



HAL
open science

Green metrics to improve sustainable networking

Md Mohaimenul Hossain

► **To cite this version:**

Md Mohaimenul Hossain. Green metrics to improve sustainable networking. Networking and Internet Architecture [cs.NI]. Université de Lorraine, 2019. English. NNT : 2019LORR0201 . tel-02499760

HAL Id: tel-02499760

<https://hal.univ-lorraine.fr/tel-02499760v1>

Submitted on 5 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Green Metrics to Improve Sustainable Networking

THÈSE

présentée et soutenue publiquement le 16 December 2019

pour l'obtention du

Doctorat de l'Université de Lorraine

(Automatique, Traitement du signal et des images, Génie informatique)

par

Md Mohaimenul Hossain

Composition du jury

<i>Rapporteurs :</i>	Dr. Pascal BERTHOU Professeur Jalel BEN OTHMAN	LAAS – UPS Toulouse III L2S-CENTRALESUPELEC
<i>Examineurs :</i>	Dr. Ah-Lian KOR Professeur Colin PATTINSON Dr. Karl ANDERSSON Dr. Jean Philippe GEORGES	Leeds Beckett University Leeds Beckett University Luleå University of Technology Université de Lorraine - CRAN
<i>Directeurs de thèse:</i>	Professeur Thierry DIVOUX Professeur Eric RONDEAU	Université de Lorraine - CRAN Université de Lorraine - CRAN

Mis en page avec la classe thesul.

Remerciements

First and foremost, I want to thank my supervisors Prof. Eric Rondeau and Prof. Thierry Divoux. This work would not have been possible without their immense support, guidance and knowledge. But a bigger thanks for being much more than just my thesis directors and for taking care of all the difficulties that we have come through along these three years. I could not have imagined having a better supervisors for my PhD.

Besides my supervisors, I would like to express my special appreciation and thanks to Dr. Jean-Philippe Georges also known as JPG. I will always be grateful for the words of wisdom and the encouragement on my research. Thanks for bringing expertise and sense of humor, even in the more tedious moments. I am honored to have you as my mentor.

I would also like to express my sincere gratitude to the reviewer of my thesis Prof. Jalal Ben Othman and Dr. Pascal Berthou, for the time and effort they have put in reviewing and assessing my work. Their ideas, feedback and questions have been invaluable.

My most sincere thanks also goes to Prof. Colin Pattinson and Dr. Karl Andersson for evaluating my work throughout the whole PhD period as external. Their guidance and directions help me to finish my work in time.

Last but by no means least, I would like to thank my parents and my wife. Words cannot express how grateful I am to them. They are the most important people in my world and I dedicate this thesis to them. Without their mental support, it would be a difficult journey for me.

I also want to thank all of my lab mates, friends and colleagues I have met during my PhD for the exchanges of knowledge and skills, helping and boosting me and for those unforgettable moments we have enjoyed and shared along these years.

Finally, I want to thank my life coaches, Mr. Katsuki Bakugo and Mr. Izuku Midoriya for teaching me, life is all about trying and not losing hope. Some difficult situations can be handled with strategies and in some cases being tenacious is the only way.

To my parents and my wife

Contents

List of Figures	ix
Chapter 1 Introduction	1
1.1 Problem statement	2
1.2 Thesis contribution	4
1.3 Thesis organization	5
Chapter 2 Green ICT and green networking: state of the art	7
2.1 Energy efficiency technology for networking devices	8
2.1.1 Network hardware level	9
2.1.2 Network software level	15
2.1.3 Network design level	17
2.2 Software defined networking: A paradigm shift in energy savings	22
2.3 Brown energy vs green energy	28
2.4 Open issues and discussion	31
Chapter 3 Analysing the power consumption of the core network: GÉANT	35
3.1 Topology and traffic analysis	36
3.1.1 Day vs night and week vs weekend traffic analysis	38
3.1.2 Testing bandwidth capacity	38
3.1.3 Why not brute-force?	41
3.2 Shared path first algorithm	42
3.2.1 Network model formalization	43
3.2.2 Network constraints	44
3.2.3 Algorithm description	45
3.3 Result analysis	47

3.4	Conclusion	49
Chapter 4 Energy aware routing: An SDN perspective		51
4.1	Accepted green policies	52
4.2	Defining planes: control, data and management plane	54
4.3	Formalization of the energy model	57
4.4	Heuristic algorithm: genetic algorithm for solving minimization problem	59
4.4.1	Algorithm description	64
4.4.2	Design of experiment	67
4.4.3	Tuning of the algorithm	69
4.4.4	Simulation setup	70
4.5	Result analysis	71
4.5.1	Tuning outcome	71
4.5.2	Fitness function analysis	77
4.6	Conclusion	78
Chapter 5 Moving towards Sustainability		79
5.1	Quality in Sustainability	80
5.2	Introducing environmental metrics	81
5.2.1	Carbon emission factor	82
5.2.2	Non-renewable energy usage percentage	82
5.3	Energy aware routing vs Pollution aware routing	84
5.4	Proposed solution	86
5.4.1	integration of environmental metrics into objective function	87
5.4.2	Design of a sustainable system	87
5.5	Evaluation of the model in terms of QiS	89
5.5.1	Result analysis	89
5.5.2	Discussion on sustainability	93
5.6	Evaluation of the model in terms of QoS	93
5.6.1	QoS parameters	94
5.6.2	Result analysis	94
5.7	Conclusion	97

Chapter 6 Green-networking and stability	99
6.1 Network stability issues	99
6.1.1 Incoherent data planes	100
6.1.2 Convergence issue in distributing a data plane	101
6.1.3 Network instability and its occurrence	103
6.2 Improving stability of the system	105
6.2.1 Introducing penalty	105
6.2.2 Plane filtering: To implement or not	109
6.2.3 Extension of the model	111
6.2.4 Case study	111
6.3 Conclusion	119
Chapter 7 Conclusion and Perspective	121
7.1 Main outcome	122
7.2 Perspectives	123
7.2.1 Variability of the static part of energy model	123
7.2.2 Implementing the solution in LAN context	124
7.2.3 Applying the solution in end to end network	124
7.2.4 Temporal analysis for environmental factors	124
7.2.5 Implementation challenges of SDN platform	125
7.2.6 Different heuristics for solving the problem	125
7.2.7 Intelligence defined network for increasing stability and QoE	126
Appendix A	127
Appendix B	129
Appendix C List of Publications	131
Bibliography	133

List of Figures

2.1	The trend of electricity consumption of the main components in the ICT Sector	8
2.2	The breakdown of a core router's power consumption	10
2.3	Software defined networking architecture	24
2.4	Classification of traffic aware approaches for SDN	25
2.5	CO_2 emission by sectors comparison between Europe and the world	29
2.6	Global Traffic Trends. (a) CDN vs Non-CDN IP-traffic, (b) worldwide IP-traffic both mobile and fixed (2002-2022)	31
2.7	Venn diagram of the state of the art	33
3.1	Approximate power consumption of different network nodes	36
3.2	GÉANT Map	37
3.3	GÉANT topology graph	37
3.4	Weekday vs Weekend 24hours traffic analysis	39
3.5	Link usage percentage (a)40Gbps (b)10Gbps	40
3.6	Routing Topology (a)Shortest path first, (b) Shared path first, (c) Shared path first sorted	48
3.7	(A) Adjacency Matrix with Shortest Path First (B) Adjacency Matrix with Shared Path First Sorted	48
4.1	Energy model	53
4.2	(a) Topology (Portion of GÉANT Network), (b) A possible Data plane, (c) control plane	55
4.3	A generic architecture of the solution	56
4.4	Genetic Algorithm Process Flow	60
4.5	Structure of the chromosome	61
4.6	Solution retrieve process RDFS vs DFS	62
4.7	Example of crossover operation	63
4.8	Example of mutation operation	63
4.9	Flowchart of the whole procedure	66

4.10	Design of Experiment general architecture	69
4.11	Main effects Plot for energy gain compare to SPF	72
4.12	Power consumption analysis of the algorithm over longer time period	73
4.13	Effect of different crossover rate over time	73
4.14	Effect of different mutation percentage over time	74
4.15	Pareto Chart of the Standardized effects	75
4.16	Energy consumption gain compare to SPF	76
5.1	Emissions of selected electricity supply technologies (gCO ₂ eq/kWh)	82
5.2	Standardization of environmental metrics for GÉANT network	83
5.3	Basic Architecture of the System.	85
5.4	Standardization of environmental metrics for GÉANT network	86
5.5	Generation of sustainable data and control plane	88
5.6	Comparison between three algorithms	90
5.7	Topology for different algorithm	92
5.8	Network topology for algorithm CO ₂ _ GA in modeler	95
5.9	E _{TE} delay comparison between algorithms	96
5.10	Packet loss comparison between algorithm	97
6.1	Data plane incoherence example	100
6.2	Changes of topologies with random demand set changes	102
6.3	Occurrence of incoherent plane in time curve	104
6.4	Example scenario for demonstrating penalty mechanism	107
6.5	First solution with higher configuration cost (Ω')	108
6.6	Second solution with lower configuration cost (Ω'')	108
6.7	Overall system architecture with penalty and filtering mechanism	112
6.8	Avoidance of incoherence plane by penalty and filtering mechanism	112
6.9	Current topology with data and control plane for case: 1	114
6.10	Rejected solution where $\Omega' > Th$ for case: 1	114
6.11	Accepted solution with $\Omega'' < Th$ for case: 1	115
6.12	Current topology with data and control plane for case: 2	117
6.13	Rejected solution where $\Omega' > Th$ for case: 2	117
6.14	Accepted solution with $\Omega'' < Th$ for case: 2	118
B.1	Countrywise energy, CO ₂ and non-renewable energy consumption values	130

Introduction

Contents

1.1	Problem statement	2
1.2	Thesis contribution	4
1.3	Thesis organization	5

In this modern era of innovations, the ever-growing information and communication technologies (ICT) sector plays an important role. Over the last decade, enhancement in technologies, the accessibility of low-cost network transmission devices and last but not the least increasing popularity of streaming services and the Internet of things have forced ICT to face exponential demand growth from an ever-increasing number of connected devices. If we look at the values, according to [IPCC, 2019] the global Internet traffic of 2022 will be equivalent to 75 times the traffic of 2007. If we consider per capita measurement, it will reach 50 GB per capita in 2022 whereas in 2017 we had only 16GB per capita. This shows how drastically the user demand is increasing. With this steady rise of the network usage causing an increasing energy usage and the amount is not non-negligible either. It has been forecasted in [Andreopoulou *et al.*, 2014] that, in 2022, the total electricity consumption due to ICT will be doubled compared to 2010 and in 2030 it will be tripled. This rise of energy consumption has a concerning impact on the environment. According to [Shuja *et al.*, 2017], currently, ICT is responsible for about 2.4-3% of global electricity consumption. And if we consider carbon footprint, ICT equipment is liable for 2-2.5% of the global emission which is the same as the aviation industry. [Wang *et al.*, 2012],[Alena and Libor, 2012]. Moreover, it is expected that the ICT-related CO_2 emissions will increase to 4% by 2020 [Maksimovic, 2018].

Among the main ICT fields, network systems, architectures, and devices are responsible for one-third of the total energy consumption of the ICT sector [Lannoo *et al.*, 2012]. Moreover, they have the highest electricity consumption growth compare to end-user equipment and data centers.

In 2016, the International Energy Agency (IEA) [IEA, 2016] published that the global electricity used by internet data transmission networks in 2015 amounted to around 184 TWh. Which is a quite large number to be neglected. As a result, the need for energy efficient networking has become a key goal involving government and industry efforts and seeking a great deal of interest from the research community. In the future, how ICT will impact on the environment depends on the acceptance and adoption of power efficient solution. Moreover, more than half of the worldwide industries are directly or indirectly is depending on ICT and largely depend on the networking system, Therefore, increasing efficiency in data network will simultaneously reduce the environmental impact of the other sectors.

1.1 Problem statement

This research aims to reduce the environmental impact of network devices. One of the main ideas behind green networking is to improve energy efficiency and reduce undesirable energy consumption which will, in fact, reduce the carbon footprint produced by ICT devices by Energy Aware Routing (EAR). EAR is implemented by different power optimization techniques like turning off the unused network devices (nodes and links), Energy Efficient Ethernet (EEE) or Adaptive Link Rate (ALR). Due to the link over-provisioning considered in the design and operation of backbone networks, a significant amount of energy can be saved by applying these techniques. However, it is important that at the same time variation of the traffic (i.e.: period with burst and period with low traffic) is also addressed. Within this perspective, a centralized approach like Software Defined Network (SDN) can be a very well-suited architecture to improve the state of the energy efficiency of the network and manage the unused links and nodes of the network in a coordinated way.

In this work, we tried to reduce the environmental impact of network devices using a centralized approach. For such purpose, we have identified a few key related challenges which have naturally guided and motivated the investigation.

The idea of green networking or energy-aware routing is based on the fact that networks are over-provisioned. We have more network devices than we actually needed. The capacity of these devices is also designed for tackling the worst case scenario and most of the time under-utilized. However, as we can see the traffic trend has been changed enormously during the last decade or so. So, our first concern is to analyze the network behavior with the demand requirements mimicking the actual demand throughput requirement and how to handle these demands robustly. It is not the same notion of increasing the number of demands. If the throughput requirement is very low compared to the bandwidth capacity of each link of a network, increasing the number of demand does not have any problem whatsoever. For example, let us consider a network where every link has a link capacity of 10Gbps. Now, if there are 100 demands with through-

put requirement all ranging from (10Mbps – 50Mbps), then even in the scenario where all the demand flows are going through a single link, the maximum possible link utilization will be 50% (50Mbps *100 = 5Gbps). On the other hand, on the same topology, if there are 25 demands with throughput requirement all ranging from (1Gbps – 5Gbps), then one link can be used maximum by 10 links and in worst case, even with two demands one link can be 100% utilized and routing algorithm needs to find out the route of other demands avoiding this route until the demands are over. Because with the increasing popularity of social media and video streaming platforms and also the increasing accessibility of the Internet to the world has changed the demand of the user. Therefore, in this work, we seek to analyze demand behavior and pursue to optimize the energy consumption of the network with this increasing demand size by providing an algorithm designed in a centralized approach to advance the state of the art.

The primary goal of a network is to provide service to the user. Therefore, in simple terms, a network can be optimized energy efficiently only if there are idle nodes and links. If everything is in use and to its full capacity then there is nothing it can be done to reduce energy. Hence, come our second research question, “Is it enough to focus only on energy consumption while trying to achieve green networking?” Because when we want to reduce energy consumption, actually the target is to decrease the carbon emission of the network equipment so that the adverse effect of networking devices to the environment is lessened. All the energy aware routing algorithms consider every network device of the same capacity consumes the same amount of energy regardless of their geographical position. And it is entirely true. But does it mean that all of the same environmental impact? Because based on their geographical position the carbon emission factor will be different for each device. This factor depends on how the electricity is generated in that region. In a huge network like GÉANT which covers a large part of the geographical region, we seek to optimize our routing algorithm focusing on reducing the overall environmental impact rather than only considering energy consumption.

Lastly, most green routing algorithms are applied to the network as soon as there is a change in demand. However, frequent change of the demand flows causes the frequent triggering of the routing mechanism which might lead to incoherent planes and convergence problems during distributing planes among the routing table of the node. These problems significantly increase the possibility of being the networking system unstable. Hence, our last research question, "Is this solution should be implemented or not". Because, the main goal of green routing is to reduce environmental impact, however, the main goal of a network is to provide service to the user. Therefore, if a green solution can lead the system to an unstable state then a mechanism must be applied which will balance out between green solution and stability. A green solution reduces the over provisioning of a network system. On the other hand, over provisioning leads to a more stable system. We have focused to optimize this problem so that the green solution

will also consider stability issues into the mix.

1.2 Thesis contribution

This thesis work contributes in accordance with the research questions and problem statement. Firstly, before jumping into proposing any solution, we have thoroughly analyzed the network attributes of a core network (i.e.: GÉANT) so that the traffic behavior can be analyzed and at the same time these analyzes provide significant information about which green policies will be suitable for implementing energy aware routing into the network. After that, the thesis represents a couple of traffic engineering algorithm to justify the findings of the analysis and at the same time show the scope of energy savings. These algorithms (i.e: Shared Path First and Shared Path First Sorted) considers a greedy approach where the savings are targeted only based on data transmission links. These first set of approaches elucidate the possibility of saving energy and also about green policies.

Therefore, in order to overcome current energy-related limitations that can be seen in different state of the art and move to a more flexible, effective and environmentally friendly traffic engineering approach, this thesis then presents as a first contribution, an energy-aware routing algorithm tailored to fit centralized system like network with SDN capabilities. Two green policies are considered to maximize savings. The first one is widely accepted, which is turning off/sleeping of unused nodes and links. And the second one is using of adaptive link rate (ALR) so that, after turning off the unused links, the links which are in use will be adapted to the bandwidth requirement. We have approached this traffic engineering problem using the genetic algorithm as our heuristic function. Our algorithm considers the user demand as input of the system, and provide a set of data and control plane which describes the best suited topology to be implemented in SDN enabled network. The data plane describes the paths that will be used by the demands. On the other hand, the control plane explains the adaptive link rate mechanism. The performance of the algorithm is also described.

Secondly, we propose an extension of the algorithm where alongside with energy, we have introduced two environmental metrics namely carbon emission factor and non-renewable energy usage percentage as an objective function for the algorithm. We have introduced these two factors as by only considering energy it is impossible to completely address the issues of pollution and resource depletion. After adding these environmental factors to the algorithm, we also provide a holistic system design to cover three pillars of sustainability to improve the quality in sustainability. In this way, data and control planes are calculated based on the objective function and significant performance improvement is attained depending on the objective function. We also conducted a quality of service (QoS) analysis of the system. Depending on the objective function, the algorithm works differently. Therefore, performance evaluation in terms of QoS parameters of each objective function is conducted and analyzed.

Thirdly, considering the unpredictable nature of incoming traffic in current networking environments, performing a dynamic energy-aware routing is one of the main technical challenges as it often leads to incoherent data planes and even an unstable system. For this purpose, in this thesis, we have also introduced a penalty and filtering mechanism to the system to improve the stability of the networking system. This proposal added a penalty extension to the original algorithm which helps to provide a better solution in terms of stability. At the same time, the proposal helps to isolate the scenarios where incoherency might occur and demonstrate how this penalty and filtering mechanism balances between system coherency and green networking.

1.3 Thesis organization

This dissertation organized as follows. Chapter 2, discusses the principles of the energy-aware routing problem and presents a literature review with the relevant research work and assesses their main contributions and drawbacks. This chapter describes the energy efficiency technology from the perspective of three broader sections: network hardware level, software level and design level. Later on, it discusses about approaches considering SDN and also renewable energy. The aim of this chapter is to provide an updated and thorough perspective of the state of the art scope in this research area, identifying the networking challenges that are covered in this thesis. Chapter 3 analyzes the power consumption behavior of GÉANT, a core network based on the traffic demand. At the same time, this chapter contains different analyses of traffic flows. In the later part of chapter two energy aware routing algorithms are described and the result analysis of these algorithms compared to the shortest path first is also presented in this chapter.

Chapter 4 introduces novel energy-aware routing solutions for SDN using the genetic algorithm as our heuristic. Before providing the algorithm it describes data, control, and management plane from our problems perspective. And later on, it describes the genetic algorithm in detail. We have conducted the design of experiment for obtaining the appropriate parameter values of the genetic algorithm. This whole setup of the experiment is also described in this chapter. In the end, this chapter provides results about parameter tuning and performance of the fitness function.

In chapter 5, we have discussed about the environmental parameters and how to integrate these parameters into our routing algorithm. It also describes a novel routing technique Pollution Aware Routing (PAR) and explains the difference between PAR and EAR. This chapter also contains the overall system design of a sustainable system considering all the parameters. Lastly, it describes the performance of the different objective function in terms of quality of service parameter: packet loss and end to end delay.

Chapter 6, introduces a novel mechanism penalty and plane filtering mechanism to provide a more stable system. It explains different instability issues like incoherent data planes and convergence problems that can occur during implementing both energy or pollution aware routing.

Then, it describes the extended model and using these mechanisms. And lastly, it discusses a few scenarios where these newly introduced mechanisms can reduce the possibility of an unstable system.

Chapter 7 summarizes the thesis contributions and presents future research directions and perspectives.

Green ICT and green networking: state of the art

Contents

2.1	Energy efficiency technology for networking devices	8
2.1.1	Network hardware level	9
2.1.2	Network software level	15
2.1.3	Network design level	17
2.2	Software defined networking: A paradigm shift in energy savings .	22
2.3	Brown energy vs green energy	28
2.4	Open issues and discussion	31

This chapter is divided into four main parts. In the first part, we have discussed about different energy efficiency technology for networking device on the light of several research works. We have categorized these energy efficiency technologies into three broader sections: network hardware level, software level, and design level. In each part, relevant research works in the literature dealing with energy efficiency are introduced. Energy-aware routing mechanisms were initially considered for traditional IP networks. However, recently networking research community moved their attention to the use of Software-defined networking (SDN) for minimizing power consumption. Therefore, in the second part, we have discussed about SDN and put emphasis on the literature, which is related to this thesis. In the third part, we have discussed the works which are focusing on green energy sources. And finally, we extract the main open issues regarding the energy awareness property in routing strategies and discussed the research opportunities based on the related works and open issues.

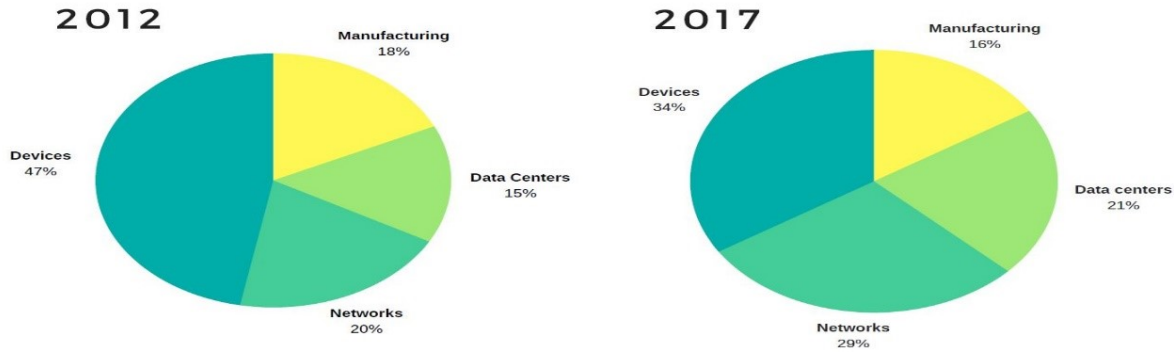


Figure 2.1: The trend of electricity consumption of the main components in the ICT Sector

2.1 Energy efficiency technology for networking devices

The availability of inexpensive and extreme capacity optical transmission and improvement in routing technologies for the network infrastructure have made possible to grow the Internet at this current unstoppable rate. This increasing demand and improvement of the ICT technologies are leading to higher energy consumption. According to European Commission’s digital agenda report in 2013, “ICT products and services are currently responsible for 8 to 10% of the EU’s electricity consumption and up to 4% of its carbon emissions.” [Europa, 2013]. And among several sectors of ICT, network architectures and devices are one of the highest energy consumers.

Figure 2.1 shows the trends of energy consumption of different components of ICT and it shows a tremendous uprising of the energy consumption of the networking devices compare to other components [Corcoran and Andrae,]. This is because of the exponential increase of the user of the Internet and most of the network architectures and resources like bandwidth, processing power and memory are designed keeping in mind the peak time network usage [Zhang *et al.*, 2010],[Addis *et al.*, 2014].

Even though it performs well during peak hours, but during off-peak hours it causes redundancy, over-provisioning of the resources and consumes extra energy which is unnecessary. As the passing of data through the network cannot be stopped, ICT engineers are needed to be tactical about how to make the network architecture more energy efficient with respect to traffic. Energy-efficient networking is in the limelight of the researchers for a quite long time. Gupta, who is one of the pioneers in this field, coined the concept of greening the internet in his work [Gupta and Singh, 2003]. In this research, they suggested the possible methods for reducing the carbon footprint of ICT network infrastructure. Authors in [Rondeau *et al.*, 2015] compare the traditional and current metrics for assessing ICT performance and their effects on the environment. In [Drouant *et al.*, 2014] a new approach for designing a green network architecture is

introduced. Numerous works have done considering the problem of energy efficiency in network and there are mainly three research directions. The first direction mainly concentrates on the network hardware level. And the second direction is more on network software level. When building an energy-efficient network infrastructure, alongside two mentioned directions there is one more direction that focuses on the practical level of green network designing and it can be identified as network design level. However, these three research directions are molded and extended into many branches. For example, especially for design level, energy savings are mainly done changing the current topology. Therefore, a design level solution can be online or offline. On-line solutions mean the solution will take place during the run time of the system and many of the variables might be unknown such as the number of demands or the throughput requirement. On the other hand, off-line solutions mean based on traffic patterns a solution will be provided which will not change within time and more importantly the traffic pattern is known beforehand. Another important thing is the approach to solving the problem. A centralized approach can be used where the state of all the nodes in a network is known by the central controller and decision can be made globally, whereas in distributed solutions each node only knows the whereabouts of its neighbor node and any decision has to be made locally. Here, different energy-saving research directions are discussed in light of different works. As our research scope bounded to the wired network, therefore all the works that are described here, are also completely based on the wired network.

2.1.1 Network hardware level

In the network hardware level, energy consumption is reduced by developing more efficient circuits and power-saving techniques. Improving the power draining components like memory or modifying the cooling system are all part of efficient device buildup. On the other hand, techniques like Energy Efficient Ethernet (EEE), Adaptive Link Rate (ALR) are all part of different power-saving techniques.

Hardware re-engineering:

In the beginning, most of the network devices are designed to keep in mind to provide the highest level of performance without giving much attention towards energy consumption. Therefore, these network devices are always consuming more energy than it actually requires. Each of these devices comprises several components like power supply and cooling device, forwarding engine, switching fabric, control plane, input/output cards, and buffers. Several studies have been conducted focusing on the consumption of the specific components and how energy savings can be done by modifying a particular component without hampering the performance level of the current network device. Figure 2.2 shows the power consumption percentage of the different

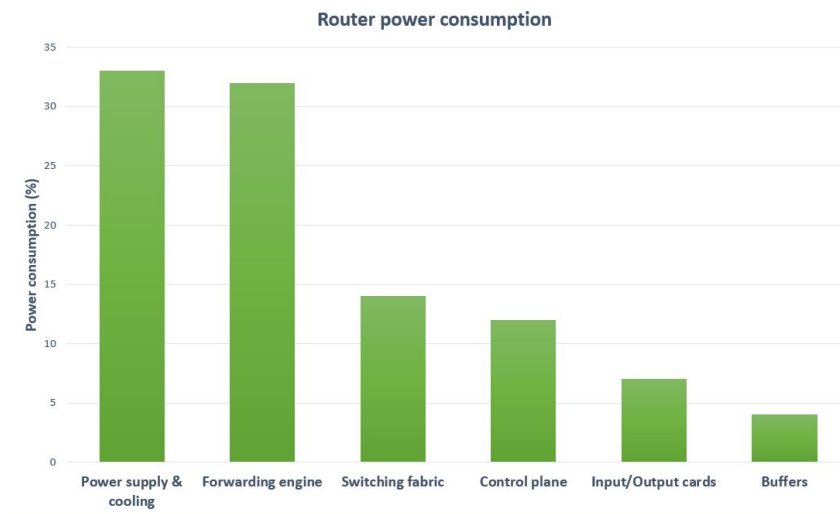


Figure 2.2: The breakdown of a core router’s power consumption

components of a core router. The values have been taken from [Hinton *et al.*, 2011].

A node can be in several states like hibernation, turned-off/on. Similarly, in a module-based design of a network device, each component can be considered separately and different energy-saving techniques can be applied to each of them. Researches have been done so that both input/output cards and buffers can be put into sleeping mode so that it can save energy while not in use. Figure 2.2 shows that power supply and cooling are responsible for more than one-third of the total energy consumption of a core router. If the router is in use then cooling is mandatory and the method of cooling is somewhat controllable. An energy-efficient approach is accepted by a few companies which is *free cooling*, proposed by [Pawlish and Aparna, 2010] where outside air is used to cool the network infrastructure. Infrastructure unlike core network, where nodes are location independent for example datacenters, many leading tech companies like Facebook moved there datacenter to Sweden in order to use the cold climate for sustainable cooling. [Orgerie *et al.*, 2014] Whereas another tech giant Google uses sea water of Finland to achieve a free cooling system [Google, 2013]. The second re-engineering is done in the memory of the network devices. Ternary content addressable memory (TCAM) is introduced for the forwarding engine. These memories can be accessed block by block and other portions can be turned off for energy saving purposes. TCAM is getting very popular to provide high throughput from the forwarding engines end. As forwarding engine is also responsible for 30% of the overall energy consumption of the core router, many works are done to improve the performance of TCAM and use it more energy efficiently [Zane *et al.*, 2003] worked on TCAM and proposed to construct a preclassifier to activate TCAM blocks separately resulting significant amount of energy savings. [Li *et al.*, 2017] on the other hand, provided a two-stage scheme for TCAM based

IP routing table lookup which efficiently places the routing table into different TCAM blocks without creating memory holes and guaranteeing both energy and memory efficient solution.

Hibernation/Turning off:

A device cannot consume energy, if it is not on. Therefore, putting the network device in hibernation or turning it off is one of the most basic yet effective ways to save energy that has been considered by many researchers. There is a slight difference between hibernation and turning off. Hibernation consumes a little bit more energy than turning off whereas at the same time the up-time of a network device from hibernation is much quicker than when a device is turned off. Hibernation or putting the device in the sleeping mode is the key concept for the modeling of the energy-efficient routing that will be discussed in the network design level. [Gupta and Singh, 2003] is one of the first work that considered hibernating the nodes and the links and if possible turn of the devices to increase the amount of savings. They showed how much energy is consumed by the internet and also identify the scope of saving energy by putting the devices into sleep. They analyzed the impact of selectively putting line cards into sleep mode on the switch protocol. Their work was one of the preliminary work in this field, where line-cards were manually chosen to put into sleep by analyzing the traffic pattern. Same authors also published similar research in [Gupta *et al.*, 2004] and [Gupta and Singh, 2007a]. In [Gupta *et al.*, 2004], they have discussed the energy saving potential in LAN switches by putting them into sleep. They examined the traffic of a LAN network and found out that there are three time-spans in a day when the traffic is lowest, both line cards and switches can be put into sleep. They have fixed a predefined time schedule for sleeping of the device and thus saving substantial amount of energy. Which can save energy but at the same time, in many cases it provides huge amount of packet loss. Moreover, this method fails to utilize all the other intermediate periods where energy can be saved. In [Gupta and Singh, 2007a], they proposed a dynamic technique to shut down the unused link and turn them back on when it is required. They have designed a dynamic algorithm that uses buffer occupancy, behavioral analysis of the previous packets and maximum bounded delay to make sleeping decisions. However they have used synthetic traffic to evaluate their model and even though in some cases they can turn off 80% of the links but the maximum traffic load is 5% of the total network which is unrealistic. So far, all the discussed work is based on sleeping or turning of the links and link ports, however in [Bolla *et al.*, 2009], authors provided an framework that can be used to optimize the energy consumption of a network device. In this framework they have included both network device and links to be put into sleep mode. They have considered the network device as multi-core routers, and based on the traffic density they proposed to turn off cores of the routers to save energy and if none of the cores are used then the whole router can be turned off to save energy.

This work showed that, comparing to idle state turning off the device can save more than 25% more energy savings. All the above mentioned works focus on saving energy by considering the nodes or links to be put into hibernation or to be turned off. Hibernation or shutting down of the device is one of the core concepts of energy efficiency networking. All the other directions of green networking like energy aware routing or virtualization are based on this primary concept.

Energy Efficient Ethernet (EEE):

Even with the advancement of wireless and mobile networks, Ethernet is still the universal networking interface in the world. Every network traffic passes through several Ethernet links before it reaches its destination. However, it has been concluded by several researchers that, majority of the Ethernet links spend a considerable amount of idle time during its whole lifetime. As network devices consume more than 10% of total ICT power consumption, Energy Efficient Ethernet (EEE) has been introduced which provides new standards and mechanisms for reducing the overall power consumption of the network devices without hampering the core function that network devices are supposed to do. The concept behind EEE is very simple. Communication links will only consume power when it is in action that means sending or receiving data. Most wired communications are using continuous transmission, that means no matter the state of the network device, busy or idle, it will be always counted as active and consumes power even it is not involved in any kind of communication. The idea was to always be ready when it needed to transmit data. Now, EEE uses this scenario in favor of saving energy by putting the links in idle mode. In order to save energy, when there is a void period in the data stream, EEE uses a signaling protocol that permits a transmitter to imply that there is a gap in the data and that the link can go idle. The signaling protocol is also used to indicate that the link requires to recommence after a predefined delay. The EEE protocol uses a modified version of the normal idle signal which is usually transmitted between data packets. This signal is known as low power idle (LPI) mode. The transmitter sends an LPI signal in place of idle signal to inform that the link can go to sleeping mode. After sending LPI for a period ($T_s = \text{time to sleep}$), the transmitter can stop signaling entirely so that the link becomes idle. Periodically, the transmitter sends some signals so that the link does not stay on idle mode for too long without a refresh. Finally, when the transmitter wants to be fully active and functional link, then it sends normal idle signals which indicate the desire of the link to be active. After a predetermined time ($T_w = \text{time to wake}$) the link becomes active and can participate in data transmission. The best part of the EEE protocol is as there is no minimum or maximum sleep interval, therefore, a link can become active at any time. This provides full control to EEE to in case of unpredictable traffic.

The authors of [Gupta and Singh, 2007b] showed how energy can be conserved by using this low power idle mode for the Ethernet LAN. They have proposed a method where they have merged

LPI with the shutting down of the link. They proposed a concept where un-utilized links will be put into LPI mode and after a certain period if there were no signal to make this link be back on, then the link will be in shut down state in order to maximize the energy gain. There are some common misconception about LPI which are described below:

1. *LPI and ALR*: Even though at the beginning several proposals have been made, but in the final version of LPI, link scaling has not been implemented. An LPI enabled link does not reduce its capacity when it is going on idle mode. When 1000BASE-T link goes on LPI mode, it still remains as 1000BASE-T link. The capacity does not change while a link is in LPI mode. It is a common misconception that EEE uses link scaling to reduce energy consumption.
2. *LPI and Sleeping/Hibernation*: LPI or low power idle mode only works for network links. It saves energy by putting the links in idle mode. However, sometimes it has been compared with putting the device in sleeping or in hibernation mode which is completely different. During LPI, network devices are always on. Only the link and that interface for that link is put on idle mode.

Adaptive link rate (ALR):

As discussed earlier most of the networks are designed over-provisioning which gives ample opportunity to reduce power consumption to a certain extent. Networks are designed to tackle the maximum request, therefore, all the network link is fixed to its maximum capacity even during the idle or low traffic period. The adaptive link rate (ALR) is downgrading or adjusting the link capacity according to the incoming traffic. Link capacity can be ranged from 10Mbps to 100Gbps depending on the router and the network. For example, in a core network, the lowest capacity of a link is 1Gbps and the highest can be 100Gbps. However, in a regular switch, the lowest capacity is usually 10Mbps and the highest can be 1Gbps. Therefore, no matter the network device, the link rate can be reduced when there is less traffic resulting in less power consumption. The key challenges in realizing ALR are:

- Defining a mechanism for swift changes in data rates of the link.
- Creating a policy to change the link data rate to maximize energy savings (that is, maximize time spent in a low link data rate) without significantly increasing the packet delay.

Several works have been done in order to handle these two challenges. According to the work of [Nedevschi *et al.*, 2008], when a negotiation process is done during the link setup, the

maximum capacity link is usually chosen even if a lower capacity link can fulfill the requirement. Therefore the adaptive link rate mechanism offers an algorithm that automatically adjusts the link speed according to the traffic. That means when there is a high level of traffic it will remain in its maximum capacity. However for a certain duration if there is no significant traffic is passing through the network then the network device will auto-negotiate to a lower level of link capacity. In their work, they discussed the technique of choosing the level of link speed according to the traffic pattern. If the traffic is high link speed will increase automatically and vice versa. They showed that this auto-negotiation process takes hundreds of milliseconds to complete. During this time communication is interrupted as the link is down which can be vital for providing better QoS or QoE. They discussed briefly how greening the network is always a trade-off between QoS and energy savings. They also showed that choosing the ideal rate for ALR is NP-hard (Non-deterministic Polynomial-time hard).

[Blanquicet and Christensen, 2007] proposed a method to select the physical layer devices (PHYs) more quickly. In this method, a frame exchange is used for renegotiating the change of the speed that withholds the need of restarting the auto-negotiation process. This method makes the transformation of the speed faster but still, it is not fast enough. The adjustments of equalizers, echo cancelers, and timing circuits are needed which takes time. Researchers are working in this field in order to make the change even quicker. Nevertheless, there is a great advantage in switching to lower data rates whenever it is possible. According to them, the main difficulty of implementing ALR is choosing the speed. It is always hard to know when and how much the speed is needed to be lowered.

[Gunaratne and Christensen, 2006] shows that 90% of the time in a small network 10Mbps links are enough to fulfill the needs of the users. However, the experiment is done in a small network with less amount of demand, therefore the value does not depict the real picture. Nevertheless, even in large networks, it is seen that the lowest possible bandwidth is actually enough for handling demands. They have proposed a mechanism to do the auto-negotiation and switching of the link rate. They have also proposed a Markov model for packet arrivals which are following Poisson distribution. This model helped to understand the relation between packet delay and energy savings.

The authors of [Gunaratne *et al.*, 2008] also proposed a MAC handshake mechanism to fix the link rate automatically by following the buffer occupancy. To follow the buffer occupancy, the number of bytes is needed to be calculated all the time which might increase the complexity of the hardware. In order to ease this problem, they proposed a heuristic where they have proposed a timer-based switching mechanism. Two time intervals were defined. One is for a high link rate and the second is for a lower link rate. The link rate must stay at a higher rate for a fixed amount of time before switching it to a lower rate regardless of the buffer occupancy. And the policy was itself an adaptive one. The duration for staying at higher rate gets double every time there is a mandatory switching in the link rate from lower to higher before the fixed interval

ends.

All these works have proved the usefulness of using ALR for saving a significant amount of energy consumption.

2.1.2 Network software level

Network software level is a research direction where energy efficiency is achieved by introducing different techniques in the software level of the network device. Network software level solutions extend to virtualization, green TCP, resource consolidation and green SNMP. A brief discussion is given below.

Virtualization:

The concept of virtualization works by dividing up the shareable resources, for example, operating system and server network connections which give the impression of network applications. This is one of the common and feasible green networking strategies at the software level that can be implemented in almost most of the network. It creates an impression that network applications are working independently with their independent resources and in reality, they are working with shared resources. In the case of network virtualization, bandwidths are split into channels where different applications are assigned to different channels. Therefore, as each channel works independently, every application gets a sense of independence. For splitting bandwidth, two types of methods are usually followed. The first one is time division multiple access (TDMA). In TDMA, every channel gets the complete bandwidth but only for a certain amount of time. It is more of a round-robin process. The second method is known as frequency division multiple access (FDMA), where every channel is assigned to a fraction of the total bandwidth. The purpose of virtualization is to increase the utilization of certain links so that it can reduce the number of active links. For example, if there are five different applications which require to use network resources, it is better to use one single link for five of them rather than assigning a single separate link to each of them where most of the time the links are idle. The virtualization is generally used in VPN-like applications, and do not permit a complete de-coupling between logical nodes and physical platforms. The authors of [Wang *et al.*, 2008] introduced the concept of router virtualization into the new level, which allows logical nodes to roam around different physical platforms maintaining zero packet loss. Similarly, the author of [Muyassarov, 2018] uses network function virtualization (NFV) to reduce the energy consumption of cloud radio access network. NFV utilizes the concept of virtualization where different network functions are decoupled from vendor proprietary hardware and deployed on distributed pool of commercial off-the-shelf servers using virtualization.

Resource consolidation:

Resource consolidation is an effective and efficient use of computer server resources. It is also a widely used method for achieving software level green networking. The main concept of resource consolidation relates to virtualization as it reduces the total number of required active machines by increasing the utilization of some of the machines. In large companies, it is a common practice to have multiple servers in order to provide better QoS. But usually, these servers are under-utilized. Resource consolidation provides the companies an opportunities to reduce energy consumption by reducing the number of active servers. However, even though consolidation can bring efficiency and increase utilization, it makes the data complicated in nature that is difficult to understand for general users. Therefore it is often applied with virtualization, as it keeps the applications isolated from each other. The author of [Srikantaiah *et al.*, 2008] showed how resource consolidation can be used in a network server in a cloud computing perspective.

Green TCP:

Green TCP is an energy-aware application proposed by [Irish and Christensen, 1998]. Generally, the task of an energy-aware application is to mainly detect the energy saving scope and in many cases predict the idle periods by analyzing its own behavior pattern so that energy can be saved. While prediction, it tries to find out the upcoming long idle period so that they can go to sleep mode or low power state that will reduce the power consumption by a certain level. Now transmission control protocol (TCP) is a connection oriented protocol that requires acknowledgment (ACK) for every transmitted data. If the sender does not receive the ACK, the data will be re-transmitted and after a certain amount of mistrials, the connection will be disconnected. Usually, when a network device is in idle mode, it is unable to send the ACK resulting connection lost. Therefore, the concept of green-TCP is to add an extra control message that will allow the client to go to sleep mode without losing the connection. They have proposed a 'sleep' message that can be sent by the client to the server which will indicate that the client is now going to sleep mode and at the same time there will be a time-stamp which will indicate the duration of the sleep period. In this way, the connection will not be lost as the server will not send any message to the client during this period. After a certain period of time or during the case of urgency, the client will send a 'wake' message indicating the server that it is now again ready to receive data and the server will send all the buffered messages to the client. However, if the duration of the time-stamp is passed and the client has not sent any wake-up message to the server then the connection will be lost. Green-TCP is implemented entirely in software level and both the client and server applications are needed to be updated to integrate this methodology.

Green SNMP:

In [Blanquicet and Christensen, 2008], the authors proposed, prototyped, and evaluated a new software level energy efficient technique which is green SNMP (Simple Network Management Protocol) extensions. They have extended the basic SNMP management information base (MIB) with SNMP Power State MIB extensions and its agent to obtain the equipment's power state to the network. This power state includes all the supported power management capabilities, current settings, total and currently active, inactive, and sleep times and statistics on wake up and sleep events. These information are vital knowledge in order to design an energy efficient network. With these power states of all the network devices, the network manager or a controller (i.e.: in a centralized network) can make efficient changes and power settings all over the network devices. Similarly, Cisco has introduced Cisco EnergyWise MIB which specifies the protocols for managing power distribution of all the network devices. It has different power level for example, shut down, sleep, hibernate, stand-by etc. to classify the power settings of each entity (i.e.: network devices).

