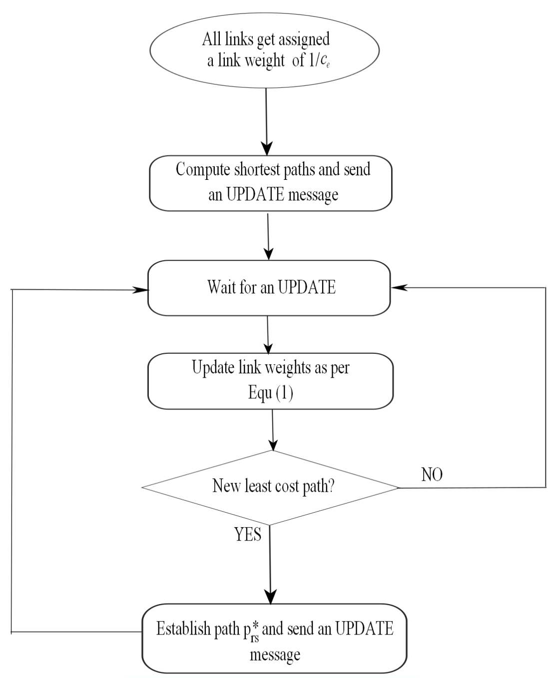


Figure 5.4: Flow chart for HotPLUZ

5.2.1 Evaluation

This section explains the methodology used to evaluate the performance of HotPLUZ. It employs, in addition to the four topologies used in Section 5.1, namely SURFnet, GEANT, AT&T and Abilene, a topology called 8N24L (8 nodes, 24 links) [115]. TMs are obtained using the gravity model. For each topology, a base TM, called TM_{Base}, is first generated and then another two TMs are calculated by multiplying TM_{Base} by 5, and 20. Each node is then randomly classified as an ER or IR. After that each IR selects *num* ERs randomly, where $1 < num < |E|$. The number of simulation runs for different number of ERs, i.e., *num*, $num = 2, 4, 6$ is 50. In all cases, an IR selects a new path as long as the new path has sufficient bandwidth to support a given IR-ER traffic demand. Only IR-ER traffic is considered. This means links that do not carry IR-ER traffic have zero utilization.

The *Savings* due to HotPLUZ is defined as $\frac{(Z-A)}{(|E|-A)} * 100$. Here, $|E|$ is the number of links. The variable A and Z are the number of links with zero utilization before running HotPLUZ, and the total number of links with zero utilization after running HotPLUZ, respectively. In other words, *Savings* correspond to the following ratio: the actual number of links with zero utilization due to HotPLUZ, denoted as $Z - A$, and the number of active links before running the proposed algorithm, i.e., $|E| - A$. The overall saving for each topology is calculated by averaging the final energy savings for each TM. In all experiments, the following metrics are recorded: *average utilization* and the average number of iterations to reach stability. The presented results are within 95% confidence interval.

Table 5.1 shows a summary of the overall savings obtained by HotPLUZ for different number of ERs, $num = 2, 4, 6$, and network loads, while Table 5.2 presents the average link utilization. Notice that energy savings increase when network load decreases. Experiment results for topologies 8N24L and ABILENE exhibit relatively constant energy savings for different network loads. These two topologies have the largest number of nodes and links. This means more paths and therefore, choices to

route respective IR-ER traffic are available to IRs. Consequently, HotPLUZ is able to shut down a higher number of links.

Table 5.3 presents the overall number of iterations for different *num* values and network loads. The highest values are observed for AT&T, GEANT and SURFnet with 2.5, 1.5 and 1.7 respectively.

Table 5.1 and 5.3 indicate that higher number of iterations lead to more energy savings. The more chances the algorithm has to run, i.e., more paths to choose from, the more energy reduction is achieved. For large topologies with rich connectivity, at low network load, HotPLUZ has more opportunities to reduce energy consumption but it will also need more iterations to stabilize. Recall that HotPLUZ adheres to link utilization constraints, which helps preventing excessive aggregation of traffic on certain links which can cause packet loss and high latencies.

Table 5.1: Overall savings (%) achieved by HotPLUZ for values of $num = 2, 4, 6$ and under different network loads.

TM	Topologies				
	8N24L	**Abilene**	**AT&T**	**GEANT**	**SURFnet**
$5 * TM_{Base}$	3.7	6.8	21.7	11.5	9.6
$20 * TM_{Base}$	3.7	0	21	6.9	1.8

Table 5.2: Average link utilization (%) exhibited by HotPLUZ for values of $num = 2, 4, 6$ and under different network loads.

TM	Topologies				
	8N24L	**Abilene**	**AT&T**	**GEANT**	**SURFnet**
$5 * TM_{Base}$	1.0	10.3	1.0	1.5	2.6
$20 * TM_{Base}$	3.7	40.17	4.2	6.0	9.8

5.3 Conclusions

This chapter has shown how two example green TE approaches induce hot-potato routing and thus, negatively interfere with BGP's operation. In other words, current green IGP techniques run the risk of exacerbating BGP convergence delays,

Table 5.3: Overall number of iterations for values of $num = 2, 4, 6$ under $TM = 5 * TM_{Base}$ and $TM = 20 * TM_{Base}$

TM	**Topologies**				
	8N24L	**Abilene**	**AT&T**	**GEANT**	**SURFnet**
$5 * TM_{Base}$	0.3	0.78	2.5	1.5	1.7
$20 * TM_{Base}$	0.3	0	2.5	0.7	0.2

and degrade the QoS of peer ASes. This conclusion is drawn from a study of two representative green IGP-WO and OSPF techniques, namely GAES and ESOL. Experimental results show that for GAES, hot-potato routing changes and the proportion of rerouted traffic are in the order of 108% and 141% greater than ESOL respectively. These results motivated the design of HotPLUZ, a distributed approach that switches off links without negatively impacting BGP. Experiments show that up to 21% overall saving is achievable under low network load.

The next chapter presents another BGP-aware techniqe that is run by egress and ingress routers. It complements HotPLUZ whereby a novel framework is introduced that allows ingress routers to collaboratively determine the set of links that will carry their respective traffic to egress routers.

Chapter 6

BGP Aware Traffic Engineering – Part II

Chapter 5 showed conclusively that shutting down the least utilized links and/or routers triggers what is known as hot-potato routing changes. These changes are well known to negatively impact the operation of BGP. Critically, they may cause packet loss due to slow BGP convergence and increased congestion. These critical issues thus motivated the design of HotPLUZ.

This chapter contributes another BGP aware approach. Specifically, it outlines a generic framework for use by Ingress Routers (IRs) that allow them to collaboratively determine the best paths that yield the best energy savings. In addition, based on the findings from Chapter 5, the proposed framework satisfies the following design considerations:

1. IRs must select the shortest Ingress-Egress Router (IR-ER) path, which equates to the goal of employing hot-potato routing. This ensures compatibility with current routing policies.

2. IRs should only use a single path per ER. This aims to reduce the size of IGP routing tables and the number of IGP update messages. This also reduces rerouting computation and hence, the CPU load of routers [126][127].

3. IRs must use a tunnel to each ER in order to avoid BGP changes due to the powering down of links/routers within an AS. However, established tunnels must be adaptive to network changes; i.e., link load, delay and QoS constraints.

4. IRs should be aware of network state and reuse already established links in their selected IR-ER paths. Hence, minimizing the number of required network resources.

5. Information about path selection and requested demands is exchanged over iBGP sessions. Note, however, in order to support large scale ISPs, the recommendation is to use route reflectors to reduce the number of iBGP sessions [128].

6. ISPs require their network to have low loss and delay. To this end, links are required to have a maximum link utilization threshold [120]. For example, in [120], the threshold is set to 80% of the link capacity.

Note, the aforementioned considerations are similar conceptually to those put forth by Teixeira et al. [124], which only consider minimizing hot potato changes as opposed to energy consumption.

The proposed framework is called Hot-Potato Low Energy Consumption (Hot-PLEC). As mentioned, it allows IRs to compute the paths, and hence the set of links, that will be carrying traffic to their selected ERs in a distributed manner. Each IR then advertises its selected paths to other IRs. Upon receiving advertised paths from peer IRs, an IR decides whether it should shift its paths. If an IR decides to shift its paths, the corresponding traffic will be shifted onto links that are also used by other IRs. In addition, each IR will ensure the utilization of all links is

within a given threshold. After convergence, there will be zero or more idle links. These links can then switched off by a network operator.

Apart from that, within the proposed framework, this chapter also presents a study on the following key factors:

1. The order in which IRs establish paths. This chapter introduces three methods: (i) most savings (MS), a technique where the IR that exhibits the largest saving establishes its respective IR-ER paths, (ii) round robin (RR), an approach where paths are established by IRs in a round robin fashion, and (iii) random order (RO), where IRs establish their IR-ER paths in a random order.

2. Path selection metric. This chapter studies the following traditional path selection metrics: i) shortest path (SP), where paths with the minimum number of hops are selected, ii) longest path (LP), a technique that prefers paths with the maximum number of hops and, iii) Random, which selects paths randomly. Note that random path selection is essentially similar to Equal-Cost Multipath (ECMP), as used by Constrained Shortest Path First (CSPF) [109].

In the section to follow, necessary notations are first introduced. After that, a mathematical model is presented in Section 6.2. Then Section 6.3 shows how HotPLEC satisfies the aforementioned design considerations. Section 6.4 presents the research methodology. This is followed by the experiment results in Section 6.5. The conclusion is presented in Section 6.6.

6.1 Network Model

The network is modeled as a directed graph $G(V, E)$, with V being the set of nodes connected by edges in E. Each link e has capacity c_e. Occasionally, $e = (i, j)$ is used to indicate the link connecting node i and j. Let $\mathcal{I} \subset V$ and $\mathcal{E} \subset V$ be the set of IRs and ERs respectively; it is assumed that $\mathcal{E} \cap \mathcal{I} = \emptyset$. The remaining routers are

denoted as $V' = V - \{\mathcal{E} \cup \mathcal{I}\}$. The set of all communication pairs between IRs and ERs is denoted as Θ. Each pair $(r, s) \in \Theta$ has demand d_{rs}. In addition, the set of links originating from an IR is denoted as $E^s = \{(i, j) \mid i \in \mathcal{I}, j \in \mathcal{E}\}$. Conversely, links terminating at an ER are recorded in the set $E^t = \{(i, j) \mid i \in \mathcal{I}, j \in \mathcal{E}\}$. Each link and router consumes $\mathcal{E}_{\mathcal{L}}$ and $\mathcal{E}_{\mathcal{R}}$ Joule/s when active.

Let P_{rs} be the set of all simple paths for pair (r, s). Note, an IR will have a different set of paths for each ER. For each IR r, its chosen path for ER s is denoted as p^*_{rs}. Let $\delta^{rs}_{e,p}$ be an indicator variable that is set to one if path $p \in P_{rs}$ uses link e to carry demand d_{rs}. The set of paths using a link e is $P_e = \{p \mid \delta^{rs}_{e,p} = 1, \forall p \in P_{rs}, \forall (r, s) \in \Theta\}$. Hence, the total traffic over a given link is $L_e = \sum_{p \in P_e} B(p)$, where the function $B(p)$ returns the demand transmitted on path p. In terms of link utilization, $u_e = L_e/c_e$. Lastly, the set of links traversed by path p is denoted as E_p, and its cost as $C_p = \sum_{e \in E_p} u_e$. The value of C_p is affected by two factors: path length and link utilization. Specifically, a longer path or a path that uses links with a higher load will increase its cost. Therefore, a lower cost path is favorable since it offers shorter hop counts and/or uses less congested links. To ease readability, Table 6.1 summarizes all common notations.

6.2 Mathematical Model

This section presents a model that captures a network operator's goal to conserve power usage and IRs' aim to use the least cost path whilst considering the maximum link load. The latter is important as a high load will lead to increased delays and possibly packet loss, despite great power savings.

The problem at hand is modelled using Bilevel Programming (BP). Briefly, BP is characterized by a two-tiered formulation, where the upper and lower levels correspond to the leader (network operator) and followers (IRs) respectively [129]. In general,

Table 6.1: Summary of notation

Variable	Description
$G(V, E)$	Nodes and edge representation
$\|V\|, \|E\|$	Cardinality of set V and E, respectively
e	Link connecting nodes i and j
c_e	Capacity of link e
p	A given path
$\mathcal{I}$	Set of IRs
$\mathcal{E}$	Set of ERs
Θ	Set of all communication pairs between IRs and ERs
(r, s)	Source-destination pair
d_{rs}	Requested demand for pair (r, s)
E^s	The set of links originating from an IR
E^t	The set of links terminating at an ER
$\mathcal{E}_{\mathcal{L}}$	Energy in Joule/s consumed by a link when active
$\mathcal{E}_{\mathcal{R}}$	Energy in Joule/s consumed by a router when active
P_{rs}	Set of all simple paths for pair (r, s)
p^*_{rs}	For each IR r, chosen path for ER s
$\delta^{rs}_{e,p}$	Set to 1 if path $p \in P_{rs}$ uses link e to carry demand d_{rs}
P_e	Set of paths using link e
L_e	Total traffic over link e
E_p	Set of links traversed by path p
u_e	Utilization of link e
C_p	Cost of path p

$$\min_{y \in Y} \phi(x(y), y) \tag{6.1}$$

where

$$x(y) = arg\ min_{x \in X} f(x, y) \tag{6.2}$$

In words, the leader wishes to select a strategy y that minimizes $\phi(.)$, where the response to y, i.e., a strategy x, minimizes the followers' objective function $f(.)$.

In this case, an operator's goal is to minimize its operating cost, which can be achieved by switching off links and routers. Let $\mathcal{X} \in \{0, 1\}^{|E|}$ and $\mathcal{R} \in \{0, 1\}^{|V'|}$ be vectors with binary variables X_e and R_j that indicate whether link e and router j are active respectively. Formally, the network operator aims to solve the following mathematical program,

$$\min_{\mathcal{X}, \mathcal{R}} \sum_{e \in E} \mathcal{E}_{\mathcal{L}} X_e + \sum_{j \in V'} \mathcal{E}_{\mathcal{R}} R_j \tag{6.3}$$

such that,

$$|I_j|R_j \geq \sum_{a \in I_j} X_a, \quad \forall j = 1 \dots |V'| \tag{6.4}$$

$$u_e \leq \gamma c_e, \qquad \forall e \in E \tag{6.5}$$

$$\sum_{e \in E^s} L_e = \sum_{e \in E^t} L_e \tag{6.6}$$

$$X_e, R_j \in \{0, 1\} \tag{6.7}$$

where I_i is the set of incident links of router i. The binary variables X_e and R_j indicate whether link e and router j are up. Constraint (6.4) ensures that a router R_j is only switched off if all its incident links are off. Constraint (6.6) ensures flow conservation. Note, in constraint (6.5), $u_e = \frac{L_e}{c_e}$ is the solution obtained from the lower level mathematical program. Here, $\gamma \in [0, 1]$, and is a constant predetermined by the network operator to set its desirable MLU.