2.1.3 Network design level

In the network design level, the whole network architecture is considered and optimization of the routing is done in order to reduce energy consumption. This is also known as energy aware routing (EAR). The basic concept behind the idea of energy aware routing is to manage the smooth transmission of the traffic across the network while reducing the number of unused resources by turning off them or putting them in a hibernation state. Another way is to maintain the network performance by applying the adaptive link rate and re-route the traffic according to available link capacity. Specially, the current network design does not imply the proportional behavior with the usage and available network resources. Normally, network infrastructures are designed with more network resources than it requires in order to tackle the situation in case of failures. A similar concept is applied while considering the link rate. Links are always equipped with maximum possible capacity to ensure flawless performance during sudden peak flows of traffic. These behaviors while designing networks lead to undesired wastage of energy during low traffic periods. The concept of energy aware routing is to minimize these unnecessary wastages of the energy from the network design level point of view using the network hardware level concepts like: Hibernation, turning off, ALR and EEE.

According to [Zhang *et al.*, 2010], there are mainly two challenges that are needed to be addressed while working with EAR. The first one is how to manipulate the routing paths to maximize the energy savings and secondly how to achieve these energy savings without significantly hampering network performance and consistency. Identifying the nodes and the links that are needed to be kept active in order to route all the incoming flows of traffic make this energy aware routing problem a NP-hard problem. Moreover, the capacity constraints of the links introduces an

additional level of complexity in the problem. And because of that, for most of the real-world topologies of the network which are large in number of nodes and links, the optimal solutions in terms of computational resources and time complexity becomes unrealistic. Hence this research field emerges and quite a few number of heuristics are developed and proposed to solve this problem by the research community.

According to the different variants considered in the model, the existing energy aware routing heuristic solutions can be divided into two groups: traffic based solutions and topology based solutions. Traffic based solutions are solutions which propose routing mechanism by aggregate traffic over a network subset in a redundant network. Usually, this kind of heuristics are designed under the assumption of knowing the characteristics of the traffic beforehand. Some heuristics are designed assuming the duration of the traffic whereas some are designed assuming the capacity requirement of the traffic. Proactive solutions are possible if all the traffic flows are known beforehand. On the other hand, topology based solutions are focused on manipulating the topology in an energy efficient way without knowing the traffic beforehand. These are mainly reactive solutions where nodes and links are turned on/off based on traffic requirements. However, in both cases, the goal is to shutdown the maximum number of nodes and links in order to save energy wastage. Later in this section, we have discussed a few of the prominent works in this field in brief.

One of the most significant works in this area is [Zhang *et al.*, 2010], in which authors propose GreenTE, an intra-domain, centralized traffic engineering mechanism that is used to identify the links of a given topology that can be turned off in order to save energy. Heuristic is designed based on the availability of the traffic load. The solution is a proactive solution as the traffic demands are routed through a set of previously computed k-shortest paths. The paths and the overall solution are calculated based on solving the mixed integer linear problem (MILP). Performance requirements such as Maximum Link Utilization (MLU), number of hops and network delay are also considered as constraints in the problem. Based on these constraints they have proposed three different variations of the problem. The first one is basic considering only the MLU constraints. The second one considers the hop count and the third one considers the delay factor. The first one considers all the solutions for a particular source and destination set whereas in the other two number of solutions are limited. From the traffic flows they have pre-calculated the candidate paths for each source and destination set. In order to solve their MILP, they have used CPLEX which is a linear programming problem solver. They have used Abilene, GÉANT, Sprint and AT&T networks to test their solution. However, Only for Abilene which is the smallest of all the topologies, CPLEX was able to solve their MILP and provide solutions for all three variations of their algorithms specially the basic one as all the solutions are considered. As they showed that, the run time of their algorithm heavily depends on the value of k.

In a similar way, [Bianzino *et al.*, 2010] aim to find the network configuration that minimizes

the network energy consumption, modeled as the sum of the energy spent by all nodes and links carrying traffic. In order to achieve this, they formulated an optimization problem for finding network subsets consuming lowest power while fulfilling network demand. They have also used CPLEX to solve their problem. They have worked with GÉANT network and their work pointed out the fact that, in a crowded network if all the nodes are used as source and destination there is not so much opportunity to save energy. They have maximized the link utilization and concerned about the reliability of the network. They have not used ALR in order to increase the savings. In [Cianfrani *et al.*, 2010], a distributed energy-aware routing (d-EAR) algorithm is proposed and the routers are known as Exporter Routers (ER) and Importer Routers (IR). ER calculate the shortest path tree (SPT) while Importer Routers (IR) takes these shortest path trees calculated by ER as the reference and make the modified path tree (MPT) by assuming ER as its root. The IR finds the links that are needed to be switched on to connect to the shortest path tree of the ER and also identifies the links that can be turned off. The selection of the ER in the network can determine the number of links to be switched off and consequently holds potential for substantial energy conservation. As the number of ERs selected will affect the network performance, this can be set based on the trade-off between the network performance and desired energy saving. The one drawback of d-EAR algorithm is that, they have not considered a real network scenario. As they do not have reliable information about the open shortest path first (OSPF) weights of the links they have assumed a mean value which can be far away from reality. An extension of d-EAR is EAR studied in [Cianfrani *et al.*, 2011] which improves the ER selection procedure and maximizes the number of ER by preserving the QoS. In both cases, they have achieved energy savings by turning off links but not nodes. Here they have used two opposite directional heuristics to improve the results. In one case they tried to improve solutions of different IR by making each of the solutions compatible with each other in order to turn off more links. And in the second case, they check for minimum used links and try to shut them down by re-route the traffic of that link to other links. In this work also they have used assumed values for OSPF link cost and made a comparison with the d-EAR and these new two heuristics.

In [Chiaraviglio *et al.*, 2011], they have formulated the problem of switching off network elements (i.e.: nodes and links) while guaranteeing full connectivity and maximum link utilization as a variant of the Multi Commodity Minimum Cost Flow (MCMF) problem which is a NP-hard problem. Their primary solution can be applicable only for trivial networks. Therefore they have provided heuristics based on greedy algorithm that can be used in large networks. They have proposed a greedy heuristic that iterates first all the network nodes and then through the links. In the beginning, they start by considering all the nodes are turned on. The algorithm starts by one node which is iterated through the set of nodes; the set of nodes are sorted in different ways like: random, least-link, least-flow and most power. And by each iteration, they turn off one link based on the sorting criteria and check if traffic can be still transmitted or not.

If yes then they keep the link turned off and move to the next node according to the sorting. And if no, then they keep the link on. They have used few synthetic topologies to evaluate their work where both topology and traffic demands are randomly generated. They have showed good outcomes in synthetic topologies. Their algorithm is applicable only when network traffic changes in 30 minutes or more than that. The computation time for their ILP is quite high. In terms of topology where traffic is steady and less changeable, the algorithm works fine for those scenarios.

Authors in [Cuomo *et al.*, 2011b] propose a routing algorithm called Energy Saving based on Algebraic CONnectivity (ESACON), using the algebraic connectivity as a metric to control the resulting network topology. The algorithm first creates an ordered list of the links of the network based on their impact on algebraic connectivity. That means if a link is responsible for transmitting the maximum number of traffic, it has a high impact on the algebraic connectivity. And if a link is not used for transmitting traffic then it has an impact below the threshold level and can be turned off in order to save energy. In this algorithm also, in the beginning, all the network devices are on. And after traffic flow has been established between the network the algorithm can be used. Therefore it is a topology-aware routing solution. They have evaluated the performance of ESACON in a synthetic network with synthetic traffic. They have shown good performance in terms of saving energy. However, they have considered 80Mbit/s as their highest traffic throughput whereas each link has a capacity of a 2.5Gbit/s and overall mean utilization of the initial network is less than 25% which is quite low for understanding the performance of the algorithm in terms of large traffic demands.

Similarly, another topology-based solution reported in [Cuomo *et al.*, 2012] which also takes into account the algebraic connectivity as a requirement to preserve the overall network connectivity. They have proposed a heuristic algorithm named Energy Saving based on TOPology control (ESTOP) which considers the 'edge betweenness' as a metric to measure the role of the link in the network. 'Edge betweenness' of an edge is the number of paths from all nodes to all other nodes that cross the edge. Therefore, if an edge has a high 'edge betweenness' that means this edge is significant for the network. Inversely, if an edge has low 'edge betweenness' that means that edge is least frequently used and therefore, the first candidate to be pruned. However, this approach is conceived to be implemented in a distributed way into each IP router. They have tested their algorithm performance on Abilene and Germany 50 with randomly generated traffic. And they also showed made comparison with ESACON and showed that ESTOP performs a little better in terms of traffic support capabilities. However, both ESACON and ESTOP both cannot be considered until a set of traffic flows is completely handled. As the algorithm does consider dynamic incoming traffic, therefore, for a set of traffic flows the algorithm provides a solution and that solution will be continued until all the flow has been reached to its

destination. For a set of new flows, all the links will be turned on and the algorithm will be applied.

The work in [Cuomo *et al.*, 2011a] also aims to improve energy efficiency reducing the number of active links. For this purpose, the authors propose mainly two variations the Energy Saving based on Occurrence of Links (ESOL) algorithm. And in the latter part, they made two more extensions in order to improve the performance of the second algorithm. The basic ESOL works in an iterative process. At first, all the links are ordered based on occurrence. To calculate the occurrence of each link, for each flow of traffic path is calculated using OSPF and then the link which is used in maximum paths has the highest occurrence. Based on that, links are removed with the lowest occurrence value one by one and check the connectivity of the network. The links are removed until the required connectivity hampers. The second variation is fast ESOL, where, in one iteration all the links which has lower occurrence value from a predefined threshold value. It performs better in terms of run time compare to b-ESOL, however in terms of energy savings it does not perform well enough. They have shown the trade-off between complexity and efficiency in powering off a great number of links. They have also used a synthesized network and traffic to make a comparison between all of their algorithms.

[Mumey *et al.*, 2012b] considered both links and nodes for saving energy in a wired backbone network. They have formulated the problem of saving energy into an integer linear programming problem and provided two heuristics to solve the problem in polynomial time. They have provided two simple heuristic ones is based on the shortest path and the second one is based on a minimum spanning tree. They have used CPLEX to calculate the ILP however, CPLEX was unable to provide results in the given time for a network with 14 nodes. The algorithm works for topology with 10 nodes. Even though they have mentioned they have checked the algorithm for heavy traffic, however, the maximum traffic flow had the throughput requirement 15 times lower than the considered capacity.

The authors of [Cianfrani *et al.*, 2012] discussed a solution called Energy Saving IP Routing (ESIR). ESIR is integrated with OSPF therefore in the worst case scenario the result will be as good as OSPF. ESIR is based on the concept of SPT exportation which allows sharing a Shortest Path Tree (SPT) between neighbor routers so that the overall set of active network links can be reduced. They have extended the solution with QoS-aware ESIR which takes into consideration QoS parameters like hop count. They have also discussed online and off-line solutions in one case routes are pre-computed in the second case, routes are computed dynamically. However, they have a fixed boundary on the load of the traffic, therefore traffic can never be higher than the assumed capacity of the links which is not practical and limits the suitability of their work for network services of varying patterns.

In order to switch off entire networking devices (both edges and nodes), the Steiner-tree-based algorithm proposed in [Matsuura, 2013] models the network topology as a graph composed of edge and core routers and their respective links. They have described an approach where the algorithm computes a Steiner tree or a set of Steiner trees to obtain the minimum number of sub-graph(s) which is enough to fulfill all the demands. And after that, it can remove all the nodes and links that are not involved in these sub-graph(s) to save energy. In order to improve the delay they have added a hop count threshold and in case of routes with hop-counts above the threshold, the original shortest path is added to the Steiner tree to minimize the path length along with energy savings mentioned in the research as a bypass route.

[Avallone and Ventre, 2012] have also proposed an energy aware routing algorithm called E2-MCRA (Energy Efficient Multi-Constrained Routing Algorithm) where each flow alongside with throughput requirement has QoS constraints like delay and jitters, and the objective of the algorithm is to find the best route considering these constraints on-line while try to turn off as many nodes and links as possible. E2-MCRA finds a QoS-satisfying path for each flow request focusing on the current network, so the minimum number of sleeping links and nodes needs to be re-switched on. However, the algorithm should be aware of the entire current topology including all nodes and links physically connected to the network. The proposed algorithm does not also care about re-sleeping the network devices after the network is re-switched to lower traffic load. They have compared their results with the initial network where no green policy is used (not even the shortest path).

2.2 Software defined networking: A paradigm shift in energy savings

In traditional networking, the networking device(i.e: switch/router) contains both the data plane and the control plane. The control plane provides the rules required for forwarding into the forwarding table. This forwarding table contains all the information regarding the next hop of each packet to reach their destination. And based on this forwarding table, the data plane forwards the packets to its destination. The intelligence in networking which includes security, forwarding, and load balancing all are done in switch level and as both planes are strongly coupled in every networking component, making any change in the network architecture is time-consuming and in many cases not feasible because every component of the network needs to be configured individually. In this regard, a novel paradigm emerges with the key design principle of separation of the control plane and the data plane, known as software defined networking or SDN. In disparity to current communication systems, it integrates a controller – a centralized control

entity containing the control plane and the data plane resides at the network devices, provides flexibility in managing network. SDN architectures generally have three components or groups of functionality: SDN applications, controller and network devices. SDN applications are the program that communicate with the controller and provide outside resources for decision making purposes. SDN controller is a logical entity which receives the instruction from the application layer and after processing the instruction, transmit it to the network devices. The controller has the functionality of configuring the forwarding tables (in other words, flow tables) of switches. And lastly, the network devices receives the instruction from the controller and act on it. This approach offers mechanisms allowing network operators to have a global point of view; therefore, the centralized approach is suitable for globally monitoring the system according to network policies and service requirements provided by application layer. SDN makes the network programmable, therefore, any changes done by the network engineers on the controller can be easily propagated throughout the entire network. The birds-eye-view of SDN is also very effective for identifying any zombie machine of the network. Zombie machine are compromised hosts (i.e.: Datacenter servers) that are connected to the network and often difficult to identify. The authors of [Hakiri *et al.*, 2014] and [Wibowo *et al.*, 2017] provide extensive state of the art about SDN research scopes, challenges, opportunities and future directions. Software defined networking provides several benefits including on-demand provisioning, automated load balancing, and the ability to scale network resources based on application requirements. SDN also helps to create completely virtualized data-center, where end-to-end network between user and servers can be created with flexibility and will have the scope to terminate the connection at any moment.

According to [Assefa and Özkasap, 2019] energy efficient approaches using SDN can be classified into three groups: traffic aware, end-system aware and rule placement. Traffic aware energy efficiency approaches are mainly designed based on the fact that, most of the network components (i.e.: Forwarding switches/routers) are under utilized. And as we have discussed in the previous section, a large number of devices can be turned off during off-peak hours. Approaches are designed to utilize the full potential of SDN and provide an energy efficient network. End-system aware approaches are mainly applicable for datacenter networks (DCN). In a DCN, most of the time there are few idle servers only kept for fail-safe purposes. Using SDN, these unused servers can be shut down and save energy. And the last approach is rule placement. Usually, when SDN is used, a set of rules is required in order to make a connection between controller and forwarding switches. Placing these rules is a NP-hard problem. And inefficient placement of rules might cause congestion and also unnecessary energy consumption. Therefore, the last energy efficient approach is to proper way of rule placement. Figure 2.3 gives a general outline of the SDN architecture and also identify the different parts of the architecture where the energy saving approaches are focused on. Using SDN can positively affect the energy consumption of the network. As the work done in Georgia Southern University proved that using SDN architecture can decrease the energy consumption significantly [Rawat and Bajracharya, 2016]. In this

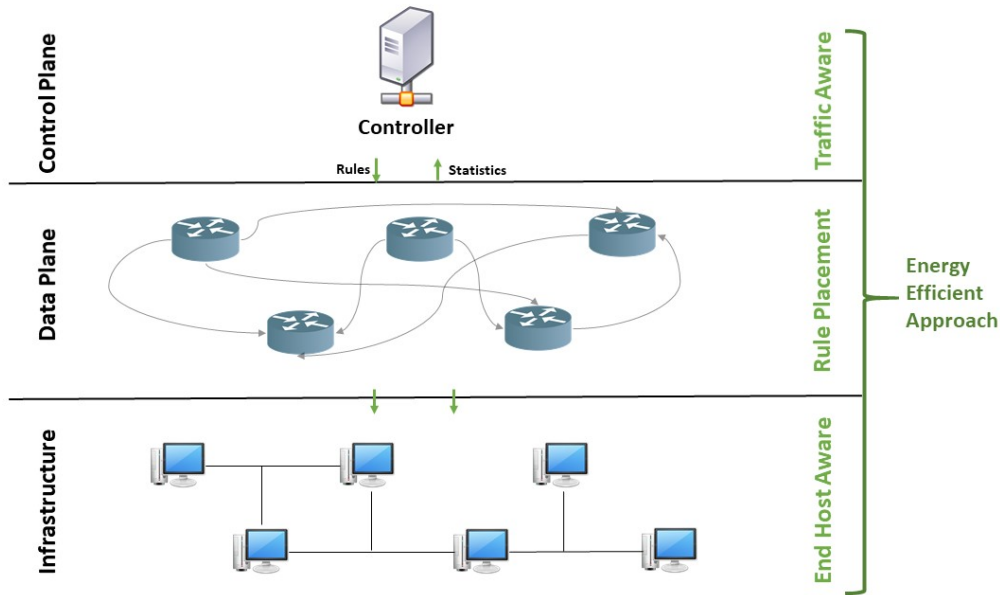


Figure 2.3: Software defined networking architecture

work, they compared the energy consumption of a conventional campus network and an SDN architecture. Accordingly, the results reveal that using SDN contributes to the reduction of the carbon emission by 25% or more.

However, as the work of this thesis focused mainly on traffic aware solutions, here we have discussed only the traffic aware approaches. These traffic aware approaches can be divided into three large groups based on the controller’s desirable characteristics. The first one is *elasticity*. Elasticity is the ability to dynamically increase or decrease the number of network components(nodes and links) to be used in order to fulfill traffic requirement. The second one is *topology awareness* which provides a global view of the topology that works as an advantage while formalizing and solving a problem of energy efficient networking. For example: in a hierarchically organized fat-tree of a data center or in a large core network with several nodes, prior knowledge of the network architecture and component’s real time characteristics like: current capacity, power status, etc. allows the network engineers to design energy efficient routing for the incoming traffic flows. And the last one is *smart sleep and off*, where the controller can turn off or put the nodes in sleep mode from a distance when they are not in use. And also turn them back on when they are needed to fulfill traffic requirements. Figure 2.4 shows the classification of different traffic aware approaches. Few prominent works addressing the traffic aware approaches of SDN are described

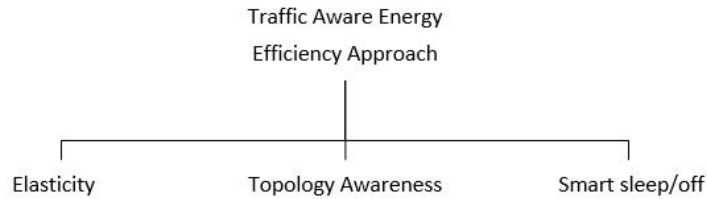


Figure 2.4: Classification of traffic aware approaches for SDN

below:

ElasticTree is one of the first power saving solutions for data center networks which is implemented on a test-bed using OpenFlow switches [Heller *et al.*, 2010]. They also formalized the common idea of turning off links and switches based on the amount of traffic load. They wanted to make energy consumption proportional to the network and it will change dynamically according to traffic. They have proposed three optimizers: formal model, greedy bin-packing, and a topology-aware heuristic. Each optimizer takes network topology (a graph), routing constraints, power model, and traffic matrix as input, and outputs a subset of links and flow routes. The first one is the formal model, which formulates the power saving problem based on objective function and constraints. The objective function is to minimize the total number of switches turned on and the number of links. However, due to the complexity of the model, it only scales up to 1000 hosts. The second one is a greedy bin-packing optimizer, which evaluates possible paths and chooses in the left to right order manner. By leftmost, it means a single layer in structured topology. Greedy bin packing improves the scalability of the formal model. However, for some traffic matrices, the greedy approach will not find a satisfying assignment for all flows. The topology-aware heuristic optimizer, on the other hand, splits the flow and finds the link subset. This approach uses Integer Programming (IP) to formalize the optimization problem. The drawback is the degradation of performance because of turning on and turning off components. In order to test the models, they have used testbeds using Openflow. ElasticTree contains more constraints, including fat-tree topology constraints, than any general model considering other topology. One drawback of elastic tree is that it does not take into account the correlation among the flows.

In order to optimize energy efficiency in SDN, authors in [Bolla *et al.*, 2013] present the GreenSDN approach, which integrates three different protocols that operate at different layers of the network: Adaptive Link Rate (ALR) [Gunaratne *et al.*, 2006], which is a chip-level protocol, Synchronized Coalescing (SC) [Mostowfi and Christensen, 2011], which is active at node-level, and Sustainability-oriented Network Management System (SustNMS) [Costa *et al.*, 2012], which operates at network level. ALR works on links and changes data rates according to the traffic

load of the network, whereas SC protocol works as LPI. However, while LPI works on individual parts of the network devices, SC can put a whole device into the idle mode. SustNMS controls the network and balances the trade-off between QoS and energy efficiency to quickly respond to changing traffic patterns. GreenSDN brings an energy efficiency simulation capability on the well known and widely used network emulator Mininet and POX controller. They have tested the performance of their algorithm in a topology with only two sets of source and sink. They used Iperf to generate traffic. The goal was to conduct the experiment in a simulation environment built on POX controller and test the effect of ALR, synchronized coalescing and sustNMS in terms of energy saving.

The author of [Rafique *et al.*, 2017] have considered ALR in their solution. They have provided a detailed benchmark implementation of the performance of centralized power aware routing heuristic algorithms. They have modeled the power-aware routing problem in SDN as a MIP problem. They implemented the problem using a general algebraic modeling system and solved it using CPLEX. They had considered the topology of Abilene for their experiment. However, they had taken only ten flows to evaluate their model showing the limit of their approach regarding scalability. As they have not mentioned how it will perform when the number of flows is larger.

In [Rahnamay-Naeini *et al.*, 2016] a novel method of traffic and resource aware energy saving is presented. Unlike most of the related work in energy efficient optimization problems which are linear, the proposed method formulates the problem as non-linear. They also wanted to use the centralized view of SDN as an advantage to design an energy efficient network by turning off nodes and links. In their objective function, they have included the price of the energy as a constraint. And alongside with nodes and links, they focused on reducing the cost of energy. They have designed an incremental heuristic to find the subset of links to be turned off. They have pre-calculated 10 shortest paths for each flows in order to compute the result of the algorithm in a reasonable time. They have tested their algorithm in Abilene and NSF network. However, they have randomly assigned the price of the electricity throughout the network.

Regarding the use of partially deployed SDN, the authors of [Wang *et al.*, 2015] faces the problem of saving energy hybrid scenarios where both SDN and IP network is considered. In this work, they have formulated an optimization model that aims to find minimum power network subsets and they have proposed a heuristic solution based on the use of several groups of spanning trees to satisfy the traffic loads. They have proposed to divide the traffic set into small clusters and for each cluster, they have proposed a minimum spanning tree to reduce the number of links that are turned on. For each spanning tree they considered one SDN controlled node will be there in order to connect to the controller. They tested in a synthetic network with 10 to 100 nodes. However, they could not test the performance of the algorithm when the number of nodes is large.

And secondly, their heuristic provides results when there are only 9 flows active at the same time.

The energy efficiency in hybrid IP/SDN networks is also addressed in the recent work of [Galán-Jiménez, 2018]. In particular, this paper introduces the energy-aware SDN nodes replacement problem aiming to improve the energy efficiency during the transition from traditional IP networks to fully deployed SDN. To solve this problem an Integer Linear Programming (ILP) formulation and a genetic algorithm are proposed. They have discussed around six transition techniques from IP network to a fully controlled SDN network. They have tested on few topologies to test which method is well suited. And according to them, MNL or most non-controllable link first, where the node with the highest number of non-controllable links connected to it is selected as SDN node at each iteration is the most effective way of transition.

Instead of assuming dedicated links between the controller and SDN nodes, the authors of [Ruiz-Rivera *et al.*, 2015] propose a model for controller-switch associations that aims to maximize the energy efficiency of the network. Although the routing of control traffic is considered in this work, they assume that controllers act as well as forwarding nodes, i.e. data plane traffic demands are routed through network controllers. Therefore, only links that belong to control paths are activated and data traffic demands are routed using these links until an MLU threshold is reached. They have randomly assigned controllers inside the network. They showed fewer controllers save more energy which is expected.

Multiple Topology Switching with Data Plane Forwarding Path Rerouting (MTSDPFPR) is an energy saving approach based on sleeping of the links [Ba *et al.*, 2018]. MTSDPFPR is a multi-level dynamic topology switching mechanism thorough finding the maximum utilized link which automatically performs topology switching based on dynamically acquired maximum link utilization(MLU). They have considered the simple fact that, in a network, the amount of traffic flows varies throughout the day. They divided the day time into a different time slot and for each time slot, they pre-defined a forwarding plane. As a heuristic, they have used k-shortest path to find out the forwarding plane. The approach is tested on real-world GÉANT network topology and traffic traces from SNDlib. However, in their approach, they have considered the traffic of a whole day at the same time which is not possible in real life. Moreover, no performance evaluation is done in terms of packet loss, as for this type of predictive approach, measuring QoS is necessary.

In [Fernandez-Fernandez *et al.*, 2016] also, authors have proposed a heuristic that addresses the problem of optimizing the power consumption in SDN using an energy-aware traffic engineering approach that minimizes the number of links that can be used to satisfy a given traffic demand. They have considered using multiple controllers. The approach considers performance constraints such as bounded delay for the control plane traffic and load balance between con-

trollers. The heuristic first finds a set of paths for source to destination and then sort the list based on the number of possible controllers to associate with and the number of possible control paths. The goal is to reduce the number of links from the network. The algorithm performs well with small topology like Abilene, but for GÉANT, it requires 3000 seconds to find a single solution. They have not used any solver they have coded the algorithm in python.

Similarly, two heuristic algorithms, namely Next Shortest Path (NSP) and Next Maximum Utility (NMU), are proposed in [Assefa and Ozkasap, 2017] to deal with the energy aware routing problem in SDN. They have first considered an initial Shortest Path Routing (SPR) which is same as OSPF. And based on that both models with heuristic seek to minimize the energy consumption of links and switches. Both heuristic tries to redirect a set of selected flows from underutilized links to more utilized links in order to turn off the underutilized links if all flows are redirected. NSP tries to put the flows to the next shortest path whereas NMU tries to put the flows which are next in line in terms of utilization. Results show that, NSP and NMU are more efficient in minimizing the average path length, and also more energy can be saved compared to SPR approach. They have considered GÉANT network however they have also considered the power consumption of a switch is equivalent to three times the power consumption of a link which is not correct in a large network like GÉANT.

2.3 Brown energy vs green energy

In the last few sections, several works have been described focusing on green networking however, one thing is common in all the above mentioned work is that all of them consider green-network the same as the energy efficient network. They have all tried actively or passively to reduce energy consumption in order to get an energy efficient network. Energy consumption is often consider as the equivalent of CO_2 emissions as worldwide 50% of the overall CO_2 emission is caused by energy sector [IPCC, 2011]. Figure 2.5 showed a comparison of the carbon dioxide emissions between the world and Europe. Even though Europe has lower the amount of carbon emission gradually with time, the value is still more than 50% of the total emission. Therefore, in terms of green or sustainable networking researchers have always focused on energy efficiency, even if its direct impact on the environment in terms of polluting earth's resources is not explicitly specified. However, an energy source can vary. An energy source can produce green energy or brown energy. All the energy produced from coal, fossil fuel or oil is considered as brown energy whereas energy produced from nuclear or renewable sources such as solar, hydro, wind, bio-fuel are considered as green energy. Depending on the source of energy carbon emission can vary a lot. For example, on a life cycle basis, burning natural gas for electricity production releases between 0.3 and 1 kgs of carbon dioxide equivalent per kilowatt-hour (CO_2 kg/kWh) and coal releases

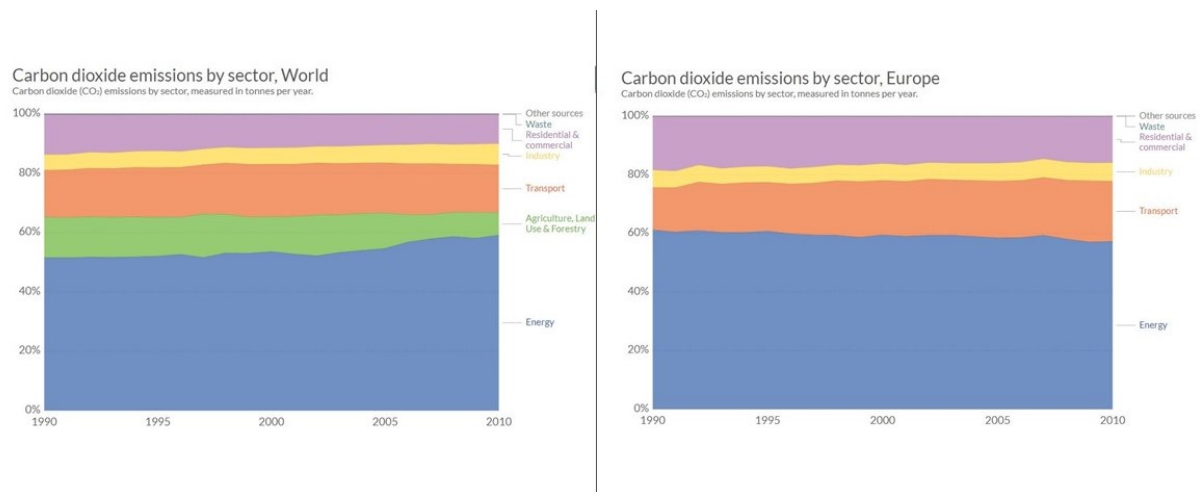


Figure 2.5: CO_2 emission by sectors comparison between Europe and the world

between 0.6 and 1.6 kgs of CO_2 /kWh. And when comes to renewable energy sources, wind is responsible for only 0.01 to 0.02 (CO_2 kg/kWh); solar 0.03 to 0.1 (CO_2 kg/kWh); geothermal 0.05 to 0.1 (CO_2 kg/kWh); and hydroelectric between 0.05 and 0.2 (CO_2 kg/kWh) [IPCC, 2011].

Because of this level of difference in terms of CO_2 emission, a new paradigm of research has been recently surfaced focusing on the green energy rather than brown energy. The sole purpose of these work is to manipulate the source of energy so that overall carbon emission can be reduced. In current times, in many cases reducing energy is not an option. Specially, when the amount of traffic is high in the network, all the users are active, usage of network is of its maximum capacity then there is no other way to reduce energy consumption. Internet service providers (ISP) are bound to yield the service at maximum capacity as reaching the packet from source to destination is the prime priority of the network. However, if we can make the changes from the end of energy source then even though with the same amount of energy consumption the carbon emission will differ.

Not so many works have been done considering alternative energy sources. Few of the prominent works are described below:

[Gattulli *et al.*, 2014], [Singh and Chandwani, 2015] and [Zhou *et al.*, 2013] all have tried to achieve green networking in data centers considering renewable energy as the alternative energy source. [Gattulli *et al.*, 2014] has provided two algorithms to route the traffic request from a user to Datacenter. The first algorithm is SWEAR (Sun and Wind Energy aware routing). It works based on the renewable energy value. They have considered a weighted graph, where each link has a weight to transmit the data. They have considered both the transmission and node

brown energy value. And chose the Datacenter which has the lowest brown energy value. In the second algorithm GEAR (Green energy aware routing), they have considered a greedy approach to find the best renewable energy-powered Datacenter. In this case, they have taken another input of how much energy is required to process the data and how much renewable energy is available in each node. And based on these they have chosen the best data center which has the lowest brown energy consumption. But as GEAR does not know about the possible changes in the availability of renewable energies in the future. Therefore, for the sake of comparison, every data center had enough renewable energy resources to fulfill the user requirement. The authors of [Singh and Chandwani, 2015] discussed a simple idea of choosing a green data center over a brown data center using SDN. Because of SDN they implied that, the controller will have all the information of a datacenter being green. However they don't define the level of green like [Gattulli *et al.*, 2014], they have differentiated only between green and non-green Datacenters. They evaluate the concept based on three criteria, nearest routing, green routing (routed to green datacenter) and priority routing (depending on the level of priority user request will be routed to different datacenter). When there is the high priority it acts as the nearest routing and when there is the low priority it acts as green routing. Even though using SDN they have a global view of all the router and decisions can be changed real time. However, as the level of renewable energy is not defined either the load of the datacenters, the difference between two green datacenters depends merely on hop count. [Zhou *et al.*, 2013] alongside with renewable energy, they have considered other criteria like geographical load balancing, Datacenter right-sizing (i.e.: energy cost and carbon emission reduced by dynamically adjusting the number of active servers) and server speed scaling. They have designed a carbon aware online control algorithm considering all the factors and evaluate their algorithm based on the energy cost vs carbon emission values of three states of the United States of America. All three works have considered renewable energy as green aware routing. However, Datacenter network(DCN) is quite different compared to core wired network as in DCN, in order to reduce the brown energy consumption they have changed the destination node based on the availability of the renewable energy which is only applicable in DCN.

The author of [Jo *et al.*, 2018] proposed a similar eco-friendly routing idea using renewable energy where they have used clustering to choose a cluster head with most renewable energy and then choose the member node accordingly. However, this solution is for wireless networks. [Mineraud *et al.*, 2016] has proposed a routing idea considering renewable energy for IP networks. They have utilized a novel gradient-based routing protocol, which favors forwarding packets along routers powered by the highest quantity of renewable energies. They have proposed an energy aware routing algorithm where one criterion of choosing the path will be the gradient value of the node in terms of renewable energy. The authors have considered different node has different level of renewable energy value and they have tested their algorithm on a network where for each router a renewable energy level is randomly selected.

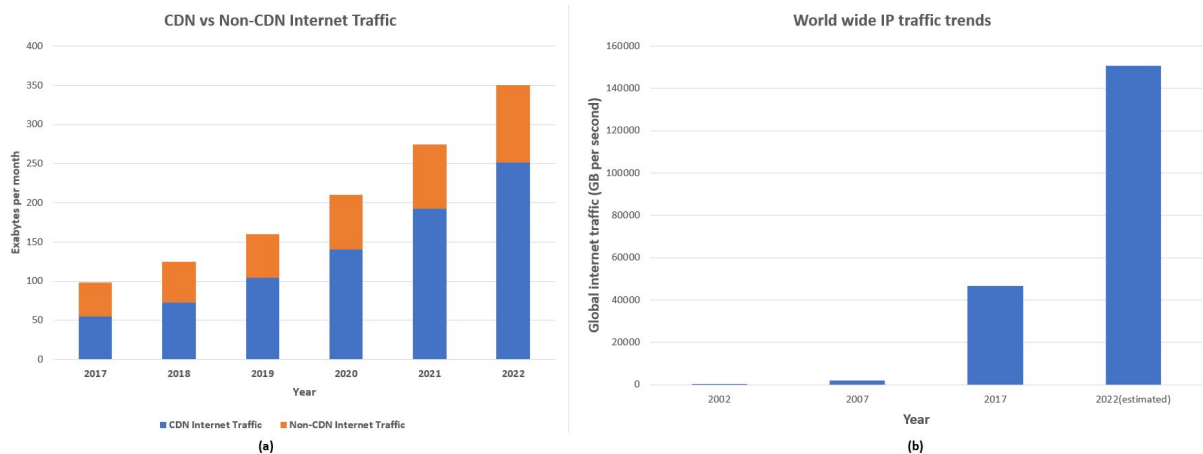


Figure 2.6: Global Traffic Trends. (a) CDN vs Non-CDN IP-traffic, (b) worldwide IP-traffic both mobile and fixed (2002-2022)

All these works have addressed reducing carbon footprint by reducing the use of brown energy. But it is pertinent to understand by the leader of the countries that, energy efficiency has its limitation. In order to improve green sustainable networking, green energy sources must be increased. Sadly this is not the case. According to world energy investment, compared to 2017, in 2018, renewable energy investment has been decreased whereas investment in oil and gas has increased [Agency, 2019]. This is high time that, alternative energy sources should be integrated into the research field of green sustainable networking.

2.4 Open issues and discussion

In this chapter, most of the techniques, specially for green routing, are covered. Every work shows different aspects of green networking and several solutions have been proposed to address the problem of energy aware routing with or without considering SDN. However, from this large collection of state of the art, still, there are few open issues that can be pointed out.

For the last decade or so, there is a massive change in the growth of the traffic. This change is brought about by the increasing role of Content Delivery Networks (CDNs) in data delivery. Large private CDN operators include Google (Youtube), Amazon, Facebook, Netflix, and Microsoft. File sharing, video streaming, live gaming, video conferencing, etc. services have made a drastic change in the global IP traffic.

Figure 2.6 shows the global traffic trends of the IP network. The figure in the left shows the booming of CDN compare to non-CDN traffic. As we can see, in the year 2017, CDN vs non-CDN has a percentage of 56 by 44 and with a cumulative total of 98 Exabytes per month.

And following the trend, it can be assumed that, in the year 2022, CDN will have 72% of the overall traffic with a cumulative total of 350 Exabytes per month. The graph on the right shows the traffic trend from the year 2002 to 2022 (estimated) for both mobile and fixed networks. In 2002, global Internet networks carried approximately 100 GB of traffic per second. Fifteen years later, in 2017, global Internet traffic amounted more than 45,000 GB/second [IPCC, 2019]. From these values, it is clearly visible that, Traffic pattern has been changed a lot in terms of quantity. But, Bandwidth has not increased in that level. For example, one of the most used sources of the traffic and topology for evaluating the algorithm is SNDlib. In SNDlib, for GÉANT network, the bandwidth that has been considered for each link is 40Gbps and the traffic is from the year 2005. According to the report of cisco, in 14 years or so, overall traffic has increased almost by 26 times but according to the latest GÉANT network, the capacity has been increased by 2.5 times. In almost all the paper has considered the traffic values of SNDlib for evaluating the performance of their algorithm. However, they have ignored the fact of this increased traffic. There are actually two reasons for ignoring this increasing traffic flows. The first one is, in a saturated network if most of the nodes and links are busy, it is difficult to show the performance of their algorithm as all the work is measuring the gain of their algorithm in terms of energy savings. And secondly, many algorithms are designed only for low traffic scenarios (i.e.: Algorithms focusing on MLU). Large size flows will increase the complexity of their algorithm by congesting links.

This increasing amount of traffic, opens up another question which is, what is the alternative if the network is saturated. The first and foremost duty of a network engineer is to route the flow from source to destination without any disruption. Green networking always comes after the network is fully functional and all the flows are able to reach the destination. But when we have a saturated network, and network elements are hard to shut down then it is very difficult to gain the performance in terms of energy savings. Therefore, it is now high time to change the way of thinking and find an alternative way to achieve the gain in green networking.

Even though the adaptive link rate is a fairly good option for saving energy, not so many green networking proposals with SDN has accepted ALR wholeheartedly. This is mainly, because of the unavailability of the tools to implement ALR in the SDN perspective. With a proper motivation from the researcher's end can influence hardware engineers to spend time on it. As ALR is a transceiver-level capability that copes with the under-utilization and over-provisioning of Ethernet links by dynamically changing data rates, a fitting integration is required between the data and control plane and ALR mechanism. Both data and control plane should be able to define the changes in different level of link rate all over the network.

Another point is that, many of the algorithms defined the problem of energy aware routing in terms of integer linear programming and as all the energy aware routing problem is NP-hard,

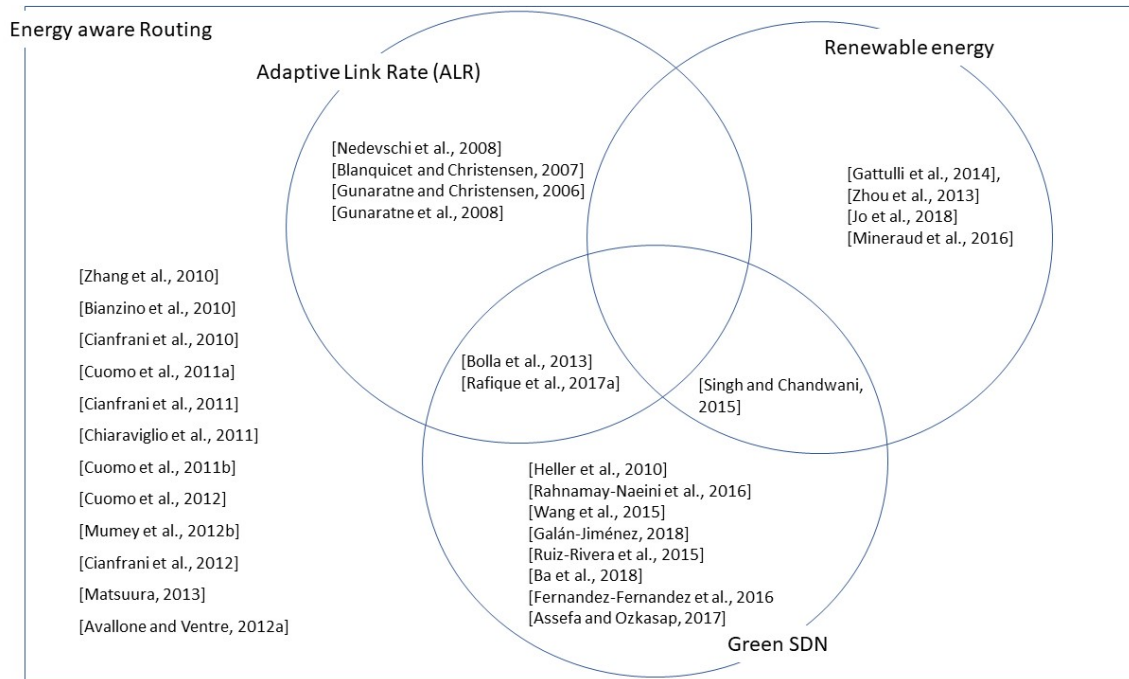


Figure 2.7: Venn diagram of the state of the art

then they have introduced their heuristic algorithm to solve the problem in polynomial time using CPLEX solver. CPLEX is a very high-end mathematical tool for solving ILP problems. In an SDN approach, if we want to build a network where the controller can process real-time data and make decisions for energy aware routing, then a very high-end processing tools are required for using CPLEX. Which is not discussed by the authors of the research works.

Reducing the number of active links or links and nodes, is the common factor for all the energy aware solutions that have been discussed in this chapter. Different energy-saving techniques are discussed but when comes to energy aware routing alongside with sleeping/LPI/turning off of the links, ALR is also another way of saving energy. Moreover, we have also discussed alternative energy sources. And last but not the least, we have discussed about the fairly new approach - SDN to deal with the energy problem. Figure 2.7 shows a Venn diagram, where the circle represents the work that we have discussed about ALR, SDN and renewable energy respectively and also shows the works that intersect the group. All the work represents a bigger collection of energy aware routing. The works outside of the circles solved the problem without using SDN, ALR or alternative energy source. This Venn diagram summarizes the overall state of the art and also helps to point out the uncharted research field.

Analysing the power consumption of the core network: GÉANT

Contents

3.1	Topology and traffic analysis	36
3.1.1	Day vs night and week vs weekend traffic analysis	38
3.1.2	Testing bandwidth capacity	38
3.1.3	Why not brute-force?	41
3.2	Shared path first algorithm	42
3.2.1	Network model formalization	43
3.2.2	Network constraints	44
3.2.3	Algorithm description	45
3.3	Result analysis	47
3.4	Conclusion	49

Network infrastructures are mainly comprised of core nodes, transport nodes, and access nodes. With the addition, there are transport links that connect every node in order to form a network topology. In every sector of the network, there are scopes that can be manipulated to save energy. According to [Bolla *et al.*, 2011], among these three kinds of nodes core network nodes consume the most. Figure 3.1 shows the approximate power consumption per device of the different parts of the network. And even though the number of core nodes is less compare to access nodes, core nodes are responsible for 30% of the overall power consumption consumed by networking devices. Moreover, there are network links that also consume energy and specially in the core network, the networking links are at its highest capacity. The power consumption of networking links are proportionate to its capacity. More it has the capacity more it will consume energy. However, in core networks, capacities of these links are established based on the peak

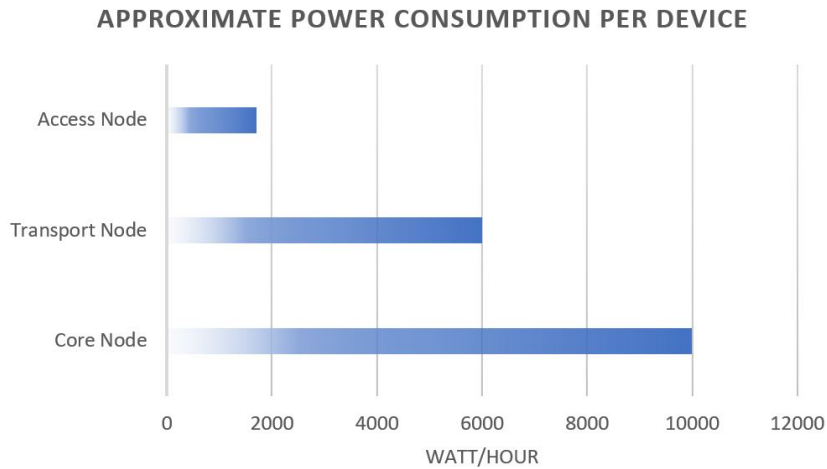


Figure 3.1: Approximate power consumption of different network nodes

traffic requirement and most of the time of the day they are underutilized and consuming more energy than their actual needs. In this chapter, we have discussed about one of the prominent core networks across the globe - GÉANT and evaluate and analyze the possibility of power savings of this core network.

3.1 Topology and traffic analysis

GÉANT project was first started in November 2000, for the purpose of serving the research and education community operating in Europe. It interconnects all the national research and education networks (NRENs) across Europe. After the success of this project, the second version of the project is launched known as GÉANT2 which had a larger coverage than the previous one. It includes thirty national networks in thirty-four countries. Then the next GÉANT project (GN3) started in 2009 and continued for four years. Currently, GN4-2 is running as a part of the GÉANT 2020 Framework Partnership Agreement (FPA). The main goal of this project is to provide a stable and advanced environment for growing the research community of Europe. Currently, the GÉANT network is the largest and most advanced research and education network in the world by gathering 38 NREN partners. It is uniting over 50 million end-users at 10,000 institutions across Europe and covering over 100 national networks worldwide. [Geant, 2017]. Figure 3.2 is the network map that has been used in the rest of this thesis work.

In this thesis, we have considered the widely accepted topology provided by the Sndlib. The topology has 22 nodes and 36 links. Each node is considered as a country. The topology is redrawn in figure 3.3 in order to visualize it more accurately. In the figure, each node has been assigned with a number. The list of countries and their indicative numbers are listed in Ap-

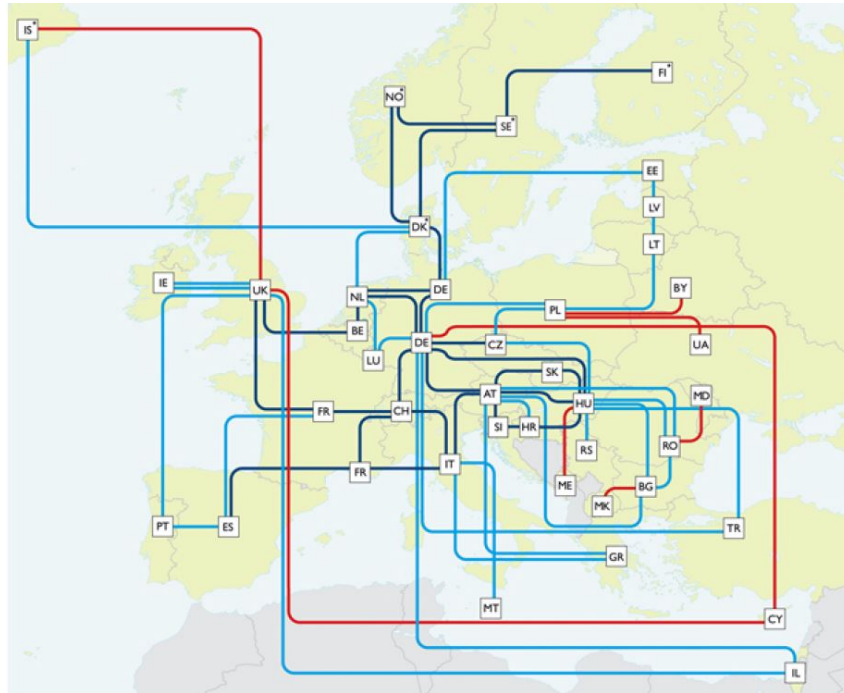


Figure 3.2: GÉANT Map

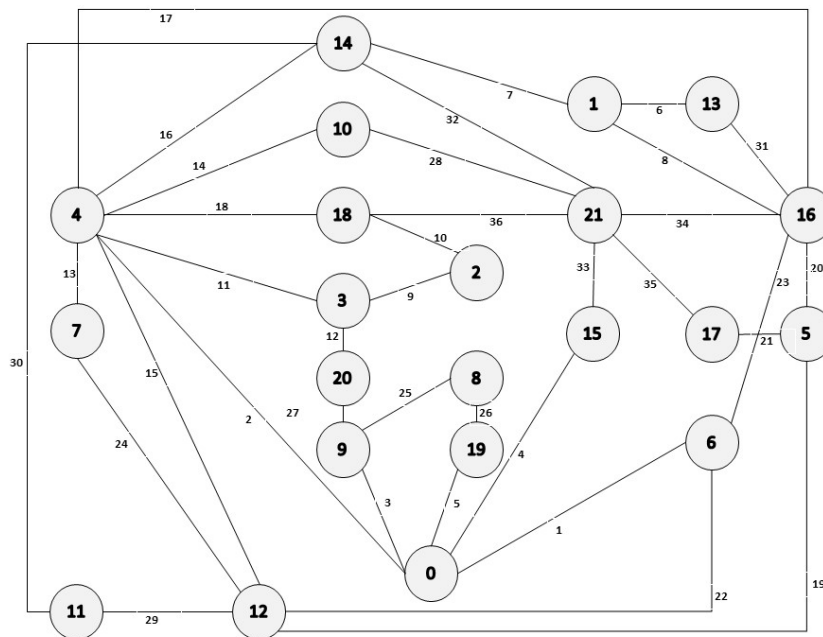


Figure 3.3: GÉANT topology graph

pendix A, table A.1. While drawing the figure, the distance of the nodes/countries are not to the scale. Links are operated with different values varied from 1Gbps, 10Gbps, 40Gbps, and 100Gbps. Sndlib provides an extensive set of traffic matrix where each traffic matrix has a set of traffics for the duration of 15 minutes from the year 2005. In the given data by Sndlib, the link capacity is considered 40Gbps for each link.

3.1.1 Day vs night and week vs weekend traffic analysis

A series of experiments are conducted in order to analyze the different characteristics of the raw data. At first with the raw traffic data, some simple analysis has been conducted in order to understand the behavior of the traffic. By behavior we mean, how the traffic load changes with time during the day and also during months. We have conducted traffic load analysis with both weekday vs weekend and also how the traffic varies during 24 hours. To do that, we have considered a whole month's data then averaged the whole month's data. The traffic load of weekdays and weekends are separated and averaged together. Then the values are plotted in figure 3.4 where x-axis shows the different times of the day and the y-axis shows the amount of average data rate transferred over the network. Figure 3.4 clearly shows the impact on traffic during the weekend and weekdays and also how traffic varies during different times of the day. For example, the traffic has been almost reduced by half at 6:30am compare to 12:30pm on weekdays. This actually also indicates the behavior of the Internet user and how office time has an impact on the load of the traffic. On the other hand, traffic remains somewhat constant in the weekend over the period of 24 hours. The data usage maximizes during the middle of the day and reduced to an extent at night time. But interestingly, during the duration of 12 a.m. to 5 a.m., the load of weekend data is comparatively higher than weekdays load. Whereas, during day time the weekend traffic is almost 40% less than weekdays. It gives a significant amount of idea about how the traffic behaves in the core network and how the network like GÉANT is underutilized during certain times of the day and also certain days of the week. This provides primarily, the idea about certain energy savings opportunities in the core network like GÉANT.

3.1.2 Testing bandwidth capacity

As mentioned earlier, in the provided Sndlib traffic matrix they have considered 40Gbps for each transmission two-directional link. Therefore, in the second experiment, we have tried to understand if there is any link that is underutilized or not for given traffic. In order to test that, we have considered a single random set of traffic flow. Each set of traffic matrix contains a set of demands(i.e: 440 demands in this case) with source and destination during the time of 15 minutes. Then we have used a simple well known routing algorithm open shortest path first (OSPF) to route all the traffic flows. The concept of OSPF is to find the shortest

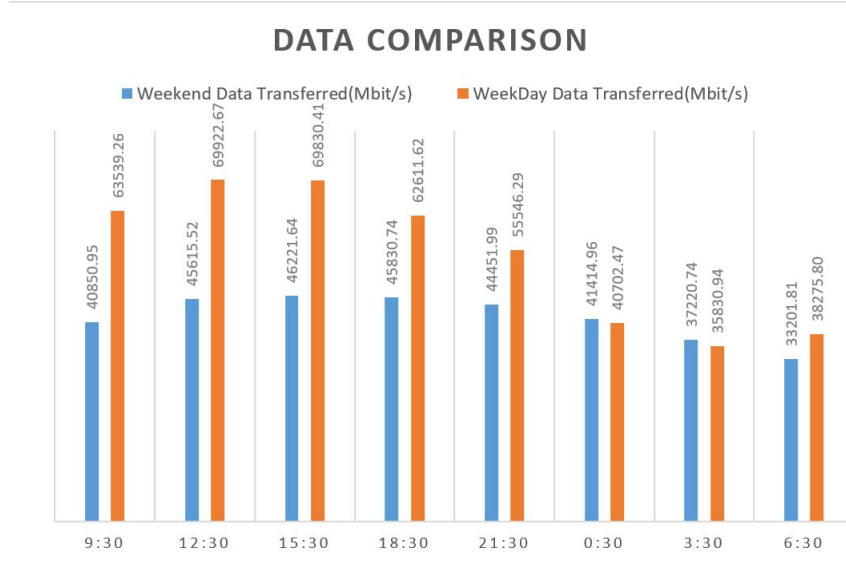


Figure 3.4: Weekday vs Weekend 24hours traffic analysis

path based on the cost of the route, taking into account bandwidth, delay and load for each traffic flow from source to destination using Dijkstra algorithm. Using this OSPF, every flows are routed through the network. As each link is considered as 40Gbps in the network, we have conducted experiments varying the bandwidths of the link of the network. We have considered four different levels for link bandwidth and each level all the links have the same bandwidth. The four different levels of bandwidths are 40Gbps, 30Gbps, 20Gbps, 10Gbps. That means, for 10Gbps bandwidth level, all the links (in the case of GÉANT 36 bidirectional links) have a link capacity of 10Gbps and so on. We have not considered 1Gbps as there are few flows larger than 1Gbps. For the same traffic matrix, we have conducted the experiment to see if OSPF can route all the traffic from source to its destination or not. As an output, we have generated a $[\text{number_of_nodes}] \times [\text{number_of_nodes}]$ (i.e: for GÉANT it is 22 x 22) matrix which actually shows the link load of all the links of the topology. If two nodes do not have any direct links in between then that cell is empty. When a cell value is 0, that means there is a link but not in use on one side. As the links are bidirectional, a link will be completely unused only when the link is unused on both sides. From a node i to j if the link is $e_{i,j}$, then in the matrix it has to be $e_{i,j} = e_{j,i} = 0$.

As the goal of this experiment is to show if the bandwidth is high or not, we have modified the OSPF according to that. We have first conducted the experiment considering 40Gbps bandwidth for each link. And found out that every demand flows are able to reach to its destination through the shortest path first. And there is no link which has been used completely to its capacity. As OSPF does not consider the capacity of the link, it only finds out the shortest path for each

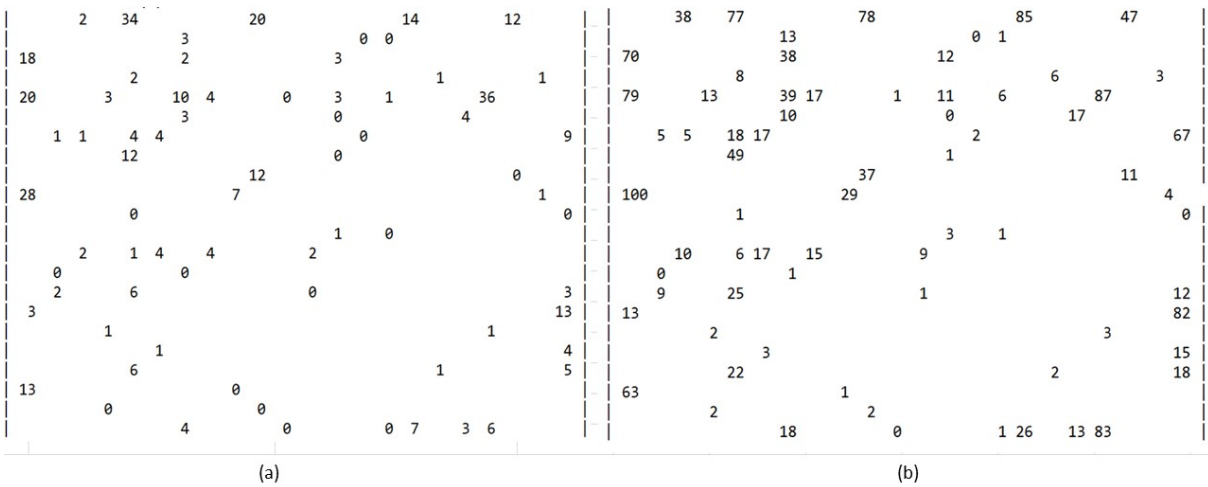


Figure 3.5: Link usage percentage (a)40Gbps (b)10Gbps

source-destination pair. Therefore, it is possible that OSPF might find a solution where the throughput of the link will be higher than its capacity which is unwanted. So, as the solution of OSPF was applicable with a topology of 40Gbps links, we wanted to check if the same solution can be applied for the other three levels of bandwidth or not. Therefore, in order to run the experiment, we have modified the code in a way that, was actually checking the OSPF that can be applicable or not for different levels. Because if any capacity constraints are introduced to OSPF, then it might provide different topology for different levels and that will harm the output of the experiment. Different capacity constraints for multicommodity flow will be discussed in the later part of the chapter.

Figure 3.5 shows the link usage percentage matrices for 40Gbps and 10Gbps. It is very interesting to see that, even with 10Gbps all over the network, a simple algorithm like OSPF can facilitate all the real flows. We have conducted the same experiment for thirty different randomly selected traffic matrix for different dates and different times of the day to validate the finding. And the outcome is the same for each case. For each case, 10Gbps is enough to route all the traffic from source to distance. The figure 3.5 also shows that for 10Gbps even though, it causes a couple of links to utilize its maximum capacity, however, there are still some links which do not reach even to its half capacity. It is very important to realize that as OSPF is used for both

cases, the total number of used links is same. That means just by reducing the bandwidth it is possible to save energy by 4.7 times which is the amount of power difference between 10Gbps and 40Gbps. This experiment clearly indicates that the core network is clearly over-provisioned based on real data. And there is clearly a scope to reduce the energy consumption of the current network topology.

3.1.3 Why not brute-force?

In the third set of experiment, we have examined the vastness of the problem by using brut-force. Brute-force is a mechanism that calculates all the possible solutions and provides the best result. In order to solve the energy efficiency problem of the GÉANT the goal is to find out all the possible forwarding/routing tables for a set of demand that will indicate which path will be used by each demand for reaching source to destination. To do that, the first is to know that for each set of source-destination pair how many paths are possible without having any loop. In order to do that, we have written a code to run an experiment where input will be the graph of GÉANT and the output will be the total number of possible paths from each node to all other nodes. In order to calculate the paths, we have used simple recursive depth-first search (DFS) to find out all the possible paths. Algorithm 1 describes the code for calculating the total paths for each node and Algorithm 2 shows the DFS algorithm.

```

total_paths
  input :  $\mathcal{G}$  an undirected graph for topology
  output: From each node to all other nodes number of possible paths

  Initialize all index usable;
  for  $i \leftarrow 1$  to number_of_nodes do
    for  $j \leftarrow 1$  to number_of_nodes do
       $target \leftarrow j$ ;
       $ans[i][j] \leftarrow \mathbf{dfs}(i)$ ;
  return ans;

```

Algorithm 1: Brute force algorithm for finding all the path for all source destination pair

Now, in order to understand the vastness of the problem let us consider a small demand set of five demands for the GÉANT topology discussed in figure 3.3. And we also assume that the cumulative throughput requirement of these demands are lower than individual link capacity. That means none of the solutions will be discarded due to capacity constraints. Table 3.1 shows the five demands with source and destination and how many possible paths are available for that pair that we have obtained by using algorithm 1 and 2. Now, as all the paths are a valid solution then the total number of valid forwarding tables will be the multiplication of all the possible

DFS

```
input : source
output: total paths for source to target

if source = target then
  return 1;
  Make the source unusable through visited array;
  adjnd  $\leftarrow$  number_of_adjacent_nodes_for_source;
  counter  $\leftarrow$  0;
  for i  $\leftarrow$  0 to adjnd - 1 do
    child  $\leftarrow$  graph[source][i];
    if !visited[child] then
      counter+  $\leftarrow$  dfs(i);
      Make source usable through visited array;
  return counter;
```

Algorithm 2: Depth First Search

paths. Therefore, only for this particular example, total number of possible forwarding table:

$$1281 \times 1105 \times 927 \times 1003 \times 1133 = 1.49e15$$

Just for five demands this is the total number of solution that all are applicable. With the increase of number of demands, the size of the solution set also grows exponentially. Moreover, when considering the capacity and energy consumption constraint, it will increase the complexity even by few folds. Therefore, solving this multi commodity flow, energy aware routing is known as NP-hard problem. The best solution find out in polynomial time is impossible. This final experiment shows even there are plenty of opportunities to save energy in a core topology like GÉANT but solving the problem is not always easy because of its enormous solution space. An effective algorithm is necessary to improve the scenario.

3.2 Shared path first algorithm

The analysis of the traffic shows there is a certain amount of savings is possible if we can turn off some unused links by maximizing the link utilization in some of the links. And as brute force is not an option, in this section we have introduced a novel algorithm named shared path first algorithm. Later on, after examining the preliminary result, we have extended it into another

Demands	Source Node	Destination Node	Total number of Paths
Demand-1	8	11	1281
Demand-2	5	19	1105
Demand-3	2	8	927
Demand-4	1	3	1003
Demand-5	9	1	1133

Table 3.1: Total number of paths for each source-destination pair

version namely shared path first sorted algorithm. This algorithm works to reduce the total number of used links. It is a greedy approach where the method only focuses on the paths that were already in use while respecting capacity constraints. And maneuver all the traffic while turning on as less number of links as possible. Now, we have discussed about the network model formalization and network constraints before explaining the algorithm in detail.

3.2.1 Network model formalization

We propose a model reflecting the physical topology of a network by an undirected graph $G = (V, E)$ that V is a set of nodes $i \in \{1, 2, \dots, |V|=n\}$ and E is a set of links (i, j) between two nodes. For a simple graph with nodes of V , the adjacency matrix is a square $V = |V| \times |V|$ such that the element $A(i, j) \in \{1, 0\}$ has the value of 1 when a link exists between two nodes i and j , and 0 when there is no edge between the nodes. The diagonal elements of the matrix are zero, as the edges from a vertex to itself are not allowed in simple graphs. The weight of matrix A is known as the capacity of each link and it is gathered in matrix C in which $C_{i,j}$ represents the current capacity of the link between node i and j . At the beginning, every link has same capacity, however, when a link is used by a flow, the current capacity updates to its residual capacity that means the capacity which is left after fulfilling the capacity requirement of the flow.

As explained earlier, there are several solutions possible in a over-provisioned graph like GÉANT. And even though the graph G represents the static part of the network, the forwarding plane is the dynamic part of the network and interacts with the changing traffic. The traffic to be transmitted via network is defined by the exchange matrix $X = |V| \times |V|$ in which $X_{i,j}$ is the flow that needs to be transported from node i to node j . Each node for a common destination might have different (sub) flows. Splitting a single flow across multiple links in the topology might reduce power by improving link utilization overall, but reordered packets at the destination (resulting from varying path delays) will negatively impact TCP performance. Therefore, we discard this fact for simplicity and only the aggregated traffic will be considered.

Let us consider K is the set of traffic demands, $K = \{k_0, k_1, \dots, k_n\}$ where n is the total number of demands. Each K_n is a list consists of a source-destination pair (s_{k_n} and d_{k_n}) and λ_{k_n} is the bandwidth requirement of that demand in Mb/s. The forwarding plane decides the output port of each node based on the destination and defines the overall path $P_{i,j}$ between node i and node j . A path is an ordered set of nodes such that $P_{i,j} \in V$ and $P_{i,j} = \{v_1, v_2, \dots, v_l\}$ with $v_1 = i$, $v_l = j$ such that v_i is adjacent to v_{i+1} . And the second notation of the path is introduced by $Q_{i,j} \in E$ with $Q_{i,j} = \{e_{v_1, v_2}, e_{v_2, v_3}, \dots, e_{v_{l-1}, v_l}\}$. Alternatively, a path can also be defined with respect to demand, P_{k_n} which defines the path for demand k_n . To show the forwarding decisions of the nodes, the matrix $F = |V|^2 \times |V|$ is introduced such that $F_{(i,j),v} = 0$, if the flow from node i to node j will not be forwarded by the node $v \in V$, i.e. $\forall v \neq j, v \notin P_{i,j}$, otherwise $F_{(i,j),v}$ gives the node $v + 1$ that is the node adjacent to the node v within the path $P_{i,j}$.

3.2.2 Network constraints

In the previous section, we have considered the cumulative bandwidth requirement of all the flow is smaller than each link's available bandwidth. Whereas, in reality, that is not the case. Among the several possible outcomes, only those can be used which respects the network constraints of the network. According to [Barnhart *et al.*, 2009] there are four network constraints for multi-commodity flow problems. First constraint is represented with equation 3.1 that ensures, the capacity of a link has to be greater or equal to the flow throughput. This is the first and foremost constraint for a flow to pass through a network.

$$\forall e_{a,b} \in Q_{i,j}, C_{a,b} \geq X_{i,j} \quad (3.1)$$

Since a single link might be participating in different paths, the capacity of the link can be shared for different flows. The second constraint ensures that a path can only be a good candidate if it fulfills equation 3.2. If the available capacity of the link is lower than the upcoming flow throughput then the alternative route has to be chosen.

$$\forall v \in P_{i,j}, \sum X_{a,b} \leq C_{v, F_{(i,j),v}} \quad (3.2)$$

Third constraint is used to choose the next hop for a given path. It ensures if the residual capacity of the entire path can bear the load. The third constraint can be represented by equation 3.3.

$$\exists v' \in \{V - P'_{(i,j)}\}, C_{(V, V')} - \sum_{(a,b) \in V^2, (a,b) \neq (i,j) | F_{(a,b),v} = v'} X_{a,b} \geq X_{i,j} \quad (3.3)$$

And lastly, fourth constraint ensures the flow conservation at source, destination and during the

Symbols	Descriptions
P_{D_i}	Path for Demand i
N	Number of demands
F	Forwarding table
D_i	Demand i
HC_{sh}	Number of new links for using $ShPF(sorted)$
HC_{spf}	Number of new links for using SPF
NU	List of already turned on nodes
PS_{D_i}	Path from source of demand i to any node of NU
PD_{D_i}	Path from destination of demand i to any node of NU
$PSPF_{D_i}$	Shortest Path for Demand i

Table 3.2: List of symbols and definitions

transmission. Equation 3.4 can be used to depict the constraint four.

$$\exists e_{i,j} \in Q_{sd}, \sum_{e_{i,j} \in E} e_{i,j} - \sum_{e_{i,j} \in E} e_{j,i} = \begin{cases} 1 & i=s; \\ -1 & i=d; \\ 0 & i \neq s,d. \end{cases} \quad (3.4)$$

3.2.3 Algorithm description

The shared path algorithm is a novel greedy algorithm which tries to reduce the total number of turned on links in order to save energy. In the beginning, we considered that all the links are off. And as soon as we received a set of demands, the algorithm routes these demands throughout the network while turning on nodes as less as possible and respecting the network constraints. In the shared path first algorithm, at first, we take the very first demand of the list and shortest path first (i.e: Dijkstra algorithm) is applied for sending the flow from source to destination. After that, from the next demand, the algorithm tries to find a path using the already turned on links. However, the total number of newly turned on links has to be lower than the number of links that will be turned on if this demand is routed using the shortest path first. In every state, capacity constraint is respected. For improving the result, we have also introduced an extension of the shared path first (ShPF) algorithm namely shared path first sorted (ShPFS) where at first the demand list is sorted and starts with the demand which has the largest number of hops in its shortest path. The whole algorithm is described in the algorithm 3. The table 3.2 contains the list of symbols for the algorithm.

```

shared_path_first_sorted
  input :  $\mathcal{G}$  an undirected graph for topology,  $D$  a demand list each demand
           containing source ( $s$ ), destination ( $d$ ) and throughput
  output: Forwarding Plane  $F$ 

  sort_spf_desc( $D$ );
   $P_{D_0} \leftarrow$  shortest_path_first( $s_{D_0}, d_{D_0}$ );
   $F \leftarrow$  add_path( $P_{D_0}, F$ );
  update_used_link_list( $NU, F$ );
  for  $i \leftarrow D_1$  to  $D_N$  do
     $PS_{D_i} \leftarrow$  shortest_path_first_of_all( $s_{D_i}, NU$ );
     $PD_{D_i} \leftarrow$  shortest_path_first_of_all( $d_{D_i}, NU$ );
     $P_{D_i} \leftarrow$  path_concatenation( $PS_{D_i}, PD_{D_i}, F$ );
     $HC_{sh} \leftarrow$  calculate_new_links( $P_{D_i}, F$ );
     $PSPF_{D_i} \leftarrow$  shortest_path_first( $s_{D_i}, d_{D_i}$ );
     $HC_{spf} \leftarrow$  calculate_new_links( $PSPF_{D_i}, F$ );
    if  $HC_{sh} < HC_{spf}$  then
       $F \leftarrow$  add_path( $P_{D_i}, F$ );
      update_used_link_list( $NU, F$ );
    else
       $F \leftarrow$  add_path( $PSPF_{D_i}, F$ );
      update_used_link_list( $NU, F$ );
  return  $F$ ;

```

Algorithm 3: Shared Path First Sorted Algorithm

Algorithm 3 has described the shared path first sorted. The only difference between these two is the *sort_spf_desc()* function which applies SPF to each demand and sorts the demand list based on the number of hop count in a descending order in order to find the demand which has the largest number of hops in their shortest path. *add_path()* function adds the path to the forwarding table and *update_used_node_list()* function updates the *NU* which is a list of already turned on nodes. This list is used in *shortest_path_first_of_all()* function. It calculates the shortest path from one node to all the nodes of a list (i.e. : *NU*) and returns the shortest path between the inputted node and one of the nodes of the list. After calculating sub-paths from source and destination the full path is merged using *path_concatenation()*. And later part of the algorithm checks if this path uses a lower number of new links or not compare to shortest path first using *calculate_new_links()*. If it uses a lower number of links compare to SPF then this path is added to the forwarding plane, if not then the path provided by SPF(Dijkstra) is added to the forwarding plane. Figure 3.6 gives a visual representation of the two versions of the algorithm alongside with shortest path algorithm. In this example, we have considered a sub-graph of the GÉANT topology with 12 nodes and 15 links. We consider a demand set of three elements, where $D = \{\{s_{D_1} = 2, d_{D_1} = 8, \lambda_{D_1} = 30Mbps\}, \{s_{D_2} = 12, d_{D_2} = 10, \lambda_{D_2} = 50Mbps\}, \{s_{D_3} = 8, d_{D_3} = 4, \lambda_{D_3} = 10Mbps\}\}$ In this scenario we have considered all the link has 100Mbps bandwidth. In figure 3.6(a) gives the solution using shortest path first where 7 links are used to fulfill these three demands. (b) gives the solution using shared path first and it uses 6 links for these 3 demands. And in the rightmost figure provides the solution with shared path first sorted which uses only 5 links to fulfill the demands. Even for three demands by sharing the link's bandwidth it can reduce the number of links. So in an over-provisioned topology like GÉANT the savings can be substantial.

3.3 Result analysis

This section describes the results obtained using the algorithm. In order to evaluate the algorithm in the GÉANT network, we considered 15 minutes of flow transmission for different times in night time and day time, which are expected to have different traffic loads. We compared shared path first with the SPF to evaluate the final gain of the proposed model. We have considered shared path first sorted version for the comparison as it performs better. The traffic data-set is provided by Sndlib. We have randomly selected a day-time time slot (on weekdays) in order to have more variance into the traffic. The bandwidth is considered here for each link is 40Gbps. At first, we have conducted the experiment for the shortest path first. For this particular demand set, in order to find the shortest path between each pair of nodes for the required flows all the edges in the network were utilized. Every demand was using the shortest possible path but at the same time, all the links were in use. Now, when we have run the experiment with shared path first sorted the result is quite different. As they were maximizing the link utilization by sharing

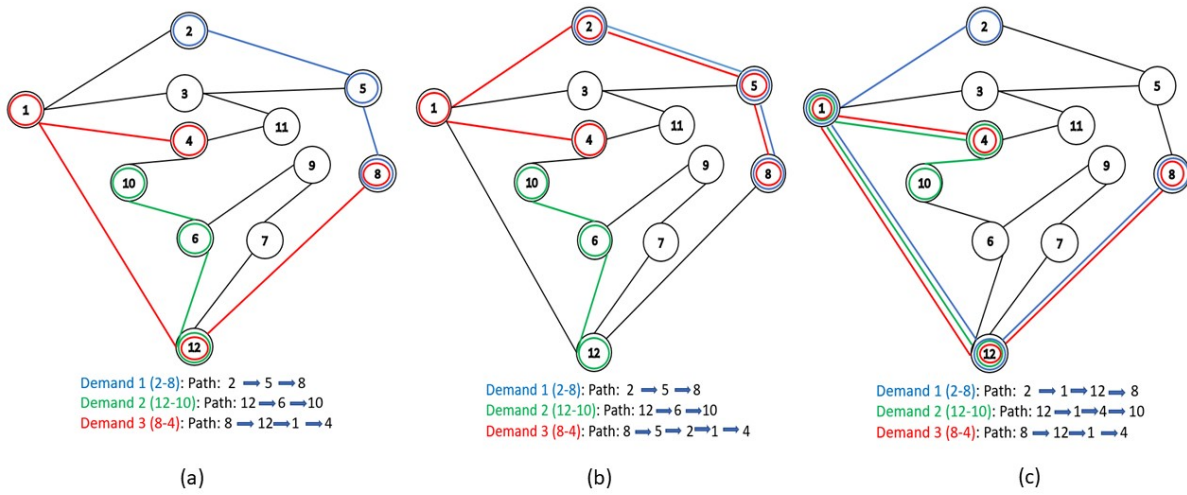


Figure 3.6: Routing Topology (a) Shortest path first, (b) Shared path first, (c) Shared path first sorted

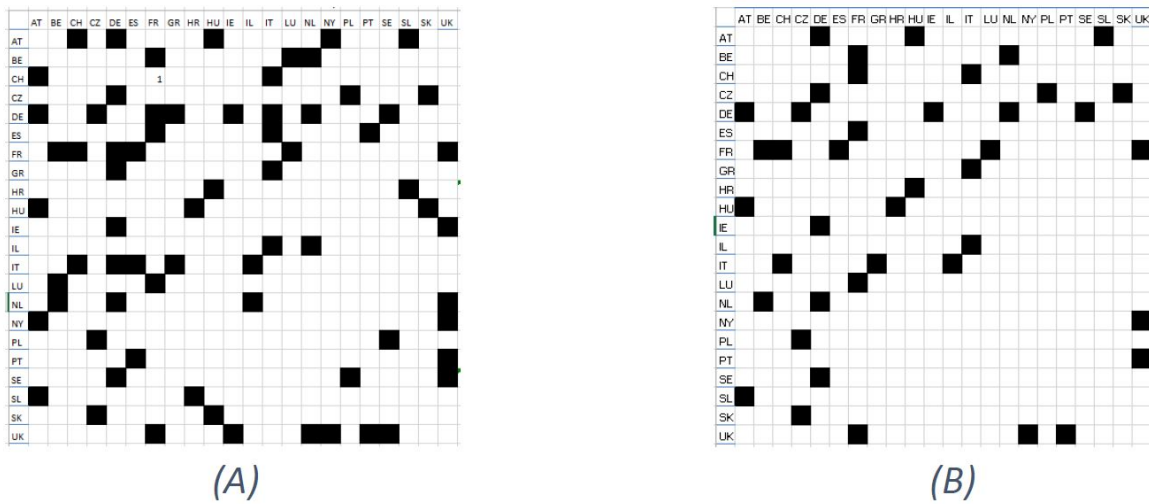


Figure 3.7: (A) Adjacency Matrix with Shortest Path First
(B) Adjacency Matrix with Shared Path First Sorted

the links by its full capacity, there are some cases where one link is fully utilized and the algorithm finds a new path with enough capacity. In the end, by applying Shared path first sorted we could save up to 41% of the links compared to the shortest path approach which is substantial.

We have provided a visual representation of the result using the adjacency matrix in figure 3.7. It demonstrates the adjacency matrix of the two mentioned approaches. The black boxes represent the active link between two nodes. The figure gives a clear contrast between the two approaches. We could fulfill the constraints and successfully transfer the flows by using only 21 out of 36 links. Since all the nodes were sending traffic it was not possible to turn off any node. From each node, there is a path to other nodes that are using the shared path only. This result for real traffic values, if 41% of links can be turned off compared to the shortest first path approach shows significant opportunity in saving energy in core networks like GÉANT.

3.4 Conclusion

In this chapter, we have analyzed the power consumption of a core network, GÉANT. The main reason behind this analysis is to point out the factors of the network that can be manipulated to save energy. We have also analyzed the available demand set and measured the compatibility of the traffic and the topology. After that, we have proposed a simple algorithm to see the energy saving potential of the network. Our Shared path first algorithm has shown promising results in saving energy by reducing the number of unused links. This result provides us the necessary incentive to forward our work. It is realized that, in order to save more energy, we need to consider both nodes and links to be turned off when they are not in use. Moreover, we have decided to include link rate adaptation for maximizing the savings. However, our shared path first algorithm does not consider these green policies. We need a more robust algorithm considering all the concerned green policies. Therefore, in the next chapter, we have formalized this energy aware routing problem considering all the green policies and also described a heuristic algorithm which is most fitted for solving this NP-hard problem.

Energy aware routing: An SDN perspective

Contents

4.1	Accepted green policies	52
4.2	Defining planes: control, data and management plane	54
4.3	Formalization of the energy model	57
4.4	Heuristic algorithm: genetic algorithm for solving minimization problem	59
4.4.1	Algorithm description	64
4.4.2	Design of experiment	67
4.4.3	Tuning of the algorithm	69
4.4.4	Simulation setup	70
4.5	Result analysis	71
4.5.1	Tuning outcome	71
4.5.2	Fitness function analysis	77
4.6	Conclusion	78

In this chapter, we have introduced a novel approach for energy aware routing using genetic algorithm designed for SDN. Before jumping into the solution, at first, we have discussed about the green policies that are accepted for the solution. After that, different planes are described and a general model of the solution is explained from SDN perspective in order to have a better grasp of the solution. Then, the problem is formalized and an energy model is designed. And then later on, the heuristic algorithm is explained. Genetic algorithm is a robust multipurpose algorithm, however designing of the algorithm (i.e: parameters, stopping criterion, design of the chromosome) based on problem formalization is critical. It has several parameters that are needed

to be tuned according to the problem. For that purpose, we have used design of experiment and run extensive experiments to fine tune the parameter settings. The whole design of experiment process is also described in detail in the chapter. Lastly the chapter concludes with the result analysis of our energy aware routing solution for GÉANT network for two different scenarios.

4.1 Accepted green policies

As described in chapter-2 there are quite a number of green policies which have been accepted by research community. During the preliminary experiments (Described in chapter 3), it have been focused on one of the most used green policy which is turning off nodes and links when they are not in use. For achieving green networking, two green policies are considered in this work. The first approach, also known as sleeping mode/turning off of nodes and links, which leads the incoming traffic flow to reach its destination over the fewest possible amount of nodes and links in order to put the remaining unused network elements to lowest power state. Here, sleeping and turning off are used interchangeably as in reality, during sleeping period the power consumption is negligible and it can be considered as turned off. And the second approach, is also known as adaptive link rate (ALR), which adjusts the speed and capacity of individual device's interfaces (i.e.: links), to meet actual traffic loads and requirements. The goal of ALR is to consume energy proportionate to traffic flows rather than consume a fixed amount of energy regardless of the amount of traffic flows.

In order to understand the motivations behind the two aforementioned energy management approaches, the power characterization of network equipment should be considered. Normally, there are two type of components that consumes the power of networking device. The first one is static component (due to power consumed by chassis, fans, line-cards, etc.) or the node part and the second one is dynamic component which is related to the rate of traffic flowing through the ports of their interfaces. In a ideal case, static part should consumes zero when not in use and the whole network consumes proportionately to the number of active dynamic ports. And the energy growth should be linear as shown in figure 4.1. But in reality, it is not the case. The energy model differs quite differently from expectation. Whenever a device is active it consumes a fixed amount of energy (E_f), regardless of it being active or idle or the load conditions. And this behavior is similar with port interfaces too. An idle 100Gbps link port consumes way more than an idle 10Gbps link port. The baseline power is increased by number of active ports and utilization of each port [Hossain *et al.*, 2015] which is shown as 'Reality' line of the figure 4.1. Here N is the total number of active links. Therefore, turning off/ sleeping of unused nodes and links is by far the most effective way of saving energy in terms of amount of energy savings. Each node and links consumes noteworthy amount of energy even if when they just on without involving in transmitting traffic (consumption of processor, ports, memory , cooling

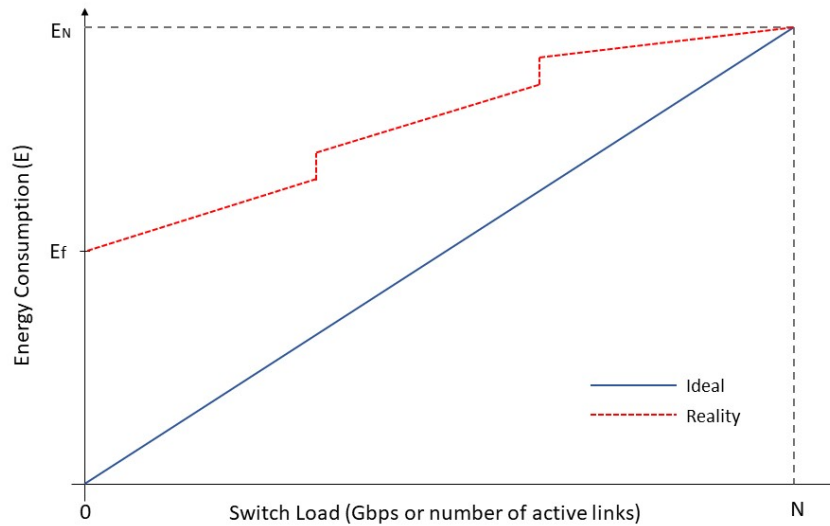


Figure 4.1: Energy model

mechanism etc). Turning off/sleeping mode of a node means saving all these energy consumption.

Second approach that we have considered is adaptive link rate (ALR). The link rate adaptation technique is based on the relation between the power consumed by a device and the data rate at which its interfaces performing. Instead of using the maximum possible link capacity, the idea of ALR is to adjust the link capacity according to the traffic flow requirements. So that, the energy consumption of the network devices are reduced and some what in accordance with the link utilization. For example, topology like GÉANT can support upto 100Gbps links which consumes around 360Watt. As most of the network is designed for handling worst traffic burst therefore, in most cases the maximum throughput ability of a device is far higher than its average usage requirement. GÉANT is no difference. Therefore the second green policy that is considered in this work is using of Adaptive link rate (ALR). ALR is a technique that network links dynamically change their data rates in response to traffic levels. For high levels of traffic, a high link data rate is necessary and must be used. For low levels of traffic, a low data rate can be sufficient and should be used. In a network like GÉANT where the lowest bandwidth is 1Gbps consuming 7Watt and highest bandwidth is 100Gbps consuming 360Watt it can create an huge impact on the overall energy savings [Addis *et al.*, 2014], [Van Heddeghem *et al.*, 2012]. However, there is always a concern that, as ALR always works in the lowest possible bandwidth;

it might takes more time to send a flow to a destination resulting less time for shutting down nodes and links. But in reality it is impossible to predict the size of the flow and in times of constant but small amount of traffic flows large bandwidth will not help much. However, from the perspective of SDN, all the traffic flows can be addressed as a service (for example: video streaming service). And if it is known the size of the required service then ALR can be adjusted accordingly focusing on maximizing the energy savings.

There are some works which prefers sleeping off a node than using adaptive link rate. for example, according to the research of [Nedevschi *et al.*, 2008] the sleeping mode approach is able to obtain higher energy savings than solutions based on adapting the link data rates. And which is completely true, however the result is completely scenario based. If the traffic flows of network is less frequent and there is enough opportunity to turn off a node then turning off a node clearly saves more energy compare to ALR as we have discussed earlier, if a link interface is on regardless its idle or involved in data passing it will consume a certain amount of energy. A core network, as we have seen from the analysis of chapter 3, is always busy to certain extent. In a recent work, the authors of [Haudebourg and Orgerie, 2017] concluded that, in a core network, a hybrid solution considering both sleeping and ALR will be able to save highest amount of energy. Therefore in this thesis, we have considered both sleeping/turning off node/switch and ALR. Moreover, in recent days, these mechanisms are more and more implemented which shows the effectiveness of these mechanisms. Our model uses a global view to asses the overall energy saving opportunities considering both approaches and provide a solution which consumes less energy.