The followers or IRs aim to route traffic to their corresponding ERs using the least cost path. Specifically, given a "new topology", which is derived from $\mathcal{X}$ and $\mathcal{R}$, by the network operator, route the demand of all ingress-egress pairs over the new topology such that the total cost, i.e., C_p, of each selected path is minimized subject to capacity constraint. Let $\mathbf{P^{rs}_p}$ be a binary path decision variable that indicates whether path p from the set of paths P_{rs} is selected to carry traffic. Formally, the following lower level mathematical program is presented,

$$\min_{\forall (r,s) \in \Theta} C_{p^*_{rs}} \tag{6.8}$$

such that,

$$\frac{L_e}{c_e} \leq \gamma c_e, \qquad \forall e \in E \tag{6.9}$$

$$\sum_{p \in P_{rs}} \mathbf{P^{rs}_p} = 1, \qquad \forall (r, s) \in \Theta \tag{6.10}$$

$$|E_p|\mathbf{P^{rs}_p} \leq \textstyle\sum_{e \in E_p} X_e, \quad \forall p \in P_{rs}, \forall (r, s) \in \Theta \tag{6.11}$$

In the above formulation, constraint (6.9) ensures the total utilization is within a given threshold; recall from Section 6.1 that L_e is equal to the sum of all demands carried by paths traversing link e. Note also that the load of each link is determined by the chosen paths and their corresponding demand. Constraint (6.10) ensures a (r, s) pair only uses one path to carry traffic. This constraint also ensures that an IR is always connected to an ER. Otherwise, if one or more links on all paths to an ER is inactive, then this constraint is not satisfied because no $\mathbf{P}_{\mathbf{p}}^{\mathbf{rs}}$ can be set to one. Lastly, constraint (6.11) ensures a path is selected only if all its links are enabled by the network operator.

Figure 6.1 is now used to illustrate some of the aforementioned constraints and variables. Figure 6.1 shows a network with node 1 and 4 as the IR and ER respectively. As mentioned, the network operator seeks to turn off as many links and nodes as possible whilst minimizing the utilization of each link. The latter objective is dependent on IRs, whilst the former objectives involve setting as many X_i to zero as possible subject to the given constraints. Note that in this example each router has two incident links. Hence, a router can only be switched off if both links are off. For example, the constraint (4) of router R2 is $2R_2 \geq X_1 + X_4$, where $R_2 \in \{0, 1\}$. If both X_1 and X_4 are zero, then R_2 can be set to zero. Otherwise, R_2 must be set to one. For link X_1, only the path P_1^{14} traversing it. Assuming $d_{14} = 1$ and $P_1^{14} = 1$, we thus have $L'_{12} = L'_{24} = 1$. An example of constraint (6.11) is as follows. Consider path P_1^{14}. In order to use it, both X_1 and X_4 need to be active, i.e., $2P_1^{14} \leq X_1 + X_4$. Hence, if X_1 or X_4 or both are zero, then P_1^{14} must be zero too.

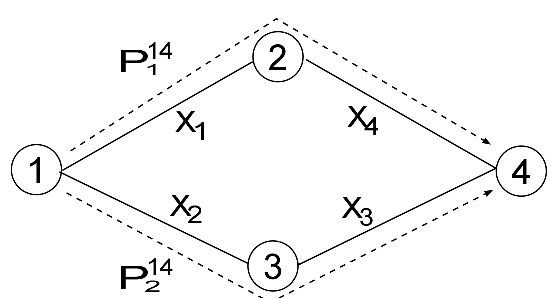


Figure 6.1: Example for problem formulation

The aforementioned BP formulation is NP-hard. In fact, as mentioned in [130], even linear versions of BP problems are NP-hard. In this case, the network operator is faced with a capacitated network design problem [131], which is a generalized version of the minimum Steiner tree problem; briefly, this problem asks for a minimum length/cost connection of *all* IRs and ERs with the possibility of adding intermediate nodes and edges. Note that the network operator's problem is similar to the NP-hard virtual private network design problem of [132]. On the other hand, the IRs are faced with the multi-commodity flow problem [133], which is NP-complete for integral solutions; in this case, the demand between a pair of IR-ER can only be routed over one path. Moreover, there may be an exponential number of paths to consider with increasing network sizes.

In light of these difficulties, the next section proposes a practical, fully distributed approach that allows IRs to decide amongst themselves the best path to establish to each ER whilst taking into account energy savings and link load.

6.3 The HotPLEC Framework

The main idea is for each IR to determine, based on updates from other IRs, a set of paths to ERs that will lead to overall network savings; i.e., use the fewest links. Specifically, each IR will first establish the least cost path to each ER of interest, and propagate information about its established paths to other IRs. Upon receiving an update from all IRs, each IR then decides, based on established paths by other IRs, whether moving or re-establishing its existing paths maximize a given benefit. For example, if the said benefit is the most savings, then the IR with the highest saving is the one that re-establishes its paths. The process then repeats and ends when no IRs deem it beneficial to change paths. In this case, it is said the IRs have *converged* onto their chosen path for each ER.

In the proposed framework, the following assumptions are made. First, each IR establishes its paths as MPLS tunnels. This ensures any changes in IGP cost do not induce any BGP changes [124]. Second, it is assumed that IRs are capable of identifying, for example using OpenFlow [134], when the network load is low. That is, the overall network load is below a given threshold predefined by a network operator. In particular, during off-peaks periods, upon detecting a drop in traffic load below a given threshold, IRs initiate HotPLEC. Finally, it is assumed that IRs and ERs are interconnected by iBGP. The resulting channel will be used to exchange information about path selection, IRs' ID, and requested demands. Note, as is the case in large scale ISPs, there may be high number of iBGP sessions. Hence, route reflectors are required [128].

Next, the key components used by HoPLEC are introduced, namely, (i) path selection, (ii) link utilization considerations, (iii) metrics, (iv) messages, and (v) saving.

1. **Path selection.** As mentioned, IRs need to decide whether to move their existing paths. In particular, after receiving updates, an IR needs to decide whether to re-establish its paths. The following options are studied:

   - *HotPLEC Most savings (HotPLEC-MS)*. In this approach, the IR r that exhibits the largest energy saving, explained later, establishes its chosen paths p^*_{rs} to each of its selected ERs s.
   - *HotPLEC Round robin (HotPLEC-RR)*. This method assumes all IRs are aware of each other's ID. IRs then establish paths in a round robin fashion; e.g., in increasing ID order.
   - *HotPLEC Random order (HotPLEC-RO)*. Each IR will input its ID into a pseudo-random function. The output is then exchanged with other IRs. This output is then used to decide the round robin order. For example,

IRs can be programmed to establish their IR-ER paths in increasing order of the said output.

Note, an IR selects a p^*_{rs} path with sufficient bandwidth to accommodate the demand d_{rs}. If this is not the case, the next shortest path in P_{rs} will be considered. If none of the paths in P_{rs} are able to serve d_{rs}, it is rejected. Note that rejection of demands may cause an increase in end-to-end delays and packet loss. In addition to bandwidth, IR also considers link utilization constraints, which will be explained next.

2. **Link utilization.** In order to avoid congestion and packet loss the framework considers a number of link utilization constraints. The maximum average link utilization (MALU) threshold is defined, where the average link utilization (ALU) is calculated as follows,

$$ALU = \frac{\sum_{e=1}^{|E|} u_e}{|E|} \tag{6.12}$$

The presented framework also restricts link utilization to a predefined maximum link utilization (MLU) value; i.e., $u_e \leq MLU$. This MLU value is a percentage of the total capacity of a given link e; i.e., γc_e, with $0 \leq \gamma \leq 1$. These parameters, namely MALU and MLU, are predefined by the network operator. Also, the MLU can be set to a value that allows a link to absorb any sudden burst in traffic. In particular, as per [120], to keep delay low, the MLU is set to 80%.

3. **Metrics.** As mentioned, IRs select paths that yield maximum energy savings. This is facilitated by the following metrics,

- *IR Saving Unit (ISU).* This metric identifies the IR that is able to achieve the highest savings. ISU is used in the *HotPLEC-MS* approach. It is

calculated as follows,

$$ISU = \sum_{h=1}^{|\mathcal{E}|} |R_h| + |S_h| - |N_h| \tag{6.13}$$

where R_h is the set of reused links, S_h is the set of links that can be put to sleep, and N_h denotes new links to be made active in order to carry IR's demands; i.e., d_{rs}. Note that ISU is an integer with no measurement unit.

- *Original-length*. The value of this metric for a given path p is defined as the number of edges that p uses. For example, if $p = [2-3-5]$, it is said that the "*Original-length*" of path p is 2, as it only contains two edges, $[2-3]$ and $[3-5]$
- *New-length*. IRs use this metric in order to identify the set of paths that will allow them to maximize energy savings. The "*New-length*" of p is calculated by decreasing its "*Original-length*" by one every time path p re-uses an already established link; i.e., one that is used by other paths.

4. **Messages.** Up to this point it has only been outlined how IRs exchange information by using messages already defined in the BGP protocol, i.e., *UPDATE* messages. However, the *HotPLEC-MS* approach employs three new messages, all of which are exchanged over iBGP. The messages are,

- *UPDATE-INIT*. This message advertises the different p^*_{rs} paths selected by each IR using the least cost path. Each p^*_{rs} path is represented as a vector of routers, each of which is identified by its IP address. The message also contains the respective d_{rs} demand of each path and the ID of the IR originating the message.
- *UPDATE-ISU*. This message carries the ISU metric of the sending IR.

- *UPDATE-WIN*. This message is sent by the IR that holds the largest ISU metric and contains the ID of the winning IR and its respective p^*_{rs} paths and d_{rs} demands.

5. **Saving.** In order to evaluate the performance of HotPLEC, see Section 6.5, the measure of the total saving (TS) between HotPLEC and an alternative approach is as follows,

$$TS = \frac{(A_O) - (A_H)}{(A_O)} * 100\% \tag{6.14}$$

where A_H is the final number of active links used by HotPLEC, and A_O the number of the active links used by an alternative approach, e.g., shortest path (SP).

6.3.1 Implementation

This section explains how all the components are integrated. Figure 6.2 depicts the flow chart of the framework that is used by each IR. At the start, i.e., *round zero*, each IR selects a path to ERs by selecting one whose intra-domain distance or IGP cost is the smallest. The IR then calculates the shortest IR-ER paths to its selected ERs and informs all other IRs. For each subsequent *rounds*, upon receiving all the relevant iBGP messages, the IR calculates, for each ER, the best path according to a given path selection criteria. For example, if the IR is using *HotPLEC-MS*, "best" refers to the shortest path that yields the maximum saving without exceeding link capacity. A new iBGP *UPDATE* message is then sent to other IRs informing them of its path selection.

Upon receiving all the updates, and according to the path selection criteria, an IR will decide if it is its turn to establish paths. Specifically, for HotPLEC-MS, an

IR that exhibits the largest savings would establish its paths, while for HotPLEC-RR and HotPLEC-RO, the next scheduled IR will establish its paths. If the IR determines that it is its turn to establish paths and that all the relevant d_{rs} demands can be satisfied, it will then establish its p^*_{rs} paths and will inform other IRs by using an *UPDATE* message. On the contrary, if it is not its turn to establish paths or a demand cannot be satisfied, the IR will return to the "wait for all messages" state. If none of the paths within the P^*_{rs} set are able to accommodate the demand d_{rs}, the IR-ER path will not be established and the requested demand is rejected. Note, the time in which an IR remains in the said wait state will increase the convergence time of each method and is directly related to the topology size. Convergence times are presented and discussed in Section 6.4.

The above process continues until no IRs make any changes, at which time IRs would have converged to a set of paths that yield the maximum savings. Upon completion, the network may have a number of links with zero utilization. The network operator then has the option to switch these links off.

A worked example for *HotPLEC-MS* is presented. The example uses the topology named *7N18L*, shown in Figure 6.3 – it consists of seven nodes and 18 links. It is assumed that the capacity of each link, c_e, is unlimited. Let $\mathcal{I} = \{1, 2, 4\}$ and $\mathcal{E} = \{7\}$ correspond to the set of IRs and ERs nodes respectively. Therefore, $\Theta = \{[1, 7], [2, 7], [4, 7]\}$ is the corresponding IR-ER pairs set. Each IR calculates $k = 3$ shortest paths to each of its respective ERs. Table 6.2 presents $k = 3$ shortest paths for demands in Θ. The selected paths p^*_{rs} are displayed in the first column, and the k-shortest paths in the second column. The third column shows the initial selected paths for each $(r, s) \in \Theta$. Each IR sends an *UPDATE-INIT* message after establishing their initial shortest path to each ER. Once an IR receives all *UPDATE-INIT* messages, it computes new IR-ER paths based on already established links by making use of the "*New-length*" parameter. For a given IR-ER pair, the IR will select as candidate the path with the smallest "*New-length*". If two paths are found

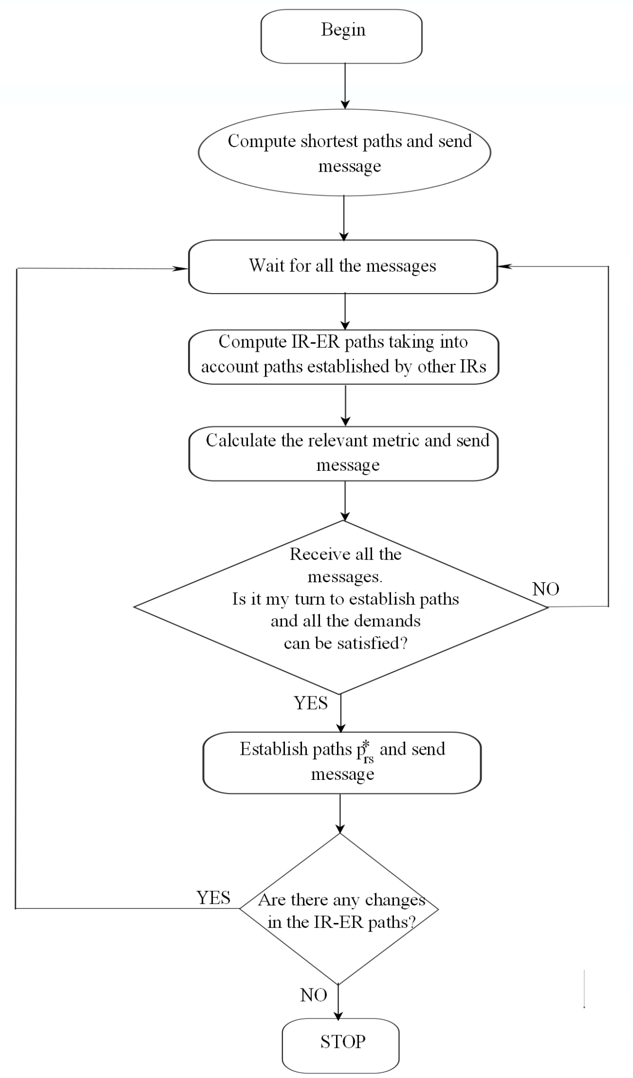


Figure 6.2: Flow chart for the proposed framework

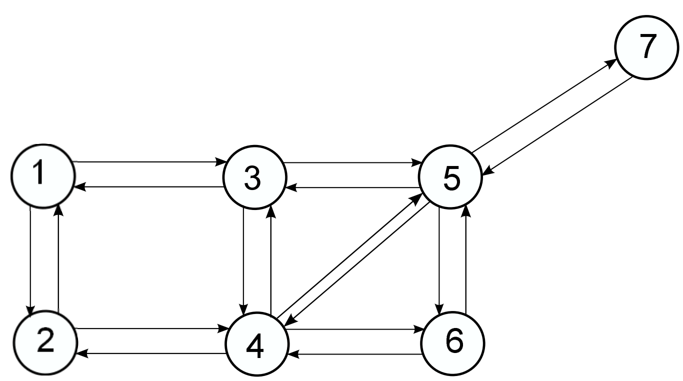


Figure 6.3: Example topology 7N18L, seven nodes and 18 links

to have the same "*New-length*", the shortest "*Original-length*" will be used to break the tie. In the case where the tie persists, i.e., the "*Original-length*" of the two paths are the same, the first of those two paths is chosen. Consider node 1. The "*Original-lengths*" of its k shortest paths $[1-3-5-7], [1-2-4-5-7]$, and $[1-3-4-5-7]$ are 3, 4, and 4 respectively. Node 1 then considers the paths established by the other IRs. That is, the path $[2-4-5-7]$ selected by node 2, and the path $[4-5-7]$ selected by node 4. Node 1 will then use these paths in order to calculate the "*New-length*" parameter of its own paths. Specifically,

- *Path 1:* $[1-3-5-7]$. This path would only reuse one single link, i.e., $[5-7]$. Therefore, its "*New-length*" is equal to $3-1=2$. This means that, if selected, the path requires the set up of two new links: $[1-3]$ and $[3-5]$.