Therefore, in the model considered in this thesis, network element like nodes can have one of two possible states: the active state at which it consumes full energy and the shut-down state with zero consumption. And for the link or port interfaces, we have five possible states: one is for sleeping/turned-off state and the rest of four is based on different possible bandwidth. The link port consumes zero energy when it is turned off and in active state, it consumes based on its assigned bandwidth. In this regard, it should be noted that when a node or link is in turned-off it is essentially removed from the network topology, which means that it is not able to receive, process and forward traffic. Therefore, a node or link can only be turned-off if it is not the source or destination of any incoming demand.

4.2 Defining planes: control, data and management plane

This section describes the general network automation technique in the basis of software defined network. As discussed earlier, SDN mainly decouples data and control plane to add the programmable capability to the network device. Here the definition of the planes are given based on the described work. First plane is data plane and also known as forwarding plane. It contains

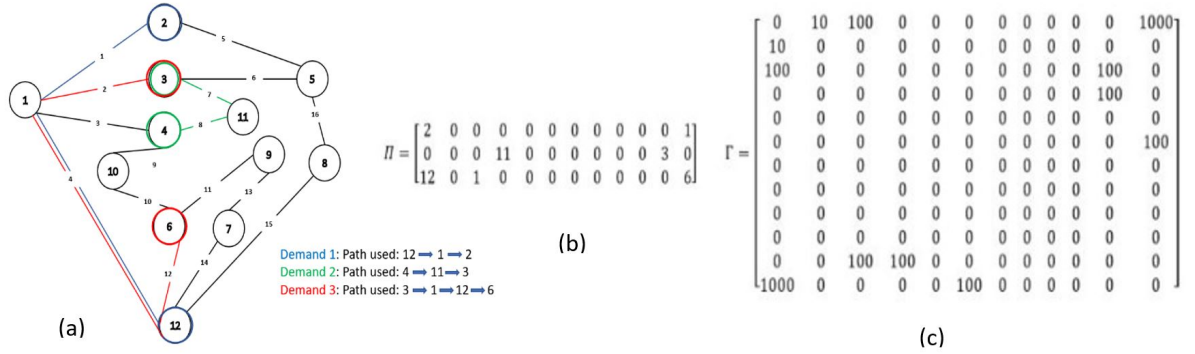


Figure 4.2: (a) Topology (Portion of GÉANT Network), (b) A possible Data plane, (c) control plane

the information of the next hop along the path to reach the destination for a demand from a source. The second plane is control plane. It is more involved in controlling the network including system configuration and link capacity allotment. It contains the routing table. Here, a third plane is also taken into account, which is the management plane. It gathers the functions used to control and monitor devices. It covers the network automation/management protocols like the historical SNMP, the famous SDN OpenFlow or even more recent like NetConf (there are usually associated to the south interfaces). Issues to be taken into account concerning real-time requirements [Silva *et al.*, 2017], detection and reconfiguration latencies, additional energy costs, controllers placement and actions ordering. Such protocols are both in charge to refresh the list of the demands K , to turn off/on (which remains a technical challenge) or put in hibernate/sleep state nodes/interfaces and to apply any new routing rule and capacity adjustment command. From a control engineering point of view, it should also address the stability of the command to switch from a plane to another one.

The idea of data and control planes will be more clear if can be shown with an example. In Figure 4.2(a) a smaller version of GÉANT network has been used. In this smaller version of GÉANT there are 12 nodes and 16 links. Now considering the three demands described in Table 4.1 the data and control plane is described.

Let us also assume, the topology has three different level of bandwidth, 10Mbps, 100Mbps and 1000Mbps respectively. For three incoming demands, three chosen paths are $\{ 12 - 1 - 2 \}$, $\{ 4 - 11 - 3 \}$ and $\{ 3 - 1 - 12 - 6 \}$ respectively. Then the possible data plane and the control plane will be look like 4.2(b) and 4.2(c). One thing to be remember that, this is one of the many possible solutions. Now if we see the control plane, for link $\{ 1,12 \}$ the used capacity is 1000Mbps even though none of the demand requires a link with bandwidth 1000Mbps. As both

Demand	Source	Destination	Throughput(Mbps)
1	12	2	8
2	4	3	60
3	3	6	95

Table 4.1: Demand set for Data plane and Control plane example

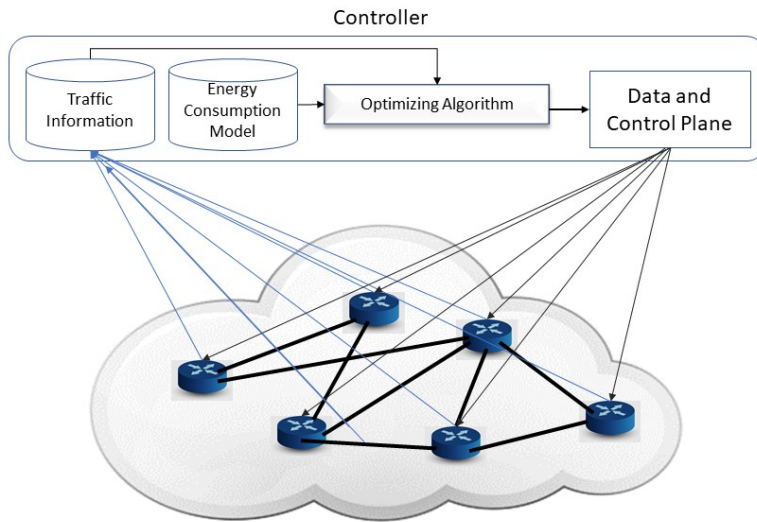


Figure 4.3: A generic architecture of the solution

demand-1 and demand-3 is using this link and the capacity has to be greater than cumulative throughput. Basically, as explained earlier, data plane contains the forwarding information. For example, for each demand there is a dedicated row for data plane showing the path by denoting the next hop of each node. On the other hand, control plane contains the information of the state of each link and how it should be managed. It shows the operating bandwidth for each link and the zero suffices that the link is turned off. Every time there is a change in the control plane and data plane due to an incoming demand or a topology change, there will be a change in the management plane which contains all the practical information of the change in the logical topology.

Figure 4.3 shows the basic architecture of our solution. As we can see, traffic information

(i.e.: demand set) are acquired from the network. In order to fulfill the demand set, controller utilizes an energy model and routing optimizing algorithm to generate data and control plane that will be implemented to the network through southbound interface. The energy model and the optimizing algorithm will be discussed in the following section.

4.3 Formalization of the energy model

Let us consider a network architecture where physical topology is defined by a directed graph $G = (V, E)$. Where V is a set of vertices $i \in \{1, 2, \dots, |V| = n\}$ and E is a set of edges (i, j) between two nodes. For a bidirectional graph with vertices of V , the adjacency matrix is a square $V = |V| \times |V|$ such that the element $A_{i,j} \in \{1, 0\}$ has the value of 1 when an edge exists between two vertices i and j , and 0 when there is no edge between the vertices. We are considering that the network infrastructure has redundancy that means for a single traffic request from source to destination or in terms of graph from one vertex to another vertex there are multiple paths available. The capacity of each edge is gathered in matrix C in which $C_{i,j}$ represents the capacity of the link between node i and j . This capacity might be seen as weight of the graph. And capacity is the only parameter of the graph G which is controllable. Now come to the dynamic part of the network which is the traffic demands. Let us consider K is the set of traffic demands, $K = \{1, 2, \dots, k\}$ where each value represents a source-destination pair. Moreover, s_k is the source and d_k is the destination of pair K and λ_k is the traffic demand in Mb/s.

In the framework of Software-Defined Networking (SDN) and more generally speaking network automation, we can consider that the topology is redundant, which means that several paths are available for each set of source and destination. However, only those paths will be considered which are satisfying both throughput demand and the capacity constraint of the links. The data plane matrix is denoted by Π , which includes all the forwarding decision for all flows over all nodes. Each row of the matrix is dedicated for one demand. $\Pi_{k,v}$ returns the node $v + 1$, which is an adjacent node of v into path p_k ($\Pi_{k,v} = v + 1$) or null set if the path does not include node v . The path ends when it returns d_k . A path $p_k = \{s_k, \Pi_{k,s_k}, \Pi_{k,\Pi_{k,s_k}}, \dots, d_k\}$ then can be retrieved. As mentioned earlier, different paths are available for every pair of source and destination therefore for each demand set several data planes are possible. The data plane should respect the following constraint for a demand i :

$$\forall v \in \{p_i - d_i\}, \quad w = \Pi_{i,v} \text{ such as } \sum_{k \in K \mid \Pi_{k,v}=w} \lambda_k \leq C_{v,w} \quad (4.1)$$

It ensures that only the links that are in the topology, are considered and no link will be assigned more throughput than its capacity. Furthermore, control plane, denoted by matrix Γ , is used to define the control-ability of the link capacity (i.e. ALR mechanism). The capacity

can be switched to different discrete values from the maximum capacity defined by C to 0. Zero is considered when both interfaces of a link are turned off. Initially, $\Gamma = C$ and later from the second flow C is replaced by Γ in equation 4.1. The value $\Gamma_{i,j}$ represents the final link capacity required to fulfill every demand requirement.

After defining the data and control plane the next step is to define the objective function. In order to minimize the overall energy consumption of the network architecture the cost of data and control plane in terms of energy consumed is formalized. Here, a simple model of energy consumption of each active device is presented. The model will be used to predict the overall energy consumption and to optimize the forwarding table. Previously, we have modeled the energy consumption of a typical network device (Cisco Ethernet Switch) by design of experiments [Hossain *et al.*, 2015]. Similar models can be found in the literature [Fithritama *et al.*, 2015], and [Gupta *et al.*, 2004] for similar devices. To simplify, it is assumed here that energy consumption of a node v , follows the linear model:

$$\varepsilon_v(t) = \int_0^t \left(\alpha_v + \sum_{w \in v} \delta_{v,w} \beta_{v,w} \right) t . dt \quad (4.2)$$

Where ε_v in Wh (Watt-hour), is the static power consumption (when no interfaces are activated), $\delta_{v,w} = 1$ if $C_{v,w} > 0$ and 0 else (to know if the interface to the neighbor w is used) and $\beta_{v,w}$ is the power consumption of the interface port itself. The model consists of two parts: a fixed/static one that must be considered each time the node is turning on and a dynamic one that will depend to the control plane. Based on that model, two green strategies have been considered that has been discussed in section 4.1. The first one is turning off any link or node that is not used by the routing configuration. And the second one is to use of adaptive link rate (ALR) to the links which are on. The goal is to turn off all the node and link that are not used and at the same use minimum possible bandwidth for the links which are on fulfilling QoS constraint, in this case which is capacity constraint. In terms of energy consumption, the cost of a data plane corresponds hence to the sum of the energy consumed by each node and each link involved in the routing strategy. Consider the two following binary variables:

$$\delta_{\Pi}^v = \begin{cases} 1, & \text{if } \sum_{k \in \mathcal{K}} \Pi_{k,v} > 0 \text{ or } v \in \Pi \\ 0, & \text{else} \end{cases} \quad (4.3)$$

$$\delta_{\Pi}^{v,w} = \begin{cases} 1, & \text{if } \exists k \in \mathcal{K}, \Pi_{k,v} = w \text{ or } \Pi_{k,w} = v \\ 0, & \text{else} \end{cases} \quad (4.4)$$

A node v is used in the data plane if it either forwards ($\sum_{k \in \mathcal{K}} \Pi_{k,v} > 0$) and receives ($v \in \Pi$) traffic or it is used as a mediatory node for a selected path for at least one demand k .

The cost of a data plane Π can be hence computed as:

$$\varepsilon_{\Pi}(t) = \int_0^t \sum_{v \in V} \delta_{\Pi}^v \left(\alpha_v + \sum_{w \in V} \delta_{\Pi}^{v,w} \beta_{v,w} \right) t \cdot dt \quad (4.5)$$

As there are redundant paths available, in a given time may be not all the nodes and links will be used. If none of the above mentioned green policy is applied, this equation will stay true whatever the traffic load since the (physical) topology will run until its dismantlement. Therefore, for our first green policy, any link which is not used will be turned off and for nodes if all the interfaces of that node is unused then it will be turned off.

In previous works of ours [Maleki *et al.*, 2017], we have shown that network devices are not acting yet as energy-proportional computing since the traffic load has a very limited (negligible) impact on the overall consumption. Furthermore, turning off nodes and even links is not always possible like in the backbone core network (as the GÉANT network) due to the end-hosts, that demand high availability and full-time connectivity even if the network is not used. Therefore, in order to maximize the energy even further, our second green policy, ALR is introduced as mentioned before. In the case of ALR, the matrix Γ gathers the decisions (Figure 4(c)), such the power consumption $\beta_{v,w}$ will now vary according to the link capacity. Hence, the problem is, while satisfying the demands requirements (elementary paths, throughput, links capacity limitations), as follows:

$$\left(\hat{\Pi}, \hat{\Gamma} \right) = \arg \min_{\Pi, \Gamma} \int_0^t \sum_{v \in V} \delta_{\Pi}^v \left(\alpha_v + \sum_{w \in V} \delta_{\Pi}^{v,w} \beta_{v,w}(\Gamma_{v,w}) \right) t \cdot dt \quad (4.6)$$

We propose here to consider for illustrative purpose that $\beta_{v,w}(\Gamma_{v,w} = 1Gb/s) = 7$ Watt and $\beta_{v,w}(\Gamma_{v,w} = 100Gb/s) = 360$ Watt. The objective will be here to develop joint (routing and forwarding) strategies by providing a data plane as well as a control plane which satisfies the demand requirements with minimum possible energy consumption.

4.4 Heuristic algorithm: genetic algorithm for solving minimization problem

As discussed in chapter 3, the problem of finding the minimum number of links and nodes are turned on while fulfilling the traffic constraint is a problem of multi-commodity flow class which falls under the class of NP-hard problem. The Shared Path Algorithm(ShPF) provides enough intel on the potential for saving energy by turning off number of nodes and links. However, a feasible implementation technique is required in order to solve this NP-hard problem. Therefore, in order to solve our optimization problem that we have formulated in the previous section we have

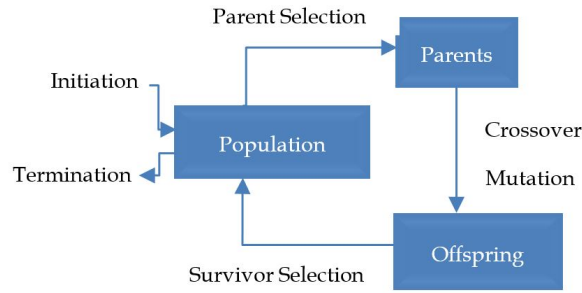


Figure 4.4: Genetic Algorithm Process Flow

used Genetic Algorithm (GA) as our heuristic algorithm. It is an evolutionary algorithm which is inspired by genetic process of biological organisms. It uses the analogies that is found in biology like, selection, reproduction, crossover and mutation and uses their behavior and mechanism to solve problem. In [Krommenacker *et al.*, 2002], [Thang Nguyen Bui and Byung Ro Moon, 1996] and [Chang Wook Ahn and Ramakrishna, 2002] all showed how this heuristic can be applied for solving NP-complete optimization problem related to networking.

[Chang Wook Ahn and Ramakrishna, 2002] uses GA to find the shortest path of routing in a network and [Krommenacker *et al.*, 2002] designed an industrial Ethernet infrastructure using GA. Genetic algorithm shown to be extremely robust and capable of dealing with highly scalable scenarios like we are dealing with. Enumerative approaches are destined to fail with a scenario where millions of possibilities are possible. GA have crossover and mutation which help the problem to not to get stuck in local minima. Rather it found a good solution within given time. Figure 4.4 shows the basic process flow of the genetic algorithm. Genetic algorithm starts with a pool of chromosome, where each chromosome is defined as a member of the solution space. The goal of GA is to find the optimal solution depending on the objective function. The optimal solution generated through the process of mutation and crossover of the chromosomes taken from the chromosome pool. Genetic algorithm thrives on the concept of survival of the fittest. Where after each iteration, weak chromosomes are discarded from chromosome pool and strong chromosomes have better possibility of selection for crossover and mutation process. The best part of the GA is that, it can be modified based on the problem. Several variations and adjustments can be made on different processes to get better result. In the following part different parts of the genetic algorithm which is designed for our problem is described in brief.

Composition of the chromosome and chromosome pool/population: It starts working with a population of individuals known as chromosome. A chromosome is a string which represents a single candidate solution of the search space. In our case, data plane is converted into chromosome. As described before, a data plane contains a solution for a demand set, mean-

Chromosome																																																
Demand 1																Demand 2																Demand 3																
L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0

Figure 4.5: Structure of the chromosome

ing each demand from the set of demand should have a path from source to destination following the capacity constraint which is defined in data plane. Therefore, to build a set of chromosomes we have to create a set of data planes. In order to create a set of solutions for a particular set of demands we have used another heuristic which is randomized depth first search (RDFS).

Random Depth First Search (RDFS): Random depth first search (RDFS) is a modified version of depth first search (DFS). We have designed this heuristic to improve the efficiency of genetic algorithm. For energy aware routing problem the solution space is huge specially for topology like GÉANT. Simple DFS works incrementally and provides a solution set which is not very diverse. The main concept of RDFS is while traversing tree for solution, for each step, RDFS randomly selects the next node for going into next level. For example figure 4.6 shows a small example differing RDFS and DFS. Here for simplicity, the tree has only three levels. Now, if we consider the third level as our solution space and for building our chromosome pool we will pick three solutions. Then for DFS, it will pick the first three solutions from the left and as a result all of them will belong to the child of same parent and diversity among them will be less. On the other hand, for RDFS as every level next nodes are chosen randomly the solution space will be largely diverse. In case of genetic algorithm, a diverse population pool always yields better solution during crossover and mutation operation. If the population pool is mostly siblings then these operation will not produce better off-spring.

When RDFS is applied it considers the current state of the topology capacity wise. So, all the solutions that are found, follow the capacity constraint. This process continues until all the demands are done and we find a single solution topology fulfilling all the demands requirements. Once we have a data plane then we can convert the data plane into chromosome. The chromosome is built in following way. If we have a set of demand $k = \{1, 2, 3, \dots, k\}$ and if a topology has E number of links $\{1, 2, 3, \dots, E\}$ then in a chromosome for each demand we have L number of 0's and 1's. Where 0 represents this link is not used for fulfilling this demand and 1 represents this link is turned on for fulfilling this demand. From the topology matrix and the source and destination node then it is easily possible to retrieve the path of each demand from this chromosome. Figure 4.5 shows the structure of the chromosome. This chromosome is actual representation of the example with three demands that we discussed in section 4.1.

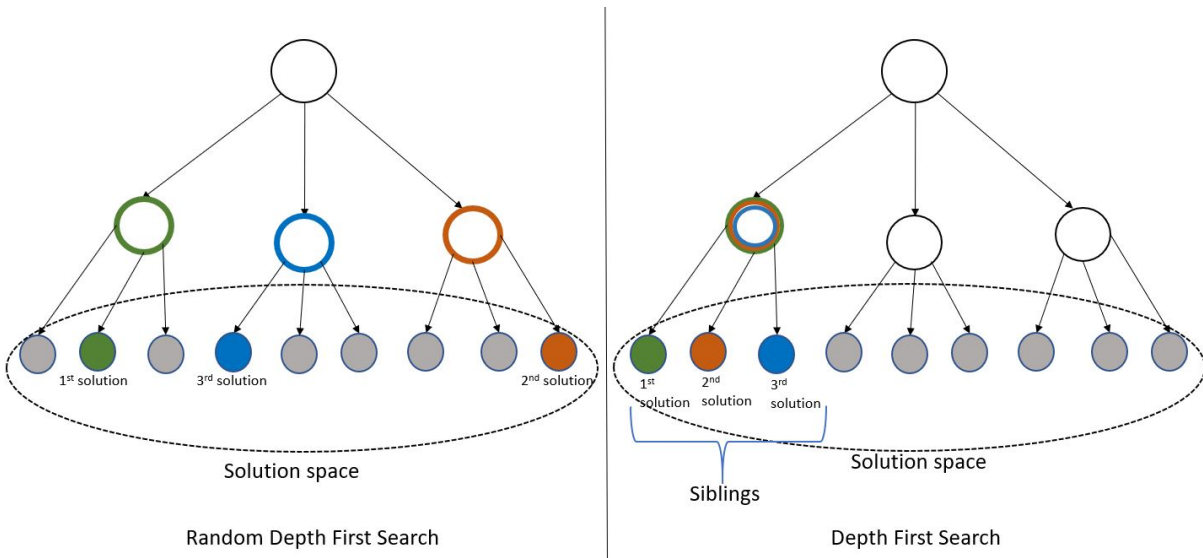


Figure 4.6: Solution retrieve process RDFS vs DFS

After creating a chromosome, the next step is creating of population also known as chromosome pool. The size of the population is one of the parameter that is needed to be adjusted. We have used Design of Experiments (DoE) to tune this parameter. However, tuning of parameter will be discussed in the later sections. Normally a population pool is created randomly. However, in order to boost up the solution and direct it to a better direction, our whole population is created from the chromosomes using RDFS except one. Suppose, if the population size is P then, $P - 1$ chromosomes are created randomly using RDFS and the other one is created using shortest path first (SPF). However, we use a modified version of SPF where even though we look for the shortest path but the total capacity of a link is always considered. So all the shortest paths are chosen in a way that for none of the link the used capacity is higher than the initial capacity. By including SPF, our algorithm will always give a solution which is as good as SPF. And as the generations goes by the result improves.

Selection Operation: Selection operation determines which chromosome is eligible for reproducing. It is done based on the fitness function. In our case which is the energy consumption value that we calculate for each chromosome based on our equation-4.6. A chromosome with less energy consumption will have the higher possibility to be selected and vice versa. Only 50% of the chromosome will be considered for the next step which is crossover and mutation. And among this selected chromosomes, some were selected for crossover and some are selected for mutation operation based on the given input percentage.

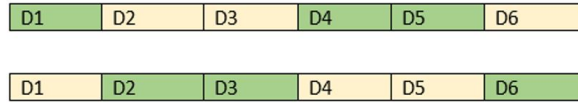


Figure 4.7: Example of crossover operation

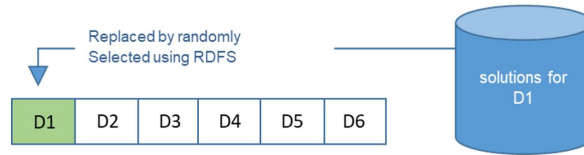


Figure 4.8: Example of mutation operation

Crossover Operation: Crossover operation usually swaps bit strings in between parents in order to breed new chromosome. As mentioned earlier, each chromosome is representing all the chosen path for all the demands. In our case, we have decided to randomly select half of the bit strings from one chromosome and rest half from the another chromosome instead of using one cut or two cut where chromosomes are divided into two equal parts or three equal parts and change the parts in between them respectively. By doing single crossover operations the process is generating two new chromosomes. The figure 4.7 is depicting the crossover operation. It improves the overall outcome compare to one cut and two cut crossover.

In the above case from parent-1 D1, D4 and D5 are selected and from the parent-2 D2, D3 and D6 are selected for the child-1 and vice versa for the child-2. The whole selection is done randomly. One important thing is crossover percentage. This values indicates how many percentage of the chromosome which are previously shortlisted by the selection process will go through this crossover operation.

Mutation Operation: Mutation is an operation when a part of the chromosome changes completely. Mutation has great importance in GA. Crossover is used for local maxima whereas mutation is used for global maxima. In our case mutation is also done demand wise. Instead of a single bit values for a single demand, the whole strings of bit of a demand is replaced. Each demand can be satisfied with hundreds of possible paths. Therefore, first of all a demand is chosen randomly. And for that demand a solution is generated using RDFS and replaced the string of the chosen demand with the generated solution. Figure 4.8 depicts the mutation operation more clearly. There are two kinds of mutation percentage. Mutation-1 indicates how many of the chromosomes which are previously shortlisted by selection process will go through the mutation process. However, from a chromosome how many demands will be selected for

mutation is denoted by Mutation-2. For example, mutation-1 equals to 10% means 10 of each 100 chromosomes are selected for mutation operation and Mutation-2, 1% means 1 of each 100 demands from a chromosome which is selected by mutation-1 will be chosen randomly for mutation operation.

Validity Check: Each new breed from crossover and mutation process has to go through the validity check. In our case, there is always a possibility that the newly breed chromosome does not fulfill the capacity constraints. Therefore, each new chromosome has to go through capacity constraint check. Any chromosome unable to fulfill the capacity constraint is discarded. There is second kind of check which is duplicity check. If after mutation or crossover operation the newly breed chromosome is already available in the population then this newly created chromosome is also discarded.

Stopping Criterion: Stopping criterion indicates when the process will end. There are several ways that a genetic algorithm can be stopped based on the problem. A threshold value, number of generations, run time, or when convergence reached, all of them separately or combinedly can be used as stopping criterion. In our case we are using running time as a stopping criterion. As for our experiment we have traffic data for every 15 minutes therefore run time is the most suitable stopping criterion for us.

4.4.1 Algorithm description

The algorithm works in a traditional manner of how a genetic algorithm usually works. However, there are certain customized tuning in to the algorithm that has been done to fulfill the need of the problem. At first demand list is received by the controller from the south interface. This list could have one demand or several demands. These demands usually have three information: source, destination and the throughput requirement. From this demand set at first fixed number of data planes and then control planes are generated as described in the previous section. After that each control plane is converted to chromosome in order to create the chromosome pool. And then the selection, crossover and mutation operation is done iteratively until the stopping criterion in this case which is runtime is met. And then the best chromosome with best fitness function value is selected as output. The last part is to convert the chromosome into data and control plane so that controller can introduce the optimized topology into the network. Now, different parts of the algorithm are described in detail.

Assume a chromosome to be represented by a vector \mathcal{X} of size $|E| \times |K|$. Where K is the traffic demand set, and each element of K is $k_{s,d}$ where s and d is the source and destination of the traffic demand respectively. Here, even though each chromosome indicate that if a link is used for a particular demand or not, however, the direction of the flow of the traffic is not defined. For

calculating the fitness of the chromosome this information is not necessary as in a bidirectional graph if a link is used in any one direction, it will be considered as in use. But in case of data plane and control plane this is a primal information. Therefore the algorithm needs to take care the information regarding direction, when converting from chromosome to data plane and then control plane. The algorithm to retrieve the data plane, Π and the control plane, Γ is given in algorithm 4. One may think that it simply corresponds to the following equation:

$$\Pi_{k,v} = \mathcal{X}_{k \times (v,w)} \times w$$

```

to_planes
  input :  $\mathcal{X}$  a chromosome
  output:  $\Pi$  the data plane and  $\Gamma$  the control plane
  initialization of  $\Pi$  such that all elements are null;
  for  $k_{s,d} \in \mathcal{K}$  do
    while  $s \neq d$  do
      foreach  $v \in V$  do
        if  $\mathcal{X}_{k \times (v,w)} = 1$  then
           $\Pi_{k,v} \leftarrow w$ ;
           $s \leftarrow w$ ;
          break;
   $\Gamma \leftarrow \text{to\_data\_plane}(\Pi)$ ;
  return ( $\Pi, \Gamma$ );
  
```

Algorithm 4: How to retrieve the data and control planes from a chromosome

The allocation of the capacities, i.e. the control plane definition, is given with the function described by the algorithm 5. It finds the minimal allocation satisfying the throughput demand gathered by the data plane. To note that the green policy, consists of shutdown v switches when $\delta_{\Pi}^v = 0$ and output ports w of switches v when $\delta_{\Pi}^{v,w} = 0$ and decreasing capacities of output ports w of switches v when $\Gamma_{v,w} < C_{v,w}$. Secondly, as control plane shows the used capacity of a link the allocation of the value that showed its capacity is symmetric. Because for a full duplex link, if one side is required 100Mbps then the other side required 10Mbps, then the bandwidth allocation of the link will be 100Mbps at both side as there is no technology available to use two different capacity in two different sides.

The flowchart given in figure: 4.9 and algorithm 6 shows the overall procedure for GA that we have discussed so far. As we can see in the algorithm, line 1-8 is used for initializing $n - 1$ chromosomes using RDFS algorithm. And in line 9-15 is the initialization of the $n - th$ chromosome using shortest path first. From line 20, the genetic algorithm starts with repetitive selection,

```

to data_plane
  input :  $\Pi$  the data plane and  $c_{v,w}$  the vector of the admissible capacities for an
           output port of a node  $v$  to a node  $w$ 
  output:  $\Gamma$  the control plane
  forall  $(v, w) \in E$  do
    if  $\delta_{\Pi}^{v,w} = 0$  then
       $\Gamma_{v,w} \leftarrow 0$ ;
    else
      forall  $c_{v,w}$  (in a decreasing order) do
        if  $c_{v,w} \geq \max\left(\sum_{k \in \mathcal{K} | \Pi_{k,v}=w} \lambda_k, \sum_{k \in \mathcal{K} | \Pi_{k,w}=v} \lambda_k\right)$  then      /* symmetric
          allocation */
           $\Gamma_{v,w} \leftarrow c_{v,w}$ ;
        else
          break;
      return  $\Gamma$ ;
  
```

Algorithm 5: Computation of the minimum control plane from a given data plane

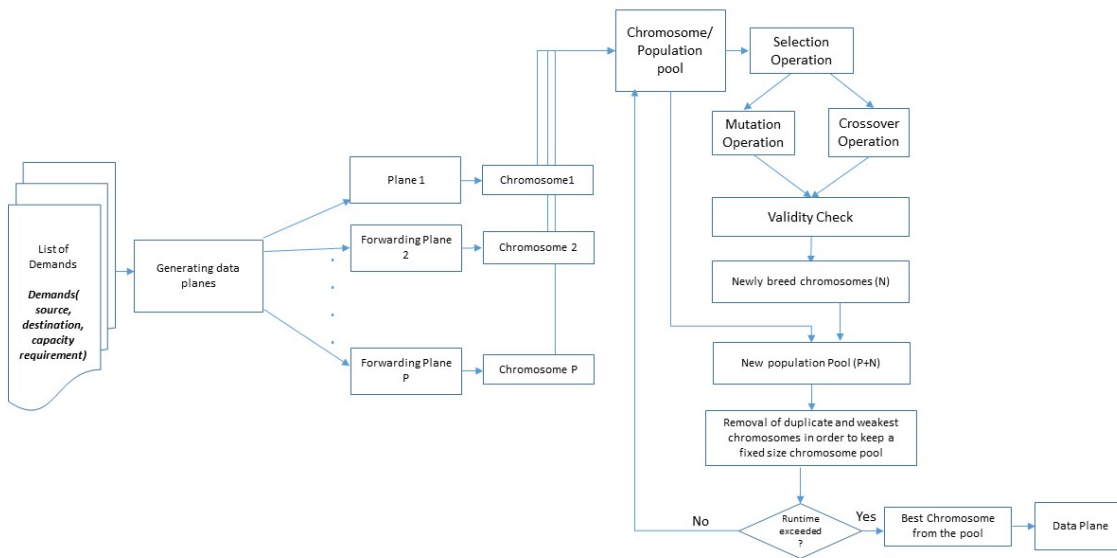


Figure 4.9: Flowchart of the whole procedure

crossover and mutation operation. After each operation set of chromosomes are evaluated and sorted and the process continues until the stopping criterion indicates.

4.4.2 Design of experiment

Genetic algorithm is a parameter sensitive algorithm like all the other evolutionary algorithm. As it is described, there are quite a few numbers of parameters (i.e. size of the population, crossover percentage, two types of mutation percentage, running time) are available when comes to genetic algorithm. And tuning of these parameters are one of the most difficult tasks in order to get the most effective point. In order to solve this problem design of experiment is introduced. Design-of-experiments (DoE) is a combined name of all sort of data collection techniques based on observation of a response from any sort of event. At the beginning of DoE's long history, the word "experiment" referred to classical (e.g., physical, chemical) experiments, whereas nowadays running a computer code of a simulation is also considered as a "computer experiment". The experiment is controlled by a set of input variables (or factors in DoE terms) and a set of output (or response) variables (or functions) can be observed. The set of all possible input combinations is known as experimental domain. For a single combination of input an execution of the experiment is done which is called an experimental run or trial. This experiment can be real experiment or computer simulated experiment. However there is a difference between these two. On one hand computer run experiment is fully deterministic. For a same value of all the inputs output will be always same. On the other hand real experiment is non deterministic. For same input value result could vary. That in the end helps to find some other hidden variable. Design of experiments is performed mainly to draw conclusions about the studied phenomenon. There could be several end result or final goal that can be set for design of experiment. For example to maximize, minimize, hidden pattern finding etc. can be set as final goal of design of experiment. To make the finding from the experiment more effective multiple runs of same experiments is done. However to get the appropriate result from these sets of runs, these runs should be executed according to an appropriate DoE method. Depending on what term information is meant and how optimal level is defined, various DoE approaches are available. The Branches of DoE that has been applied in this work is briefly described in the Methodology section. Design of experiment is generally used for complex experiments. However there is no harm to use in for simple experiments. Design of experiments helps to understand the inner structure of the experiment and the relation between the input variables and response variables, whose are cannot exactly be defined by just by seeing the raw results. The unavailability of the knowledge of the inner relationships between the variables points to black box models. Black box model are those where there is ample information about input variable and output variable but the functionality inside the system is unknown. Design of experiments is a suitable choice for black box modeling.

Figure 4.10 shows the general architecture of design of experiment. Where we have few fixed

Data: \mathcal{K} the initial ordered set of demands, C the initial adjacency matrix weighted by the maximum capacity of each port

Result: $\hat{\Pi}$ the optimal data plane and $\hat{\Gamma}$ the optimal control plane

```

1 initialization of the  $n - 1$  chromosomes through the RDFS algorithm;
2 for  $i \leftarrow 1$  to  $n - 1$  do
3      $\mathcal{K}' \leftarrow \text{random\_order}(\mathcal{K})$ ;
4      $\mathcal{A} \leftarrow C$ ;
5     for  $k \in \mathcal{K}'$  do
6          $\Pi_{i_k} \leftarrow \text{rdfs}(\mathcal{A}, \lambda_k)$ ;
7         forall  $(v, w) \mid \Pi_{i_k, v} = w$  do
8              $\mathcal{A}_{v, w} \leftarrow \mathcal{A}_{v, w} - \lambda_k$ ;
9 initialization of the  $n$ -th chromosome through the SPF algorithm;
10  $\mathcal{K}' \leftarrow \text{random\_order}(\mathcal{K})$ ;
11  $\mathcal{A} \leftarrow C$ ;
12 for  $k \in \mathcal{K}'$  do
13      $\Pi_{n_k} \leftarrow \text{spf\_a\_k}(\mathcal{A}, \lambda_k)$ ;
14     forall  $(v, w) \mid \Pi_{n_k, v} = w$  do
15          $\mathcal{A}_{v, w} \leftarrow \mathcal{A}_{v, w} - \lambda_k$ ;
16 for  $i \leftarrow 1$  to  $n$  do
17      $\mathcal{X}_i \leftarrow \text{to\_chromosome}(\Pi_i)$ ;
18 evolution of the population;
19 eval( $\mathcal{X}$ ); sort( $\mathcal{X}$ );
20 repeat
21      $\mathcal{X}_s \leftarrow \mathcal{X}_s \subset \mathcal{X}$  such that  $|\mathcal{X}_s| = p_s |\mathcal{X}|$ ; /* selection */
22     forall  $x_i \in \mathcal{X}_s$  do /* cross-over */
23         if random( $1$ ) <  $p_c$  then
24              $\mathcal{K}' \leftarrow \mathcal{K}' \in \mathcal{P}(\mathcal{K})$  such that  $|\mathcal{K}'| = |\mathcal{K}|/2$ ; /* any  $\mathcal{P}(\mathcal{K})$  */
25             forall  $k \in \mathcal{K}'$  do  $x'_k = x_i$ ;
26              $x_j \leftarrow x_j \in \mathcal{X}$  and  $x_j \neq x_i$ ; /* any  $x_j$  */
27             forall  $k \notin \mathcal{K}'$  do  $x'_k = x_j$ ;
28              $\mathcal{X} \leftarrow \mathcal{X} \cup x'$ ;
29     forall  $x_i \in \mathcal{X}_s$  do /* mutation */
30         flag  $\leftarrow$  false;
31         if random( $1$ ) <  $p_{m_1}$  then
32             forall  $k \in \mathcal{K}$  do
33                  $x' \leftarrow x$ ;
34                 if random( $1$ ) <  $p_{m_2}$  then
35                     flag  $\leftarrow$  true;
37                      $z \leftarrow$  a possible subpart (path) of chromosome for demand  $k$ ;
38                     forall  $(v, w) \in E$  do
39                          $x'_{k \times (v, w)} \leftarrow z_{v, w}$ ;
40             if flag then  $\mathcal{X} \leftarrow \mathcal{X} \cup x'$ ;
41     eval( $\mathcal{X}$ ); sort( $\mathcal{X}$ );  $\mathcal{X} \leftarrow n$  first elements of  $\mathcal{X}$ ;
42 until time occurred (time over);
43  $(\hat{\Pi}, \hat{\Gamma}) \leftarrow \text{to\_planes}(\mathcal{X}_1)$ ;
    
```

Algorithm 6: The full algorithm

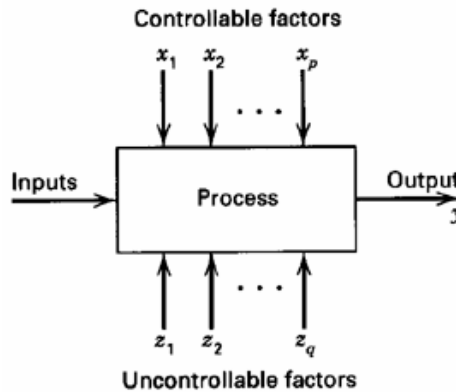


Figure 4.10: Design of Experiment general architecture

inputs, some controllable factors that will be controlled during experiment, few uncontrollable factors for example noise and an output or response variable. However this structure can vary depending on the type of experiment that is run. A well planned and executed experiment can offer a new perspective about the response variable. The behavior of the response variable can be depended on one or several factorials.

4.4.3 Tuning of the algorithm

As [Hallam *et al.*, 2010] mentioned, every parameter of GA is problem specific and there is no way to know the value beforehand without doing proper tuning and adjusting. According to [Eiben and Smit, 2012] there are two kinds adjusting in GA. One is parameter control and second one is parameter tuning. In our case as none of the parameters value will change during run time, therefore only parameter tuning is needed. All the quantitative parameters of the algorithm are tuned using Design of Experiment (DoE). In this case, DoE is used in order to find which set of parameters value can maximize our gain through GA. Full factorial method for DoE is used which means every single parameters performance will be judged individually as well as all the possible combinations between the parameters. An extensive set of experiments for tuning the parameters has been done to finalize the parameters. Five parameters are needed to be adjusted which are time, population size, crossover percentage, mutation-1 percent-age and lastly mutation-2 percentage. Few fixed values for each parameter for design of experiment is selected. Each values of each parameter are preliminary tested before used in DoE. Table 4.2 contains the different level of values for each parameter is given. As for the output of DoE, Energy gain compared to shortest path fast (SPF) is calculated. All the parameters are tuned in order to maximize this value. For each demand set ten sets of experiments are conducted

varying demand order and average value is taken into account for analyzing.

Factor	Values
Time	10s, 20s, 30s
Population Size	20, 40, 80
Crossover Percentage	20%, 40%, 80%
Mutation-1 Percentage	20%, 40%, 80%
Mutation-2 Percentage	4%, 8%, 12%

Table 4.2: DoE parameter summary

4.4.4 Simulation setup

Before testing the analysis we need to conduct design of experiment. GÉANT network structure and the traffic has been taken from SNDlib [SNDlib, 2008]. They have provided a network architecture with real data from 2005. However according to the current report of GÉANT provided in their website, it is now using 100-gbps optical fiber network in order to connect between nodes whereas in 2005 the maximum link capacity was 40Gbps And alongside with the increase capacity of network equipment, network overall traffic has also increased significantly. From the report of [Cisco, 2015] and [Cisco, 2012] it can be concluded that, in 12 years, traffic has increased by almost 28 times. In order to keep up with the time, the real data from 2005 is been multiplied with the increasing traffic coefficient. Network Detail summary can be seen in table 4.3. We have used two sets of demands: 400 demands and 25 demands per fifteen minutes respectively. Two demand sets are used in order to fully comprehend the capability of the algorithm.

Network	GÉANT
Nodes	22
Links	36
Link type	Full-Duplex
Demand Structure	Source, Destination, Throughput
Demand Type	Aggregated. (One demand request for one source destination couple)
Bandwidth Capacity	1Gbps, 10Gbps, 40Gbps, 100Gbps

Table 4.3: Network topology summary

Power measurement values for the experiments have been taken from [Addis *et al.*, 2014] and

[Van Heddeghem *et al.*, 2012]. These values are given in table 4.4

Type	Power in watts
Static Node	10000
OTN 1-Gbps port	7
OTN 10-Gbps port	34
OTN 40-Gbps port	160
OTN 100-Gbps port	360

Table 4.4: Power consumption values for node and links

4.5 Result analysis

This result section is divided into two parts. The first part contains the result of tuning the genetic algorithm. It gives the justification for the values of the parameter that has been chosen for the later part of the experiment. The second part explains the behavior in terms of fitness function and how it performs in different scenarios.

4.5.1 Tuning outcome

The running the full factorial experiment on our parameters with above mentioned settings, provides us significant information about the parameter tuning. Although for both 25 demands and 400 demands DoE shows similar kind of pattern, here only the results with 25 demands is shown. Figure 4.11 shows the main effects plot for energy gain compare to SPF. The first parameter population size has almost same impact with all three levels of value. However, population size 40 shows the better outcome for this scenario. As crossover percentage and mutation-1 percentage is directly dependent on the size of the population therefore an optimum value is needed to be found. Crossover percentage shows clear pattern that 40% has the highest impact on energy gain compare to other values. Mutation-1 also shows the higher the value of mutation-1 percentage the more is the gain. Mutation-2 percentage is the number of demands in a chromosome that goes through mutation process. Here, mutation-2 rate 4%, 8% and 12% is considered for 25 demands so that 1, 2 and 3 demands from each chromosome are changed by mutation operation respectively.

It also showed that lower value of Mutation-2 has better effect on the result than higher value for this scenario. And lastly, the time shows an increasing pattern for giving better outcome. However, we have conducted the experiment for longer period of time for multiple occasions and

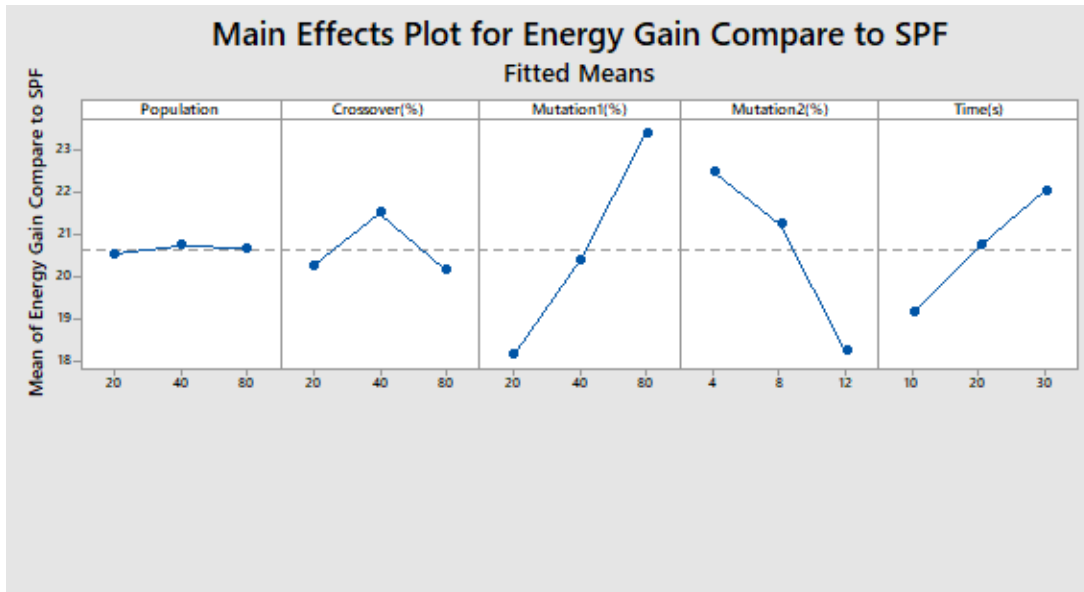


Figure 4.11: Main effects Plot for energy gain compare to SPF

in most cases, it reaches convergence after certain period and the outcome does not improve.

Figure 4.12 shows that, we have conducted the experiment for upto 120 seconds, and the overall result does not improve significantly after 30 seconds period. Therefore, we have chosen our upper limit based on these data. But, by only seeing the effect of main parameters does not imply actual result. It is the combination of different parameters that provides the desired outcome. We have analyzed the effects of most vital parameters of genetic algorithm (i.e.: crossover and mutation) separately. For a fixed population size which is 20 in this case, figure 4.13 shows the behavior of crossover operation alone when mutation operation is not considered. First of all, with compare with 20%, 40% and 80% the 20% consumes lowest amount of power, whereas according to design of experiment, the best crossover operation percentage is 80. The values clearly indicate that, when mutation operation is alongside with crossover, crossover acts differently. Secondly, the lowest possible value by crossover operation is still few hundred watts higher than the lowest possible solution. It proves the importance of mutation operation. And finally, time has little impact if not zero when crossover is considered alone. These values also indicates the random behavior of the overall system. As populations are chosen randomly. It is very possible that for a particular run the initial population was not as good as the other run. And if there is no mutation operation, even with higher run time it is impossible for the algorithm to find out better solution, instead it will find out local minima.

Figure 4.14 shows the impact of mutation operation when crossover operation is not used. It clearly indicates few things. First of all, for mutation-2, 1% is the best choice if not less.

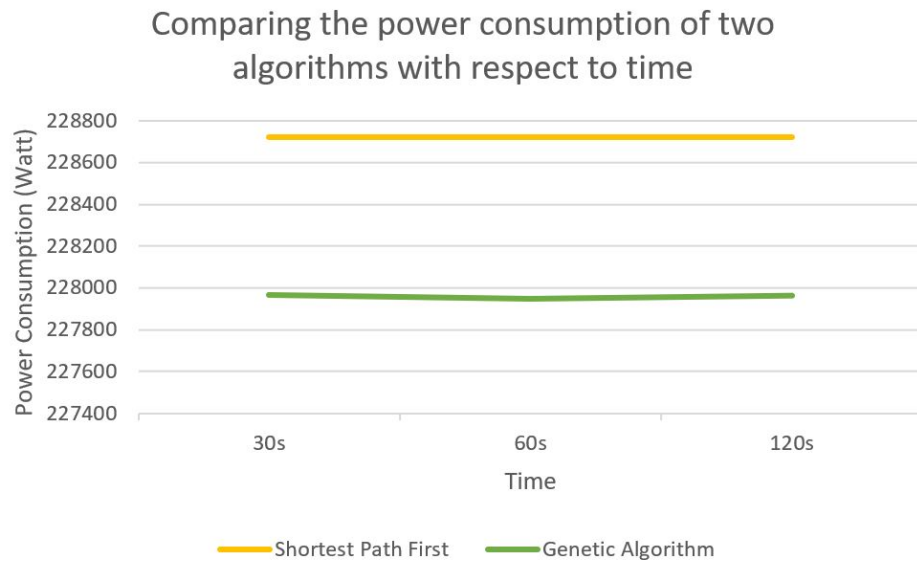


Figure 4.12: Power consumption analysis of the algorithm over longer time period

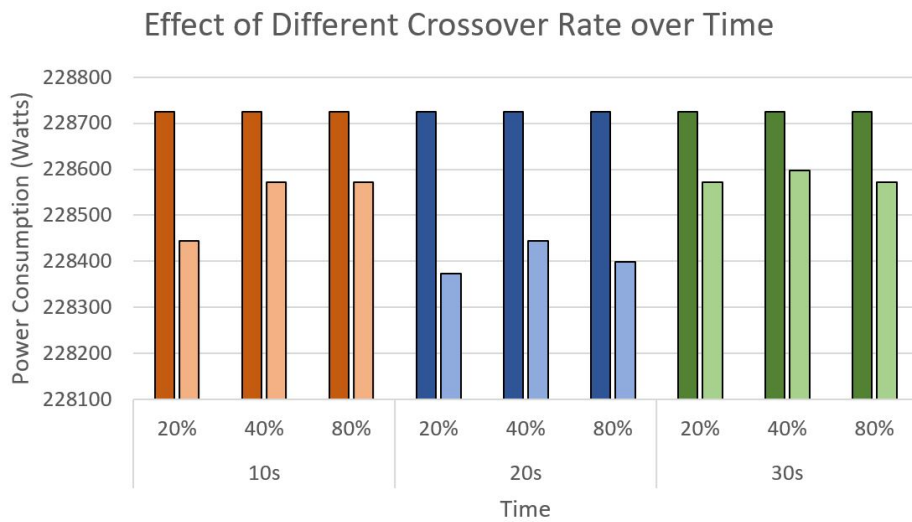


Figure 4.13: Effect of different crossover rate over time

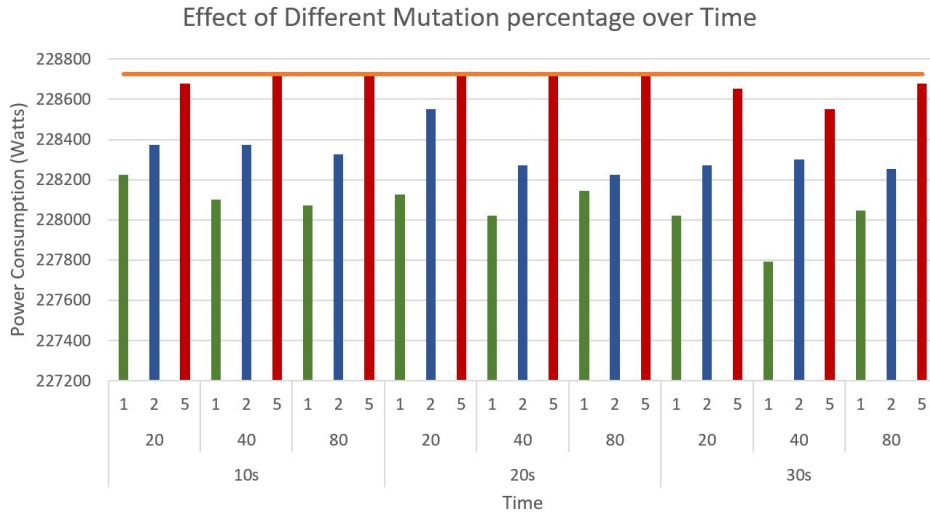


Figure 4.14: Effect of different mutation percentage over time

Secondly, in almost all three cases, 40% mutation rate performs best. Compare to other two. Which is same as our previous findings from design of experiment. Thirdly if we just focus when the mutation-2 is 1% we can see, with time mutation improves in all three cases. This is a clear indication that, even starting with a moderate or average set of population, mutation can bring out a better solution with time. Finally, the final solutions provided by the algorithm, is still few hundred watts short than the best possible value that we have achieved with time. It shows the overall behavior of the genetic algorithm. And how important is to tune the parameters in order to get best possible value.

Figure 4.15 shows the Pareto chart for the effect of all the combination of the parameters. It clearly shows that, the right combination of crossover and mutation-1 has the highest impact on the outcome than any other parameter. In terms of impact, mutation-1 is in the second position. By changing mutation-1 rate as it is possible to find out the global-minimum by changing the rate of mutation-1, the graph is merely showing the obvious. 1.965 is the standardized mean, any parameter or combination of parameters which has value more that has significant impact on the outcome of the result. Another important find is that, size of population has less impact on the overall success of the algorithm. This is due to our solution space. As discussed in chapter 3, we have tremendously large solution space and the values we have considered for solution space is negligible compare to the actual solution space. But, in order to keep the run time of the solution practical and applicable into the real networks these values for the population size are considered.

Based on the outcome of the design of experiment, following parameter values described in

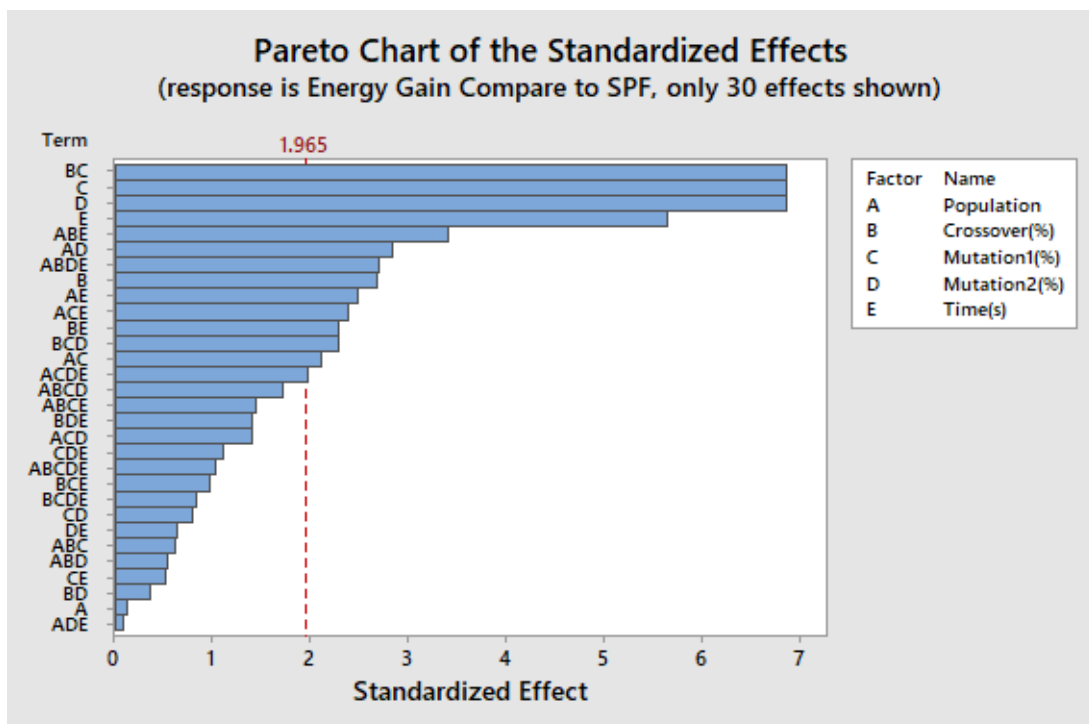


Figure 4.15: Pareto Chart of the Standardized effects

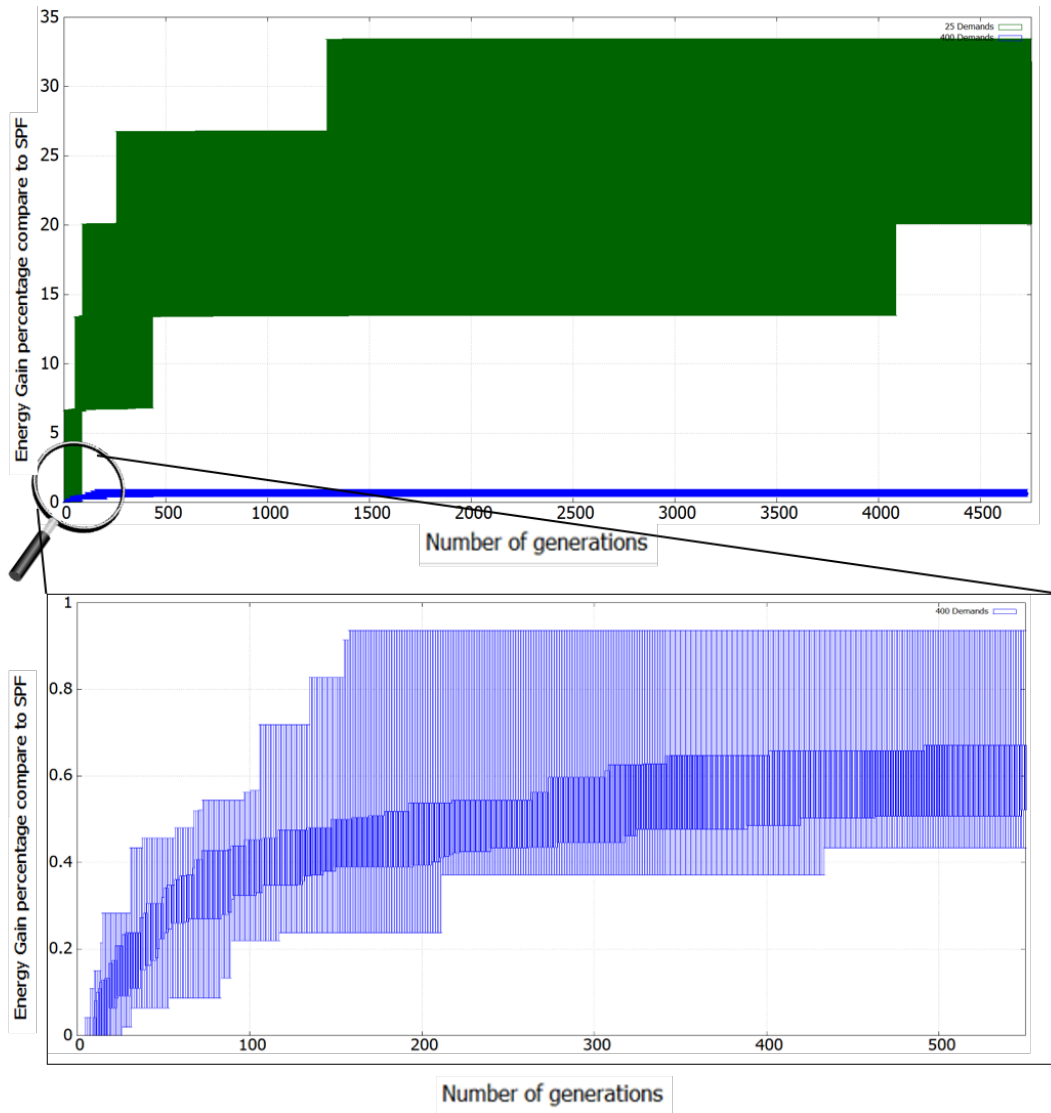


Figure 4.16: Energy consumption gain compare to SPF

table 4.5 are chosen for genetic algorithm. The rest of the experiments are conducted considering these values.

Factor	Values
Time	30s
Population Size	40
Crossover Percentage	40%
Mutation-1 Percentage	80%
Mutation-2 Percentage	4%

Table 4.5: Parameters for Genetic Algorithm

4.5.2 Fitness function analysis

As mentioned earlier, we have conducted the experiments for two sets of demands 25 and 400 respectively and tried to see how much our algorithm over performs compare to shortest path first with ALR mechanism. The fitness function of our algorithm is energy consumption. Therefore, it tries to improve the result based on energy consumption. The top part of the figure 4.16 shows the gain of the algorithm compare to SPF with green policies for two different cases. And the bottom part shows magnified version for 400 demands. Here, 30 runs for each demand set were experimented where every time the demand order is different and showed as a box and whisker. For 25 random demands our algorithm achieved at most 33.4% energy gain compare to SPF with green policies. Whereas, when comes to 400 demands we achieve a gain of around 1%. Now if we consider the current establishment without any green considerations: no device is turned off and no ALR mechanism is also implemented. Compare to current establishment, our algorithm can save up to 8% of energy consumption, when the demand size is 400 and 59% of energy consumption, when the demand size is 25. This phenomenon happened because even for as less as 100 random demands, all the nodes are used either as source or destination and none of them can be turned off. So, when demand size is 400 the only way to save energy is by turning off links or implementing adaptive link rate. Moreover, when we consider the traffic multiplied by traffic coefficient, the network become massively dense. From the result it can be understood that, when the traffic rate is high and network is saturated ALR is more applicable on the other hand when low amount of traffic is passing though network then turning off node is definitely better solution.

There is another important issue, that is needed to be addressed. For this particular scenario with GÉANT, if we consider, all the network devices (i.e.: node and links) are switched on in its full capacity, then the energy consumption ratio between nodes and the links is almost 1:8.

Therefore, as we are considering both links and nodes into the energy equation, during heavy traffic, the savings does not highlight the efficiency of the algorithm. Let us consider a scenario. In GÉANT network, all the network devices (i.e: 22 nodes and 36 links) are switched on in its full capacity and consuming energy accordingly. Now if eight links are turned down, according the values that we have considered, the overall energy gain will be 2%. However, if we do not consider the nodes energy consumption in our energy model as in heavy traffic network these are not always possible to turn down. Then in terms of energy consumed by links only, the gain will be 22%. Which might look large in paper, but in reality the savings will be the same. We wanted to design a robust system with a birds-eye view that will considers both links and nodes and also both sleeping and ALR and provide the best solution in order to save highest possible energy respecting capacity constraint and also within reasonable time.

4.6 Conclusion

This chapter provides an energy aware solution for a SDN enabled network, which is robust enough to handle the large traffic flows mimicking the actual demand and at the same time reduce significant amount of energy consumption when possible. It also describes the genetic algorithm that has been used as a heuristic algorithm for our energy aware solution. The solution provides a set of data and control plane which is suitable for implementing the solution in SDN platform. However the question remains, if is it enough to achieve green networking. Because even though we consider energy consumption is same as environmental pollution but this is not always the case. To investigate this phenomenon we have introduced environmental parameters into the equation in order to achieve more sustainable way of routing than energy aware routing. The next chapter describes about this above mentioned topic.

Moving towards Sustainability

Contents

5.1	Quality in Sustainability	80
5.2	Introducing environmental metrics	81
5.2.1	Carbon emission factor	82
5.2.2	Non-renewable energy usage percentage	82
5.3	Energy aware routing vs Pollution aware routing	84
5.4	Proposed solution	86
5.4.1	integration of environmental metrics into objective function	87
5.4.2	Design of a sustainable system	87
5.5	Evaluation of the model in terms of QiS	89
5.5.1	Result analysis	89
5.5.2	Discussion on sustainability	93
5.6	Evaluation of the model in terms of QoS	93
5.6.1	QoS parameters	94
5.6.2	Result analysis	94
5.7	Conclusion	97

In this chapter, we have discussed about the sustainability aspect of our proposed solution. A sustainable solution is much more than saving only energy consumption. Next, we have introduced two environmental metrics into our equation and proposed a holistic solution with respect to sustainability. After that, we have compared our results that we have achieved in chapter four with the results that we achieved after adding these environmental metrics. The results open up a new dimension in the field of network sustainability. Finally, we wanted to check the quality of service of our proposed solution. Therefore, we have tested all of our solutions

for end to end delay and packet loss to see if they provide satisfactory results or not in terms of quality of service. The contribution of this chapter has been published in [Hossain *et al.*, 2019].

5.1 Quality in Sustainability

When we refer to green sustainable networking or greening ICT, most of the work is done to achieve energy efficiency which passively has an impact on the environment. Nevertheless, the relationship between sustainability and greening ICT has many other layers which are needed to be unfolded. The author of [Hilty *et al.*, 2011] explained a very important point mentioning that, achieving sustainability is not a straightforward approach of saving resources like energy. Even if from a technical point of view, it might seem a direct approach, but the economical and behavioral perspective of the society is needed to be included while trying to achieve a sustainable solution. In fact, sustainability is a unified concept which considers environmental, ethical, and economic aspects as three fundamental pillars. In order to overcome the biasedness towards only energy savings, authors of [Rondeau and Lepage, 2010] have introduced the term Quality in Sustainability (QiS) which encompasses these three pillars. QiS parameters are energy dissipation, emission of CO_2 and other greenhouse gases, waste generation (mass of total wastes or hazardous waste mass), visual pollution (not directly measurable) or electromagnetic disturbances (EMF and SAR measurements). Even though many of the parameters can be measured with values however measuring the ethical impact of these parameters is completely different scenario. Sustainability pillars are a great means of explaining a complete sustainability problem. They indicate that, liable development requires the understanding of nature, society and economic capital or colloquially speaking the planet, people, and profits [Jeurissen, 2000], [Kajikawa, 2008], [Hansmann *et al.*, 2012]. Therefore, in order to design a sustainable green networking architecture, it is very important to somehow consider all three pillars. For example, considering the last experiment done in chapter 4, if the analysis of gain is extended from the point of view of Quality in sustainability the results are given in Table 5.1. Energy consumption costs money and it also produces CO_2 . Here we have summarized the energy consumption, cost and CO_2 emission for one year for our particular scenarios. For the cost of the electricity, average non household unit price and for CO_2 emission average energy to carbon emission rate for European Union countries are been considered [(EEA), 2016].

The result shows how our algorithm improves the outcome of both economical and environmental point of view. However, the people pillar is difficult to measure in values. In our case, it corresponds to the availability of the service for any number of user demands. If we see the values from table 5.1 and also from the discussion in the previous chapter, it clearly shows that when the number of demand is high and the network is saturated the savings are comparatively less than when the number of demand is low. However, this does not mean it is not performing

	Scenario 1 (400 Demands)			Scenario 2 (25 Demands)		
	Energy (MWh)	Cost(Euro)	CO ₂ (Tons)	Energy (MWh)	Cost(Euro)	CO ₂ (Tons)
Current Infrastructure	2154	246016	594	2154	246016	594
SPF	2006	229053	553	1324	151217	365
GA	1987	226881	548	881	100651	243

Table 5.1: Energy ,Cost and CO₂ emission analysis for a year

enough, it actually shows the commitment of the algorithm towards the third pillar in this case which is a guarantee of service for all the demands. During the formalization of the problem, we have mentioned that our algorithm always bound to fulfill the constraint of QoS. Therefore, when the number of demands is higher, as the priority is to provide a connection to all the users the savings become on the lower side. However, when the number of demands is low it performs in all three levels of sustainable development.

The next sections contain the key feature of our solution that has been introduced through my work and how it differs from existing solutions. The holistic view of the solution is explained in the later part of the chapter.

5.2 Introducing environmental metrics

Green networking has been surfaced in the research community for more than one decade or so. The main idea behind green networking is to improve energy efficiency and reduce undesirable energy consumption which will, in fact, reduce the carbon footprint produced by ICT devices. However, it is needed to be remembered that, energy consumption does not always proportionate to carbon emission. The same amount of energy production can have a very different impact on the environment in terms of carbon emission. For example, if we assume an average value of 400g of CO₂ emissions for 1KWh of electricity production [Brander *et al.*, 2011], France produces less than one fifth of the average value whereas Poland produces three times the average value. Therefore, for these two countries, for consuming the same amount of electricity has a quite different result in terms of environmental impact. Which begs the question: *“Is it enough to focus only on energy consumption while trying to achieve green networking?”* In order to address this question, the term, pollution aware routing (PAR) has been introduced which uses energy efficiency techniques and environmental variables combined to find a more pollution aware solution. In this work, two environmental metrics are introduced namely carbon emission factor (CEF) and non-renewable energy usage percentage (NRE) in order to understand the impact of green networking from the perspective of QiS. In the following subsection, these two metrics are described in brief.

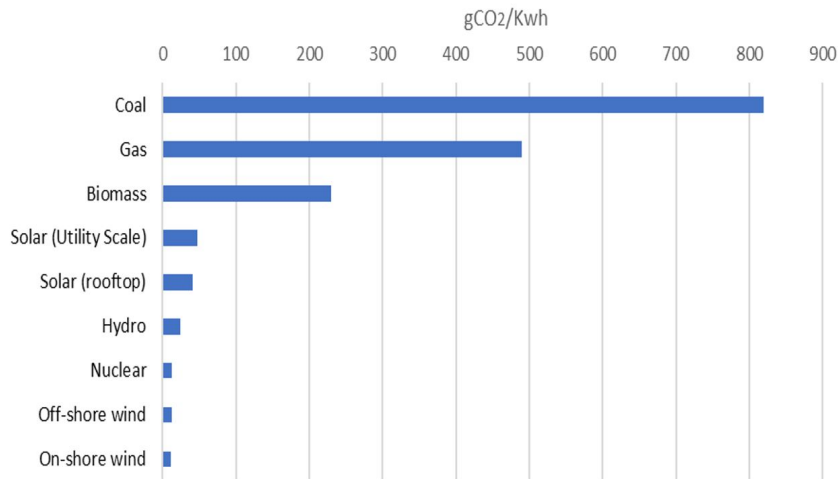


Figure 5.1: Emissions of selected electricity supply technologies (gCO₂eq/kWh)

5.2.1 Carbon emission factor

Carbon emission factor (CEF) means how much CO₂ in grams is emitted while producing 1Kwh of electricity. In a geographically diverse network like GÉANT, it is very common to have different means of generating electricity hence producing a different level of CO₂. Even in US different states use different resources to produce electricity. Now in a network, as it is assumed that all the network devices require same amount of energy and each operation also cost same amount of energy however when we will consider the amount of CO₂ that it will produce will be completely different. Figure 5.1 shows the variation of values of CO₂ emission in terms of energy production methods. The data is collected from [Schlömer *et al.*, 2014]. It can be clearly seen that for same amount of electricity production the CO₂ emission can vary a lot. For example, Coal produces almost 70 times more CO₂ compared to the nuclear power plant. Therefore for producing the same amount of energy carbon emission can be completely different based on which means of production has been used. As discussed earlier, the goal should be to increase the usage of green energy and reduce the usage of brown energy as much as possible. Therefore, it is crucial to consider CO₂ emission if the target is to achieve sustainable green networking.

5.2.2 Non-renewable energy usage percentage

The second environmental metric is non-renewable energy usage percentage (NRE). The idea of NRE is to rank the node (in case of GÉANT is country) in a topology based on NRE. A node that has lower non-renewable energy usage percentage has definitely a better impact on the environment. There are two reasons for choosing NRE as the second factor. Firstly, as described in chapter-2 a number of research has emerged in terms of renewable energy specially

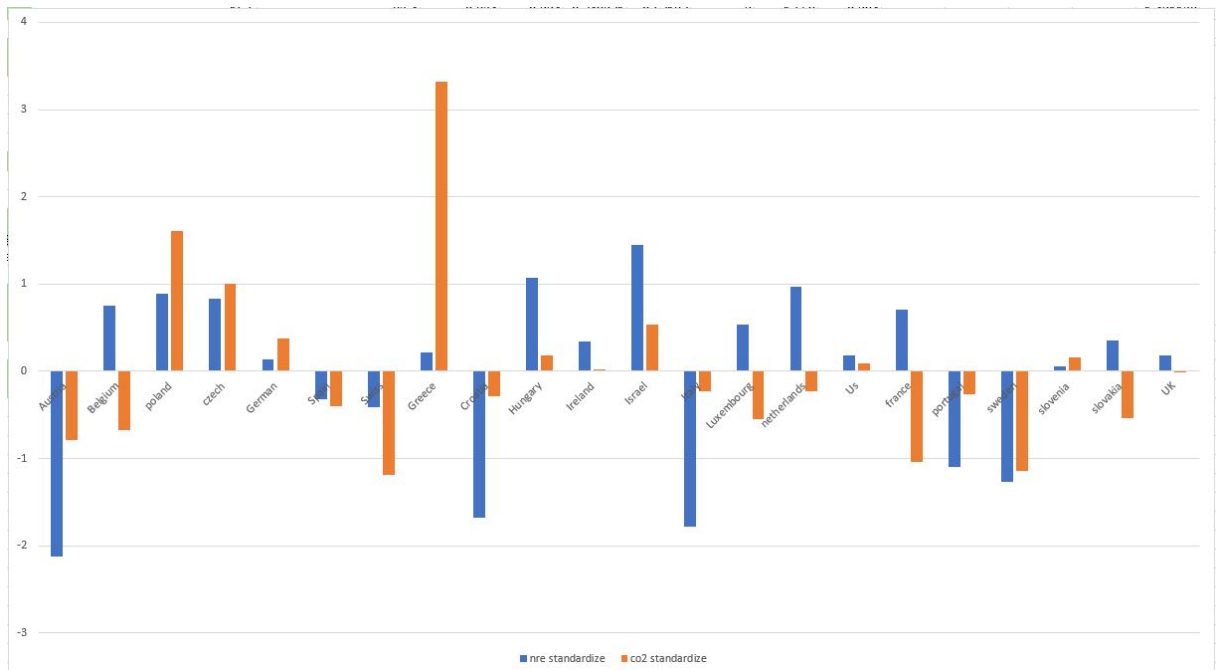


Figure 5.2: Standardization of environmental metrics for GÉANT network

in datacenters. However, in a datacenter network, destination can be changed and it does not hamper the QoS whereas in a wired network changing the destination is not an option! Secondly, the figure 5.1 clearly shows the impact of renewable energy in terms of CO_2 emission. Each country has a different level of energy mix of brown and green energy. And there is a third kind of energy producing technique which is the nuclear power plant which considers as a source of green energy however it does not fall into the category of renewable energy. Because it produces a significant amount of nuclear wastage which has an adverse impact on the environment. Therefore, the second environmental metric is NRE which will be applied to reduce the non-renewable energy usage percentage.

Figure 5.2 provides a standardization of all the countries CEF and NRE which are involved in GÉANT network. Data has been collected from the report of [Brander *et al.*, 2011] and [Van Heddeghem and Idzikowski, 2012]. Table A.1 contains the value of the parameters for each country. It shows that, even though these two factors both focusing on environment and carbon footprint, one country can rank completely differently based on which metric is considered. For example, countries like France, Greece, Italy has very high difference between these two environmental metrics.

Reducing energy consumption always has been beneficial for the environment. However, the energy parameter does not directly concern about the pollution or resource depletion. For ex-

ample, if for one node the power production method is changed from solar to coal, energy wise it does not have any difference but environmentally, it has a quite different outcome. Therefore our two introduced parameters yield more focus on pollution and renewable energy rather than only energy. If we consider the environment, it may seem that considering only renewable energy is sufficient for improving environmental impact as it covers both CO_2 emission and resource depletion. However, the nuclear power plant is not renewable and it emits less CO_2 than most of the renewable energy sources. For this reason, considering the carbon emission factor as a parameter is also necessary.

The main difference between the currently used metric (i.e., energy consumption) and our introduced metric is that ours consider external factors (like power production) whereas energy consumption only reflects network efficiency. Secondly, CEF and NRE will vary based on the location (i.e., country), in this regard, we can say that we have addressed the problem with a more holistic approach considering, in fact, the political engagement of each country regarding the production of the energy. As we have considered our solution from the point of view of software defined network (SDN), based on three different indicator/metrics three different objective functions are formulated, which are used as a fitness function for genetic algorithm which is providing a set of data and control plane as output. These planes are providing operational topology for the network but by focusing either on reducing energy consumption, or total carbon emission or non-renewable energy usage percentage respectively. The controller receives the traffic information from the network, and based on the geographical position of the nodes, the values of environmental metrics like CO_2 emission and non-renewable energy usage are received. Then controller based on the optimizing algorithm provides a data plane and a control plane for the network. The detail description of how a control plane and a data plane are designed are already described in section 4. Figure 5.3 gives an overview of the system.

5.3 Energy aware routing vs Pollution aware routing

In order to get a clear view of the proposed system, it is better to start with EAR and PAR. While implementing green networking, one of the most common and effective solution is to shut down as much network equipment as possible while keeping the operational network. This approach is known as energy aware routing (EAR). EAR has been a focus field for the energy efficient network community for a long time and numerous works have been done in this field. Some works based on backbone networks [Mumey *et al.*, 2012a], [Manjate *et al.*, 2018]; whereas some are for data centers. All the works focused on EAR try to reduce the number of network equipment which is in most cases nodes and links. This drop in the number of active devices reduces the energy consumption which has, in fact, a positive impact on the carbon footprint. That is the main goal while trying to design a green network. The fundamental difference between a classical routing approach like shortest path first (SPF) and EAR is that, SPF considers the number of hops and

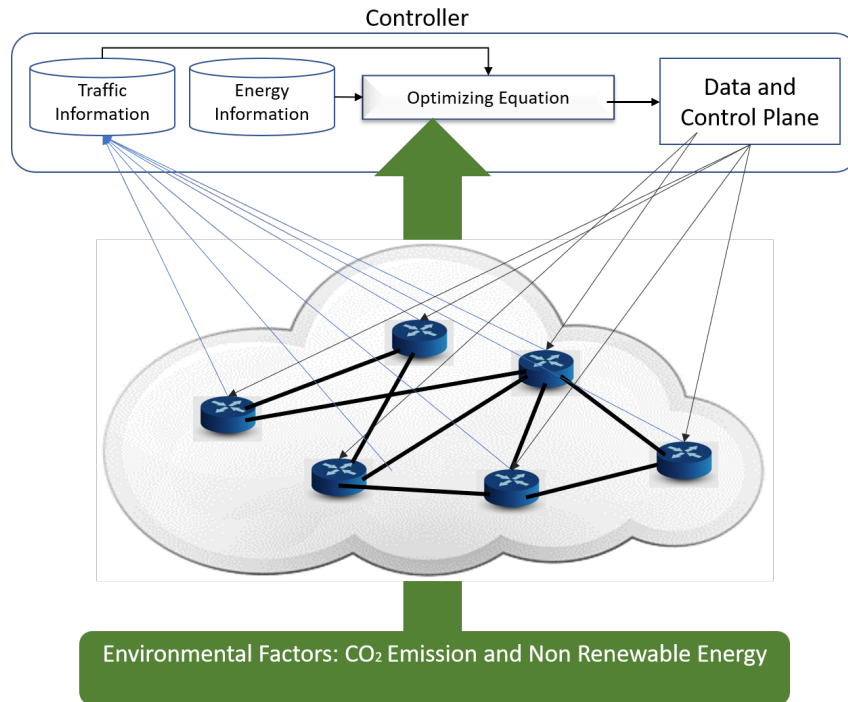


Figure 5.3: Basic Architecture of the System.

tries to reach the destination as quick as possible without focusing on energy consumption of the network. Whereas, EAR tries to reduce the overall energy consumption of the network, which might result a longer path for some demands and similar length paths for other demands.

In this work, a different type of strategy of routing is introduced which is pollution aware routing (PAR). The main concept of PAR is to route the demands with different paths based on environmental metrics such as carbon emission factor (CEF) and the non-renewable energy usage percentage (CEF) of a node and try to reduce the targeted goal of reducing the CO_2 emission or the usage percentage of non-renewable energy of the total network architecture. This routing is applicable for geographically-diverse, distributed network architecture where every node has different means of energy production. The example depicted in figure 5.4 will give a better grasp of the concept. Figure 5.4(a) shows a topology with 12 nodes. For this example, we have considered CO_2 emission as a factor for PAR. For simplicity, there are only 2 demands. One is from X to Z and the second one is from Y to Z. And again, for avoiding complicity, the summation of two demand requirements is less than any available links bandwidth is considered. Three different colors are used instead of values for carbon emission factor. Green means the node has the lowest amount of CEF, orange means moderate and red means it has the highest amount of CEF. Now, 5.4(b) shows the topology for SPF. The lowest number of hops are considered for each demand. Regardless of the nodes CEF or even without considering the total energy

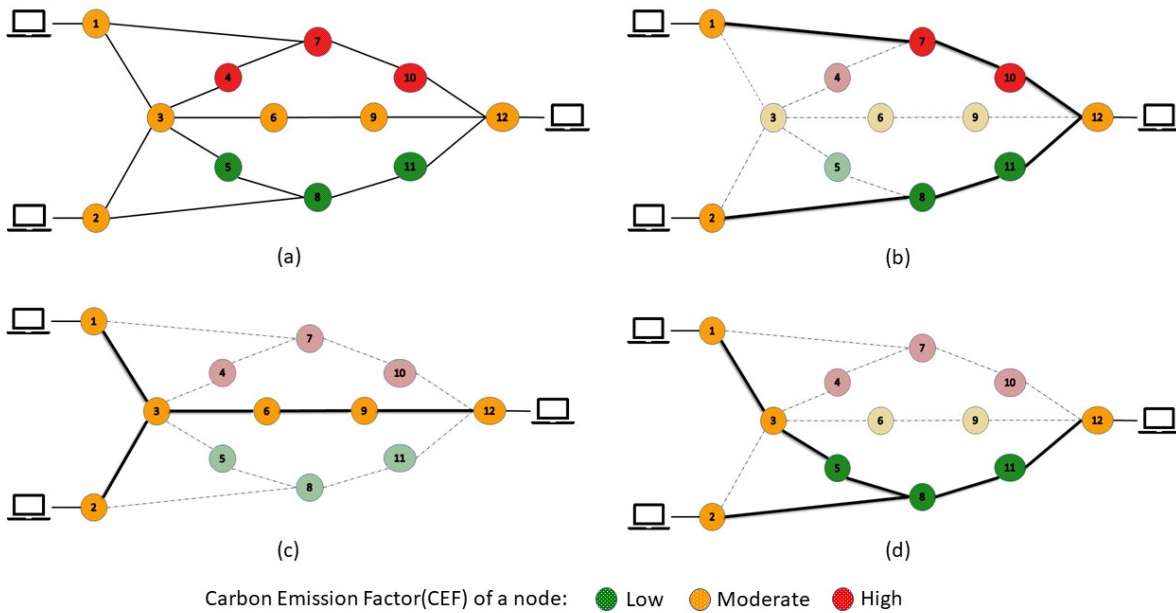


Figure 5.4: Standardization of environmental metrics for GÉANT network

consumption. 5.4(c) shows how EAR will act on this scenario. It will choose a common link in order to reduce the number of active nodes and links. It will reduce one node and two links compare to SPF. However, this will also not be concerned about the CO_2 emission. In 5.4(d) we have a different topology compare to (b) and (c). It didn't look for shortest path or lowest number of nodes and links. Rather it chooses a topology where the total amount of CO_2 emission will be low. It will not consider the total number of turned on nodes and links if the overall CO_2 emission is on the low side.

In the later section of this chapter consists of the result of EAR with two different approaches of PAR and an effort has been made to try to understand the place of EAR and PAR in terms of sustainable green networking.

5.4 Proposed solution

This thesis work presents a solution system which that focuses on sustainable green networking focusing beyond saving energy consumption. in this section, firstly the objective function has been modified to facilitate the environmental metrics. Secondly, a design of the sustainable system is provided respecting the concept of quality in sustainability. The solution provides a holistic view towards the research question of sustainable green networking.

5.4.1 integration of environmental metrics into objective function

Previously in the chapter 4 we have designed the fitness function which focuses on reducing the energy consumption of the data and control plane in order to minimize the energy consumption of the total system. Now, the objective function has been shifted towards our introduced environmental metrics namely CEF and NRE. Now we need to provide an objective function which takes care of both energy metric and also environmental metrics. At first, the current objective function is stated which has been modeled in the previous chapter.

$$\left(\widehat{\Pi}, \widehat{\Gamma}\right) = \arg \min_{\Pi, \Gamma} \int_0^t \sum_{v \in V} \delta_{\Pi}^v \left(\alpha_v + \sum_{w \in V} \delta_{\Pi}^{v,w} \beta_{v,w} (\Gamma_{v,w}) \right) t \cdot dt \quad (5.1)$$

Now, suppose, Λ is a set of carbon emission factors (CEFs) for all the nodes. Where, $\Lambda = \{\Lambda_1 \Lambda_2, \dots, \Lambda_v\}$. And Ψ is a set of non-renewable energy usage percentage (NREs) for all the nodes. $\Psi = \{\Psi_1 \Psi_2, \dots, \Psi_v\}$. Here one thing to notice is that for all, the goal is to reduce it. As mentioned earlier, that both environmental parameters are a multiplicative factor of the original energy consumption model, therefore, objective function can be rewritten for CEF and NRE respectively in the following way:

$$\left(\widehat{\Pi}, \widehat{\Gamma}\right) = \arg \min_{\Pi, \Gamma} \int_0^t \sum_{v \in V} \Lambda_v \times \delta_{\Pi}^v \left(\alpha_v + \sum_{w \in V} \delta_{\Pi}^{v,w} \beta_{v,w} (\Gamma_{v,w}) \right) t \cdot dt \quad (5.2)$$

$$\left(\widehat{\Pi}, \widehat{\Gamma}\right) = \arg \min_{\Pi, \Gamma} \int_0^t \sum_{v \in V} \Psi_v \times \delta_{\Pi}^v \left(\alpha_v + \sum_{w \in V} \delta_{\Pi}^{v,w} \beta_{v,w} (\Gamma_{v,w}) \right) t \cdot dt \quad (5.3)$$

The goal is to provide data and control planes which will reduce the energy consumption or carbon emission or non-renewable energy percentage of the overall system. Then from the above mentioned equations a generalized version of the objective function for system would be:

$$\left(\widehat{\Pi}, \widehat{\Gamma}\right) = \arg \min_{\Pi, \Gamma} \int_0^t \sum_{v \in V} \Phi_v \times \delta_{\Pi}^v \left(\alpha_v + \sum_{w \in V} \delta_{\Pi}^{v,w} \beta_{v,w} (\Gamma_{v,w}) \right) t \cdot dt \quad (5.4)$$

The value of Φ_v will be 1 if the minimizing criteria is energy consumption, Λ_v if the criteria is carbon emission or Ψ_v if non-renewable energy percentage is considered as objective function.