- *Path 2:* $[1-2-4-5-7]$. This path would reuse links $[2-4]$, $[4-5]$ and $[5-7]$. Therefore, its "*New-length*" is equal to $4-3=1$. Here, only one new link, $[1-2]$, is necessary.

- *Path 3:* $[1-3-4-5-7]$. This path would reuse two links, $[4-5]$ and $[5-7]$, hence its "*New-length*" is equal to $4-2=2$, meaning it requires two new links $[1-3]$ and $[3-4]$ if selected.

Given the above results, node 1 will then define path $[1-2-4-5-7]$ as one with the smallest "*New-length*" and therefore, it will select that path.

For the second IR, node 2, paths established by the other IRs, nodes 1 and 4, are $[1-3-5-7]$ and $[4-5-7]$ respectively. Using the same procedure as node 1, it is found that a tie exists between paths $[2-4-5-7]$ and $[2-1-3-5-7]$ as their "*New-lengths*" are equal. That is, path $[2-4-5-7]$ can reuse two links, $[4-5]$ and $[5-7]$, and its "*New-length*" is equal to $3-2=1$. On the other hand, path $[2-1-3-5-7]$ can reuse three paths $[1-3]$, $[3-5]$ and $[5-7]$, which results in a "*New-length*" of $4-3=1$. In order to break the tie, the shortest of these two paths, $[2-4-5-7]$, is selected. For the third IR, node 4, it considers

the paths selected by node 1 and 2, $[1-3-5-7]$ and $[2-4-5-7]$ respectively. The "*New-length*" of its shortest k paths, see Table 6.2, are $2-2=0$, $3-2=1$, and $3-1=2$, therefore, the selected candidate path is $[4-5-7]$.

Every IR then calculates its respective ISU metric. Concretely,

- *Node 1.* Current IR-ER path is $[1-3-5-7]$ and the candidate path $[1-2-4-5-7]$. Links reused, $[2-4]$, $[4-5]$ and $[5-7]$. Links to be shut down, $[1-3]$ and $[3-5]$. Finally, new links to be set up $[1-2]$. Therefore, $ISU=(3)+(2)-(1)=4$,

- *Node 2.* The metric is calculated as $ISU=(2)+(0)-1=1$, and

- *Node 4.* $ISU=(2)+(0)-(0)=2$.

Hence, the IR that exhibits the largest ISU is node 1, and it is the only one that is allowed to establish its selected candidate path $[1-2-4-5-7]$. Node 2 and 4 keep their original shortest paths, $[2-4-5-7]$ and $[4-5-7]$, see fourth column of Table 6.2. Notice that for this particular example, there are no more changes in the selected (r,s) paths, so the algorithm converges.

The number of active links then used by HotPLEC is 4, $[1-2]$, $[2-4]$, $[4-5]$ and $[5-7]$. When the exercise is repeated using the SP traditional approach, it is found that SP will end up with 5 active links, $[1-3]$, $[3-5]$,$[5-7]$, $[2-4]$ and $[4-5]$. By employing Equ (6.14), as compared to SP, the total saving achieved by HotPLEC is $[(5-4)/4]*100\%=20\%$.

6.4 Evaluation

This section starts by introducing the *optimum solution* in the form of a binary integer programming (BIP). Figure 6.1 will be used to show an instance of a BIP formulation. Let $C_e=C_{ij}$ be the capacity of link e with end nodes i and j. For each demand, K shortest paths are defined, denoted as p^k_{rs} and indexed by k. Let

Table 6.2: $k = 3$ shortest paths for $\Theta = \{[1,7],[2,7],[4,7]\}$

IR-ER pair	Shortest paths	Selected shortest path (round zero)	Selected shortest path (round one)
[1,7]	[1-3-5-7] [1-2-4-5-7] [1-3-4-5-7]	[1-3-5-7]	[1-2-4-5-7]
[2,7]	[2-4-5-7] [2-1-3-5-7] [2-4-3-5-7]	[2-4-5-7]	[2-4-5-7]
[4,7]	[4-5-7] [4-3-5-7] [4-6-5-7]	[4-5-7]	[4-5-7]

P_e contain the decision variables, i.e., p^k_{rs}, corresponding to the paths that traverse link e. Let Θ_e contains all demands (r, s) with a path that traverses link e. Finally, let x_e be a binary link decision variable that indicates whether link e is active. Consequently, the following BIP is presented,

$$\min_{\forall e \in E} \sum x_e \tag{6.15}$$

Subject to,

$$\sum_{k=1}^{K} p^k_{rs} = 1, \quad \forall (r,s) \in \Theta \tag{6.16}$$

$$x_e \geq p^k_{rs}, \quad \forall e \in E, \forall p^k_{rs} \in P_e, \tag{6.17}$$

$$\sum_{f \in \Theta_e, p^k_f \in P_e} d_f * p^k_f \leq C_e, \quad \forall e \in E \tag{6.18}$$

In words, the objective is to minimize the number of active links by selecting a path for each demand. The constraints include: (i) only one path can be selected for each demand (Equ. 6.16), (ii) a link is active only if there is a path that uses it (Equ. 6.17), and (iii) the total demands on each link does not exceed the link's capacity (Equ. 6.18). This BIP formulation thus serves as a baseline that allows us to evaluate the performance of HotPLEC against a centralized scheme that yields a

set of routes resulting in maximum energy saving. In all the experiments, the value of K is set to 3.

Similar to the evaluation of HotPLUZ, the performance of all methods is validated by using five topologies: SURFnet (50 nodes, 138 links), GEANT (40 nodes, 122 links), AT&T (25 nodes, 112 links), Abilene (11 nodes, 28 links) [112] [117] [113] and a the fabricated topology 8N24L (8 nodes, 24 links). Traffic matrices (TMs) are obtained using the well known gravity model [118] [119]. Simulations are conducted in MATLAB [114]. For each topology, a base TM, called TM_{Base} is generated. Each node is then randomly classified as an ER or IR. After that, each IR selects num ERs randomly, where $1 < num < |E|$. The number of simulation runs is 100 for different number of ERs, i.e., $num = 2, 3, 4, 5, 6$, which result in a different number of IR-ER pairs per topology. In all the cases, an IR selects a new path as long as the new path has sufficient bandwidth to support a given IR-ER traffic demand. Only IR-ER traffic is considered. This means links that do not carry IR-ER traffic have zero utilization. The average final number of active links for each topology is reported.

It is also reported and compared against the results obtained by the framework, the average final number of active links used by three well known approaches: i) Shortest path (*SP*). This approach will select as candidate path, p, for a given (r, s) pair, the path with the minimum number of hops among the corresponding P_{rs} paths. Path p will become the selected path p^*_{rs}, only if it contains enough bandwidth to accommodate the requested d_{rs} demand. If this is not the case, the next shortest path in P_{rs} will be considered, ii) Longest path (*LP*). In contrast to SP, LP will select among the P_{rs} paths for a given (r, s) pair, the path with the maximum number of hops as the candidate path p. Path p will only be promoted to selected candidate path, p^*_{rs}, if it is able to accommodate the requested demand d_{rs}. In case p is not able to do so, the next longest path in P_{rs} will be considered, and iii) Random. For a given (r, s) pair, this algorithm will randomly select one of

the paths, p, in the P_{rs} set to be the candidate path to carry the corresponding d_{rs} demand. If p does not have the sufficient bandwidth to carry such demand, another randomly selected path in P_{rs} will be selected as the new candidate path. For all the three approaches, if none of the paths in P_{rs} are able to serve the requested demand, the path for that particular (r, s) pair will be declared as "empty" and the corresponding d_{rs} demand rejected. All the evaluated approaches comply with predefined link utilization constraints, i.e., $MLU = 80\%$. In addition to the number of active links for all the approaches, the *convergence times* and number of *rounds* for all the methods introduced by the framework are also presented. The presented results are within 95% confidence interval.

6.5 Results

The results of the investigation on the impact of varying IR-ER pairs are now presented; different number of IR-ER pairs are obtained by varying the number of ERs, $num = 2, 3, 4, 5, 6$. Figure 6.4 and 6.7 show the number of active links for the 8N24L and Abilene topologies respectively. The HotPLEC variants show similar behaviors and achieve better performance than SP. Specifically, HotPLEC-MS, HotPLEC-RR and HotPLEC-RO use 3% and 4% fewer active links in 8N24L and Abilene respectively as compared to SP. HotPLEC-MS, HotPLEC-RR and HotPLEC-RO also use a smaller number of links than Random with 35% fewer links for 8N24L and 33% for Abilene. Similar results are also observed when compared against the LP approach; specifically, the presented approaches use 45% and 40% fewer number of links for 8N24L and Abilene respectively.

HotPLEC-MS, HotPLEC-RR and HotPLEC-RO were also found to use only 17% for 8N24L and 26% for Abilene more links than the optimal solution, i.e., BIP. Figure 6.4 shows that for 8N24L, all the HotPLEC variants present a similar performance in regards to the number of active links used. However, Figure 6.7

shows that for Abilene, the best two heuristics, HotPLEC-MS and HotPLEC-RR, exhibit different behaviours. In particular, HotPLEC-RR has 1% fewer number of active links than HotPLEC-MS.

Figure 6.5 and 6.8 show the convergence times of HotPLEC variants for the 8N24L and Abilene topologies respectively. HotPLEC-RR has the longest convergence time. Specifically, for 8N24L, HotPLEC-RR incurs the same convergence time as HotPLEC-MS and 13% longer than HotPLEC-RO. For Abilene, HotPLEC-RR takes 1% and 39% longer to converge than HotPLEC-MS and HotPLEC-RO respectively.

Figure 6.6 and 6.9 present the number of rounds incurred by HotPLEC variants for the 8N24L and Abilene topologies. In particular, for 8N24L, HotPLEC-RR requires 2% and 8% more rounds than HotPLEC-MS and HotPLEC-RO respectively. Similar results are observed for Abilene where the number of rounds employed by HotPLEC-RR surpasses the number of rounds of HotPLEC-MS and HotPLEC-RO by 2% and 30% respectively. These results are consistent with the results observed for the convergence times in Figure 6.5 and 6.8.

Table 6.3 presents the running time of BIP for 8N24L whereas Table 6.4 shows the BIP convergence times for Abilene. Notice that the convergence times for Abilene are longer than for 8N24L. Notice also that any results for the larger topologies, e.g., AT&T, GEANT, SURFnet are not included because BIP is intractable on these topologies. In particular, on average, the convergence times recorded for the BIP in 8N24L is 1150% shorter than those for the Abilene topology. For large topologies, no solutions can be obtained for the BIP model.

The reason BIP's convergence times increase with topology size can be explained via the developed BIP model; see Section 6.2. For each k-shortest IR-ER path and link, a binary integer variable is defined. If a topology with $|E|$ links and $|\Theta|$ IR-ER pairs is considered, where k shortest paths exist between each IR-ER pair, the resulting number of binary integer variables is $k \times |\Theta| + |E|$. The number of con-

Table 6.3: Running times (s) for the BIP in the 8N24L topology(8 nodes, 24 links).

# of IR-ER pairs	8N24L
4	0.2
6	0.55
8	0.9
10	1.3
12	1.2
Avg.	0.8

Table 6.4: Running times (s) for the BIP in the Abilene topology(11 nodes, 28 links).

# of IR-ER pairs	Abilene
10	3.5
15	9.5
20	11.5
25	11.9
30	9.7
Avg.	9.2

straints will be equal to $|E|$. Tables 6.3 and 6.4 show that, for a given topology, an increase in the number of IR-ER pairs results, on average, in longer convergence times. Similar results are observed when topologies have a large number of links. This is to be expected as BIP is NP-hard in general and computationally intractable for moderate-sized problem instances [135]. In contrast, for a given set of all communication pairs between IRs and ERs, denoted as Θ, the introduced heuristics need to explore for each of the (r, s) pairs in Θ, the different $|P_{rs}|$ shortest paths. Each of them with a maximum length of N. Therefore, the presented algorithms have a run time complexity of $O(|\Theta||P_{rs}|N)$.

Figure 6.10, 6.11 and 6.12 present the results for AT&T. Figure 6.10 shows how HotPLEC-MS and HotPLEC-RR exhibit the best performance, using 2% and 3% respectively, fewer links than the SP approach. Random and LP exhibit the worst performance. HotPLEC-MS and HotPLEC-RR use around 30% fewer number of links than Random and 33% fewer number of links than LP. Figure 6.10, 6.11 and 6.13 show the tradeof between performance and convergence times. The figures show

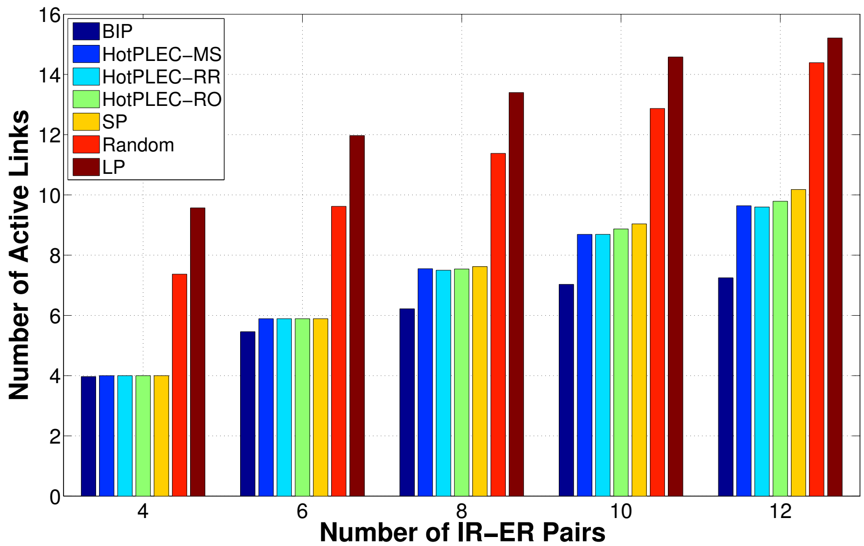


Figure 6.4: Average number of active links for the evaluated approaches in the 8N24L topology(8 nodes, 24 links).

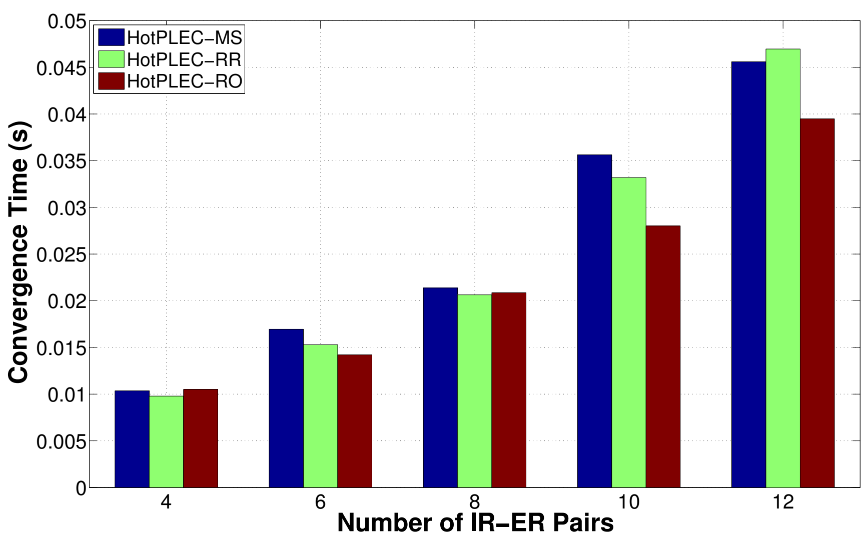


Figure 6.5: Average convergence times for the HotPLEC variants in the 8N24L topology(8 nodes, 24 links).

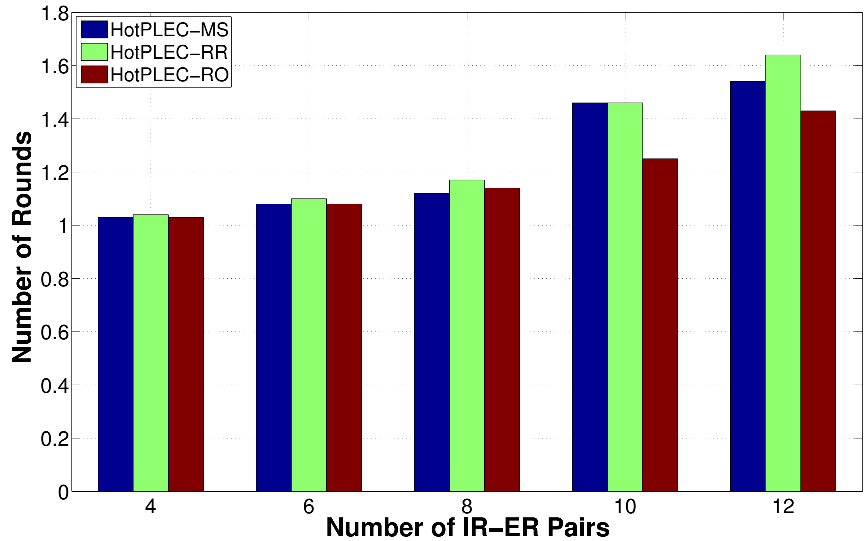


Figure 6.6: Average number of rounds for the HotPLEC variants in the 8N24L topology (8 nodes, 24 links).

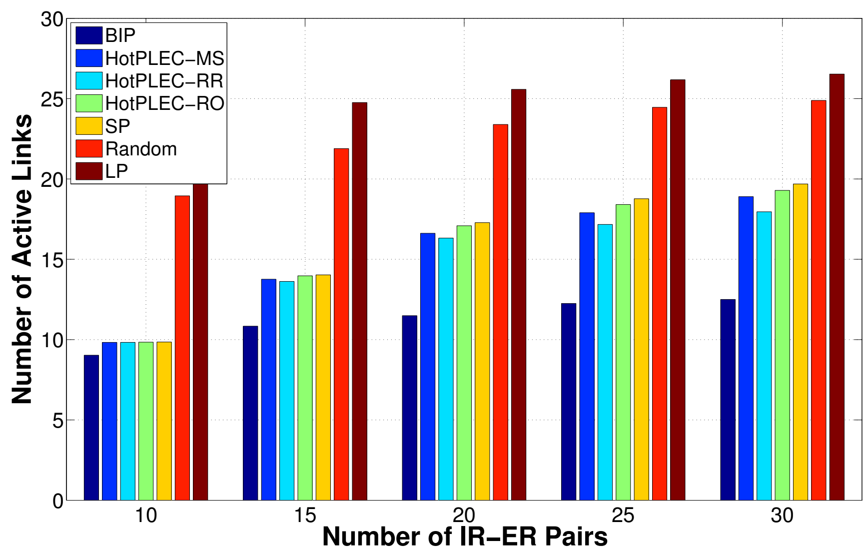


Figure 6.7: Average number of active links in the Abilene topology (11 nodes, 28 links).

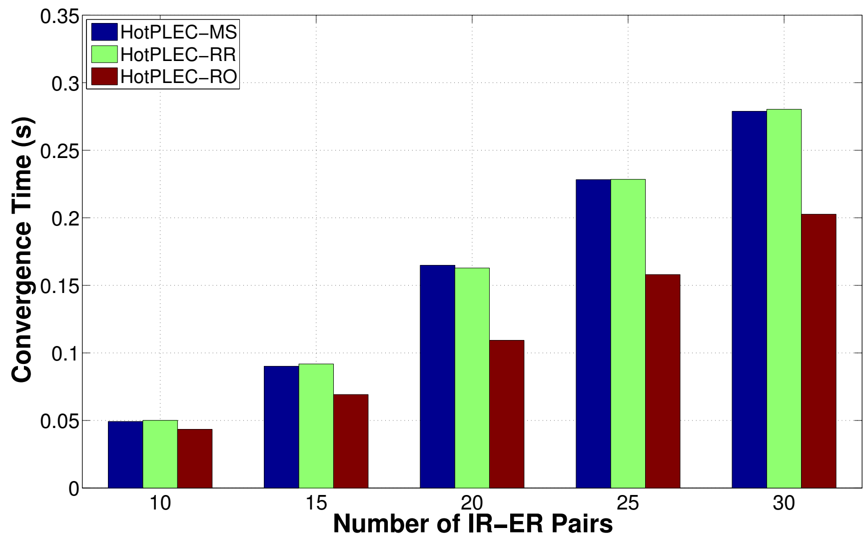


Figure 6.8: convergence times for the HotPLEC variants in the Abilene topology(11 nodes, 28 links).

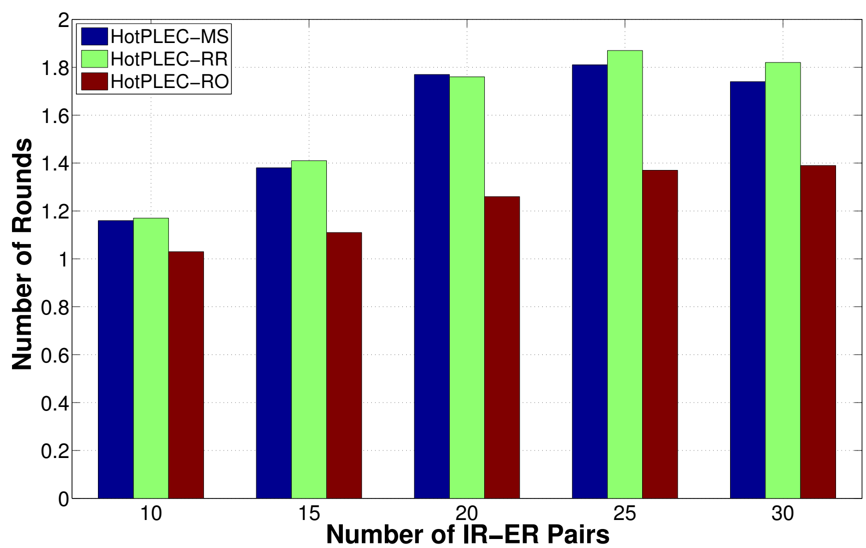


Figure 6.9: Average number of rounds for the HotPLEC variants in the Abilene topology(11 nodes, 28 links).

how HotPLEC-RR uses 1% fewer number of links than HotPLEC-MS, and takes 8% more time to converge than HotPLEC-MS requiring 10% more number of rounds.

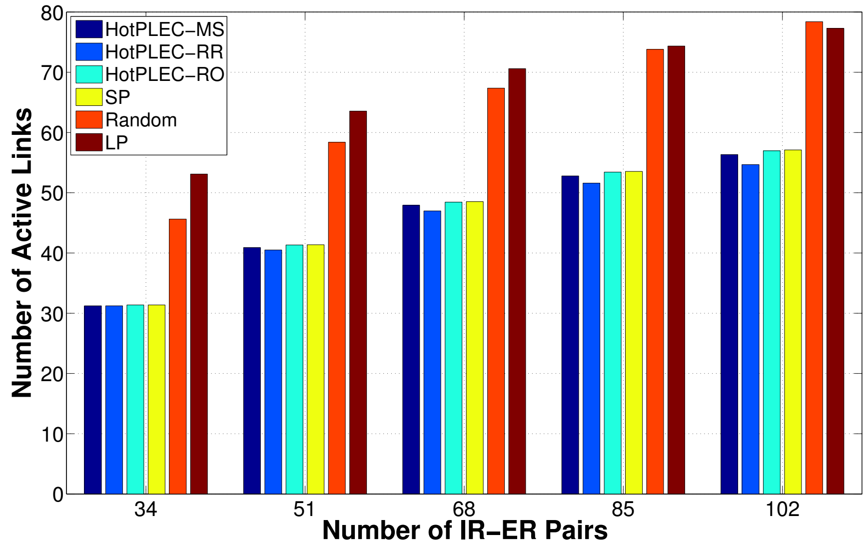


Figure 6.10: Average number of active links in the AT&T topology (25 nodes, 112 links).

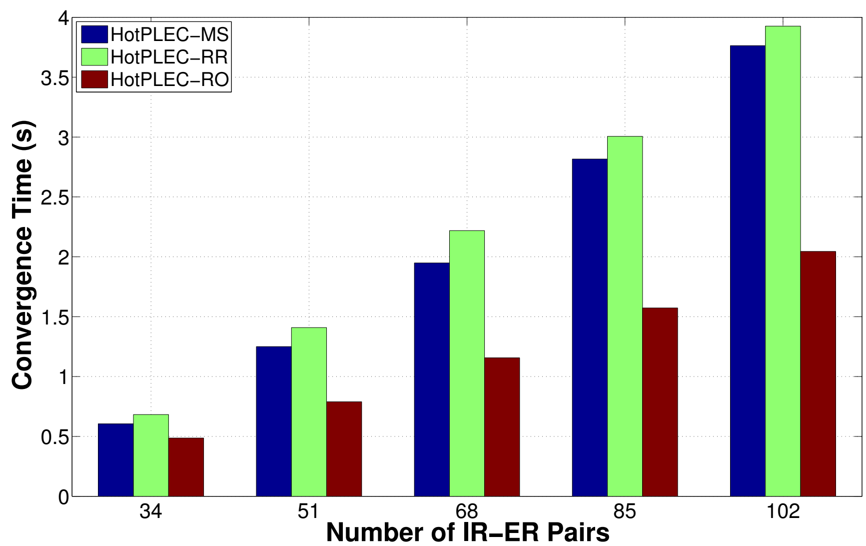


Figure 6.11: Average convergence times for the HotPLEC variants in the AT&T topology (25 nodes, 112 links).

Figure 6.13 and 6.14 and 6.15 present the obtained results for GEANT. Similar to the results obtained for AT&T, HotPLEC-MS and HotPLEC-RR are the best approaches using only respectively 2% and 3% fewer number of active links than the SP approach. On the other hand, the figures also show that Random and LP

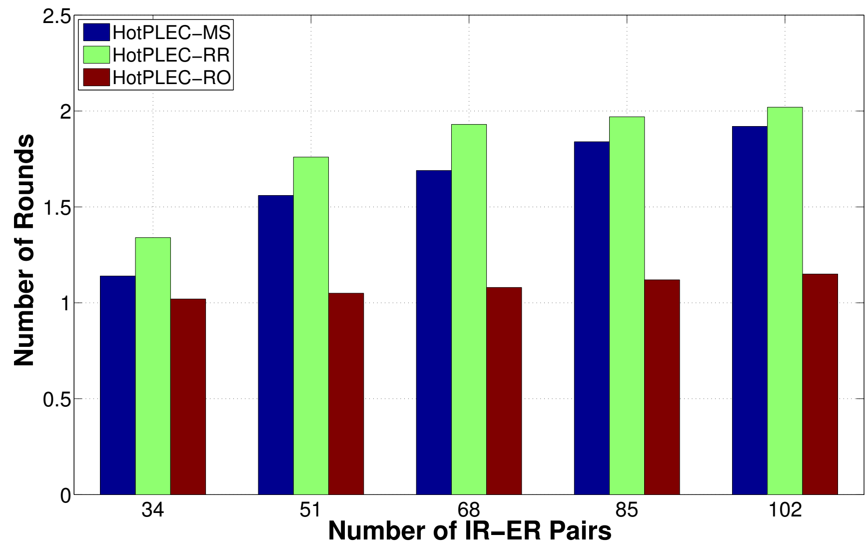


Figure 6.12: Average number of rounds for the HotPLEC variants in the AT&T topology (25 nodes, 112 links).

are the worst approaches. HotPLEC-MS and HotPLEC-RR use around 33% and 40% fewer links than Random and LP respectively.

Figure 6.13 also shows that the difference in performance between HotPLEC-RR and HotPLEC-MS is only 1%, with HotPLEC-RR using fewer number of active links. Despite the small 1% difference, Figure 6.14 shows HotPLEC-RR taking almost 26% more time to converge than HotPLEC-MS. This larger convergence time is the result of HotPLEC-RR requiring 25% more rounds than HotPLEC-MS.

Lastly, the results for SURFnet are shown in Figure 6.16 and 6.17 and 6.18. HotPLEC-MS and HotPLEC-RR are again the best approaches, whereas Random and LP are again the worst heuristics. In this case, HotPLEC-MS and HotPLEC-RR use approximately 2% and 5% respectively fewer active links than the SP approach. HotPLEC-MS uses 37% fewer links than the Random approach and 44% fewer links than LP. It was also found that HotPLEC-RR uses 40% fewer number of links than Random and 46% fewer links than LP. Similar to the results observed for AT&T, HotPLEC-RR presents 1% better performance than HotPLEC-MS with HotPLEC-RR taking 7% more time to converge than HotPLEC-MS while requiring 12% more number of rounds.

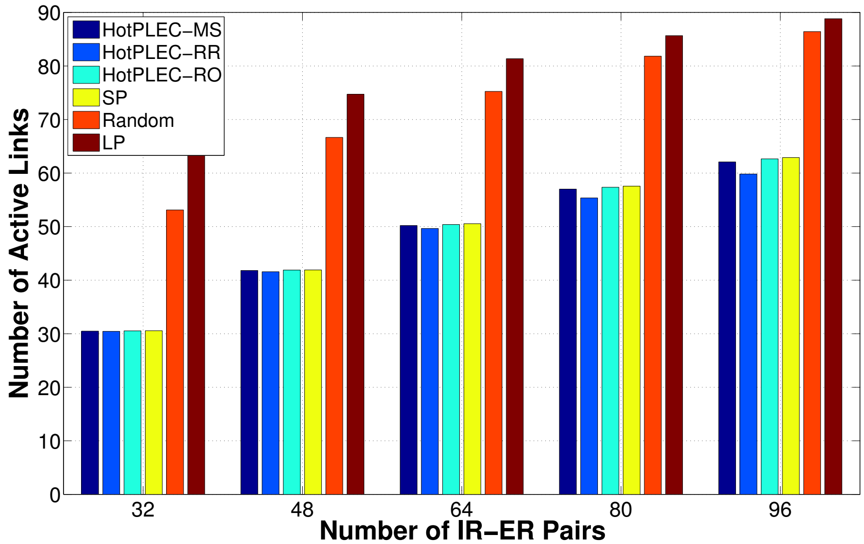


Figure 6.13: Average number of active links in the GEANT topology (40 nodes, 122 links).

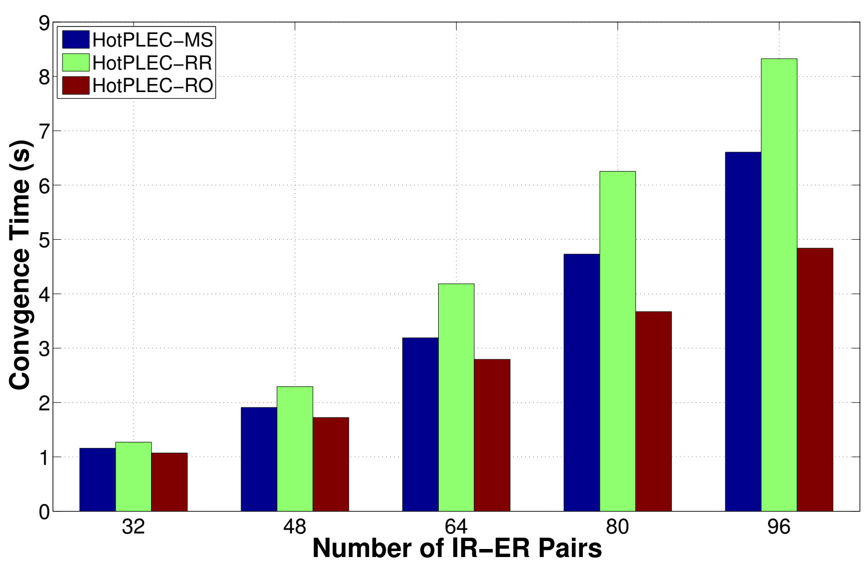


Figure 6.14: Average convergence times for the HotPLEC variants in the GEANT topology (40 nodes, 122 links).