5.4.2 Design of a sustainable system

As explained in section 5.1, a solution cannot be sustainable without addressing the three pillars of sustainability. And in the above section, EAR and PAR are described. The solution, depicted in figure 5.5, have integrated all these elements in order to address green networking problem. And through our solution, we would like to investigate the performance of the proposed approach

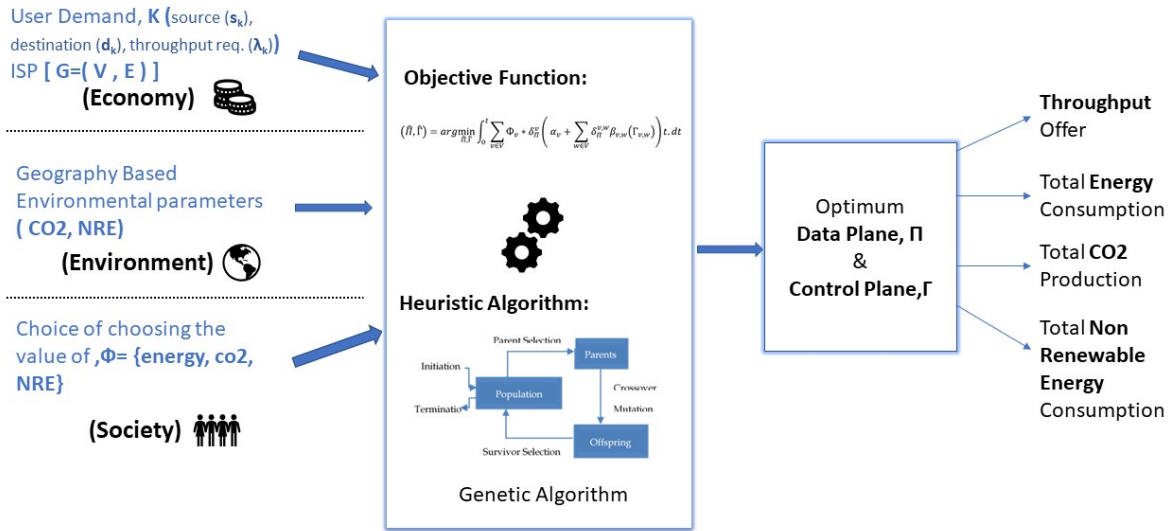


Figure 5.5: Generation of sustainable data and control plane

(PAR) and the available approach (EAR). The solution has been designed keeping in mind the three pillars of sustainability namely economy, environment and society. Figure 5.5 shows the architecture of the proposed system. In the following part, how the three pillars are integrated into our solution is described briefly. Firstly, our system receives a set of traffic flows as demand and the topology of the network from the Internet service provider (ISP). Demands are coming from the users of ISP. Both the user and ISP are the stakeholder for our solution. ISP has direct relation with the profit whereas users are mainly concern about the QoS for example, throughput requirement.

We also take the geography-based environmental information as input such as carbon emission factor (CEF) and non-renewable energy usage percentage (NRE). This information is one of the key factors for providing a green outcome. These environmental parameters are acting as the second pillar of sustainability - the environment. And lastly, our system provides a choosing ability to the client of this system (i.e.: ISP) to select a solution focusing on either reducing energy consumption or CO_2 emission or non-renewable energy usage percentage (the value of Φ in the equation 5.4) form north interface. The social aspect of sustainability focuses on balancing the needs of the individual with the needs of the group. By giving the choosing ability to the internet service providers, it integrates the social entity of the sustainability to our

system. For example, in case of GÉANT network, any environmental strategy or law defined and imposed by European Union can be implemented in the network by tuning the optimizing equation. After getting these inputs our system provides a solution based on the derived equation in the previous section. It uses genetic algorithm as the heuristic algorithm in order to solve the optimization problem. The heuristic algorithm has been described in the following section. And after that, an optimum data and control plane are provided to implement the solution into the topology. These data and control planes are calculated based on the inputs given into the system. The main goal of these planes is to minimize the given objective function while fulfilling the QoS constraint. Alongside with the solution topology settings, these planes also provide vital information about energy consumption, CO_2 emission and, non-renewable energy usage percentage of the topology. This information is essential specially for the people pillar of the sustainable development. Because it provides a clear information about the impact of network users on environment. Which by default provides them a guideline to act accordingly. This kind of feedback always makes the people of the society more involved in the system which is a prerequisite for creating a complete sustainable system.

5.5 Evaluation of the model in terms of QiS

After modifying the objective function based on the newly introduced environmental metrics a new set of experiments are done while all the parameters of the genetic function remain the same. Here the idea is to determine the effect of considering different fitness function into the system and see the change in savings in terms of energy, CO_2 emission and non-renewable energy usage percentage.

5.5.1 Result analysis

A set of experiments have been conducted to analyze the performance of the three different objective functions. The results are compared with Shortest Path First (SPF) with the same green policies that our algorithm has been considered. 25 demands are randomly taken for the experiment as we want to see how the algorithm uses the two green policies according to the objective function. If the number of demands is higher and every node is used as either source or destination, then there will be no scope to analyze the performance of the algorithm by turning of the node. Figure 5.6 shows the graph which gives the comparison of three different approaches. As we can see, every algorithm outperforms the other two when their minimizing factor has been chosen.

When energy is considered as the minimizing factor, that means for our EAR with GA (E_GA), we have maximum energy savings compare to the other two approaches. EAR saves around 24% of energy consumption compared to SPF for this scenario. Whereas both approaches of

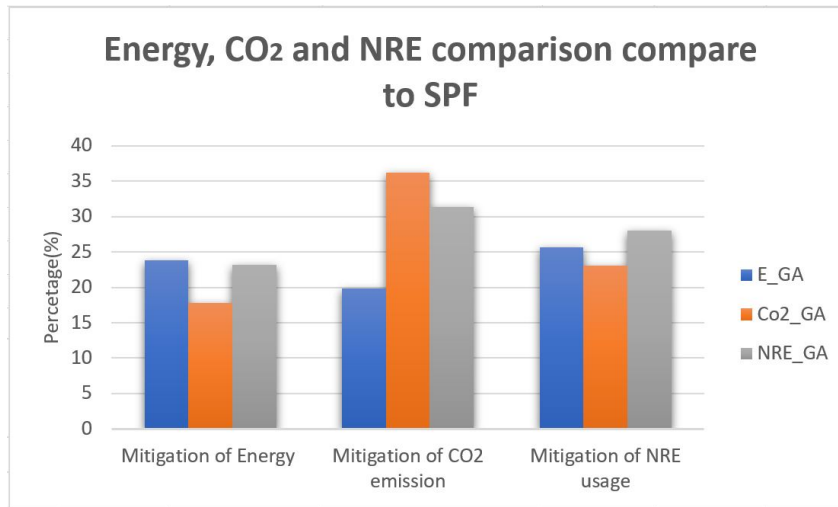


Figure 5.6: Comparison between three algorithms

PAR (CO_2_GA and NRE_GA) saves around 18% and 23% energy consumption respectively. However, when comes to reduce of the CO_2 emission, both CO_2_GA and NRE_GA reduce significant amount compare to E_GA . CO_2_GA reduces 36% of the CO_2 emission compare to SPF. Whereas, NRE_GA and E_GA reduce 31% and 20% respectively. Even though our EAR (E_GA) has over performed SPF by a large margin in all three aspects, in terms of reducing CO_2 emission both pollution-based algorithm performs better than it. In fact, when the optimizing function focuses on reducing CO_2 emission CO_2_GA reduces almost double of the E_GA in terms CO_2 emission. Figure B.1 in the appendix shows all obtained results for SPF and the other three algorithms. It contains the country-wise consumption value of energy, CO_2 and non-renewable energy for each algorithm in order to understand the result more clearly. And lastly, while for reducing non-renewable energy usage NRE_GA performs better than the other two. NRE_GA reduces the non-renewable energy usage by 28% compared to the shortest path. All these results raise the question, as the goal of green networking is to reduce the carbon footprint from the environment then how much effective will be designing a green network by only considering energy efficiency. Even if a solution focusing on reducing CO_2 emission consumes more energy but if the total CO_2 emission of the system is lower that means the system is consuming more green energy than brown energy (energy sources produce a higher amount of CO_2).

Based only on the consumption values that are used in the paper, table 5.2 provides an annual emission of energy, CO_2 and non-renewable energy for the set of randomly chosen 25 demands for this section. Here, at first, we have shown if no green policy is used then what will be the total emission and then the consumption values for shortest path first and our three algorithms are given. These values only reflect the emission due to the devices and links. However, the actual

Method	Energy(MWh)	CO ₂ (Tons)	Non-renewable energy(MWh)
Without any green policy	2154	1088	1470
Shortest Path First (SPF)	1522	684	1074
E_GA	1160	548	799
CO ₂ _GA	1252	436	824
NRE_GA	1170	470	777

Table 5.2: Energy ,CO₂ and NRE emission analysis for a year

amount will be much higher considering the establishment, cooling and many other criteria.

Figure 5.7 gives the different topologies for different solutions. The first one in the top left gives the topology for SPF. As it is clearly visible that, maximum number of nodes and links are used in this topology. Now if we analyze the differences between different topologies some interesting changes in the topology. Energy based solution have not used node six and node sixteen which is Switzerland and France. And have used the node 14 which is Netherlands. For energy-based solution every node is considered as same as every node consumes the same amount of energy. Whereas, for CO₂ based solution for the same number of nodes the emission could be completely different. As for our example, Netherlands emits 130 times more CO₂ gm per kw-h compared to Switzerland and 6 times more compared to France which has a huge impact on the overall result. In the same way, node-2 (Poland) is used by E_GA whereas both of the PAR solutions have avoided Poland as it has huge CEF and NRE usage percentage. These choices made by E_GA might have reduced the overall energy consumption of the network but at a cost of a high carbon emission.

P. No.	Paths	NRE% sum of the intermediate nodes	CEF sum of the intermediate nodes
1	5-16-4-14	1.535	0.742
2	5-16-1-14	1.659	0.294
3	5-12-4-14	1.337	1.082
4	5-12-11-14	1.602	1.15

Table 5.3: Paths for Demand from node-5 to node-14. 1-Belgium, 4-Germany, 5-Spain, 11-Israel, 12-Italy, 14-Netherlands, 16-France.

Now, if we compare the solution of the two PAR based solutions, as mentioned above they both have avoided node-2 (Poland), However, for NRE_GA majority of the traffic went through node-4 (Germany) and node-16 (France) is completely avoided whereas for CO₂_GA a substan-

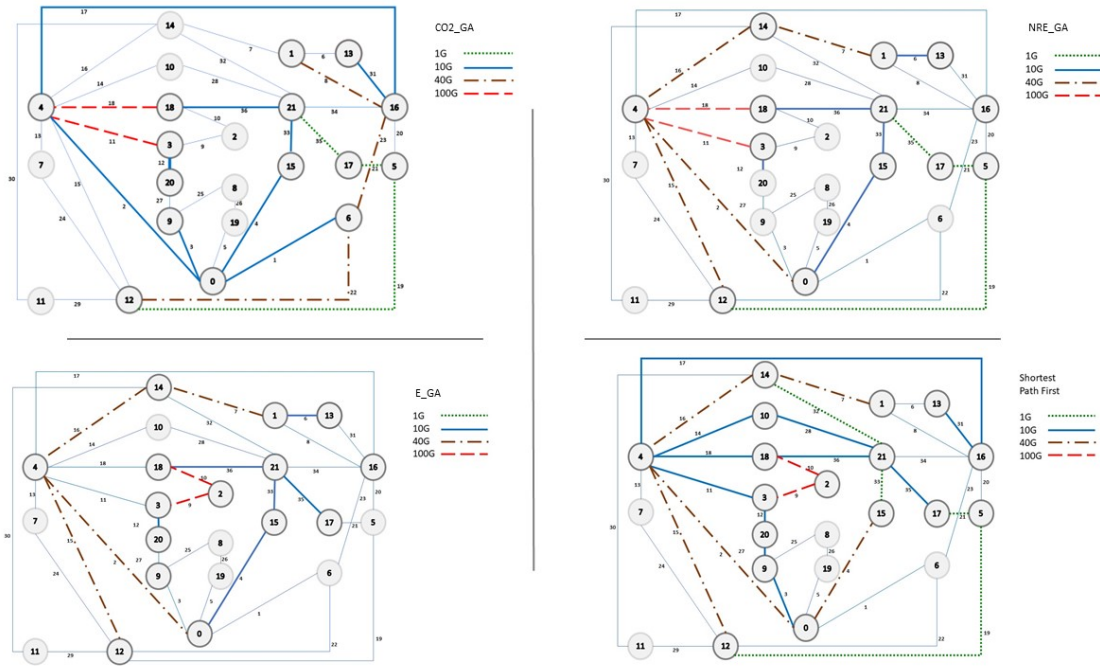


Figure 5.7: Topology for different algorithm

tial amount of traffic went through France even though it has very low carbon footprint. It is because NRE_GA only focuses on non-renewable energy percentage and as France uses nuclear power plant for producing electricity it has very high non-renewable energy usage percentage. For example, for the same topology, let's consider a demand from node 5 to node 14. There are several possible solutions. However, we will consider only the solutions which are shortest that means lowest number of hops. Now, there are four different possible paths. The table 5.3 gives a summary of all four shortest paths with carbon emission factor and non-renewable energy usage percentage. Now, here only the NRE and CEF values of intermediate nodes are considered. As based on the NRE_GA the optimal solution would be path-3 (with Italy and Germany) even though this path has one of the highest amounts of carbon emission factor. Whereas, the optimal solution based on CEF would be path-2 but as this path has maximum amount of NRE percentage, therefore, this will not be chosen by NRE_GA. Based on this analysis of the three algorithms, it can be said, even though the outcome of the algorithm depends on demand set, but it is very much a possibility that in order to achieve a sustainable solution focusing only on energy efficiency might not be the wise decision. Our result shows that for this particular scenario both of our algorithms outperform energy-based solution in terms of reducing carbon emission. Specially CO_2_GA reduces 20% more carbon emission compared to energy-based solution which is definitely not negligible.

5.5.2 Discussion on sustainability

The three pillars of sustainability are a great means of explaining the complete sustainability problem. We have included all the three pillars into our system in order to provide a balanced sustainable solution. Our system provides information about the throughput of the different parts of the network. And whichever objective function is selected it gives an insight about the three parameters for measuring the state of the green networking namely energy consumption, CO_2 emission and percentage of non-renewable energy usage. But, as the goal of green networking is to reduce the carbon emission, therefore the objective function with minimizing the CO_2 has the utmost importance. As we can see, CO_2_GA reduces around 8% of CO_2 emission compared to our E_GA and 40% of CO_2 emission compared to SPF with two green policies. Which is definitely a non-negligible value. However, a more tangible example might give a better understanding of the situation. For our scenario, if we compare the result of SPF and CO_2_GA in one year it can save up to 271 tons of CO_2 . It is, in fact, equivalent to the carbon emission if a person makes 416 times round trip to JFK, New York from CDG, Paris by plane. And even when comparing with E_GA the carbon savings 57 times round trip by plane for same source and destination. And if we consider the savings in terms of money, even if we consider an average rate for cost of electricity production, all three can save more than 25% of the cost compared to SPF. Lastly, the proposed system respects both the user traffic demand and the objective function choice and provides a data and control plane and hence integrate the social attribute of the sustainability.

5.6 Evaluation of the model in terms of QoS

It is always claimed that green networking always comes with the cost of quality of service. When a customer orders a service from a service provider, a service level agreement(SLA) is negotiated. This SLA is defining the third pillar of the sustainability namely people. In the SLA contract, QoS parameters are included in order to specify the service level that will be provided by the internet service provider. Therefore it is pertinent to see the performance of the proposed algorithm in terms of QoS parameters. There are many QoS parameters when comes to networking. However, there are mainly two QoS parameters which define the provided quality of service namely end to end delay and packet loss. In the following chapter, the parameters are described briefly. And then a thorough test has been done to analyze the performance of the algorithm in terms of QoS.

5.6.1 QoS parameters

End to End delay

End to end delay is the time required to reach a packet from source to destination. Network delay can be further divided into three parts:

Transmission delay, that is, the time it takes to transmit a packet through the transmission medium. This part can be significant for large packets over low-speed access links such as DSL or T1 lines. For example, a 1500-byte packet transmitted over a 380 Kbps DSL upstream link would take about 32 ms. Transmission delay is insignificant for high-speed links. For example, a 1500-byte packet transmitted over a 155 Mbps STM-1/OC-3 link would take only 0.08 ms.

Packet processing delay, that is, the time it takes to process a packet at a network device, for example, queueing, table lookup, etc.

Propagation delay, that is, the time it takes for the signal to travel over the distance. Because electromagnetic signal (for example, light) that carries the data does not propagate as fast in fiber or copper as in a vacuum, a rule of thumb is a propagation delay of 5 ms will be added every 1000 km of fiber route [Zekauskas *et al.*, 1999].

Packet Loss

The second parameter that is considered for the QoS analysis is packet loss. Packet loss is when packets traveling through a network medium get lost before getting to their destination. There are a couple of reasons why packet loss happens.

The first one is due to link congestion. As both of the green policies tend to decrease the number of links and also keep the bandwidth in a minimum value. There is a possibility that it will cause link congestion and then packet loss. Sometimes if a link is shared by multiple flows there might be some packet loss.

The second cause of packet loss similar to network congestion is Over-Utilized devices. This means that a network device is operating at a capacity it was not designed for. In a network, packets may arrive faster than they can be processed/sent out. To handle this type of situation, many devices have buffers where they hold packets temporarily until they are able to be processed and sent out. However, in the case of an Over-Utilized device, the buffer will probably fill up quickly, resulting in excess packets being dropped.

In the next section result analysis of the QoS parameter for the proposed algorithm is given.

5.6.2 Result analysis

For analyzing the performance of the algorithm based on quality of service (QoS) networking metrics: end-to-end delay and packet loss network modeler have been used. In the modeler, the average packet size is considered 1500 Bytes. The probability distribution function (PDF) of

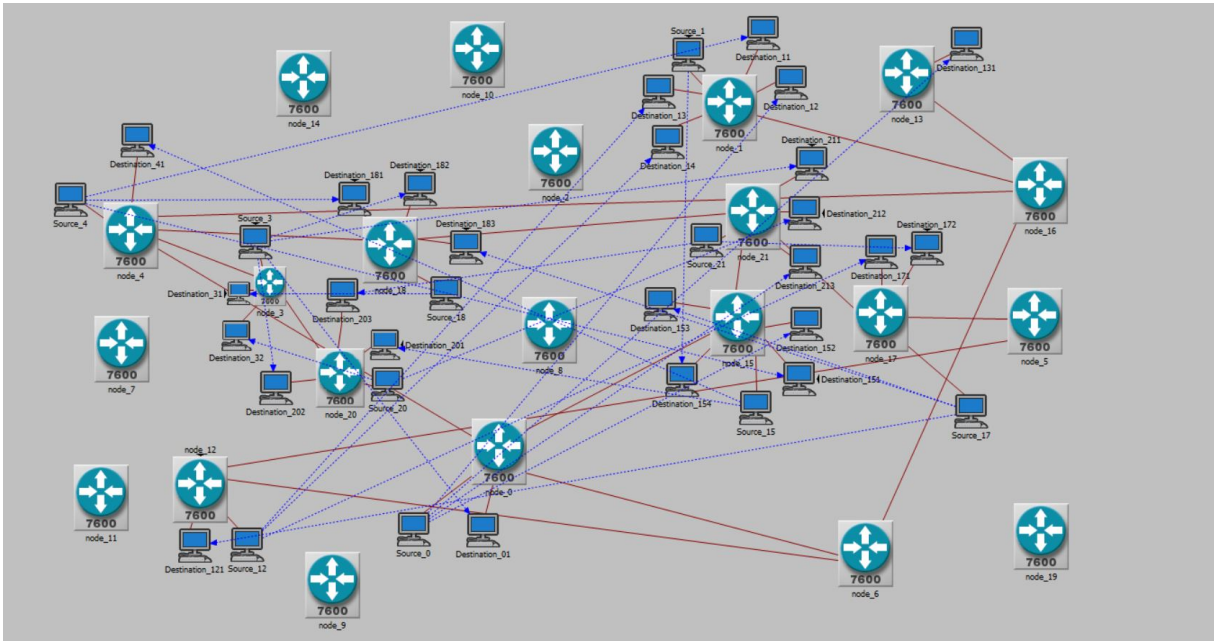


Figure 5.8: Network topology for algorithm CO_2_GA in modeler

the packet size is constant. Whereas, PDF of packet inter-arrival time is exponential. We have designed the network on modeler based on our algorithm provided data and control plane. For each algorithm output, a different topology is designed in the modeler. Figure 5.8 shows the network topology for the CO_2_GA in the network modeler.

End to end delay End to end delay is one of the most crucial QoS parameters to analyze an algorithm. We have compared and analyze the ETE delay ratio with SPF for all three algorithms that can be seen in a box and whisker plot in figure 5.9 to have a better understanding of the distribution of the data. Usually, SPF has good performance in terms of ETE, therefore, we kept SPF value as a comparison unit for our three algorithms. The graph clearly shows that, NRE_GA has higher ETE delay for packets compare to other two. We observe that, the degradation rate of NRE_GA compared to SPF in terms of average ETE delay of packets is more than 5%. And GA_E has the lowest ETE delay ratio. The placement of nodes in GÉANT network and specially the CEF and NRE values of each node has a very high influence on this result. Figure 5.7 shows the different topologies for different solutions which are later designed in modeler for QoS measurements. Now, if we see the topology of NRE_GA, out of 25 demands, 20 demands are passing through node-4 (Germany). The main reason behind this kind of bottlenecking is the NRE value of the other nodes. For example, node-16 (France) uses nuclear power plant for producing energy and because of that, it has a very high NRE value. Similarly, node-2 (Poland) uses coal for producing energy and has also very high NRE value. Therefore, when

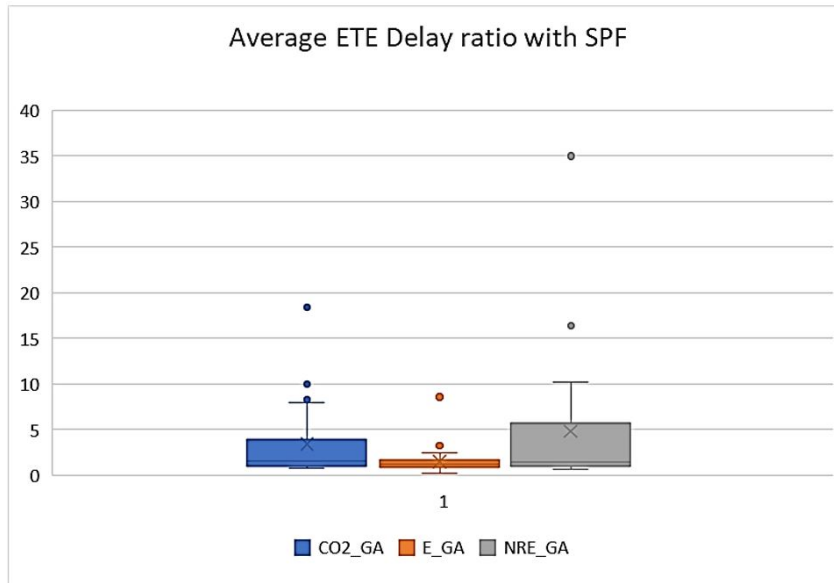


Figure 5.9: ETE delay comparison between algorithms

NRE is considered as an objective function all the paths avoid using node-16 and node-2 which creates a bottleneck in the topology that causes the unexpected ETE delay. For E_GA also, this kind of scenario might occur but for our test case, it was not the case. And secondly, for both CO₂_GA and NRE_GA each node has different CEF or NRE values. Which makes the routing more complex than straight forward energy aware routing where for every node same amount of energy consumption is considered. For a similar sort of reason, CO₂_GA underperforms compare to E_GA in terms of ETE delay.

Packet loss: Figure 5.10 shows the packet loss of all three algorithms plus the packet loss of SPF. In terms of overall packet loss, all of them perform similarly even though there were few outliers in the data set. All of them have an average of less than 5%. And for at least 75% of the overall demands for all algorithms, the average packet loss percentage is close to zero. The outliers are expected as we have saturated the network with a maximum demand size of 97Gbps whereas the maximum amount for bandwidth was 100Gbps. This has been done in order to see the robustness of the algorithm. Packet loss or ETE delay for packets is the result of when the buffer is full or there is a congestion in the link. For both CO₂_GA and NRE_GA, packet loss and ETE delay are slightly higher than E_GA and E_GA has a slightly higher value than SPF. It is expected as all of them try to reduce the number of nodes and links to maximize the greening of the network. Additionally, ALR is maximizing the link utilization percentage and minimizing energy consumption at the same time. For E_GA, the goal is to reduce the number of nodes and links regardless of which node or link, for CO₂_GA and NRE_GA the goal is also to shut

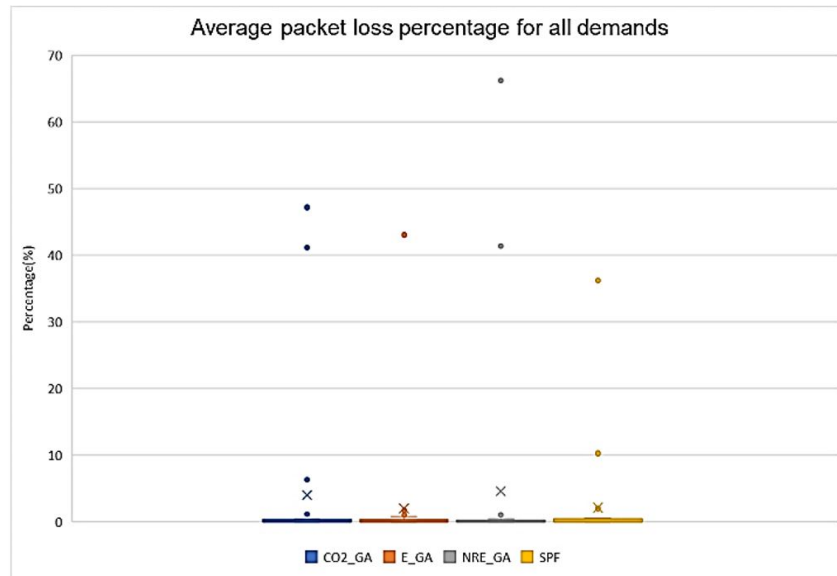


Figure 5.10: Packet loss comparison between algorithm

down nodes and links as both of them, are a multiplicative function of energy and shutting down reduce the overall value of CEF and NRE. However, as different node has different green metrics values, it curves the network even more.

To conclude, the packet loss is more or less similar for all the cases and in worst-case scenario (for NRE_GA) we get an average delay ratio of 5% compare to SPF. Whereas, when comes to savings of CO_2 emission and energy consumption, the same algorithm can save more than 31% and 21% respectively compare to SPF. Now, if we analyze the trade-off between quality of service and quality in sustainability, our proposed algorithms clearly show significant improvement in terms of environmental impact with a slight effect on time performance, offered by the network. Additionally, we can say that, by using our proposed metrics (carbon and renewable energy) in the objective function provides better opportunity to reach the goal of greening network compared to using only energy as objective function which is a common practice in the literature for green networking.

5.7 Conclusion

This chapter explains our novel approach for sustainable routing. In this chapter, we have introduced two environmental parameters namely, CEF and NRE. In this chapter, we have provided an overall architecture of the system where sustainability pillars are respected. Internet service provider (i.e.: user) will provide the priority of their green routing mechanism from northbound interface and based on the choice the algorithm will select its fitness function and solution will

be provided by the controller and implemented through southbound interface to the network. We have also tested our algorithm performance based on QoS parameters like end-to-end delay and packet loss. Our approach of including environmental parameters is simple and feasible. More information like how the variation of energy demands will impact on carbon emission, green energy rate, and electricity price can also be integrated into the system. Here we have provided the algorithm ready to implement for achieving sustainable routing. However, there is one concern, which is how the implementing of the algorithm will impact on the stability of the system. Therefore, in the next chapter, we have analyzed the relation between implementing of routing algorithm and stability of the system.

Green-networking and stability

Contents

6.1	Network stability issues	99
6.1.1	Incoherent data planes	100
6.1.2	Convergence issue in distributing a data plane	101
6.1.3	Network instability and its occurrence	103
6.2	Improving stability of the system	105
6.2.1	Introducing penalty	105
6.2.2	Plane filtering: To implement or not	109
6.2.3	Extension of the model	111
6.2.4	Case study	111
6.3	Conclusion	119

6.1 Network stability issues

In this chapter, we will discuss about the stability issue of a network and how stability of a network can be affected by green networking. Our algorithm succeeded to provide better sustainable solution. But, now it is time to take step back and see the whole picture from also implementation point of view. While focusing on achieving the maximum energy efficient (brown/green energy) or CO_2 efficient networking topology, stability of the network is often overlooked by the researchers which is a key factor while implementing the algorithm into the actual system. Using of dynamic algorithm like us or any reactive energy aware routing might cause stability problem of the network. For simplicity, stability of the network can be defined as a state when a network fully functional. Inversely, an unstable network is a network which is not operating according to the requirement. Before going deep into network stability it is important

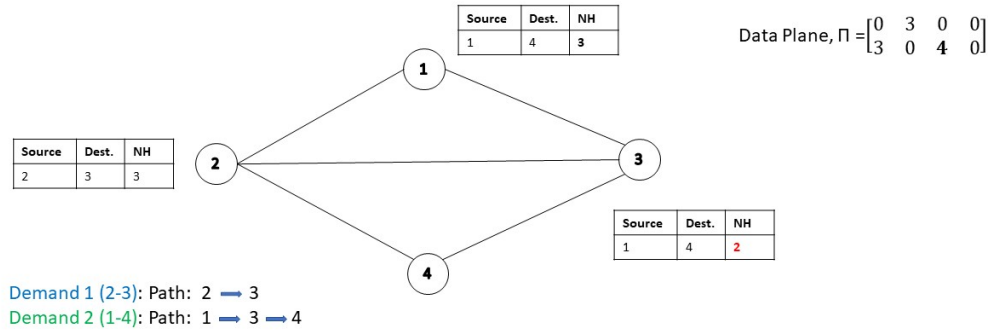


Figure 6.1: Data plane incoherency example

to know more about the reasons behind it. There are two main reasons behind instability of the network, incoherency and convergence issues. In the following section we have discussed more about these issues from our problem’s perspective and then stability and it’s control mechanism is discussed.

6.1.1 Incoherent data planes

Coherency means two routers of a network shares same topological information. Coherence of the data planes (i.e: forwarding table) are a requirement for a network to work accurately. Incoherence can lead to routing loops and inconsistent traffic forwarding and also future inconsistent routing table entries. As in SDN perspective, data plane refers to all the functions and processes that forward packets/frames from one interface to another, every data plane should be coherent content wise.

For example, for simplicity, lets consider a small network to demonstrate the coherency problem. Now figure 6.1 shows an example of incoherent data planes. Let us consider, we have currently two demands that are in use as flows. For simplicity, here, each forwarding table consists of only three fields source, destination(Dest) and next hop(NH). The first demand flow is going from node-2 to node-3 and the second one is going from node-1 to node-4. In the figure we have also given the path, that the demand should follow that has been calculated by the algorithm. But there is a clear incoherency between the data plane and the forwarding table of

node-3 (marked red). Because of that even though demand-1 is able to reach to its destination, there is not enough information available for demand-2 to reach its destination. The flow for the second demand will reach to node-3, the flow will redirect to node-2 due to the incoherency between the data plane and forwarding table. And as node-2 does not have any information about what to do with demand-2, the packets will be lost. This example demonstrate, the severity of the incoherency of the data plane. This kind of incoherency usually occurs when there is rapid changes in the data plane due to frequent changes of the demand set. Which results incomplete updation of forwarding table and data plane and residue information from previous data plane and forwarding table leads to path that might not exist in the current solution. In a large network like GÉANT, incoherent data planes can lead to several packet loss and also the congestion of the network.

6.1.2 Convergence issue in distributing a data plane

Generally, convergence means reaching to a set point. In networking point of view, convergence is to reach to a state, when all the routers have same knowledge about the arrangement of the element of the communication network in which they operate. In other word, data planes of all the routers have coherency and they all are indicating same topology. In order to consider all the routers to be converged, they must have obtained the same topology information from each other via the implemented routing protocol. Any contradict, of the information will result different routing problems like packet loss, zombie nodes, or infinite loops. But in SDN or any centralized approach, controller is the sole responsible to calculate the data planes for each nodes (i.e: routers) and it disseminates this information to all the nodes and convergence depends on the information provided from the controller, rather than collecting from neighboring nodes. The main difference between coherency and convergence is that, coherence indicates rather a static state whereas convergence is more of a dynamic state. When the convergence is occurred, all the forwarding table became coherent.

Every time, when there is an update of the demand set, the routing algorithm provides an updated topology and a new set point is established. All the router should have this updated topology to be coherent to each other. In order to reach to that set point or to be in a state of convergence, it requires certain amount of time, known as convergence time. This time is a measure of how long it takes to update all the routers from current topology information to new topology information. In SDN, as mentioned earlier, a controller sends the topology information to all the routers. However, this dissemination is done one by one. It is not possible for the controller to change every nodes forwarding table according to data plane, simultaneously. Any update may also require changes in data plane or control plane or both which also requires time. Therefore a convergence time is required to update data plane, control plane and all the forwarding table and achieve coherency. Many of the works, does not consider this time as they

$2Gbps$ }, $\{12, 10, 0.5Gbps\}$, $\{8, 4, 2Gbps\}$. Π and Γ gives the data and control plane for this topology. Now in figure (b), there is a new demand added to the demand list $k_4 = \{s_4 = 1, d_4 = 8, \lambda_4 = 1Gbps\}$. And because of that one demand, both data plane and control plane is changed completely. And as data plane is changed, in order to achieve coherency, forwarding table of each node has also changed quite a bit. The topology figure provides a visual aid about the change of the topology. And then in the third figure (c), one more demand is added to the list, $k_5 = \{s_5 = 9, d_5 = 2, \lambda_5 = 14Gbps\}$. And because of its large throughput requirement, it creates another round of all sort of updation in data plane, control plane and forwarding table. And lastly in (d), another new demand is added but an old demand is also removed at the same time, keeping the demand set equal of the previous scenario, however, the data plane, control plane and forwarding table are change quite extensively compare to the previous state. This shows, how a small change in the data plane can trigger large amount of change over the network. For actual GÉANT network, the scenario would be much more adverse. Now it can be considered that, each of this update requires certain amount of time which will be added and increase the overall convergence time that might lead to network instability.

6.1.3 Network instability and its occurrence

In a computer network, data flows may travel through different paths to arrive to their destination. While from user perspective, it might look that, transportation is only end to end as in the demand list only the source and the destination are mentioned but in fact it is the job of the network protocol for the faultless transfer mechanism in order to avoid congestion problem and data-loss. A network is said to be a stable network, if the network routing protocol can perform its job without any interruption. As explained earlier, coherency and convergence of the data plane and the forwarding tables of the nodes are the basis to have a stable network. However, defining an unstable network can be difficult. Because instability of a network cannot be measured, it is a state when network fails to perform. For dynamic routing protocol, achieving convergence is always a challenge specially because of the both unknown occurrence of the demands and duration of these demand flows. In terms of coherency and convergence, a network is unstable if frequency of triggering of the algorithm (i.e.: a new data plane) is larger than the computation and convergence time of the new data plane. This relation is similar to Nyquist-Shannon sampling theorem, which denotes that, sampling frequency of a signal should be at least double of its highest frequency component. If the total computation time is denoted by C_t and convergence time is denoted by C'_t and the time period between an update of the demand set is $T_{K-K'}$ then, equation 6.1 has to be true in order to be a stable system.

$$C_t + C'_t < T_{K-K'} \quad (6.1)$$

For a reactive routing algorithm like ours, algorithm starts computing and provides a new

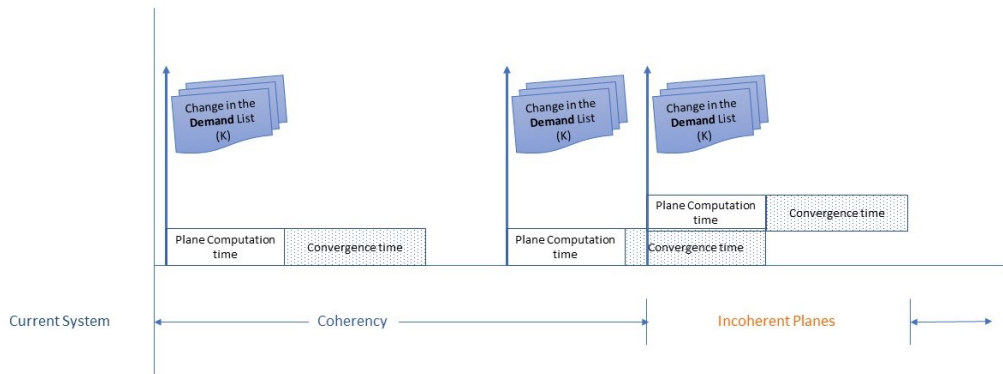


Figure 6.3: Occurrence of incoherent plane in time curve

topology as soon as there is a change in the demand set. Therefore, after an updation of the demand set, there is a certain period of time required for computing a new data plane and then distributing that data plane to the forwarding table (i.e.: convergence time). In order to be a stable system the next update of the demand set must occur after the convergence state has reached for the current topology. If the second updation starts before the convergence is reached then there is a high possibility that incoherency will occur and that will lead to a unstable network. Without the convergence of the network, it will never be ready to transmit data flows and fulfill user requirement.

This phenomena can be explained better with a time-line figure. Figure 6.3 depicts a scenario when there is a possibility of unstable system. Every time there is a change/update in the demand set, K , the algorithm automatically starts computing and generate a new data and control plane for the updated topology. And after computation it requires a convergence time. In the figure it can be seen that, during the convergence time of the second updation of the demand set, a new change in the demand set is occurred. Therefore, convergence time of the second update and computation time for the third update merged together resulting incoherent data planes and forwarding tables. Which might often lead to an unstable system.

Stability of a network system largely depends on the frequency of change of the demand set. Now, as explained earlier, the demand is an unknown variable and cannot be controlled. In the chapter 3, we have analyzed the pattern of the incoming demand flows. But frequency of the incoming demand and the duration of the demand is somehow impossible to predict beforehand as it entirely depends on the user and we have no control over it. The only thing that we can do is to control and moderate the behavior of our algorithm so that it can lessen the possibility of an

unstable system. In the following section we have introduced few mechanisms that will attempt to reduce the percentage of being an incoherent system and by default, improve the stability of the system.

6.2 Improving stability of the system

In this section, we have discussed about two mechanisms that we have integrated into our system in order to improve stability. The first one is a penalty mechanism that is added to the objective function. The idea behind adding this penalty mechanism is to reduce the number of updation related to planes and forwarding table. And secondly, we have also introduced a filter to address this reconfiguration penalty, and will work as a determining factor to apply the reconfigured topology or not. In the following subsections work methodology of both mechanism are described. In the later part, the integration of these mechanisms with the current system is described alongside with the scenarios where these mechanisms are applicable.

6.2.1 Introducing penalty

The first mechanism that we want to take into account is a penalty that will be added to the objective function which will address the problem of reconfiguration latency and additional energy/ CO_2 /Non-renewable energy cost. As explained earlier, every time there is a change in the demand set, topology might go through a series of updation of namely data plane, control plane and forwarding table. Sometimes the number of updations can be small and some times it can be quite large number. Our optimizing algorithm focuses on the fitness function (i.e.: energy, CO_2 or NRE) and based on the current demand set provides a topology which performs best in terms of our saving criteria regardless of number of updation. Previously, as we can see in figure 6.2 for a single change in the demand set the overall topology might change quite a bit. Each increasing updation for controller increases convergence time and elevate the chance of getting an unstable state. Therefore, we have introduced a reconfiguration cost, Ω as our penalty function. From a control engineering point of view, it will address the stability of the command to switch from a plane to another one and keep in check the number of updation for the controller. As each updation requires some time, if the number of updation is in check, then convergence time will also be reduced . It will lead to a new objective function for the system:

$$\left(\widehat{\Pi}, \widehat{\Gamma}\right) = \arg \min_{\Pi, \Gamma} \int_0^t \sum_{v \in V} \Phi_v \times \delta_{\Pi}^v \left(\alpha_v + \sum_{w \in V} \delta_{\Pi}^{v,w} \beta_{v,w} (\Gamma_{v,w}) \right) t \cdot dt + \Omega(\Pi, \Gamma) \quad (6.2)$$

$\Omega \geq 0$ will correspond to the cost to switch from the actual control and forwarding planes to the optimal new ones. If we consider \mathcal{U} is the set of total number of updation in data plane, control plane and the forwarding tables of the topology, then the Ω can be written as:

$$\text{Reconfiguration cost, } \Omega = \frac{\mathcal{U}}{|\mathcal{E}_{\Pi',\Gamma'} - \mathcal{E}_{\Pi_0,\Gamma_0}|}$$

Where, $\mathcal{E}_{\Pi',\Gamma'}$ is the energy/ CO_2 /NRE consumption of the new planes and $\mathcal{E}_{\Pi_0,\Gamma_0}$ is the energy/ CO_2 /Non-renewable energy consumption of the current planes. Here for difference, absolute value is considered for the case, when due to change in the demand set new nodes and links are required and the current value is lower than the new solution. It is a ratio of gain between new and current planes and the number of updation to achieve that gain. The gain here is interchangeable between the considered fitness function. For simplicity an update can be considered as addition of a new rule, deletion of a old rule or replacing the value of current rule in the planes and forwarding table. The goal of the objective function will be now provide a solution (i.e: data and control plane) which provides best solution in terms of fitness function at the same time this new solution will have lowest reconfiguration cost compare to all the other solution.

Every time a new set of flow(s) are added to the flow list, the whole flow list is considered all together in order to provide a best result with respect to fitness function regardless of the current topology. For example, if we consider energy savings as our fitness function, multiple topologies with same number of node and same capacity links will consume exactly same amount of energy. And all of them can be a candidate solution for the demand set. And as the all consumes same amount of energy, from algorithms point of view, all the solutions are equal but from the data planes perspective the solutions are might be far away from each other. In this kind of scenario, previously algorithm might choose a solution which will cause more updates but by doing that we might gain in terms of objective function but with the increasing number of changes in the planes might nullify the gain that we are achieving in terms of stability. Because, the primary goal of a green network is to provide service to the user while ensuring as much energy or carbon savings as possible. However, if frequent changes of the topology lead to a unstable system, then it will hamper the primary goal. Therefore, this penalty will bound the objective function to provide a better solution in terms of stability as it will definitely reduce the convergence time of the network.