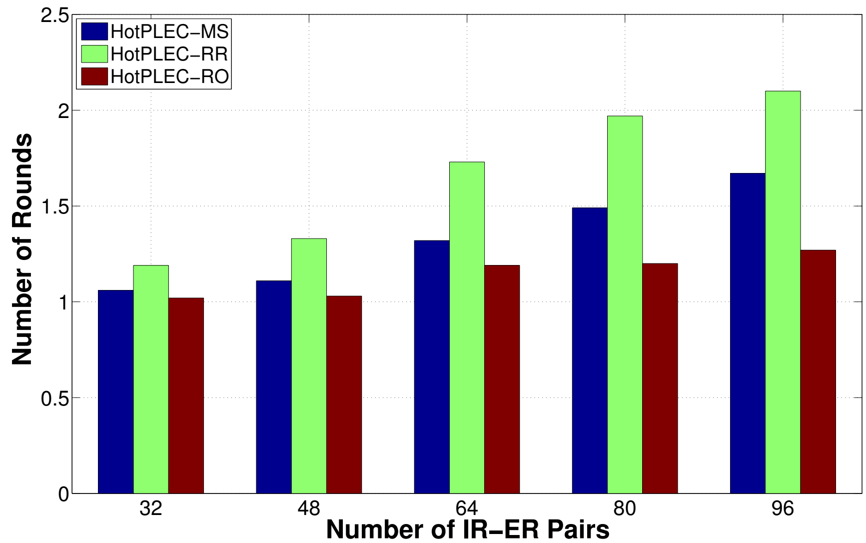


Figure 6.15: Average number of rounds for the HotPLEC variants in the GEANT (40 nodes, 122 links).

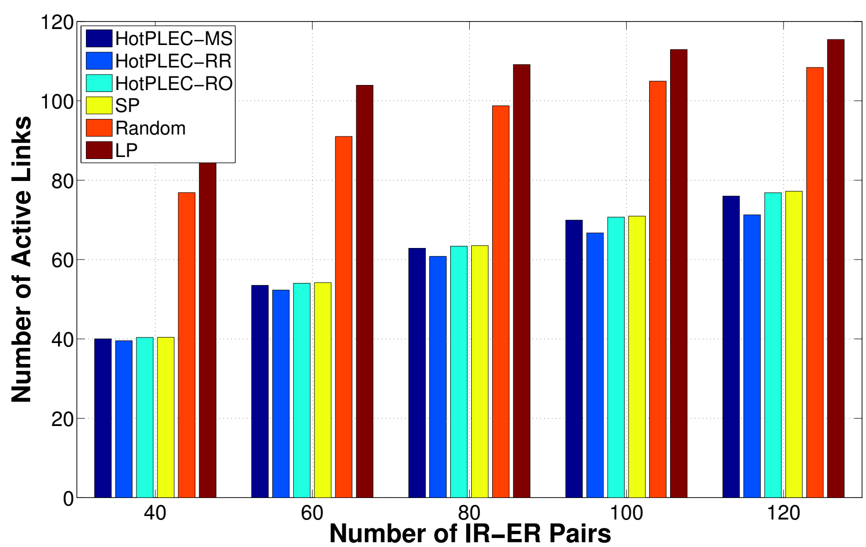


Figure 6.16: Average number of active links in the SURFnet toplogy (50 nodes, 138 links).

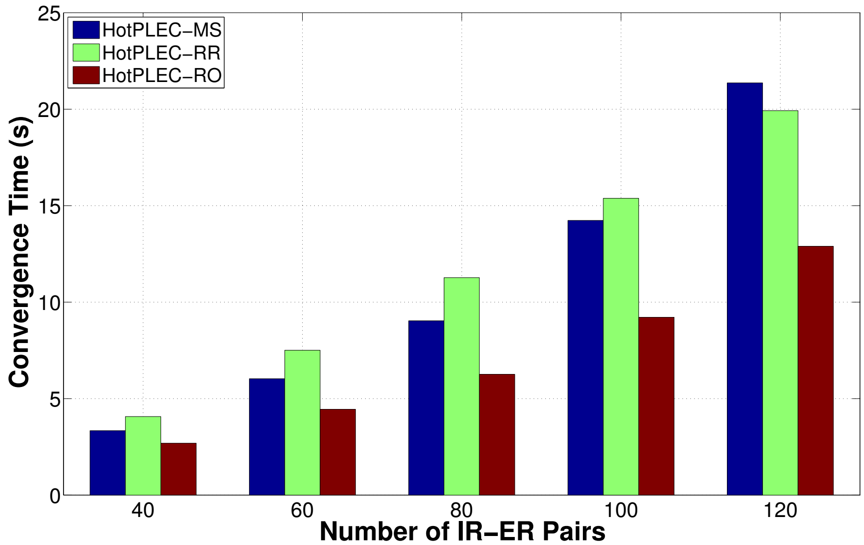


Figure 6.17: Average convergence times for the HotPLEC variants in the SURFnet topology (50 nodes, 138 links).

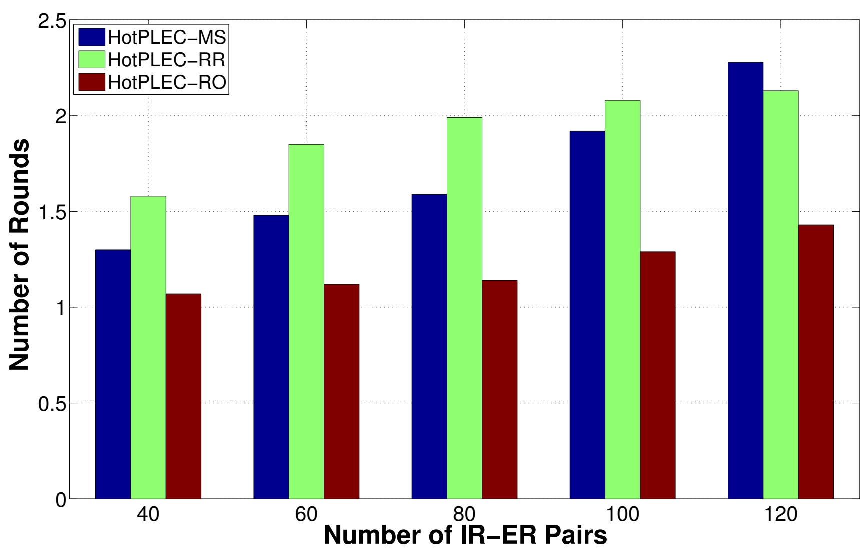


Figure 6.18: Average number of rounds for the HotPLEC variants in the SURFNET topology (50 nodes, 138 links).

6.5.1 Discussion

Overall, results indicate that HotPLEC-RR has the best performance using around 1% fewer links than HotPLEC-MS. However, results also indicate that HotPLEC-RR takes up to 23% more time to converge than HotPLEC-MS in some cases. IRs running HotPLEC-MS require the one with the largest number of *ISUs* to establish its IR-ER shortest paths first in each round; however, a larger number of ISUs does not necessarily indicate a larger number of reused links nor fewer number of active links. That is, *ISU* calculation considers inactive links as these directly affect the overall capability of an IR to save energy when routing IR-ER traffic.

On the other hand, HotPLEC-MS converges once the set of shortest paths that uses the fewest number of active links is obtained. As a result, HotPLEC-RR ends up with 1% fewer active links than HotPLEC-MS, but it also takes longer to converge. As topology size increases, the number of IR-ER pairs also increases. This affects convergence time as shown in Figures 6.5, 6.8, 6.11, 6.14 and 6.17.

The longer convergence times are mainly the result of larger number of rounds as observed in Figures 6.6, 6.9, 6.12, 6.15 and 6.18. Specifically, as the number of IRs increases, the total number of UPDATE messages also increases. Each IR therefore receives a larger number of UPDATEs containing information such as the shortest paths calculated by other IRs and their respective *ISU* metric. This means each IR needs to consider more IR-ER paths when deciding which links to reuse in order to route its respective IR-ER traffic. Consequently, convergence time increases.

6.6 Conclusions

This chapter has contributed another BGP-aware green TE approach. The proposed framework, called HotPLEC, is a distributed approach that allows IRs to reuse as many active links as possible. Unused links can then be shut down. Experiments show that up to 46% overall savings are achieved under low network load when

compared to establishing longest paths only, and up to 5% when compared to using shortest paths. For small topologies, HotPLEC uses as little as 17% more active links than the optimal solution, i.e., BIP, taking almost 27% less time to converge.

The green TE approaches presented so far have not taken into account uncertain traffic demands. In fact, current green TE techniques are not robust against random traffic demands. Therefore, the next chapter introduces the first robust, green TE solution that considers demands defined by a polyhedral set. Advantageously, the solution is robust to all demands while ensuring link utilization does not go above a pre-defined threshold.

Chapter 7

Green-PolyH: A Green Traffic Engineering Solution Over Uncertain Demands

Uncertainty in traffic demands plays a critical role in the design of TE techniques. There are two key components in the design of TE techniques: i) understanding traffic flows; i.e., traffic matrices, and ii) configuration and design of routing protocols. These two components are related. That is, an understanding of the traffic matrix and dynamics of traffic flows leads to better routing and network designs [136]. Unfortunately, as noted in [39], obtaining an accurate traffic matrix is expensive due to the volume in traffic. Moreover, predicting traffic demands is a difficult problem as they change over time, flow measurements are rarely available, and it is even harder to estimate origin-destination flow aggregates [137].

As indicated in Chapter 2, current green TE approaches have not considered uncertainty in traffic demands. Henceforth, this chapter reports the first robust, green TE solution, called Green-PolyH, that considers demands defined by a given

polyhedral set. Advantageously, Green-PolyH ensures all such demands do not cause the utilization of links to exceed a given threshold. Also, as the resulting solution is robust for all demands within the polyhedra set, a network operator does not need to recompute a new solution whenever the TM changes.

7.1 Network Model

A connected network is modeled as a directed graph $G(V, E)$. The set V contains nodes/routers/switches and E is the set of edges. Each edge e has capacity c_e. Let $\Theta = \{(s,t) : s,t \in V, t \neq s\}$ be the set of commodities where source s has demand d_{st} for destination t. Moreover, the TM is denoted as $d \in \mathcal{R}^{|\Theta|}$ and d_{st} is a component of d; note, d is technically a vector but the term *traffic matrix* is ubiquitous in the literature. The possible values that d can take are governed by a polytope $\mathfrak{D}$ [33] that is defined as follows: $\mathfrak{D} = \{d \in \mathcal{R}^{|\Theta|} : Ad \leq \alpha, d \geq 0\}$. Here $A \in \mathcal{R}^{K \times |\Theta|}$ and $\alpha \in \mathcal{R}^K$, where K is the number of constraints or inequalities.

As an example, consider a triangle topology with $V = \{1,2,3\}$ and the set of edges $E = \{(1,2),(2,1),(1,3),(3,1),(2,3),(3,2)\}$. Assume a special instance of the polyhedra set called the Hose model [138], where each node v has a total outgoing and incoming bandwidth that is denoted as C_v^+ and C_v^-, respectively. Moreover, $\Theta = \{(1,2),(2,1),(1,3),(3,1),(2,3),(3,2)\}$. In this example, a possible inequality is $d_{12}+d_{13} \leq C_1^+$, meaning the total outgoing demands from node 1 must not exceed C_1^+. Conversely, for incoming demands into node 2, we have $d_{12} + d_{32} \leq C_2^-$. In general, we have

$$\sum_{j \neq i, (i,j) \in \Theta} d_{ij} \leq C_i^+, \ \forall i \in V \tag{7.1}$$

$$\sum_{i \neq j, (i,j) \in \Theta} d_{ij} \leq C_j^-, \ \forall j \in V \tag{7.2}$$

The set containing the first K simple shortest paths for commodity (s,t) is written as P_{st}; each of which is indexed by k. In other words, the k-th path is p_k^{st}.

Note, K is an upper bound as some (s,t) pairs may have no more than K paths. P_e will be used to denote the set of paths that use link e. Hence, the total demand over a given link e is $L_e^d = \sum_{p \in P_e} B(p)$, where the function $B(p)$ returns the demand transmitted on path p given TM $d \in \mathfrak{D}$. Each link has utilization $u_e = L_e^d / c_e$ and it is bounded by γ. Lastly, it is assumed that all links consume the same amount of energy; see [15].

7.2 The Problem

The problem at hand is to minimize the number of links used to route demands from the given set $\mathfrak{D}$ such that for *any* $d \in \mathfrak{D}$ (i) the demand for each commodity $(s,t) \in \Theta$ is routed over one path in P_{st}, and (ii) u_e of all link $e \in E$ is no more than γ. In other words, the utilization of each active link must be no more than γ for any TM in $\mathfrak{D}$.

Reconsider the topology shown in Figure 7.1; all the links are undirected and have unit capacity. In this topology, the required link utilization is no more than $\gamma = 0.8$. The polyhedral set is described by the inequalities, $d_{15} + d_{16} \leq 0.8$, $d_{15} \leq 0.8$, and $d_{16} \leq 0.8$. Given these inequalities, we thus have a polyhedron with extreme points $(0,0)$, $(0,0.8)$ or $(0.8,0)$. The aim is to switch off as many links as possible whilst supporting any demand realizations from the given polyhedron subject to the utilization of all links being less than or equal to γ.

Let $x_e \in \{0,1\}$ be a decision variable that is set to one if link e is active, and zero otherwise. With a slight abuse of notation, p_k^{st} will also be used to denote a binary decision variable that indicates whether the said path is chosen. We thus have,

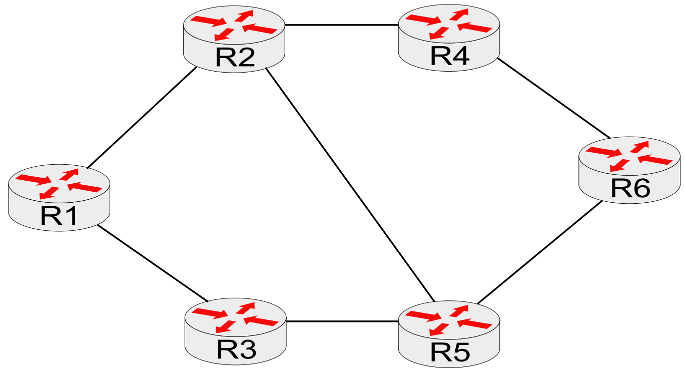


Figure 7.1: Uncertain demands. Demands from router *R1* need to be supported whilst guaranteeing that link utilization remains below γ.

$$\begin{array}{lll} \text{MIN} & \sum_{e \in E} x_e & \\ \text{subject to} & \sum_{k=1}^{|P_{st}|} p_k^{st} = 1, & \forall (s,t) \in \Theta \\ & x_e \geq p_k^{st} & \forall p_k^{st} \in P_e, \forall e \in E \\ & \max_{d \in \mathfrak{D}} \left\{ L_e^d \right\} \leq \gamma c_e, & \forall e \in E \end{array}$$

The objective is to minimize the number of active links. Notice that: (i) switches can be considered by introducing a decision variable $s_j \in \{0, 1\}$ for a switch j and setting it to zero when all its incident links are off. To ease exposition, this extension will be ignored, and (ii) maximizing the number of switched off links albeit in the context of deterministic traffic has been considered in [15]. The first constraint ensures only one path for each commodity is chosen. The second constraint ensures x_e is one, i.e., a link is up, if there is a path using it. The third constraint ensures that for a given set of chosen paths or active links, *all* TM d in $\mathfrak{D}$ do not cause the load over edge e to exceed γc_e.