Let us use an example to demonstrate the usage of penalty system more clearly. Figure 6.4 depicts a portion of GÉANT topology with five demands where $K = \{k_1 = \{s_1 = 8, d_1 = 2, \lambda_1 = 2Gbps\}, \{12, 10, 0.5Gbps\}, \{8, 4, 2Gbps\}, \{1, 8, 1Gbps\}, \{3, 9, 3Gbps\}\}$. We have considered only five demands in order to display every aspect of the topology more clearly. This is depicting the current state of the network which is in a convergence state. Π_0 and Γ_0 represent the current data plane and control plane of the controller. The paths are also given in order to follow the route. Now at a certain time, there is an update of the demand set. From the current

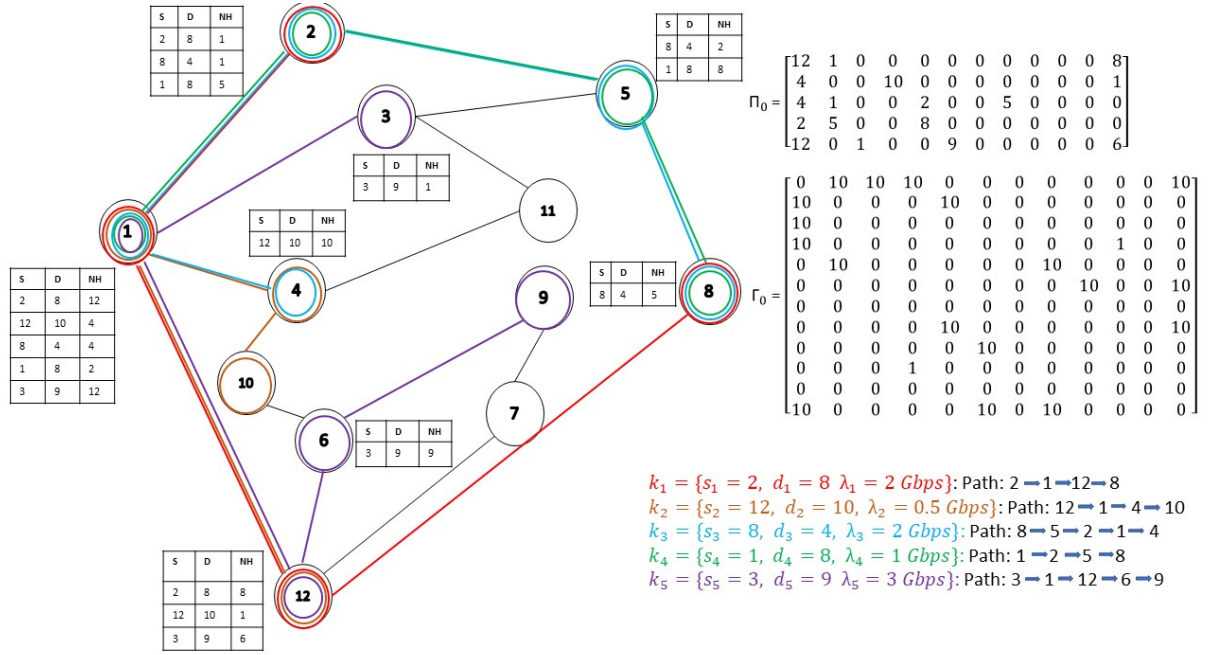


Figure 6.4: Example scenario for demonstrating penalty mechanism

demand list k_5 has been removed and at the same time another new demand has been added $\{s_5 = 9, d_5 = 2, \lambda_5 = 3 \text{ Gbps}\}$. Now let us consider, the fitness function for the algorithm is energy consumption. After adding this demand the algorithm computes new planes and provide two set of solutions with exact same amount of energy consumption. Figure 6.5 and 6.6 show the two solutions that our algorithm provided. (Π', Γ') and (Π'', Γ'') are the data and control plane of first and second solution receptively. Even though their planes and forwarding table look completely different from each other, they both use same number of nodes and links with similar capacity settings. Therefore, both consume exactly same amount of energy. However they have quite different values in terms of updation of planes and forwarding table. Now if we do the calculation, for figure 6.5 the configuration cost would be,

$$\Omega' = \frac{\mathcal{U}'}{|\mathcal{E}_{\Pi', \Gamma'} - \mathcal{E}_{\Pi_0, \Gamma_0}|} = 0.003242925 \quad (6.3)$$

And for the figure 6.6 the configuration cost would be,

$$\Omega'' = \frac{\mathcal{U}''}{|\mathcal{E}_{\Pi'', \Gamma''} - \mathcal{E}_{\Pi_0, \Gamma_0}|} = 0.000884434 \quad (6.4)$$

As we can see that, even though both solution consume same amount of energy, the first solution's configuration cost is 3.5 times higher than the second solution's configuration cost.

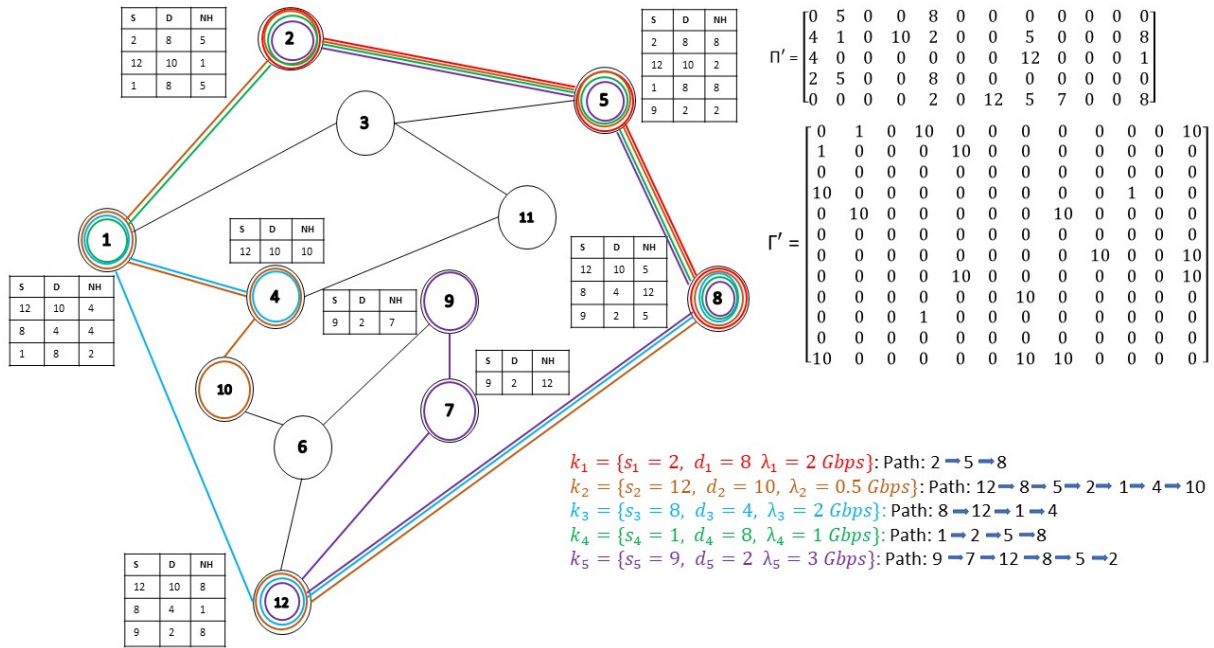


Figure 6.5: First solution with higher configuration cost (Ω')

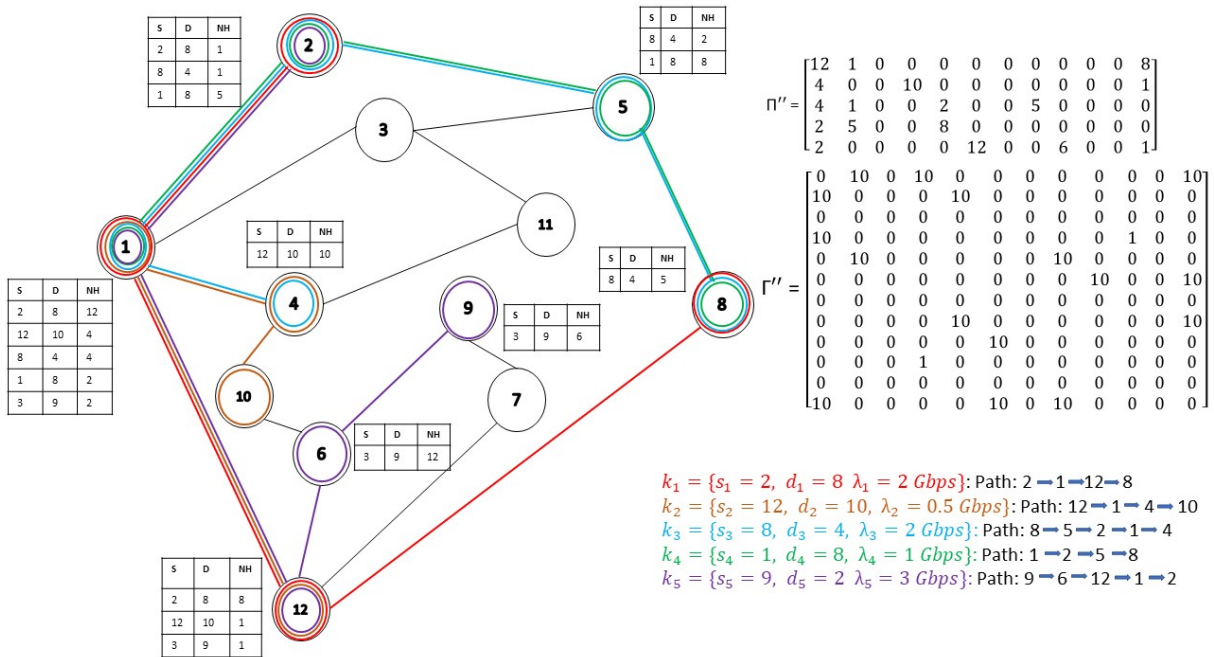


Figure 6.6: Second solution with lower configuration cost (Ω'')

This penalty mechanism assists the genetic algorithm to find out a solution which is green and at the same time less time to converge while applied by the controller.

6.2.2 Plane filtering: To implement or not

Introducing penalty improves the optimal solutions in terms of number of updation, but still it does not ensure not to have a situation where incoherence will occur. Because the new topology might reduce the number of updation and by doing that convergence time also, but it does not mean that during that convergence time another update of the demand set will not occur and will create a chance of incoherent system. Therefore, we have introduced a plane filtering mechanism to reduce the possibility of incoherent system even more. Plane filtering is a simple mechanism that decides, whenever system provides a new update of the topology, the controller will apply this change or not.

With the change of a demand set there are two things that are needed to be considered. Firstly, if this update is an obvious requirement or not. Because sometimes a new flow requires a set of nodes and links that are mandatory to be turned on. The previous topology was not sufficient (i.e: new node/links are needed to be turned on) for fulfilling the new set of flows. But some other times it may be possible that, in order to fulfill the requirement of updated demand set, previous topology was sufficient to fulfill the demand. And if the second case implies then comes to the point of how much gain can be achieved by making the changes of the plane as previous setup was sufficient to fulfill demand requirement. In order to make this kind of decisions, plane filtering is introduced. Specially, during frequent demand changes, if current plane is sufficient then may be its better not to implement new plane and push to system towards incoherence. In the next part we have described the logic and threshold policy for plane filtering.

Filtering logic

The logic behind this filtering mechanism is quite simple. Our first and foremost concern is to provide service to the user. Therefore, if a demand requires an update of the current planes then the logic is to just accept the new planes and implement it through the controller. Because, if it is not implemented, then the system is unable to serve user, which is primary cause of a networking system. Now In the second part, of the logic concerns, if the current plane is sufficient for new demand set. This can happen in scenarios like deletion of demands from the current demand set or change in the capacity requirement or both. The detail of each scenario is discussed in the section 6.2.4. Now in these scenarios, we again consider the configuration cost, and we compare this configuration cost with a threshold value. This threshold value works as a filter for a new plane to be implement or not. If the threshold value is higher than the configuration cost of the new plane then the current planes are replaced with new planes and if not then the current

Symbols	Descriptions
Π_0	Current data plane
Π'	New computed data plane
Γ_0	Current control plane
Γ'	New computed control plane
D	Demand list
$Th(t)$	Threshold at time t
Ω	New plane reconfiguration cost

Table 6.1: List of symbols and definitions

planes are kept for the topology. The list of symbol for the algorithm 7 is given in table 6.1.

Plane_filtering

```

input :  $D$  updated demand list each demand containing source ( $s$ ), destination
         ( $d$ ) and throughput ( $\lambda$ ),  $\Pi_0$ ,  $\Gamma_0$ ,  $\Pi'$ ,  $\Gamma'$ 
output: Decision of accept/reject the new computed plane
if satisfying_demands( $\Pi_0$ ,  $\Gamma_0$ ,  $D$ ) == False then
  | Accept  $\Pi'$ ,  $\Gamma'$ ;
else
  | if  $\Omega < Th(t)$  then
  | | Accept  $\Pi'$ ,  $\Gamma'$ ;
  | else
  | | Reject  $\Pi'$ ,  $\Gamma'$ ;

```

Algorithm 7: Plane Filtering Algorithm

Threshold policy

One important part of this filtering is the value of the threshold and how it will be controlled. As we know for configuration cost $\Omega \geq 0$. Therefore for threshold, Th , also, $Th \geq 0$. Now, even though, Th is compared with configuration cost, the value of the Th is independent to configuration cost. The objective of using the threshold value is to minimize the incoherence of the system. As updation frequency of demand set is unknown, we considered to use additive-increase/multiplicative-decrease (AIMD) algorithm to guide our threshold value. AIMD is a

feedback control algorithm best known for its use in TCP congestion control. In TCP congestion control, this algorithm increases the transmission rate (window size), probing for usable bandwidth, until loss occurs. And as soon as congestion detected, the window size is reduced multiplicative of a given value. In our case also, Whenever there is a incoherency detected between planes and forwarding table, the threshold is reduced and reducing the accept rate of the new planes and demand flows are managed using current planes. The following equation is used for defining threshold policy. Let $th(t)$ be the threshold during time slot t , a ($a > 0$) be the additive increase parameter, and b ($0 < b < 1$) be the multiplicative decrease factor.

$$th(t+1) = \begin{cases} th(t) + a, & \text{if convergence is achieved} \\ th(t) \times b, & \text{if convergence is not achieved} \end{cases} \quad (6.5)$$

The starting value of Th , a and b is something that is needed to be used based on analyzing quite a few number of configuration cost for several scenarios. Even though the value of a and b are static, it will lead the threshold value for better coherency of the system.

6.2.3 Extension of the model

By adding penalty and plane filtering mechanism our new algorithm provides more stability to the network system. It reduces convergence time by adding the penalty and reduce the possibility of incoherent planes by adding the filtering. Figure 6.7 provides an overlook of the overall architecture with newly added extensions. In the plane selection part, optimization by fitness function is described in chapter 5. That part is combined with the penalty analysis to provide a better topology for the new demand set. And secondly, the new planes are accepted and implemented by the controller. A plane filtering stage is added to avoid changes in the plane with low gain in terms of objective function or high configuration cost.

Figure 6.8, depicts the scenario where incoherence can be avoided. Figure 6.3 depicted an scenario where there was a possibility of an unstable system, whereas 6.8 shows the same scenario and how incoherence is avoided and possibility of an unstable system is reduced by adding penalty and filtering mechanism.

Next section describes the scenarios about how a plane is either accepted or rejected for increasing stability.

6.2.4 Case study

An update of a demand set can be considered as addition of a new demand, deletion of a current demand or change in the capacity requirement or any combination of the mentioned event. Every

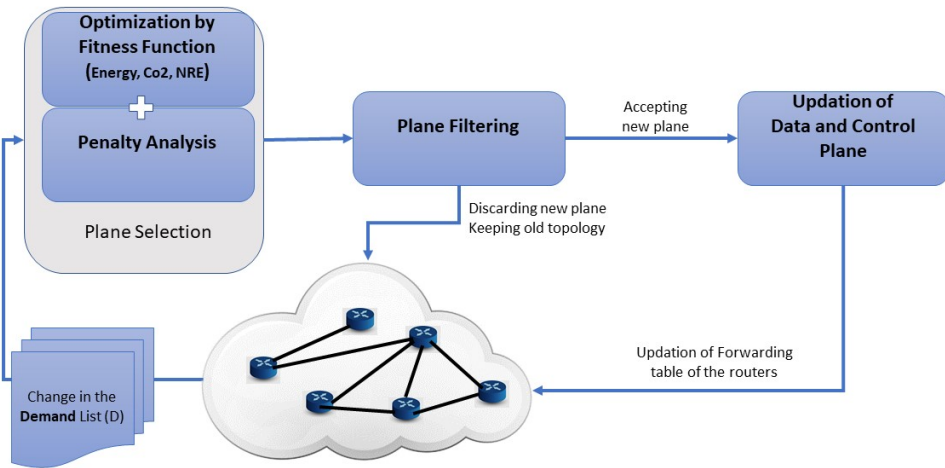


Figure 6.7: Overall system architecture with penalty and filtering mechanism

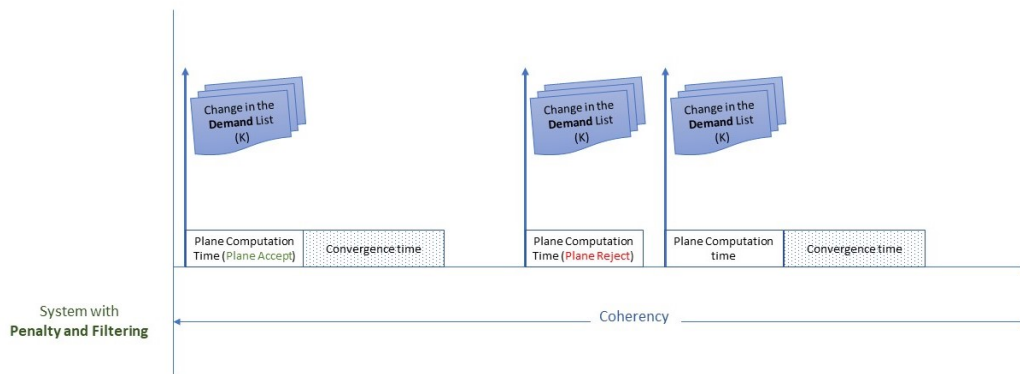


Figure 6.8: Avoidance of incoherence plane by penalty and filtering mechanism

time an updation of a demand set is occurred, a new plane is created and the decision is needed to be taken if this new plane will be implemented or not. As explained earlier, there are few scenarios where penalty and filtering can be applicable. Here we have described two particular cases with both accept and reject scenarios. However, according to filtering logic, one thing has to be common in all the scenario is that current data plane and forwarding table have to able to fulfill the new demand set.

Case 1: Deleting of demand(s) from demand set

The first case that we are discussing is deleting of a demand or multiple demands from the demand set. When few demands of the demand set is removed two things are obvious. Firstly, the data plane, control plane and forwarding tables definitely have more rules then it actually needed. regardless of that, both planes and forwarding tables are definitely able to fulfill new demand set's requirement. Secondly, Deletion of demands from the demand set is most of the time beneficial from green perspective. Few demand means possibility of fulfilling user requirement with less number of nodes and links. And current solution is most probably consuming more energy then the optimal one. Now let us provide an example, about how this new extension of the system will act in this kind of scenario. Let us consider an example with trimmed GÉANT topology with 12 nodes and 16 links that we have considered before. The figure 6.9 shows the current data plane (Π_0), control plane (Γ_0) and forwarding table of the nodes, as well as the path used for each demand. Currently we have a demand set of 8 demands. Where, $K = \{k_1 = \{s_1 = 2, d_1 = 8, \lambda_1 = 2Gbps\}, \{12, 10, 0.5Gbps\}, \{8, 4, 0.4Gbps\}, \{2, 12, 14Gbps\}, \{6, 1, 45Gbps\}, \{3, 4, 21Gbps\}, \{1, 8, 4Gbps\}, \{6, 3, 1Gbps\}\}$. And for this example, we have considered energy as our fitness function. But any fitness function would have similar outcome. And lets assume the current threshold value is 0.05.

Now, in the first scenario, lets assume, four demands $\{8, 4, 0.4Gbps\}$, $\{2, 12, 14Gbps\}$, $\{1, 8, 4Gbps\}$ and $\{6, 3, 1Gbps\}$ are deleted and the algorithm with penalty analysis, provide one of the best optimal solution in terms of configuration cost for the remaining four demands. Figure 6.10 shows the new optimal solution with new data plane (Π') and control plane (Γ') for the updated demand set. The figure also shows the expected topology and how it will look, if the plane filtering accept this solution. Now the configuration cost of this solution is:

$$\Omega' = \frac{\mathcal{U}'}{|\mathcal{E}_{\Pi', \Gamma'} - \mathcal{E}_{\Pi_0, \Gamma_0}|} = \frac{26}{348} = 0.074712644 \quad (6.6)$$

Configuration cost is calculated based on number of updates to make the change of the topology and how much gain we are getting compare to current topology. As this value is higher than the threshold value this new planes will be rejected by filtering and will not be applied by the controller. The main reason behind this is, even four demands are removed, the solution new

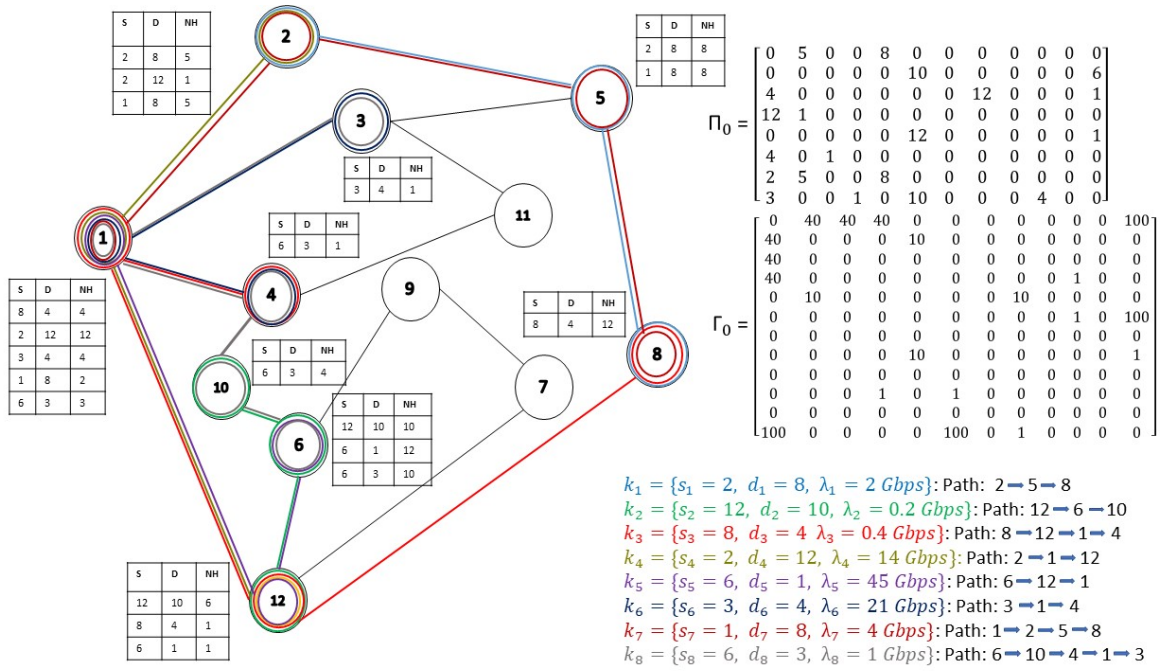


Figure 6.9: Current topology with data and control plane for case: 1

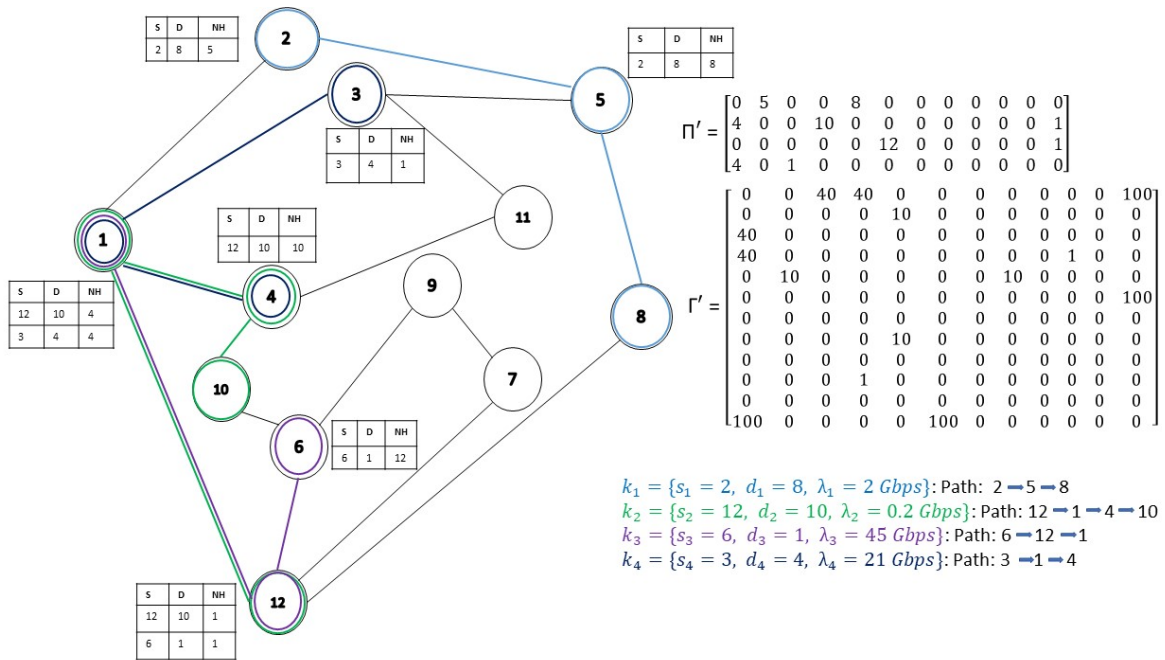
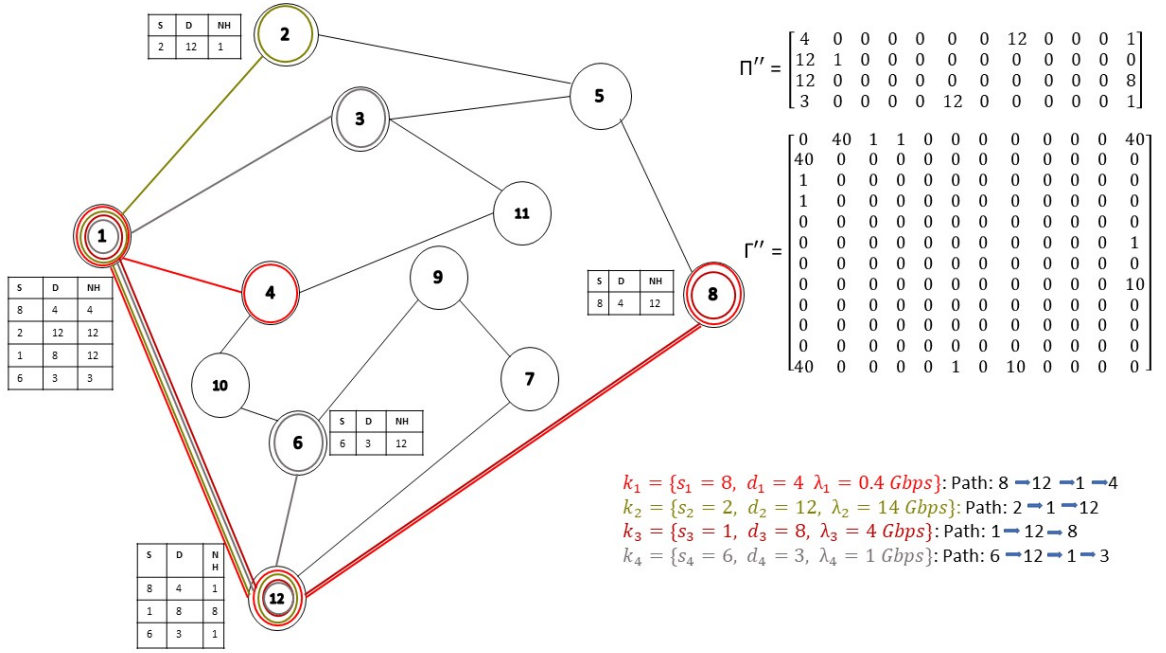


Figure 6.10: Rejected solution where $\Omega' > Th$ for case: 1

Figure 6.11: Accepted solution with $\Omega'' < Th$ for case: 1

topology does not save enough energy. But there are quite a number of changes in the planes and forwarding table from the current solution resulting configuration cost higher than the threshold value. Because of that, the new solution is rejected.

Now let's consider the second scenario. In this scenario, four different demands $\{2, 8, 2\text{Gbps}\}$, $\{12, 10, 0.2\text{Gbps}\}$, $\{6, 1, 45\text{Gbps}\}$, $\{3, 4, 21\text{Gbps}\}$ are removed from the list. And same as before, for the remaining four demands the algorithm with penalty analysis provides one of the best optimal solution with lower configuration cost. Figure 6.11 shows the new optimal solution with new data plane (Π'') and control plane (Γ'') for the updated demand set. The figure also shows the expected topology and how it will look, if the plane filtering accept this solution. Now the configuration cost of this solution is:

$$\Omega'' = \frac{\mathcal{U}''}{|\mathcal{E}_{\Pi'', \Gamma''} - \mathcal{E}_{\Pi_0, \Gamma_0}|} = \frac{31}{21828} = 0.001420194 \quad (6.7)$$

As we can see here the configuration cost of this new solution less then threshold value, therefore this solution will be selected. This new solution also has quite a few number of updation like previous scenario, however the energy gain between the current and new solution is quite substantial, therefore, this solution is selected.

This is one of the cases where plane filtering is applicable. One important thing to notice is

that, here we have depicted two general scenarios of accepting and rejecting new solution. But as we know, acceptance and rejection of new solution entirely depends on the threshold value, And the value of threshold is always changing based on the network state. For a particular time period, if the demand sets are frequently updating and multiple incoherence events are occurred in quick succession, then due to AIMD algorithm, the threshold of accepting solution will be quite low and during that time the accepted solution of the above scenario might not get accepted. Similarly, if the network does not faces incoherency for a long period of time, naturally the threshold will go high, and the solution of the first scenario might get accepted.

Case 2: Reducing of the capacity requirement of a demand set

The next case that we have discussed here is the reducing of the capacity requirement of a demand set. If the current topology serving a set of demands and at a particular time capacity requirement of a flow or multiple flows are reduced. One thing is sure that, current topology will be able to handle this update as there will be no change in the data plane. However, it is possible that, because of the few flows with large capacity requirement might turning on extra links and may be as well nodes to fulfill their requirement. Reducing of this kind of capacity requirement might open few options of substantial amount of savings. In this case also we have considered energy as our objective function and threshold value is 0.05. Now let us discuss two scenarios of accepting and rejecting new plane.

Now with the same topology of previous case, let us consider the current planes are dealing with a network with demand set of five demands where, $K = \{k_1 = \{s_1 = 3, d_1 = 6, \lambda_1 = 65Gbps\}, \{5, 10, 40Gbps\}, \{2, 12, 21Gbps\}, \{8, 3, 37Gbps\}, \{6, 2, 1Gbps\}\}$. As we can see few demands have large capacity requirement. The figure 6.12 shows the current data plane (Π_0), control plane (Γ_0) and forwarding table of the nodes, as well as the path used for each demand.

Now, in the first scenario, capacity requirement of demand-3 and 4 (k_3 and k_4) are reduced. λ_3 is reduced from 21Gbps to 1Gbps and λ_4 is reduced from 37Gbps to 14Gbps. Based on this updation algorithm provides the best solution given in figure 6.13 with new data plane (Π') and control plane (Γ').

Now, whereas, in the second scenario, capacity requirement of demand-1 and 2 (k_1 and k_2) are changed. λ_1 is reduced from 65Gbps to 40Gbps and λ_2 is reduced from 40Gbps to 20Gbps. Figure 6.14 shows the solution topology with new data plane (Π'') and control plane (Γ'') for this updated demand list generated by the algorithm. Now if we see the configuration cost for

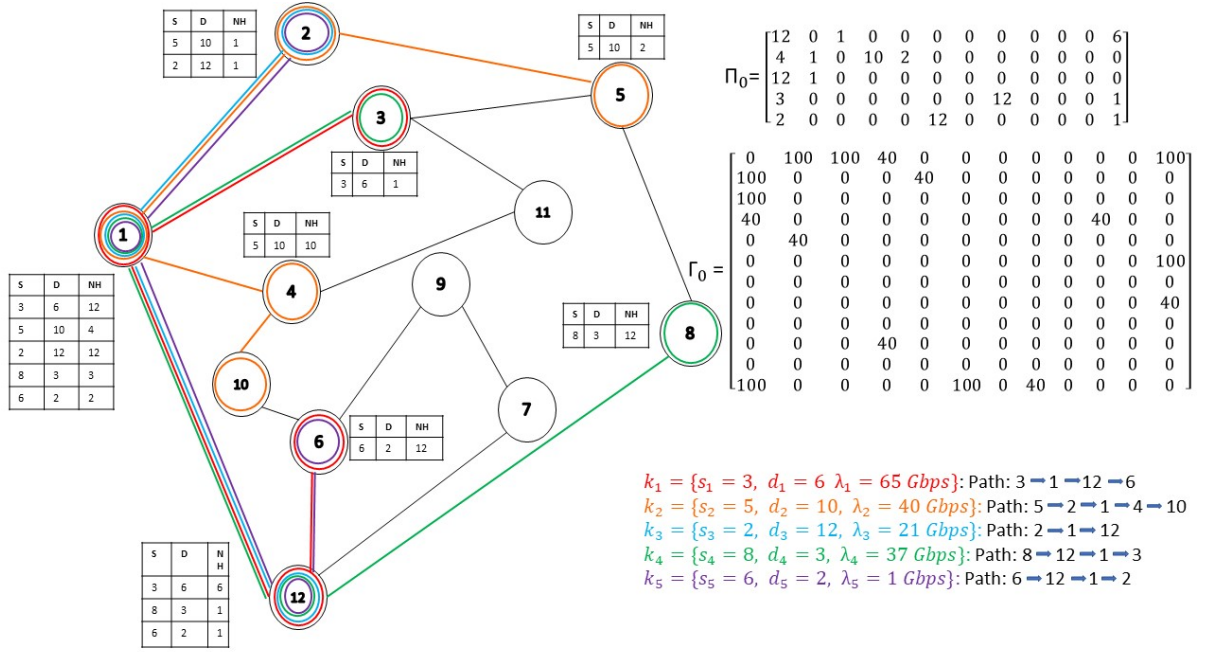


Figure 6.12: Current topology with data and control plane for case: 2

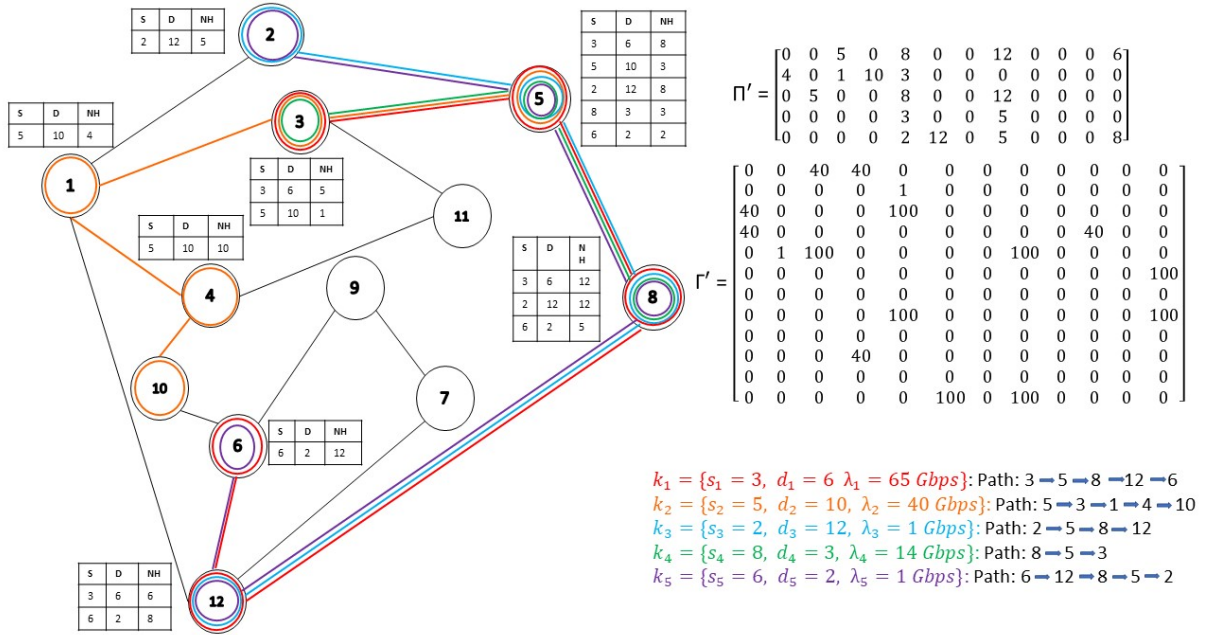
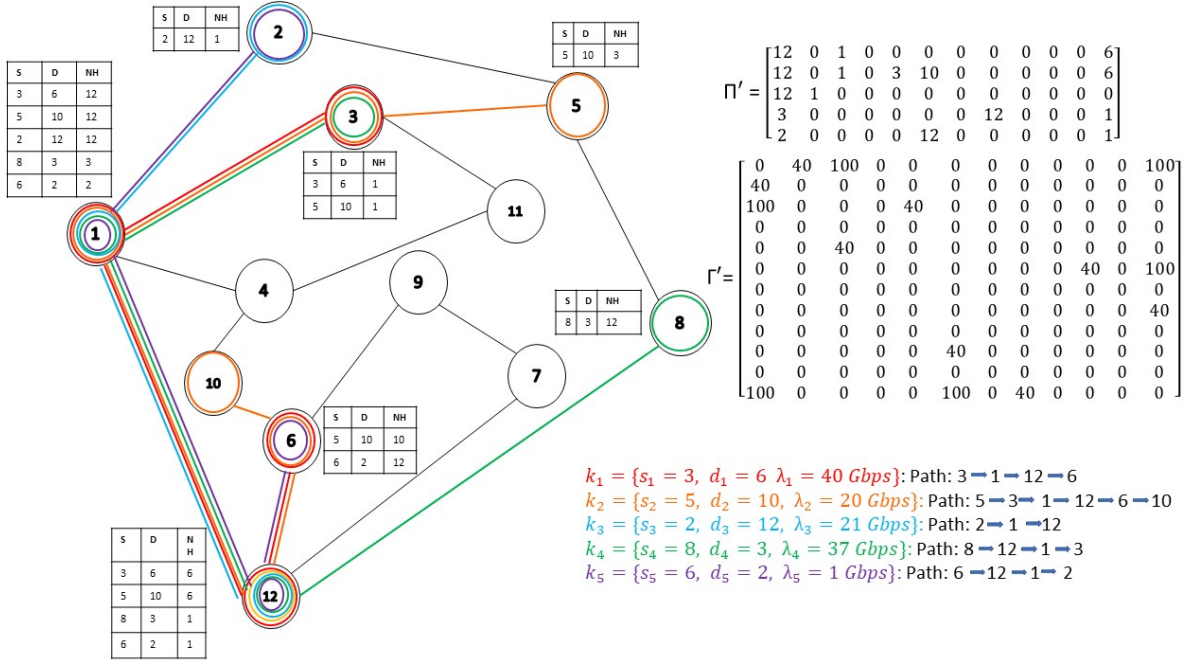


Figure 6.13: Rejected solution where $\Omega' > Th$ for case: 2


 Figure 6.14: Accepted solution with $\Omega'' < Th$ for case: 2

both scenarios,

$$\Omega' = \frac{\mathcal{U}'}{|\mathcal{E}_{\Pi', \Gamma'} - \mathcal{E}_{\Pi_0, \Gamma_0}|} = \frac{30}{306} = 0.098039216 \quad (6.8)$$

$$\Omega'' = \frac{\mathcal{U}''}{|\mathcal{E}_{\Pi'', \Gamma''} - \mathcal{E}_{\Pi_0, \Gamma_0}|} = \frac{15}{10720} = 0.001399254 \quad (6.9)$$

Like before, the first scenario is rejected and the second scenario is accepted. If we analyze the scenarios, in both cases two demands have reduced its capacity requirement and quantity wise reduction is quite similar. However, the solutions are quite different. Specially the accepted scenario is almost identical to the current scenario, and because of that number of updatation is quite low. In fact second scenarios configuration cost is almost 70 times higher. If the first scenario had same amount of updatation as second scenario, then even with this savings, it would have been selected. And lastly, like previous discussion, the value of threshold depends on the current state of the network, at any different scenario acceptance and rejection of the new planes might be completely different from this scenario.

6.3 Conclusion

This chapter addresses system stability which is a major concern before implementing a solution. The idea developed in this chapter is very unique and to our knowledge, this is the lone contribution considering stability for controlling green routing in core network. This chapter also opens up few important topic that are needed to be addressed. From the overall study, one thing is sure to be mentioned is that, an over provisioned system with every link and nodes are in use is most of the time stable, as controller does not need to make any substantial changes into the forwarding table. Only change might be required, when a link has reached to its maximum capacity usage and a new path is needed to be assigned to the upcoming demands which are using that link. In other words, solution of stability is over provisioning. And as we can see in our two case study, we have rejected two new solutions which clearly had energy gain compare to current solution. The algorithm chooses an over provisioned solution in order to improve stability. So, it can be said that, any green networking algorithm increases the possibility of a network being unstable. Therefore, it is important to integrate stability issue into the objective function in order to find a more optimal solution regarding both green efficiency and stability.

Secondly, as discussed before, due to huge size of GÉANT network, dedicated link from controller to each node seems illogical. Therefore, hybrid network is needed to be considered. Which begs another important concern that can expressed with the expression of, "Do not chop a branch you're sitting on". That means, while routing of the demand flows we need to also take of the control message flows. Before turning off a link or nodes it should be checked that if that node or link is used for sending a control message from controller to some node or not. Moreover, routing of this message flows are also needed to be addressed in the forwarding table.

Lastly, in our algorithm we have used AIMD algorithm to adjust the threshold value. However, it does not take into account the frequency of change of the demand set which is the main reason behind an incoherent system. AIMD merely adjust the value of threshold by seeing the outcome. However, using machine learning technique if the incoming demand set can be predicted and then that value is adjusted to the threshold value then a much more stable system can be achieved. It will lead to a full autonomic network management which is reaching the concept of Intelligent Defined Networking (IDN) [Georges, 2019]. Currently both demand flow requirement and when it will arrive is unknown, but if some sort of prediction can be done, then pro-active measures can be taken for reducing consumption as well as increasing stability and enhancing users quality of experience (QoE).

Conclusion and Perspective

Contents

7.1	Main outcome	122
7.2	Perspectives	123
7.2.1	Variability of the static part of energy model	123
7.2.2	Implementing the solution in LAN context	124
7.2.3	Applying the solution in end to end network	124
7.2.4	Temporal analysis for environmental factors	124
7.2.5	Implementation challenges of SDN platform	125
7.2.6	Different heuristics for solving the problem	125
7.2.7	Intelligence defined network for increasing stability and QoE	126

Reducing the environmental impact of modern communication networks has been identified as an ever-increasingly vital need, given the notable energy consumption by the ICT sector. With growing users' needs, networking models should evolve not only in terms of service provisioning but in the capability to improve energy efficiency and mitigate environmental impact by reducing CO_2 emissions. Acceptable solutions to this challenge are required in order to provide the green support to the network. Throughout this thesis, we have addressed some of the most challenging issues regarding the design of energy-aware routing solutions by reducing the number of active network elements and implementing the adaptive link rate in SDN. It should be highlighted that all the contributions of this research stand as novel and original proposals which were designed from scratch. This thesis is focused on green routing and has proposed a few methods for reducing the environmental impact of wide-range core networks. We here briefly describe the main outcomes of this research in order to assess the fulfillment of research objectives. In addition, we discuss about the possible directions in which our work can be extended.

7.1 Main outcome

In this thesis, our main focus has been reducing the environmental impact of the data transmission network of core networks. For this purpose, at first, we analyzed an existing core network, GÉANT and its traffic behavior to point out the energy saving scope and also we have figured out the green policies that will be most beneficial. We have proposed a non complex algorithm which considered the network constraints and showed the vast possibility of saving energy. The solution has considered only links as energy reducing element and only policy is considered is turning off of unused links. This algorithm has provided enough insight for designing an algorithm in order to maximize savings in SDN perspective. Therefore, in the following section, we have provided a full model of a solution which has considered a centralized approach for improving the energy efficiency of a network. We have designed data and control plane based on our two selected green policies: shutting down of both unused nodes and links and secondly, the use of adaptive link rate. We have used genetic algorithm as a heuristic function. For fine-tuning of the parameters of the genetic algorithm, we have also conducted a design of experiment, an extensive set of experiments to identify the optimal parameter values for the experiment. We have introduced traffic coefficients based on the traffic increasing pattern and our algorithm is tested with this large demand load to see the robustness of the algorithm.