Recall that each commodity in Θ has up to K paths. The search space in terms of paths is thus of size $K^{|\Theta|}$. Another key challenge is the number of extreme points in the polyhedra set $\mathfrak{D}$ grows exponentially with the number of demands; i.e., checking the third constraint is computationally expensive. Thus, it is not surprising that routing over polyhedral sets is NP-hard [139].

7.3 Proposed Solution

This section proposes a heuristic algorithm, called Green-PolyH, that iteratively moves paths away from links with low utilization. In each iteration it removes a link with low utilization that is not in the final solution from the network. It also reroutes all paths that traverse the link. It then checks the utilization of the remaining links when fed demands from the given polyhedra set. If the resulting link utilization exceeds γ, the link is added into the final solution and Green-PolyH reverts back to the previous routing solution. Otherwise, the link is removed permanently.

With the aid of Algorithm 4, the details of Green-PolyH are now presented. The set Δ_t is used to record links that have been removed temporarily, and $\mathcal{A}$ is a set containing links included in the final solution. Initially, for each commodity, its demand is routed over its shortest path. Unused links are added into Δ_t and removed from the network temporarily; see lines 2 to 6. Given the initial routing, denoted as R, Green-PolyH then solves an LP called *LP-MaxUTIL()* to determine the total demand traversing a link (see line 9 to 12); the formulation for *LP-MaxUTIL()* will be explained later. The total demand of link e is stored in l_e; the set $\mathcal{L}$ is used to store the total demand of each link. Note that in practice *LP-MaxUTIL()* is only applied on links that have at least one path.

Green-PolyH then checks the link utilization of all links. In particular, if the maximum utilization across all links is less than γ, then there is an opportunity to reroute commodities away from the least loaded link and thereby switch off said link. This is the goal of lines 14 to 20. First, in line-14, temporary links are removed permanently. This is reasonable because the current network $G(V, E)$ has sufficient capacity to handle all demands in $\mathfrak{D}$. Then in line-15, the current routing is saved; the algorithm will need to revert back to this routing if removing the link selected in line-16 causes high utilization. After that, the function *Reroute()* is called to reroute commodities as follows: (i) Select a link e^* with the *lowest* utilization in G and not in the set $\mathcal{A}$. If there is no such link, then return “DONE”, (ii) Reroute *all*

commodities corresponding to paths in P_{e^*} onto another path that does not involve link e^*. If not successful, then add e^* into $\mathcal{A}$ and go back to Step (i). Otherwise, i.e., all paths in P_{e^*} have been rerouted, add e^* into Δ_t and return the new routing R, $\mathcal{A}$, and Δ_t. Lastly, in line-20, it removes the link in Δ_t from $G(V, E)$ before proceeding back to the start of the *while* loop.

Green-PolyH may have removed a critical link; i.e., one that is required to ensure the worst case demand from $\mathfrak{D}$ does not cause link utilization to exceed γ. To this end, lines 23 to 27 address the scenario where one or more links have utilization beyond γ. First, Green-PolyH adds the links in Δ_t, so called critical links, to $\mathcal{A}$. This means in the next iteration, *Reroute()* will no longer consider these links. Then Green-PolyH restores these critical links, see line 23, and the previous routing solution; see line-26.

As mentioned, Line-10 of Algorithm 4 calls an LP solver to compute *LP-MaxUTIL*. Its aim, for a given routing and link, is to determine the maximum aggregated demand that traverses the said link. By iterating over each link, *LP-MaxUTIL* can be used to determine the utilization of links in G. Let $I^{st}_{e,R}$ be an indicator function that returns one if commodity (s,t)'s path, as determined by routing R, traverses link e. Then the following LP solves for a TM d in $\mathfrak{D}$ that maximizes the load of edge e.

$$\underset{d \in \mathfrak{D}}{\text{MAX}} \sum_{(s,t) \in \Theta} I^{st}_{e,R} \times d_{st} \tag{7.3}$$

Note, it is possible that the value of (7.3) exceeds link e's capacity. This is a not a concern because the goal is to identify whether the current routing R causes edge e's utilization to exceed γ.

This section concludes with a few key facts.

Proposition 1. *Green-PolyH ensures all commodities in Θ remain connected at all times; i.e., the resulting graph induced by links in the set $\mathcal{A}$ is connected.*

Algorithm 4: Green-PolyH

input : $G(V, E)$, Θ, P_{st}, $\mathfrak{D}$
output: $\mathcal{A}$ – the set of active links

$\Delta_t = \mathcal{A} = R = \emptyset$
for $(s, t) \in \Theta$ **do**
    $R = R \cup$ RouteShortest(G, P_{st})
end
$\Delta_t =$ GetUnusedLinks(G, R)
G = RemoveLinks(G, Δ_t)
while *true* **do**
    $\mathcal{L} = \emptyset$
    for $e \in E$ **do**
        $l_e =$ `Solve` LP-MaxUTIL(G, e, R, $\mathfrak{D}$)
        $\mathcal{L} = \mathcal{L} \cup l_e$
    end
    if $MAX\{\frac{l_e}{c_e} \mid l_e \in \mathcal{L}\} \leq \gamma$ **then**
        $\Delta_t = \emptyset$
        $R_{temp} = R$
        [$\mathcal{A}$, Δ_t, R, Code] = Reroute(G, $\mathcal{A}$, $\mathcal{L}$, R)
        if *Code ==‘DONE’* **then**
            Return $\mathcal{A}$
        else
            G = RemoveLinks(G, Δ_t)
        end
    else
        $\mathcal{A} = \mathcal{A} \cup \Delta_t$
        G = RestoreLinks(G, Δ_t)
        $\Delta_t = \emptyset$
        $R = R_{temp}$
    end
end

Proof. In line-3, each (s,t) is routed on the shortest path. As the network is connected, the proposition is true. Consider an arbitrary iteration k where all $(s,t) \in \Theta$ are connected, and the utilization of all links is below γ. Assume link e_1 has the lowest utilization. Then *reroute()* either (i) successfully establishes an alternative path for all commodities traversing e_1, meaning all rerouted commodities remain connected, or (ii) commodities over e_1 cannot be rerouted and thus link e_1 is added into $\mathcal{A}$. As no links on paths traversing link e_1 have been removed, all commodities remain connected. Observe also that if a commodity has only one path, then all links on the path will eventually be included into $\mathcal{A}$. Lastly, assume case (i) and e_1 has been removed temporarily from G in iteration line-20, and at iteration $k+1$, least one link's utilization exceeds γ. Then as per line-24, Green-PolyH reverts back to the previous routing of iteration k, which by assumption connects all commodities. Also, Green-PolyH restores the link in Δ_t. Hence, the proposition is true. □

As noted in Section 7.1, the poyhedra set $\mathfrak{D}$ is defined by K inequalities and $|\Theta|$ commodities. If we assume the Hose model [138], then for a given network, LP-MaxUTIL() contains $|\Theta|$ decision variables and $2|V|$ inequalities; each node has a constraint that bounds its aggregated incoming and outgoing demands. This means the size of LP solved by LP-MaxUTIL() is proportional to the network size and number of commodities. The number of times Green-PolyH calls LP-MaxUTIL() is stated in the next proposition.

Proposition 2. *LP-MaxUTIL() is called at most $|E|^2$ times.*

Proof. The *while* loop, i.e., line 7-28, repeats $|E|$ times. In each iteration, a link is either added into $\mathcal{A}$ or removed permanently from G; the former occurs when rerouting is unsuccessful or a new routing causes high utilization. The latter happens if Green-PolyH successfully reroutes all paths from the link. Recall that Step (i) of *Reroute()* ignores links in $\mathcal{A}$. Hence, in both cases, in subsequent iterations, Green-

PolyH ignores links in $\mathcal{A}$. Now, as LP-MaxUTIL() is carried out on a link-by-link basis, Green-PolyH thus calls LP-MaxUTIL() no more than $|E|^2$ times. □

7.4 Evaluation

The experiments are conducted in MATLAB over two well known topologies, Abilene (11 nodes, 28 links) and AT&T (25 nodes, 112 links), and two synthetic topologies, 7N18L (7 nodes, 18 links), and 8N24L (8 nodes, 24 links) [140]. Results are conducted for two cases: *Exp-1* and *Exp-2*. In both cases, without loss of generality, the Hose model [138] is considered. Also, $C_i^+ = C_i^-$; this symmetric case is denoted as $C_i^\pm$. In *Exp-1*, the impact of $C_i^\pm$ on energy savings is studied, in terms of percentage of shut down links, where $|\Theta| = 10$ and $|\Theta| = 50$. In *Exp-2*, $C_i^\pm$ is fixed to a random value from the range $[0, \gamma]$ at the start of each experiment. Then, the impact of increasing number of (s,t) pairs is studied. Each experiment is run 20 times. The presented results are within 95% confidence interval. Lastly, as per [120], the MLU (γ) is set to 80%.

Figure 7.2 presents the result from Exp-1. As expected, when the network load is low, i.e., $|\Theta| = 10$, more savings are observed for all the evaluated topologies. In particular, for the small topologies, namely 7N18L and 8N24L, see Figure 7.2a, savings above 50 % and 40% are observed for 8N24L and 7N18L, respectively.

For AT&T and Abilene, see Figure 7.2b, savings achieved by Green-PolyH decrease when $C_i^\pm$ approaches γ. In the case of AT&T, savings decrease from above 80% to 67%. As for Abilene, the observed decrease is from 44% to 41.4%. When $|\Theta| = 10$, many links will have a low utilization, and thus, there are more opportunities to reroute flows. However, as the load of these ten (s, t) pairs increases, it becomes increasingly difficult to reroute a flow to reduce the number of operating links without causing one or more links to exceed the specified MLU. However, note that for the small topologies, almost constant energy savings are observed. This

can be explained by the small number of links in these topologies which reduce the possibilities of rerouting flows.

The case where $|\Theta| = 50$ is now discussed. As link utilization tends to be high, even when $C_i^{\pm} = 0.1$, there is little or no opportunity to reroute flows in order to reduce the number of active links. Hence, for all tested values of $C_i^{\pm}$, the percentage of shut down links corresponds to the maximal number of links that can be switched off safely without impacting robustness.

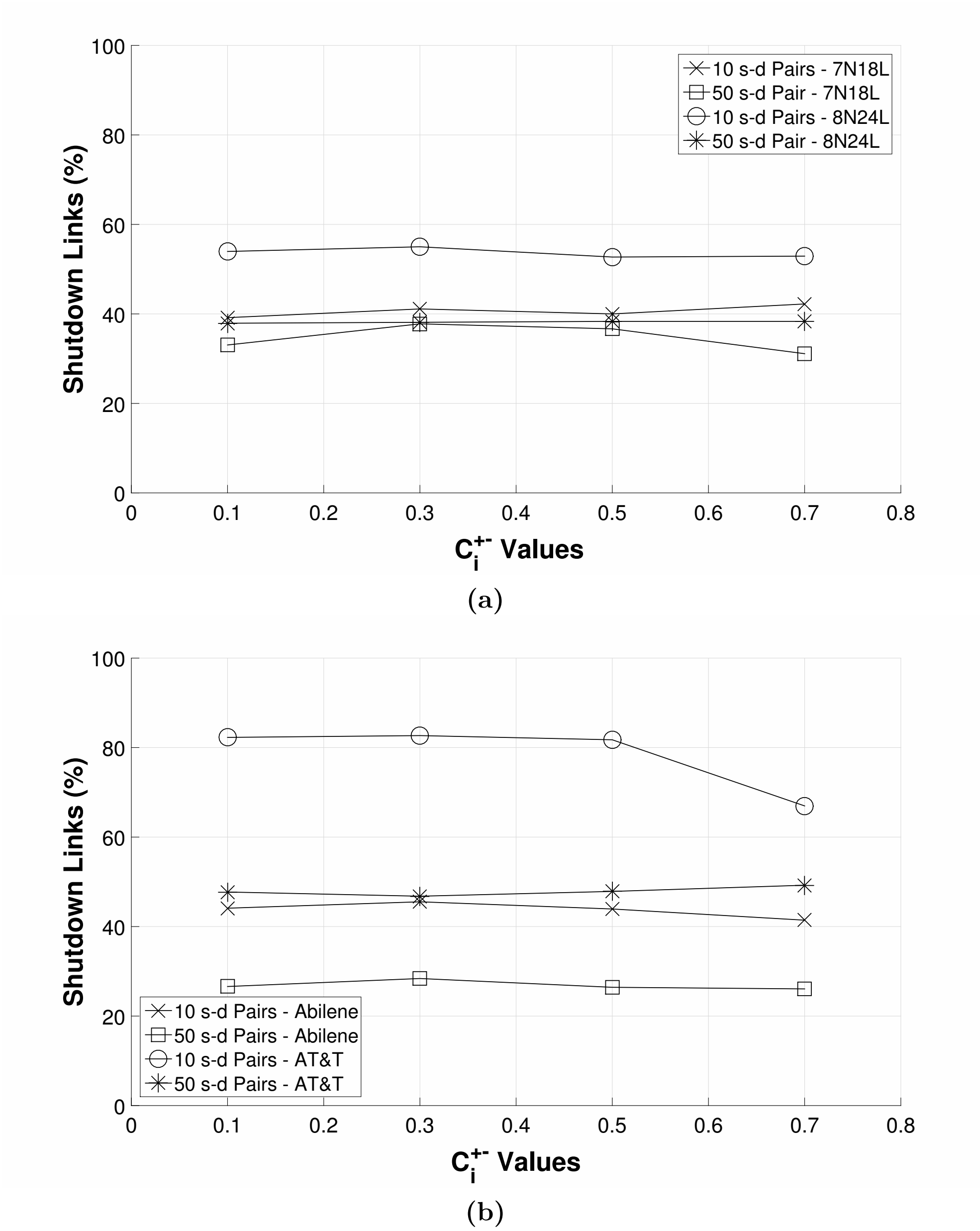


Figure 7.2: Percentage of shut down links versus increasing load. a) 7N18L and 8N24L and b) Abilene and AT&T

The result from Exp-2 is shown in Figure 7.3. For all the evaluated topologies, i.e., AT&T, 8N24L, 7N18L, and Abilene, energy savings reduce when the network load increases. In particular, for the lowest network load, i.e., $|\Theta| = 10$, Green-PolyH is able to shut down 82%, 57%, 48% and 43.4% of the links for AT&T, 8N24L, 7N18L, and Abilene, respectively. For the highest network load, i.e., 35 (s,t) pairs, energy savings reduce to 56%, 36%, 29%, and 26% for AT&T, 8N24L, 7N18L, and Abilene, respectively.

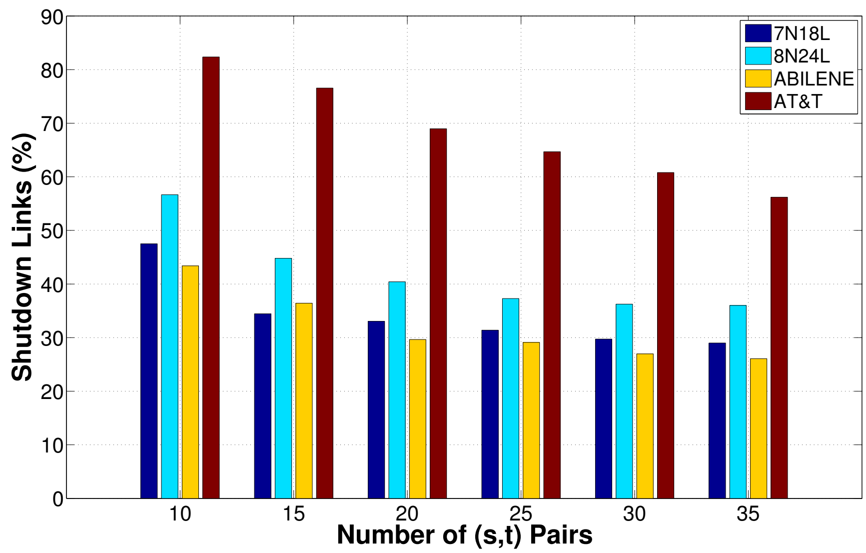


Figure 7.3: Percentage of shut down links versus increasing (s,t) pairs or $|\Theta|$ values

7.5 Conclusion

This chapter has introduced the first green and robust TE solution that ensures active links have the capacity to support *any* demand from a given polyhedral set. This is significant because the presented solution considers random traffic matrices when shutting down links. Experimental results over two well-known topologies confirm the efficacy of the solution in terms of saving energy and ensuring the utilization of all links is below a given threshold.

Chapter 8

Conclusions

The main motivation for this thesis is due to recent reports on the significant energy consumption attributed to the ICT industry. A key reason is that current ICT infrastructure is designed to handle the "worst-case" scenario, meaning the Internet is not energy efficient. To this end, in the last few years, there have been intensive research activities on developing green approaches. Of interests are those that (i) focus on establishing LSPs in an energy efficient way, (ii) associate switches to controllers using paths that involve the fewest number of switches, (iii) do not adversely affect the operation of BGP, and (iv) are robust to varying traffic demands.

Henceforth, in Chapter 3, this thesis studies a number of energy-efficient LSP establishment methods. It investigates six heuristics and also introduces a novel ratio, called ρ, which is defined as the percentage of shut down links (PSL) over the number of LSP accepted requests (LAR). This ratio is then used to quantify the performance of each heuristic. Advantageously, it allows network operators to evaluate whether a solution with a large acceptance rate is also energy efficient. Apart from that, Chapter 3 also presents a key innovation by introducing a heuristic called *Online-R*. It utilizes already established links and additionally favors paths that require the fewest new links. Results show that, during off-peak times, savings beyond 20% are achievable with LSP acceptance rates above 90%.

Chapter 4 studies the novel problem of jointly shutting down links and determining controller-switch association. The proposed algorithm, called GreCo, takes into account elongated propagation delays resulting from shutting down links on the shortest path from a switch to a controller. In addition, GreCo also guarantees that all controllers have a similar number of switches and that links have utilization no more than a given threshold. Results show that GreCo achieves up to 55% energy savings during off-peak hours and uses as few as 20% more links as compared to the optimal solution.

A key observation identified in this thesis is that existing green TE works do not consider BGP. To this end, Chapter 5 contains the first study that quantifies the impact that green TE techniques have on the operation of BGP. Results indicate that running a green TE technique can cause an increase of 29% in egress/border router selection changes when compared to non-green approaches. The results also show more than 25% shift in traffic. To mitigate these negative effects, this thesis presents two novel BGP aware approaches. The first one, called HotPLUZ and presented in Chapter 5, aggregates ingress router (IR) traffic destined to their corresponding egress routers (ERs) from lowly utilized links onto highly utilized links, while minimizing changes to the corresponding ERs of a given destination. In addition, HotPLUZ considers link utilization in order to avoid packet loss and increase in latencies. HotPLUZ is able to achieve up to 21% in terms of energy savings during off-peak periods. The second BGP aware approach called HotPLEC; see Chapter 6. This technique allows IRs to collaborate with the aim to consolidate ingress-egress traffic onto the minimal number of links. The main idea is for IRs to exchange information with each other in order to determine if changing paths yield better energy savings. Experimental results show that HotPLEC can use as little as 25% more active links than the optimal solution.

The last contribution is concerned with random traffic demands. This is an important consideration because in practice traffic varies temporally and spatially.

Henceforth, Chapter 7 presents Green-PolyH, the first green and robust TE solution that considers demands that are characterized by a polyhedral set. Advantageously, Green-PolyH ensures that in the worst case, the total demands over a given link does not exceed a threshold. Results over four well known topologies show that savings above 80% are possible whilst being robust to traffic changes

There are many avenues for further research. For example, with regards to the green LSP establishment methods proposed in Chapter 3, one possible investigation is to consider one or more additive constraints such as delay when establishing LSP paths. Another is to determine the competitive ratio of existing heuristics; i.e., how close is the performance of online heuristics as compared to their offline counterparts. In terms of green SDNs, a logical extension to the work in Chapter 4 is to consider random PACKET_IN requests from switches. Specifically, the worst case number of requests from the switches associated to a controller should not exceed a given threshold. Apart from that, another future work is deploying the smallest number of controllers that are robust against random PACKET_IN requests. This thesis has shown that BGP-awareness is important. To this end, possible future works include collaborative solutions between different ASes running BGP-aware green network approaches. Apart from that, an interesting future work is to employ cooperative game theory to model the behavior of IRs and to derive the best policy that yields the most energy savings. As pointed out earlier, a key limitation of existing green TE works is that they do not consider uncertain traffic demands. A key future work is to determine faster algorithms, preferably ones that do not involve calling a LP solver.

Current TE approaches run during off-peak hours and links are only shut down as long as the resulting network remains connected. However, once the final topology has been reached, i.e., the topology that yields the minimum number of network elements, a node or link failure may cause a disruption in service. Therefore, a key

future research direction is to explore mechanisms that take into account failure scenarios whilst also saving the maximum amount of energy.

As a final remark, this thesis has motivated the necessity of reducing energy consumption. It has also demonstrated that energy savings, particularly, during off-peak hours, are achievable. However, the presented techniques pose interesting challenges. Existing network infrastructure, protocols and even applications are not energy-aware. Therefore, a shift in hardware and software industries is required to provide the necessary support needed to implement green techniques.

Bibliography

[1] "Enabling the low carbon economy in the information age." Available: www.smart2020.org/publications/, 2008.

[2] F. Ramos, R. Gibbens, F. Song, P. Rodriguez, J. Crowcroft, and I. White, "Reducing energy consumption in IPTV networks by selective pre-joining of channels," in *ACM SIGCOMM workshop on Green networking*, (New Delhi, India), Aug 2010.

[3] A. Bacioccola, C. Cicconetti, and G. Stea, "User-level performance evaluation of VoIP using ns-2," in *Proceeding of the International Conference on Performance evaluation Methodologies and tools*, (Nantes, France), Oct 2007.

[4] B. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Fifth International Joint Conference on Networked Computing (INC), Information Management and Service (IMS) and Digital Content, Multimedia Technology and its Applications (IDC)*, (Seoul,Korea), Aug 2009.

[5] D. Singh, G. Tripathi, and A. Jara, "A survey of internet-of-things: Future vision, architecture, challenges and services," in *IEEE World Forum on Internet of Things (WF-IoT)*, (Seoul, Korea (South)), Mar 2014.

[6] "The Zettabyte era: Trends and analysis." Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI_Hyperconnectivity_WP.html, 2014.

[7] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *ACM/USENIX NSDI*, (San Francisco, CA), Apr 2008.

[8] "Make IT green: Cloud computing and its contribution to climate change." Available: http://www.greenpeace.org/international/en/publications/reports/make-it-green-cloud-computing/, 2010.

[9] G. Ananthanarayanan and R. Katz, "Greening the switch," in *Proceedings of Power aware computing and systems*, (San Diego, CA), Dec 2008.

[10] M. Gupta and S. Singh, "Greening of the Internet," in *ACM SIGCOMM*, (Karlsruhe, Germany), Aug 2003.

[11] M. Gupta, S. Grover, and S. Singh, "A feasibility study for power management in LAN switches," in *IEEE ICNP*, (Berlin, Germany), Oct 2004.

[12] C. Gunaratne, K. Christensen, and B. Nordman, "Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed," *International Journal of Network Management*, vol. 15, no. 5, pp. 297–310, 2005.

[13] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the energy consumption of Ethernet with adaptive link rate (ALR)," *IEEE Transactions on Computers*, vol. 57, no. 4, pp. 448–461, 2007.

[14] K. Christensen, C. Gunaratne, B. Nordman, and A. George, "The next frontier for communications networks: power management," *Computer Communications*, vol. 27, no. 18, pp. 1758–1770, 2004.

[15] M. Zhang, C. Yi, B. Liu, and B. Zhang, "GreenTE: Power-aware traffic engineering," in *IEEE ICNP*, (Kyoto, Japan), Oct 2010.

[16] A. Bianzino, L. Chiaraviglio, and M. Mellia, "GRiDA: A green distributed algorithm for backbone networks," in *IEEE Online Conference on Green Communications (GreenCom)*, Sep 2011.

[17] A. Cianfrani, V. Eramo, M. Listanti, M. Marzza, and E. Vittorini, "An energy saving routing algorithm for a green OSPF protocol," in *IEEE INFOCOM*, (San Diego, CA), Mar. 2010.

[18] G. Athanasiou, K. Tsagkaris, P. Vlacheas, D. Karvounas, and P. Demestichas, "Multi-objective traffic engineering for future networks," *IEEE Communications Letters*, vol. 16, pp. 101 –103, Jan 2012.

[19] G. Lin, S. Soh, K. Chin, and M. Lazerescu, "Efficient heuristics for energy-aware routing in networks with bundled links," *Elsevier Compupter Networks*, vol. 57, pp. 1774–1788, Jun 2013.

[20] N. Wang, K. Ho, G. Pavlou, and M. Howarth, "An overview of routing optimization for internet traffic engineering," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 1, pp. 36 –56, 2008.

[21] S. Avallone and G. Ventre, "Energy efficient online routing of flows with additive constraints," *Elsevier Computer Networks*, vol. 56, no. 10, pp. 2368–2382, 2012.

[22] F. Francois, N. Wang, K. Moessner, S. Georgoulas, and R. Schmidt, "Leveraging MPLS backup paths for distributed energy-aware traffic engineering,"

IEEE Transactions on Network and Service Management,, vol. 11, pp. 235–249, Jun 2014.

[23] A. Coiro, F. Iervini, and M. Listanti, "Distributed and adaptive interface switch off for internet energy saving," in *IEEE ICCCN*, (Maui, Hawaii), Jul 2011.

[24] A. Coiro, M. Listanti, A. Valenti, and F. Matera, "Energy-aware traffic engineering: A routing-based distributed solution for connection-oriented IP networks," *Elsevier Computer Networks*, vol. 57, no. 9, pp. 2004–2020, 2013.

[25] H.-W. Chu, C.-C. Cheung, K.-H. Ho, and N. Wang, "Green MPLS traffic engineering," in *Australasian Telecommunication Networks and Applications Conference, ATNAC*, (Melbourne, Australia), Nov. 2011.

[26] A. Cianfrani, V. Eramo, M. Listanti, and M. Polverini, "An OSPF enhancement for energy saving in IP networks," in *IEEE Conference on Computer Communications Workshops*, (Shanghai, China), Apr. 2011.

[27] F. Cuomo, A. Abbagnale, and S. Papagna, "ESOL: Energy saving in the internet based on occurrence of links in routing paths," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, (Lucca, Italy), Jun 2011.

[28] E. Amaldi, A. Capone, L. Gianoli, and L. Mascetti, "Energy management in IP traffic engineering with shortest path routing," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, (Lucca, Italy), Jun 2011.

[29] D. Arai and K. Yoshihara, "ECO-friendly distributed routing protocol for reducing network energy consumption," in *International Conference on Network and Service Management (CNSM)*, (Niagara Falls, Canada), Oct 2010.

[30] "Multiprotocol label switching architecture." RFC 3031. Available: https://www.ietf.org/rfc/rfc3031.txt, Jan 2001.

[31] D. Kreutz, F. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, pp. 14–76, Jan 2015.

[32] "A Border Gateway Protocol 4 (BGP-4)." RFC 4271. Available: https://www.ietf.org/rfc/rfc4271.txt, Jan 2006.

[33] W. Ben-Ameur and H. Kerivin, "Routing of uncertain traffic demands," *Optimization and Engineering*, vol. 6, pp. 283–313, 2005.

[34] F. Hu, Q. Hao, and K. Bao, "A survey on software defined networking (SDN) and openflow: From concept to implementation," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2014.

[35] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proceedings of the 1st workshop on Hot Topics in SDNs*, (Helsinki, Finland), Aug 2012.

[36] G. Yao, J. Bi, Y. Li, and L. Guo, "On the capacitated controller placement problem in software defined networks," *IEEE Communications Letters*, vol. 18, pp. 1339–1342, Aug 2014.

[37] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford, "Impact of hot-potato routing changes in IP networks," *IEEE/ACM Transactions on Networking*, pp. 1295 –1307, Dec 2008.

[38] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford, "Dynamics of hot-potato routing in IP networks," in *ACM SIGMETRICS*, (New York), Jun 2004.

[39] P. Tune and M. Roughan, "Internet traffic matrices: A primer," in *Recent Advances in Networking (Volume I)* (H. Haddadi and O. Banaventure, eds.), ACM SIGCOMM, Aug. 2013.

[40] A. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, "A survey of green networking research," *IEEE Communications Surveys Tutorials*, vol. 14, no. 1, pp. 3 –20, 2012.

[41] S. Balon and G. Leduc, "BGP-Aware IGP link weight optimization in presence of route reflectors," in *IEEE INFOCOM*, (Rio de janeiro, Brazil), Apr 2009.

[42] K. S. Sohn and D. K. Sung, "A distributed LSP mechanism to reduce spare bandwidth in MPLS networks," in *IEEE ICC*, (New York City, USA), Apr 2002.

[43] "Key considerations for MPLS/IP-VPN success." Available: http://www.xo.com/resources/white-papers/key-considerations-for-mpls-ip-vpn-success.pdf, Aug 2012.

[44] "MPLS/IP VPN services market update 2014." Available: http://www.researchandmarkets.com/research/s2wtfz/mplsip_vpn, Sep 2014.

[45] D. Hong, C. Hong, and G.-H. Lee, "M_CSPF: A scalable CSPF routing scheme with multiple QoS constraints for MPLS traffic engineering," *ETRI journal*, vol. 27, no. 6, pp. 733–746, 2005.

[46] "Constrained shortest path first." Internet-draft. Available: http://tools.ietf.org/search/draft-manayya-constrained-shortest-path-first-00, Feb 2009.

[47] D. Adami, D. Callegari, S. Giordano, and M. Pagano, "A new path computation algorithm and its implementation in NS2," in *IEEE ICC*, (Glasgow, Scotland), Jun 2007.

[48] A. Elwalid, D. Mitra, I. Saniee, and I. Widjaja, "Routing and protection in GMPLS networks: from shortest paths to optimized designs," *Journal of Lightwave Technology*, vol. 21, pp. 2828 – 2838, Nov 2003.

[49] K. Hao and Z. Jin, "An on-line routing algorithm based on the off-line optimal computing in MPLS," in *IEEE WiCom*, (Beijing, China), Sep 2009.

[50] S. Nelakuditi and Z. Zhang, *Localized quality of service routing for the Internet.* Springer, 2003.

[51] R. Pant and T. Sanguankotchakorn, "A heuristic approach for bandwidth constraint path selection in MPLS based networks," in *International Conference on Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON)*, (Chiang Mai, Thailand), May 2010.

[52] K. Kar, M. Kodialam, and T. Lakshman, "Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2566–2579, 2000.

[53] M. Kodialam and T. Lakshman, "Minimum interference routing with applications to MPLS traffic engineering," in *Proceedings of the Nineteenth Annual Joint Conference of the IEEE INFOCOM*, vol. 2, (Tel Aviv, Israel), IEEE, Mar 2000.

[54] A. Bestavros and I. Matta, "Load profiling for efficient route selection in multiclass networks," in *Proceedings International Conference on Network Protocols ICNP*, (Atlanta, GA), IEEE, Oct 1997.

[55] Z. Wang and J. Crowcroft, “Quality-of-service routing for supporting multimedia applications,” *IEEE JSAC*, vol. 14, no. 7, pp. 1228–1234, 1996.

[56] J. de Oliveira, F. marinelli, and C. Scoglio, “SPeCRA: A stochastic performance comparison routing algorithm for LSP setup in MPLS networks,” in *IEEE GLOBECOM*, (Taipei, Taiwan), Nov. 2002.

[57] S.-i. Kuribayashi and S. Tsumura, “Optimal LSP selection method in MPLS networks,” in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, (Victoria, B.C., Canada), Aug 2007.

[58] Z. Zenghua, S. Yantai, L. Zhang, and Y. Oliver, “Flow-level multipath load balancing in MPLS network,” *IEICE transactions on communications*, vol. 88, no. 5, pp. 2015–2022, 2005.

[59] H. Ishii and K. Nagami, “Proposal and evaluation of new LSP autobandwidth algorithm,” *IEICE transactions on communications*, vol. 89, no. 10, pp. 1894–1901, 2006.

[60] A. Elwalid, C. Jin, S. Low, and I. Widjaja, “MATE: MPLS adaptive traffic engineering,” in *Proceedings of the Twentieth Annual Joint Conference IEEE INFOCOM*, (Anchorage, Alaska), Apr. 2001.

[61] H. Saito, Y. Miyao, and M. Yoshida, “Traffic engineering using multiple multipoint-to-point LSPs,” in *IEEE INFOCOM*, (Tel Aviv, Israel.), Mar 2000.

[62] J. Reich, M.C, N. Foster, J. Rexford, and D. Walker, “Modular SDN programming with pyretic,” *IEEE Communications Magazine*, vol. 38, no. 5, pp. 128–134, 2013.

[63] C. Anderson, N. Foster, A. Guha, J.-B. Jeannin, D. Kozen, C. Schlesinger, and D. Walker, “Netkat: Semantic foundations for networks,” *ACM SIGPLAN Notices*, vol. 49, no. 1, pp. 113–126, 2014.

[64] M. Casado, M. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, “Ethane: Taking control of the enterprise,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 1–12, 2007.

[65] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovtion in campus networks,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[66] S. Schmid and J. Suomela, “Exploiting locality in distributed SDN control,” in *Proceedings of the 2nd ACM SIGCOMM workshop on Hot topics in SDN*, (Hong Kong, China), Aug 2013.

[67] Y. Jimenez, C. Cervello-Pastor, and A. Garcia, “On the controller placement for designing a distributed SDN control layer,” in *IFIP Networking*, (Trondheim, Norway), Jun 2014.

[68] R. Ahmed and R. Boutaba, “Design considerations for managing wide area software defined networks,” *IEEE Communications Magazine*, vol. 52, pp. 116–123, Jul 2014.

[69] J. Fu, Y.and Bi, J. Wu, Z. Chen, K. Wang, and M. Luo, “A dormant multi-controller model for software defined networking,” *IEEE China Communications*, vol. 11, pp. 45–55, Mar 2014.

[70] Y. Hu, W. Wang, X. Gong, X. Que, and S. Cheng, “On the placement of controllers in software-defined networks,” *The Journal of China Universities of Posts and Telecommunications*, vol. 19, pp. 92–171, 2012.

[71] Y. Hu, W. Wang, X. Gong, X. Que, and S. Cheng, “On reliability-optimized controller placement for software-defined networks,” *Communications, China*, vol. 11, pp. 38–54, Feb 2014.

[72] H. Wu, R. W. P. Luk, K. Wong, and K. Kwok, “Interpreting TF-IDF term weights as making relevance decisions,” *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 3, p. 13, 2008.

[73] F. Ros and P. Ruiz, “Five nines of southbound reliability in software-defined networks,” in *Proc. of the 3rd Workshop on Hot Topics in SDN*, (New York, USA), Aug 2014.

[74] M. Bari, A. Roy, S. Chowdhury, Q. Zhang, M. Zhani, R. Ahmed, and R. Boutaba, “Dynamic controller provisioning in software defined networks,” in *9th International Conference on Network and Service Management, CNSM*, (Zürich, Switzerland), Oct 2013.

[75] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran-Gia, “Pareto-optimal resilient controller placement in SDN-based core networks,” in *25th International Teletraffic Congress, ITC*, (Shanghai, China), Sep 2013.

[76] S. Yeganeh, A. Tootoonchian, and Y. Ganjali, “On scalability of software-defined networking,” *IEEE Communications Magazine*, vol. 51, pp. 136–141, Feb 2013.

[77] Y. Hu, W. Wang, X. Gong, X. Que, and S. Cheng, “Balanceflow: Controller load balancing for Openflow networks,” in *2nd IEEE International Conference on Cloud Computing and Intelligent Systems, CCIS*, (Hangzhou, China), Oct 2012.

[78] G. Yao, J. Bi, and L. Guo, “On the cascading failures of multi-controllers in software defined networks,” in *21st IEEE International Conference on Network Protocols (ICNP)*, (Göttingen, Germany), Oct 2013.

[79] S. Khuller and Y. Sussmann, “The capacitated k-center problem,” *SIAM J. on Discrete Mathematics*, vol. 13, pp. 403–418, May 1996.

[80] J. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. The University of Michigan Press, 1975.

[81] I. Sapountzis, "Traffic monitoring for green networking," Master's thesis, KTH Royal Institute of Technology, Stockhol, Sweden, 2014.

[82] B. Nordman, "Networks, energy, and energy efficiency," in *Cisco Green Research Symposium*, (Berkeley, California), Mar 2008.

[83] S. Gobriel, C. Maciocco, and T.-Y. Tai, "Long idle: Making idle networks quiet for platform energy-efficiency," in *Fifth International Conference on Systems and Networks Communications (ICSNC)*, (Nice, France), Aug 2010.

[84] "OSPF version 2." RFC 2328. Available: http://www.ietf.org/rfc/rfc2328.txt, Apr 1998.

[85] M. Shen, H. Liu, K. Xu, N. Wang, and Y. Zhong, "Routing on demand: Toward the energy-aware traffic engineering with OSPF," in *IFIP NETWORKING*, (Prague, Czech Republic), Nov 2012.

[86] "Traffic engineering (TE) extensions to OSPF version 2." RFC 3630. Available: https://tools.ietf.org/html/rfc3630, Sep 2003.

[87] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *IEEE INFOCOM*, (Tel-Aviv, Israel), Mar 2000.

[88] A. Bianzino, L. Chiaraviglio, and M. Mellia, "Grida: A green distributed algorithm for backbone networks," in *IEEE Online Conference on Green Communications (GreenCom)*, Sep 2011.

[89] K.-H. Ho and C.-C. Cheung, "Green distributed routing protocol for sleep coordination in wired core networks," in *6th International Conference on Networked Computing (INC)*, (yeongju, Korea (South)), May 2010.

[90] J. Restrepo, C. Gruber, and C. Machuca, “Energy profile aware routing,” in *IEEE GreenComm*, (Dresden, Germany), Jun 2009.

[91] T. Ye, G. Micheli, and L. Benini, “Analysis of power consumption on switch fabrics in network routers,” in *Proceedings of the 39th annual Design Automation Conference*, (New Orleans, LA), June 2002.

[92] “Active/idle toggling with Low-Power idle.” Available: http://www.ieee802.org/3/az/public/jan08/hays_01_0108.pdf, Jan 2008.

[93] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, “Theoretical and practical limits of dynamic voltage scaling,” in *Proceedings of the 41st annual Design Automation Conference*, (San Diego, CA), June 2004.

[94] A. Yamada, S. Imai, and M. Kakemizu, “A path-based traffic control method for green networks,” in *8th Asia-Pacific Symposium on Info and Telecommunications Technologies (APSITT)*, (Kuching, Sarawak, Malaysia), Jun 2010.

[95] “Rip version 2.” RFC 2453. Available: https://tools.ietf.org/html/rfc2453, Nov 1998.

[96] G. Di Caro and M. Dorigo, “AntNet: Distributed stigmergetic control for communications networks,” *Artificial Intelligence Research*, pp. 317–365, 1999.

[97] B. Addis, A. Capone, G. Carello, L. Gianoli, and B. Sanso, “Energy management through optimized routing and device powering for greener communication networks,” *IEEE/ACM Transactions on Networking*, vol. 22, pp. 313–325, Feb 2014.

[98] B. Addis, A. Capone, G. Carello, L. Gianoli, and B. Sanso, “Energy-aware multiperiod traffic engineering with flow-based routing,” in *IEEE International Conference on Communications (ICC)*, (Ottawa, Canada), pp. 5957–5961, Jun 2012.

[99] A. Raj and O. Ibe, "A survey of IP and multiprotocol label switching fast reroute schemes," *Computer Networks*, vol. 51, no. 8, pp. 1882–1907, 2007.

[100] N. Vasic and D. Kostic, "Energy-aware traffic engineering," in *Proceedings of 1st International Conference on Energy-Efficient Computing and Networking*, (Passau, Germany), Citeseer, Apr 2010.

[101] F. Cuomo, A. Abbagnale, A. Cianfrani, and M. Polverini, "Keeping the connectivity and saving the energy in the internet," in *IEEE INFOCOM*, (Shanghai, P.R. China), Apr 2011.

[102] A. Jamakovic and S. Uhlig, "On the relationship between the algebraic connectivity and graph's robustness to node and link failures," in *3rd EuroNGI Conference on Next Generation Internet Networks*, (Trondheim, Norway), May 2007.

[103] N. Wang, C. Michael, and K. H. Ho, "Disruption-Free green traffic engineering with NotVia Fast Reroute," *IEEE Communications Letters*, vol. 15, pp. 1123 –1125, Oct 2011.

[104] "IP fast reroute using Not-Via addresses." RFC 6981. Available: http://tools.ietf.org/html/draft-ietf-rtgwg-ipfrr-notvia-addresses-07, Apr 2011.

[105] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing ISP network energy cost: Formulation and solutions," *IEEE/ACM Transactions on Networking*, vol. 20, pp. 463–476, Apr 2011.

[106] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *IEEE ICC*, (Dresden, Germany), Jun 2009.

[107] I. Ghamlouche, T. Crainic, and M. Gendreau, "Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design," *Operations research*, vol. 51, no. 4, pp. 655–667, 2003.

[108] H. Yonezu, S. Gao, S. Shimizu, D. Ishii, S. Okamoto, E. Oki, and N. Yamanaka, "Network power saving topology calculation method by powering off links considering QoS," in *15th OptoElectronics and Communications Conference (OECC)*, (Sapporo, Japan), July 2010.

[109] E. Osborne and A. Simha, *Traffic engineering with MPLS*. Cisco Press, 2002.

[110] M. Pioro and M. Deepankar, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Elsevier, 2004.

[111] E. Bouillet, G. Ellinas, J. Labourdette, and R. Ramamurthy, *Path Routing in Mesh Optical Networks*. Wiley, 2007.

[112] A. M. C. A. Koster and M. Kutschka, "Network design under demand uncertainties: A case study on the abilene and GEANT network data," *Proceedings of Photonic Networks*, May 2011.

[113] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE JSAC*, vol. 29, pp. 1765 –1775, Oct 2011.

[114] MATLAB, *version 7.12.0.635 (R2011a)*. Natick, Massachusetts: The MathWorks Inc., 2011.

[115] A. Ruiz-Rivera, K.-W. Chin, R. Raad, and S. Soh, "HotPLUZ: A BGP-aware green traffic engineering approach," in *IEEE ICC*, (Sydney, Australia), Jun 2014.

[116] J. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, pp. 712–716, 1971.

[117] K. Neggers, "Network design for large data flow," in *Proceedings of the Optical Fiber Communication Conference*, (Anaheim, CA), Mar 2006.

[118] J. P. Kowalski and B. Warfield, "Modelling traffic demand between nodes in a telecommunications network," in *ATNAC*, (Sydney, Australia), Dec 1995.

[119] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *ACM SIGMETRICS*, (San Diego, CA), Jun 2003.

[120] A. Nucci, N. Taft, P. Thiran, H. Zang, and C. Diot, "Increasing the link utilization in IP over WDM networks," in *SPIE ITCom*, (Boston, MA), Jul 2002.

[121] S. Balon, J. Lepropre, O. Delcourt, F. Skivee, and G. Leduc, "Traffic engineering an operational network with the TOTEM toolbox," *IEEE Transactions on Network and Service Management*, vol. 4, pp. 51 –61, Jun 2007.

[122] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware backbone networks: A case study," in *IEEE International Conference on Communications Workshops*, (Dresden, Germany), Jun 2009.

[123] C. Panarello, A. Lombardo, G. Schembra, L. Chiaraviglio, and M. Mellia, "Energy saving and network performance: a trade-off approach," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, (Passau, Germany), Apr 2010.

[124] R. Teixeira, T. Griffin, M. Resende, and J. Rexford, "TIE breaking: Tunable Interdomain Egress selection," *IEEE/ACM Transactions on Networking*, vol. 15, pp. 761 –774, Aug 2007.

[125] "Best practices in core network capacity planning." Available: www.cisco.com/c/en/us/solutions/collateral/service-provider/quantum/white_paper_c11-728551.html, 2013.

[126] D. Adami, C. Callegari, S. Giordano, and M. Pagano, "Single-path and multi-path label switched path allocation algorithms with quality-of-service constraints: performance analysis and implementation in NS2," *IET communications*, vol. 6, no. 4, pp. 398–407, 2012.

[127] Y. Ganjali and A. Keshavarzian, "Load balancing in ad hoc networks: single-path routing vs. multi-path routing," in *IEEE INFOCOM*, (Hong Kong), Mar 2004.

[128] "BGP route reflection: An alternative to full mesh internal BGP (IBGP)." RFC 4456. Available: https://tools.ietf.org/rfc/rfc4456.txt, Apr 2006.

[129] B. Colson, P. marotte, and G. Savard, "An overview of bilevel optimization," *Annal of Operations Research*, vol. 153, pp. 235–256, Apr 2007.

[130] A. Migdalas, "Bilevel programming in traffic planning: Models, methods and challenge," *Journal of Global Optimization*, vol. 7, pp. 381–405, Dec 1995.

[131] R. Hassin, R. Ravi, and F. S. Salman, "Approximation algorithms for a capacitated network design problem," in *Proceedings of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization*, (Germany), Sep 2000.

[132] A. Gupta and A. K. T. Roughgarden, "Simpler and better approximation algorithms for network design," in *ACM STOC*, (San Diego, CA, USA), Jun 2003.

[133] T. H. Cormen, C. E. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms (2nd ed)*. MIT Press, 2001.

[134] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: traffic matrix estimator for OpenFlow networks," in *Passive and Active Measurement*, pp. 201–210, Springer, 2010.

[135] D. S. Johnson and M. Garey, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman&Co, NY, 1979.

[136] D. Applegate and E. Cohen, "Making routing robust to changing traffic demands: algorithms and evaluation," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 6, pp. 1193–1206, 2006.

[137] "Overview and principles of internet traffic engineering." RFC 3272. Available: http://www.rfc-editor.org/rfc/rfc3272.txt, May 2002.

[138] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, and J. E. van der Merwe, "A flexible model for resource management in virtual private networks," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 95–108, 1999.

[139] M. Minoux, "Robust network optimization under polyhedral demand uncertainty is np-hard," *Discrete Applied Mathematics*, vol. 158, pp. 597–603, Mar. 2010.

[140] A. Ruiz-Rivera, K.-W. Chin, and S. Soh, "A novel framework to mitigate the negative impacts of green techniques on BGP," *Journal of Network and Computer Applications*, vol. 48, pp. 22–34, 2015.