Next, we have introduced two environmental parameters namely, carbon emission factor and non-renewable energy usage percentage. At the same time, we have introduced a new routing technique which is pollution aware routing based on the environmental parameters. This approach, unlike energy aware routing, does not focus on energy saving rather than it focuses on reducing the overall CO_2 emission or non-renewable energy usage. After that, we provided a model of a sustainable system respecting all three pillars of sustainability that consider both energy aware and pollution aware routing and offer a more holistic solution in terms of quality in sustainability. The result shows that, depending on the objective function the savings and environmental impact can differ a lot. At the same time, it provides an answer to the second research question, showing the importance of introducing environmental parameters. The performance of each parameter is also evaluated and analyzed separately in terms of quality of service performance metrics like end to end delay and packet loss.

In the later part, we have introduced the penalty analysis and plane filtering mechanism into the system, in order to reduce the possibility of an unstable system. We have analyzed different issues regarding stability like coherency and convergence of the plane and provided these two mechanisms to cope with the scenario. We also provided an extended version of the model and how these mechanisms can be implemented. Penalty analysis and plane filtering help the system to avoid unstable scenarios and provide the answer to our third research question. We have also explained the case scenarios where these mechanisms will be effective in order to have a better grasp of the concept.

In general, solutions proposed along this manuscript hold the potential to deliver substantial reduction of the environmental impact of the data transmission network for their deployment in real-world networks while considering important features like realistic demand size and stability concerns. Extensive simulation experiments demonstrate the viability and efficacy of proposed approaches. These findings provide a new direction of sustainable networking of large backbone networks using the SDN architecture.

7.2 Perspectives

In this work, we have proposed a novel system for sustainable networking. This work leads to some new directions that can be explored in the near future. As well as there is room for further improvement in our work. Here, overall perspectives are described.

7.2.1 Variability of the static part of energy model

One thing that is considered by us and also by all the state of the art works is the static part of an energy model. The static part of an energy model contains the energy consumption of the node (i.e.: switch or router). And as traffic load has almost no impact on the energy consumption of the node it is always considered as a static portion of the energy model. However, there are few things that should be considered which might change the course of modeling of the energy equation. In a core network like GÉANT, each node represents a different country. Now, a node comprises of power supply, cooling device, forwarding engine, switching fabric, control planes, input/output cards, etc. Now every product produced by different companies as well as it might have different settings and capacity. For example, while analyzing the traffic as we can see, France and German are responsible for the largest portion of the overall traffic whereas Greece is generating quite less amount of traffic. So, it is quite possible and safe to assume that the devices that are used by Greece might be quite different compared to France. Moreover, there are also issues of new vs old products. An updated energy efficient product might consume much less energy compared to an old product. Lastly, there are also cooling techniques and the countries' geographical position. As we explained in chapter two, power supply and cooling are responsible for more than 33% of the overall power consumption of a network device (node), so a country like Sweden will spend much less in terms of cooling compare to Spain due to their geographical position. The same theory is applicable for other core networks, for example, in Abilene network, every node is in different states of United States of America. Therefore, it will be unfair to assume that every node consumes the same amount of energy. A research could be conducted based on the different factors of the network devices and can be added to the static part of the current energy model to provide a more realistic and efficient result.

7.2.2 Implementing the solution in LAN context

Our solution is completely focused on greening the routing of the core network. However, our first algorithm concerning only energy can also be implemented in any local area network of an enterprise, data-center or even in a campus network. It considers two green policies namely shutting down of unused network devices and link rate adaptation. In [Rondeau *et al.*, 2017] we have explained that, by simply cabling optimization (i.e.: serialization) we can achieve a significant amount of energy efficiency. This algorithm can be tested against the finding of our previous work. Our work can also be compared with the current state of the art solution for the local area network. This will at the same time test the robustness of the algorithm.

7.2.3 Applying the solution in end to end network

Our overall sustainable routing solution is entirely focused on the core network routing regardless of the host. However, we could apply our algorithm holistically by considering the end nodes into our equation. For example, a service provider can have several data-centers around the globe to accommodate its user. When a user requests a service, based on different QoS parameters such as: the load of the data-center, distance between the user and the data-center, etc. a data-center is chosen to serve that particular demand. And while the traffic flow is running through the network to fulfill this demand, our algorithm provides the optimal sustainable routing solution. However, our algorithm currently does not consider which data-center to choose for improving green networking. Therefore, an extended version of our algorithm can be implemented where the green performance of the end nodes (i.e.: Data-centers) will be considered. Based on the location of the data-center the overall carbon emission or non-renewable energy usage can vary a lot. So, by considering the end nodes alongside with networking devices could improve the sustainability of the network significantly.

7.2.4 Temporal analysis for environmental factors

In this work, we have introduced two environmental metrics, carbon emission factors and non-renewable energy usage percentages. For pollution aware routing we use these metrics as a multiplicative factor of the energy for finding out the best routing path. However, the values that we have considered are static and fixed for each country. However, in reality, it is not the case. During different time period of a day, electricity production technique can vary and so does environmental metrics. For example, a country might use solar energy for producing electricity on day time and coal for the night. Similarly, another country might use coal during the peak of energy demands for compensating the missing of electricity produced by nuclear plants. This mixing of electricity production technique can significantly impact on the environmental factors that are needed to be considered. These values should be in real-time while making routing

decisions in order to have better solutions. Moreover, there are few renewable energy production techniques which are dependent on the weather. For example, solar power will produce less energy during a cloudy day. Therefore, temporal analysis of these environmental parameters should be integrated into the system in order to have a better environmental impact. Moreover, alongside with real time weather information, we can also introduce weather forecast as our input parameters. The availability of any weather dependent power source can be known beforehand and a pro-active decision can be made based on that. This kind of temporal analysis of the environmental factors will also have an impact on the energy policy strategy of a country which may have a direct impact on their digital business when considering sustainability.

7.2.5 Implementation challenges of SDN platform

Our solution provides an algorithm that can be implemented any centralized controlled or SDN enabled system. However, there are few issues regarding the implementation of the algorithm in the real-life environment. The first thing that is needed to consider is the placement of the controller. As we described before, due to the geographical coverage of GÉANT network we have considered that, the controller will use the current network for communicating with the nodes. That means all the control messages will be added to the demand list for routing. Therefore, it is crucial to place the controller in a way so that the number of control messages can be reduced. At the same time, the carbon emission factor and non-renewable energy usage percentage should be considered while placing the controller as the controller also has an environmental impact. Moreover, if a hybrid network is used for both routing traffic flows and control messages, then while removing the links and nodes we have only considered the demand flows but the control messages should also be considered. All these issues should be implemented on SDN enabled platform and evaluate the change in the performance of the algorithm if any.

7.2.6 Different heuristics for solving the problem

For solving our sustainable routing problem, we have used genetic algorithm as our heuristic. However, there are several other heuristic algorithm like genetic algorithm which might provide different result. By including environmental parameters into our problem formalization we have introduced a new dimension in green routing. Moreover, when stability analysis is also added, it makes the problem even more unique. Therefore, it is important to test with other heuristic to see if better performance can be achieved or not. There are quite a few bio-mimic optimization algorithms that can be used. For example: particle swarm optimization (PSO), ant colony optimization algorithm (ACO) or artificial bee colony algorithm (ABC). All these algorithms are evolutionary algorithms that iteratively try to improve a candidate solution with regard to a given measure of objective function. They have been used for several optimizing problems including routing. Therefore, it could be interesting to apply some of these algorithms for solving

our problem and compare the result with our genetic algorithm.

7.2.7 Intelligence defined network for increasing stability and QoE

In chapter six, we have introduced the penalty and filtering mechanism. But as we have discussed there, the stability of the system largely depends on the incoming traffic flows and the frequency of it. Even with penalty and filtering mechanisms, there are scenarios where incoherent planes cannot be avoided. Therefore, one solution could be to predict the incoming traffic flow and proactively make the change in the system according to that to avoid instability completely. For this purpose, intelligent defined network (IDN) concept can be introduced where based on the machine learning technique system will be trained by the previous traffic pattern so that it can forecast a change in the traffic before it's occurrence. Even though, the prediction might be wrong in some cases, but including this traffic prediction alongside with penalty and filtering mechanism will definitely reduce the percentage of a system being unstable by a large margin which will also improve the quality of service. Moreover, there is another matter of concern that can also be tried to solve using IDN, which is the green cost of the solution. While optimizing algorithm is running to find out the best possible solution for the green routing, it is also consuming energy. Therefore, for stopping criterion of genetic algorithm, instead of time, the green cost of the solution can be introduced. So that, the computational green cost will not lessen the impact of the actual solution. A machine learning technique can be introduced to predict the probability of finding a better solution after a certain number of iteration. So, if the probability of finding a better solution is low, then the algorithm will be stopped and the green cost of the overall algorithm will be reduced.

A

Table A.1 shows the national characteristics in terms of carbon emission factor and non-renewable energy usage factor.

Node No.	Country	CEF	NRE Factor
0	Austria	0.176	0.257
1	Belgium	0.224	0.834
2	Poland	1.196	0.863
3	Czech Republic	0.938	0.85
4	German	0.672	0.71
5	Spain	0.342	0.619
6	Switzerland	0.003	0.602
7	Greece	1.921	0.726
8	Croatia	0.386	0.348
9	Hungary	0.589	0.899
10	Ireland	0.521	0.753
11	Israel	0.74	0.975
12	Italy	0.41	0.327
13	Luxembourg	0.276	0.792
14	Netherlands	0.413	0.879
15	United States	0.547	0.72
16	France	0.07	0.825
17	Portugal	0.4	0.465
18	Sweden	0.023	0.429
19	Slovenia	0.578	0.694
20	Slovakia	0.282	0.755
21	United Kingdom	0.508	0.721

Table A.1: List of nodes of the GÉANT network with carbon and non-renewable energy factors

B

Figure B.1 shows the obtained results for each countries' consumption of energy CO2 and non-renewable energy for our scenario.

Country	Algorithm			SPF			GA_E			GA_CO ₂			GA_NRE		
	Energy (Watt)	CO ₂ (gCO ₂ /kWh)	NRE (Watt)	Energy (Watt)	CO ₂ (gCO ₂ /kWh)	NRE (Watt)	Energy (Watt)	CO ₂ (gCO ₂ /kWh)	NRE (Watt)	Energy (Watt)	CO ₂ (gCO ₂ /kWh)	NRE (Watt)	Energy (Watt)	CO ₂ (gCO ₂ /kWh)	NRE (Watt)
Austria	10354	1822.30	2660.97	10194	1794.14	2619.85	10102	1777.95	2596.21	10194	1794.14	2619.85	10194	1794.14	2619.85
Belgium	10160	2275.84	8473.44	10194	2283.45	8501.79	10160	2275.84	8473.44	10194	2283.45	8501.79	10194	2283.45	8501.79
Poland	10720	12821.1	9251.36	10720	12821.1	9251.36	0	0	0	0	0	0	0	0	0
Czech Republic	10428	9781.46	8863.8	10294	9655.77	8749.9	10394	9749.57	8834.9	10394	9749.57	8834.9	10394	9749.57	8834.9
German	10616	7133.95	7537.36	10480	7042.56	7440.8	10788	7249.53	7659.48	11200	7526.4	7952	11200	7526.4	7952
Spain	10014	3424.78	6198.66	0	0	0	10014	3424.78	6198.66	10014	3424.78	6198.66	10014	3424.78	6198.66
Switzerland	0	0	0	0	0	0	10354	31.062	4162.30	0	0	0	0	0	0
Greece	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Croatia	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hungary	10068	5930.05	9051.13	0	0	0	0	0	0	0	0	0	0	0	0
Ireland	10068	5245.42	7581.20	0	0	0	0	0	0	0	0	0	0	0	0
Israel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Italy	10167	4168.47	6374.70	10160	4165.6	6370.32	10167	4168.47	6374.70	10167	4168.47	6374.70	10167	4168.47	6374.70
Luxembourg	10034	2769.38	7946.92	10034	2769.38	7946.92	10034	2769.38	7946.92	10034	2769.38	7946.92	10034	2769.38	7946.92
Netherlands	10327	4265.05	9077.43	10320	4262.16	9071.28	0	0	0	10320	4262.16	9071.28	10068	5507.19	8588.00
United States	10167	5561.34	8672.45	10068	5507.19	8588.00	10068	5507.19	8588.00	10068	5507.19	8588.00	10068	5507.19	8588.00
France	10068	704.76	8306.1	0	0	0	0	0	0	10388	727.16	8570.1	0	0	0
Portugal	10041	4016.4	4669.06	10034	4013.6	4665.81	10014	4005.6	4656.51	10014	4005.6	4656.51	10014	4005.6	4656.51
Sweden	10428	239.844	4473.61	10394	239.062	4459.02	10394	239.062	4459.02	10394	239.062	4459.02	10394	239.062	4459.02
Slovenia	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slovakia	10068	2839.17	7601.34	10034	2829.58	7575.67	10034	2829.5	7575.67	10034	2829.5	7575.67	10034	2829.5	7575.67
United Kingdom	10116	5138.9	7293.63	10102	5131.816	7283.542	10075	5118.1	7264.075	10075	5118.1	7264.075	10075	5118.1	7264.075

Figure B.1: Countrywise energy, CO2 and non-renewable energy consumption values

C

List of Publications

Articles en revue internationale

Md Mohaimenul Hossain, Jean-Philippe Georges, Eric Rondeau, Thierry Divoux. "Energy, Carbon and Renewable Energy: Candidate Metrics for Green-aware Routing", *Sensors*, 19 (13), pp.2901. (10.3390/s19132901)

Communications dans des conférences internationales avec actes

Md Mohaimenul Hossain, Eric Rondeau, Jean-Philippe Georges, Thierry Bastogne. "Modeling the power consumption of Ethernet switch", International SEEDS Conference 2015: Sustainable Ecological Engineering Design for Society, Sep 2015, Leeds, UK

Eric Rondeau, Jean-Philippe Georges, **Md Mohaimenul Hossain**, Habibullah Khan Mohammad. "Is Green Networking a new workload for ICT engineers? A network topology example", 20th IFAC World Congress, IFAC 2017, Jul 2017, Toulouse, France

Atefeh Maleki, **Md Mohaimenul Hossain**, Jean-Philippe Georges, Eric Rondeau, Thierry Divoux. "An SDN Perspective to Mitigate the Energy Consumption of Core Networks – GÉANT2", International SEEDS conference 2017, Sep 2017, Leeds, UK

Md Mohaimenul Hossain, Jean-Philippe Georges, Eric Rondeau, Thierry Divoux. "Using SDN for Controlling the Carbon Footprint of The Internet", (Submitted to 21th IFAC World Congress, IFAC 2020, Berlin, Germany)

Communications nationales sans actes

Md Mohaimenul Hossain, Jean-Philippe Georges, Eric Rondeau, Thierry Divoux. “Greening the Internet by building sustainable network infrastructure using SDN”, SDN Day 2018, Nov 2018, Paris, France

Bibliography

- [Addis *et al.*, 2014] Bernardetta Addis, Antonio Capone, Giuliana Carello, Luca G. Gianoli, and Brunilde Sanso. Energy Management Through Optimized Routing and Device Powering for Greener Communication Networks. *IEEE/ACM Trans. Netw.*, 22(1):313–325, February 2014.
- [Agency, 2019] International Energy Agency. World Energy Investment 2019. <https://www.iea.org/wei2019/>, 2019.
- [Alena and Libor, 2012] Buchalcevova Alena and Gala Libor. Green ICT adoption survey focused on ICT life-cycle from the consumer’s perspective (SMEs). *Journal of competitiveness*, 4(4):109–122, 2012.
- [Andreopoulou *et al.*, 2014] Zacharoula Andreopoulou, Emmanouil Stiakakis, and Maro Vlachopoulou. Green Ict applications towards the achievement of Sustainable Development. In *E-innovation for sustainable development of rural resources during global economic crisis*, pages 11–21. IGI Global, 2014.
- [Assefa and Ozkasap, 2017] B. G. Assefa and O. Ozkasap. Link utility and traffic aware energy saving in software defined networks. In *2017 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 1–5, June 2017.
- [Assefa and Özkasap, 2019] Beakal Gizachew Assefa and Öznur Özkasap. A survey of energy efficiency in SDN: Software-based methods and optimization models. *Journal of Network and Computer Applications*, 137:127–143, 2019.
- [Avallone and Ventre, 2012] Stefano Avallone and Giorgio Ventre. Energy efficient online routing of flows with additive constraints. *Computer Networks*, 56(10):2368–2382, 2012.
- [Ba *et al.*, 2018] J. Ba, Y. Wang, X. Zhong, S. Feng, X. Qiu, and S. Guo. An SDN energy saving method based on topology switch and rerouting. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–5, April 2018.
- [Barnhart *et al.*, 2009] Cynthia Barnhart, Niranjan Krishnan, and Pamela H. Vance. Multicommodity flow problems. *Encyclopedia of Optimization*, pages 2354–2362, 2009.

- [Bianzino *et al.*, 2010] Aruna Prem Bianzino, Claude Chaudet, Federico Larroca, Dario Rossi, and J. Rougier. Energy-aware routing in IP networks: A reality check. In *IEEE, GLOBECOM Workshops*, pages 1422–1427, 2010.
- [Blanquicet and Christensen, 2007] Francisco Blanquicet and Ken Christensen. An Initial Performance Evaluation of Rapid PHY Selection (RPS) for Energy Efficient Ethernet. *32nd IEEE Conference on Local Computer Networks (LCN 2007)*, 2007.
- [Blanquicet and Christensen, 2008] Francisco Blanquicet and Ken Christensen. Managing energy use in a network with a new SNMP power state MIB. In *2008 33rd IEEE Conference on Local Computer Networks (LCN)*, pages 509–511. IEEE, 2008.
- [Bolla *et al.*, 2009] Raffaele Bolla, Roberto Bruschi, Franco Davoli, and Andrea Ranieri. Performance constrained power consumption optimization in distributed network equipment. In *Green Communications Workshop, Dresden, Germany, 2009*.
- [Bolla *et al.*, 2011] Raffaele Bolla, Roberto Bruschi, Franco Davoli, and Flavio Cucchietti. Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures. *IEEE Communications Surveys & Tutorials*, 13(2):223–244, 2011.
- [Bolla *et al.*, 2013] R. Bolla, R. Bruschi, F. Davoli, L. Di Gregorio, P. Donadio, L. Fialho, M. Collier, A. Lombardo, D. Reforgiato Recupero, and T. Szemethy. The Green Abstraction Layer: A Standard Power-Management Interface for Next-Generation Network Devices. *IEEE Internet Computing*, 17(2):82–86, March 2013.
- [Brander *et al.*, 2011] Matthew Brander, Aman Sood, Charlotte Wylie, Amy Haughton, and Jessica Lovell. Technical Paper| Electricity-specific emission factors for grid electricity. *Ecometrica, Emissionfactors.com*, 2011.
- [Chang Wook Ahn and Ramakrishna, 2002] Chang Wook Ahn and R. S. Ramakrishna. A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation*, 6(6):566–579, December 2002.
- [Chiaraviglio *et al.*, 2011] Luca Chiaraviglio, Marco Mellia, and Fabio Neri. Minimizing ISP network energy cost: Formulation and solutions. *IEEE/ACM Transactions on Networking*, 20(2):463–476, 2011.
- [Cianfrani *et al.*, 2010] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini. An Energy Saving Routing Algorithm for a Green OSPF Protocol. In *2010 INFOCOM IEEE Conference on Computer Communications Workshops*, pages 1–5, March 2010.

-
- [Cianfrani *et al.*, 2011] Antonio Cianfrani, Vincenzo Eramo, Marco Listanti, and Marco Polverini. An OSPF enhancement for energy saving in IP networks. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 325–330, Shanghai, China, April 2011. IEEE.
- [Cianfrani *et al.*, 2012] Antonio Cianfrani, Vincenzo Eramo, Marco Listanti, Marco Polverini, and Athanasios V. Vasilakos. An OSPF-integrated routing strategy for QoS-aware energy saving in IP backbone networks. *IEEE Transactions on Network and Service Management*, 9(3):254–267, 2012.
- [Cisco, 2012] Cisco. White paper on Global IP Traffic Forecast and Methodology, 2006–2011. http://www.hbtf.org/files/cisco_IPforecast.pdf, 2012.
- [Cisco, 2015] Cisco. VNI Global Fixed and Mobile Internet Traffic Forecast. <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html>, 2015.
- [Corcoran and Andrae,] Peter Corcoran and Anders Andrae. Emerging Trends in Electricity Consumption for Consumer ICT. page 56.
- [Costa *et al.*, 2012] C. H. A. Costa, M. C. Amaral, G. C. Januário, T. C. M. B. Carvalho, and C. Meirosu. SustNMS: Towards service oriented policy-based network management for energy-efficiency. In *2012 Sustainable Internet and ICT for Sustainability (SustainIT)*, pages 1–5, October 2012.
- [Cuomo *et al.*, 2011a] F. Cuomo, A. Abbagnale, and S. Papagna. ESOL: Energy saving in the Internet based on Occurrence of Links in routing paths. In *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–6, June 2011.
- [Cuomo *et al.*, 2011b] Francesca Cuomo, Anna Abbagnale, Antonio Cianfrani, and Marco Polverini. Keeping the connectivity and saving the energy in the internet. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 319–324. IEEE, 2011.
- [Cuomo *et al.*, 2012] Francesca Cuomo, Antonio Cianfrani, Marco Polverini, and Daniele Mangione. Network pruning for energy saving in the Internet. *Computer Networks*, 56(10):2355–2367, 2012.
- [Drouant *et al.*, 2014] Nicolas Drouant, Eric Rondeau, Jean-Philippe Georges, and Francis Lepage. Designing green network architectures using the ten commandments for a mature ecosystem. *Computer Communications*, 42:38–46, April 2014.

- [(EEA), 2016] European Environment Agency (EEA). Overview of electricity production and use in Europe, 2016.
- [Eiben and Smit, 2012] A. E. Eiben and S. K. Smit. Evolutionary Algorithm Parameters and Methods to Tune Them. In Youssef Hamadi, Eric Monfroy, and Frédéric Saubion, editors, *Autonomous Search*, pages 15–36. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [Europa, 2013] Europa. European Commission - PRESS RELEASES - Press release - Digital Agenda: global tech sector measures its carbon footprint. https://europa.eu/rapid/press-release_IP-13-231_en.htm, 2013.
- [Fernandez-Fernandez *et al.*, 2016] Adriana Fernandez-Fernandez, Cristina Cervello-Pastor, and Leonardo Ochoa-Aday. Achieving Energy Efficiency: An Energy-Aware Approach in SDN - IEEE Conference Publication. IEEE, December 2016.
- [Fithritama *et al.*, 2015] A. Fithritama, E. Rondeau, V. Bombardier, and J. Georges. Modeling fuzzy rules for managing power consumption of ethernet switch. In *2015 23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 42–47, September 2015.
- [Galán-Jiménez, 2018] Jaime Galán-Jiménez. Exploiting the control power of SDN during the transition from IP to SDN networks. *International Journal of Communication Systems*, 31(5):e3504, 2018.
- [Gattulli *et al.*, 2014] M. Gattulli, M. Tornatore, R. Fiandra, and A. Pattavina. Low-Emissions Routing for Cloud Computing in IP-over-WDM Networks with Data Centers. *IEEE Journal on Selected Areas in Communications*, 32(1):28–38, January 2014.
- [Geant, 2017] Geant. GÉANT Project Home. <https://geant3plus.archive.geant.net/Pages/default.aspx>, 2017.
- [Georges, 2019] Jean-Philippe Georges. Network Automation: Opportunities for Industrial Internet of Things – IFAC Blog. <http://blog.ifac-control.org/network-automation/network-automation-opportunities-for-industrial-internet-of-things/>, 2019.
- [Google, 2013] Google. Data center efficiency – Data Centers – Google. <https://www.google.com/about/datacenters/efficiency/>, 2013.
- [Gunaratne and Christensen, 2006] Chamara Gunaratne and Ken Christensen. Ethernet adaptive link rate (alr): Analysis of a buffer threshold policy. In *Proceedings of IEEE GLOBECOM*, 2006.

-
- [Gunaratne *et al.*, 2006] C. Gunaratne, K. Christensen, and S. W. Suen. NGL02-2: Ethernet Adaptive Link Rate (ALR): Analysis of a Buffer Threshold Policy. In *IEEE Globecom 2006*, pages 1–6, November 2006.
- [Gunaratne *et al.*, 2008] Chamara Gunaratne, Kenneth Christensen, Bruce Nordman, and Stephen Suen. Reducing the energy consumption of Ethernet with adaptive link rate (ALR). *IEEE Transactions on Computers*, 57(4):448–461, 2008.
- [Gupta and Singh, 2003] Maruti Gupta and Suresh Singh. Greening of the Internet. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, pages 19–26, New York, NY, USA, 2003. ACM. event-place: Karlsruhe, Germany.
- [Gupta and Singh, 2007a] Maruti Gupta and Suresh Singh. Dynamic ethernet link shutdown for energy conservation on ethernet links. In *2007 IEEE International Conference on Communications*, pages 6156–6161. IEEE, 2007.
- [Gupta and Singh, 2007b] Maruti Gupta and Suresh Singh. Using low-power modes for energy conservation in Ethernet LANs. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pages 2451–2455. Citeseer, 2007.
- [Gupta *et al.*, 2004] Maruti Gupta, Satyajit Grover, and Suresh Singh. A Feasibility Study for Power Management in LAN Switches. In *ICNP*, volume 4, pages 361–371. Citeseer, 2004.
- [Hakiri *et al.*, 2014] Akram Hakiri, Aniruddha Gokhale, Pascal Berthou, Douglas C. Schmidt, and Thierry Gayraud. Software-Defined Networking: Challenges and research opportunities for Future Internet. *Computer Networks*, 75:453–471, December 2014.
- [Hallam *et al.*, 2010] Joshua W. Hallam, Olcay Akman, and Füsün Akman. Genetic algorithms with shrinking population size. *Computational Statistics*, 25(4):691–705, December 2010.
- [Hansmann *et al.*, 2012] R. Hansmann, H. A. Mieg, and P. Frischknecht. Principal sustainability components empirical analysis of synergies between the three pillars of sustainability, June 2012.
- [Haudebourg and Orgerie, 2017] Timothée Haudebourg and Anne-Cécile Orgerie. On the Energy Efficiency of Sleeping and Rate Adaptation for Network Devices. In *ICA3PP 2017 - 17th International Conference on Algorithms and Architectures for Parallel Processing*, volume 10393, pages 132–146, Helsinki, Finland, August 2017. Springer.
- [Heller *et al.*, 2010] Brandon Heller, Srinivasan Seetharaman, Priya Mahadevan, Yiannis Yiakoumis, Puneet Sharma, Sujata Banerjee, and Nick McKeown. Elastictree: Saving energy in data center networks. In *Nsdi*, volume 10, pages 249–264, 2010.

- [Hilty *et al.*, 2011] Lorenz Hilty, Wolfgang Lohmann, and Elaine M. Huang. Sustainability and ICT - An overview of the field. *Notizie di Politeia*, 27(104):13–28, June 2011.
- [Hinton *et al.*, 2011] Kerry Hinton, Jayant Baliga, Michael Feng, Robert Ayre, and Rodney S. Tucker. Power Consumption and Energy Efficiency in the Internet. *IEEE Network*, 25(2):6 - 12, May 2011.
- [Hossain *et al.*, 2015] Md Mohaimenul Hossain, Eric Rondeau, Jean-Philippe Georges, and Thierry Bastogne. Modeling the power consumption of ethernet switch. In *International SEEDS Conference 2015: Sustainable Ecological Engineering Design for Society*, 2015.
- [Hossain *et al.*, 2019] Md Mohaimenul Hossain, Jean-Philippe Georges, Eric Rondeau, and Thierry Divoux. Energy, Carbon and Renewable Energy: Candidate Metrics for Green-Aware Routing? *Sensors*, 19(13):2901, January 2019.
- [IEA, 2016] IEA. Digitalization and Energy. <https://webstore.iea.org/digitalization-and-energy>, 2016.
- [IPCC, 2011] IPCC. IPCC, 2011: IPCC Special Report on Renewable Energy Sources and Climate Change Mitigation. Prepared by Working Group III of the Intergovernmental Panel on Climate Change. <https://www.mcc-berlin.net/en/research/publications/publications-detail/article/ipcc-2011-ipcc-special-report-on-renewable-energy-sources.html>, 2011.
- [IPCC, 2019] IPCC. Cisco Visual Networking Index: Forecast and Trends, 2017–2022. <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf>, 2019.
- [Irish and Christensen, 1998] Liam Irish and Kenneth J. Christensen. A “Green TCPDP ” to Reduce Electricity Consumed by Computers. *Proceedings IEEE Southeastcon’98’Engineering for a New Era’*. IEEE, 1998.
- [Jeurissen, 2000] Ronald Jeurissen. Cannibals With Forks: The Triple Bottom Line of 21st Century Business. *Journal of Business Ethics*, 23(2):229–231, January 2000.
- [Jo *et al.*, 2018] S. Jo, L. Wang, J. Kangasharju, and M. Muehlhaeuser. Cost-Effective and Eco-Friendly Green Routing Using Renewable Energy. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8, July 2018.
- [Kajikawa, 2008] Yuya Kajikawa. Research core and framework of sustainability science. *Sustainability Science*, 3(2):215–239, October 2008.

-
- [Krommenacker *et al.*, 2002] N. Krommenacker, E. Rondeau, and T. Divoux. Genetic algorithms for industrial ethernet network design. In *4th IEEE International Workshop on Factory Communication Systems*, pages 149–156, August 2002.
- [Lannoo *et al.*, 2012] Bart Lannoo, Contributors Bart Lannoo, Sofie Lambert, Ward Van Heddeghem, Mario Pickavet (iminds, O. Kuipers (tudelft, George Koutitas, Harris Niavis, Anna Satsiou (certh, Michael Till, Andreas Fischer, Hermann De Meer (uni Passau, Paul Alcock (ulanc, Nguyen Hoang Viet, Thomas Plagemann (uio, Javier Aracil (uam, and Ros Tassiulas Certh. *D8.1: Overview of ICT energy consumption*. 2012.
- [Li *et al.*, 2017] Wenjun Li, Xianfeng Li, and Hui Li. MEET-IP: Memory and energy efficient TCAM-based IP lookup. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8. IEEE, 2017.
- [Maksimovic, 2018] M. Maksimovic. Greening the future: Green Internet of Things (G-IoT) as a key technological enabler of sustainable development. In *Internet of things and big data analytics toward next-generation intelligence*, pages 283–313. Springer, 2018.
- [Maleki *et al.*, 2017] Atefeh Maleki, Md Mohaimenul Hossain, Jean-Philippe Georges, Eric Rondeau, and Thierry Divoux. An SDN Perspective to Mitigate the Energy Consumption of Core Networks – GÉANT2. September 2017.
- [Manjate *et al.*, 2018] J. A. Manjate, M. Hidell, and P. Sjödin. Can Energy-Aware Routing Improve the Energy Savings of Energy-Efficient Ethernet? *IEEE Transactions on Green Communications and Networking*, 2(3):787–794, September 2018.
- [Matsuura, 2013] H. Matsuura. Energy-saving routing algorithm using Steiner tree. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pages 378–386, May 2013.
- [Mineraud *et al.*, 2016] J. Mineraud, L. Wang, S. Balasubramaniam, and J. Kangasharju. Hybrid renewable energy routing for ISP networks. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.
- [Mostowfi and Christensen, 2011] M. Mostowfi and K. Christensen. Saving energy in LAN switches: New methods of packet coalescing for Energy Efficient Ethernet. In *2011 International Green Computing Conference and Workshops*, pages 1–8, July 2011.
- [Mumey *et al.*, 2012a] B. Mumey, J. Tang, and S. Hashimoto. Enabling green networking with a power down approach. In *2012 IEEE International Conference on Communications (ICC)*, pages 2867–2871, June 2012.

- [Mumey *et al.*, 2012b] Brendan Mumey, Jian Tang, and Saiichi Hashimoto. Enabling green networking with a power down approach. In *2012 IEEE International Conference on Communications (ICC)*, pages 2867–2871. IEEE, 2012.
- [Muyassarov, 2018] Feruzjon Muyassarov. Network Function Virtualization Deployment Scenario Dimensioning. Master’s thesis, 2018.
- [Nedevschi *et al.*, 2008] Sergiu Nedevschi, Lucian Popa, Gianluca Iannaccone, Sylvia Ratnasamy, and David Wetherall. Reducing Network Energy Consumption via Sleeping and Rate-adaptation. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI’08, pages 323–336, Berkeley, CA, USA, 2008. USENIX Association. event-place: San Francisco, California.
- [Orgerie *et al.*, 2014] Anne-Cecile Orgerie, Marcos Dias de Assuncao, and Laurent Lefevre. A Survey on Techniques for Improving the Energy Efficiency of Large-scale Distributed Systems. *ACM Comput. Surv.*, 46(4):47:1–47:31, March 2014.
- [Pawlish and Aparna, 2010] Michael Pawlish and S. Varde Aparna. Free cooling: A paradigm shift in data centers. In *2010 Fifth International Conference on Information and Automation for Sustainability*, pages 347–352. IEEE, 2010.
- [Rafique *et al.*, 2017] Yousef Rafique, Mohamad Khattar Awad, and Ghadeer Neama. A benchmark implementation for evaluating the performance of power-aware routing algorithms in practical software-defined networks. In *2017 Fourth International Conference on Software Defined Systems (SDS)*, pages 118–124. IEEE, 2017.
- [Rahnamay-Naeini *et al.*, 2016] Mahshid Rahnamay-Naeini, Sonali Sen Baidya, Ehsan Siavashi, and Nasir Ghani. A traffic and resource-aware energy-saving mechanism in software defined networks. In *2016 International Conference on Computing, Networking and Communications (ICNC)*, pages 1–5. IEEE, 2016.
- [Rawat and Bajracharya, 2016] D. B. Rawat and C. Bajracharya. Software Defined networking for reducing energy consumption and carbon emission. In *SoutheastCon 2016*, pages 1–2, March 2016.
- [Rondeau and Lepage, 2010] Eric Rondeau and Francis Lepage. Vers une ingénierie de réseaux respectueuse de l’environnement. In *Sixième Conférence Internationale Francophone d’Automatique, CIFA 2010*, page CDROM, Nancy, France, June 2010.
- [Rondeau *et al.*, 2015] Eric Rondeau, Francis Lepage, Jean-Philippe Georges, and Gérard Morel. Measurements and Sustainability. In Colin Pattinson Mohammad Dastbaz and Babak Akhgar,

-
- editors, *Green Information Technology. A Sustainable Approach*, pages 29–59. Elsevier, March 2015.
- [Rondeau *et al.*, 2017] Eric Rondeau, Jean-Philippe Georges, Md Mohaimenul Hossain, and Habibullah Khan Mohammad. Is Green Networking a new workload for ICT engineers? A network topology example. In *20th IFAC World Congress, IFAC 2017*, Toulouse, France, July 2017.
- [Ruiz-Rivera *et al.*, 2015] Alejandro Ruiz-Rivera, Kwan-Wu Chin, and Sieteng Soh. GreCo: An energy aware controller association algorithm for software defined networks. *IEEE communications letters*, 19(4):541–544, 2015.
- [Schlömer *et al.*, 2014] S. Schlömer, T. Bruckner, L. Fulton, E. Hertwich, A. McKinnon, D. Perczyk, J. Roy, R. Schaeffer, R. Sims, P. Smith, and R. Wiser. *Climate Change 2014 Mitigation of Climate Change: Working Group III Contribution to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, 2014.
- [Shuja *et al.*, 2017] Junaid Shuja, Raja Wasim Ahmad, Abdullah Gani, Abdelmuttlib Ibrahim Abdalla Ahmed, Aisha Siddiqa, Kashif Nisar, Samee U. Khan, and Albert Y. Zomaya. Greening emerging IT technologies: techniques and practices. *Journal of Internet Services and Applications*, 8(1):9, 2017.
- [Silva *et al.*, 2017] L. Silva, P. Gonçalves, R. Marau, and P. Pedreiras. Extending OpenFlow with industrial grade communication services. In *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, pages 1–4, May 2017.
- [Singh and Chandwani, 2015] D. Singh and G. Chandwani. Carbon-aware routing in software defined inter data center network. In *2015 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6, December 2015.
- [SNDlib, 2008] SNDlib. Sndlib—library of test instances for survivable fixed telecommunication network design. <http://sndlib.zib.de>, 2008.
- [Srikantaiah *et al.*, 2008] Shekhar Srikantaiah, Aman Kansal, and Feng Zhao. Energy aware consolidation for cloud computing. 2008.
- [Thang Nguyen Bui and Byung Ro Moon, 1996] Thang Nguyen Bui and Byung Ro Moon. Genetic algorithm and graph partitioning. *IEEE Transactions on Computers*, 45(7):841–855, July 1996.
- [Van Heddeghem and Idzikowski, 2012] Ward Van Heddeghem and Filip Idzikowski. Equipment power consumption in optical multilayer networks-source data. *Photonic Network Comm*, 2012.

- [Van Heddeghem *et al.*, 2012] Ward Van Heddeghem, Filip Idzikowski, Willem Vereecken, Didier Colle, Mario Pickavet, and Piet Demeester. Power consumption modeling in optical multilayer networks. *Photonic Network Communications*, 24(2):86–102, October 2012.
- [Wang *et al.*, 2008] Yi Wang, Eric Keller, Brian Biskeborn, Jacobus Van Der Merwe, and Jennifer Rexford. Virtual routers on the move: live router migration as a network-management primitive. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 231–242. ACM, 2008.
- [Wang *et al.*, 2012] Xiaofei Wang, Athanasios V. Vasilakos, Min Chen, Yunhao Liu, and Ted Taekyoung Kwon. A survey of green mobile networks: Opportunities and challenges. *Mobile Networks and Applications*, 17(1):4–20, 2012.
- [Wang *et al.*, 2015] Huandong Wang, Yong Li, Depeng Jin, Pan Hui, and Jie Wu. Saving energy in partially deployed software defined networks. *IEEE Transactions on Computers*, 65(5):1578–1592, 2015.
- [Wibowo *et al.*, 2017] Franciscus XA Wibowo, Mark A. Gregory, Khandakar Ahmed, and Karina M. Gomez. Multi-domain software defined networking: research status and challenges. *Journal of Network and Computer Applications*, 87:32–45, 2017.
- [Zane *et al.*, 2003] Francis Zane, Girija Narlikar, and Anindya Basu. CoolCAMs: Power-efficient TCAMs for forwarding engines. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, volume 1, pages 42–52. IEEE, 2003.
- [Zekauskas *et al.*, 1999] Matthew J. Zekauskas, Sunil Kalidindi, and Guy Almes. A One-way Delay Metric for IPPM. Technical report, 1999.
- [Zhang *et al.*, 2010] Mingui Zhang, Cheng Yi, Bin Liu, and Beichuan Zhang. GreenTE: Power-aware Traffic Engineering. In *Proceedings of the The 18th IEEE International Conference on Network Protocols, ICNP '10*, pages 21–30, Washington, DC, USA, 2010. IEEE Computer Society.
- [Zhou *et al.*, 2013] Zhi Zhou, Fangming Liu, Yong Xu, Ruolan Zou, Hong Xu, John C.S. Lui, and Hai Jin. Carbon-Aware Load Balancing for Geo-distributed Cloud Services. In *2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 232–241, San Francisco, CA, USA, August 2013. IEEE.

Résumé

Le monde actuel reposant sur une économie et des activités personnelles de plus en plus digitalisées a pour conséquence que la quantité d'information circulant à travers internet ne cesse de croître d'année en année. Les architectures de communication doivent être dimensionnées en conséquence, voire surdimensionnées pour anticiper les besoins futurs ou pour pallier des pics de trafic ou encore des défaillances éventuelles. A cela s'ajoute des systèmes de refroidissement pour garantir le bon fonctionnement des équipements d'interconnexion. Aussi, actuellement les réseaux représentent 20% de la consommation d'énergie de la sphère numérique sachant que si cette sphère était un pays, elle sera au cinquième rang des pays en terme de consommation d'énergie. Il est donc essentiel de trouver des stratégies d'optimisation considérant à la fois cette consommation tout en maintenant la Qualité de Services exigée par les utilisateurs de l'internet. Cette thèse propose tout d'abord un premier un algorithme de routage permettant de réduire l'électricité nécessaire pour alimenter le réseau en utilisant deux mécanismes : l'endormissement des nœuds et l'adaptation des débits des lignes de transmission. Le problème étant NP difficile, des algorithmes génétiques sont employés. Les résultats montrent l'intérêt de cette approche sur le réseau Européen Géant-2 et cela en se reposant sur des architectures SDN (Software Defined Network). Cependant, une optimisation basée uniquement sur l'électricité peut être une vue tronquée sur l'impact environnemental de l'internet. Dans la suite de cette thèse, le type d'énergie utilisé pour produire l'électricité a été pris en compte pour analyser le routage selon deux autres indicateurs : le taux d'énergie renouvelable et le taux d'émission carbone. Les algorithmes ainsi développés sont de nouveau appliqués à Géant-2 pour illustrer l'incidence de ces nouveaux indicateurs sur le routage. Finalement, la reconfiguration du plan de contrôle du réseau doit se faire au rythme de l'évolution du trafic. Mais cela peut générer une instabilité du système de communication et donc une rupture de Qualité de Service. Aussi, des mécanismes de filtrage et de pénalité sont introduits dans les précédents algorithmes pour éliminer des changements intempestifs de reconfiguration néfastes au maintien global de la performance du réseau.

Mots-clés: Réseau à faible empreinte environnementale, Routage durable, Plan de contrôle et données, Métriques environnementales

Abstract

Achieving energy efficiency has in recent times become a major concern of networking research due to the ever-escalating power consumption and CO₂ emissions produced by large data networks. This problem is becoming more and more challenging because of the drastic traffic increase in the last few years and it is expected to be increased even more in the coming years. Using efficient energy-aware strategies that could overturn this situation by reducing the electricity consumption as well as mitigating the environmental impact of data transmission networks. However, CO₂ and energy consumption cannot be considered proportionate if the means of electricity production differs. This research work focuses on reducing the environmental impact of data transmission network by implementing energy aware routing, where unused network devices will be put into sleep/shut down and high capacity links will be adapted according to demand requirement. But, alongside with energy, this work has introduced two different metrics namely carbon emission factor and non-renewable energy usage percentage, which are considered as objective functions for designing green network. Here a centralized approach like using Software-Defined Networking (SDN), is used for designing to solve this problem as they allow flexible programmability suitable for this problem. Our proposal proposes a routing technique using genetic algorithm that minimizes the number of network-elements required and at the same time adapt the bandwidth capacity while satisfying an incoming traffic load. Different from existing related works, we focus on optimizing not only energy consumption but also carbon emission and non-renewable energy consumption in SDN in order to close this important gap in the literature and provide solutions compatible with operational backbone networks. Complementing the general aim of improving the environmental impact of data transmission network, this research is also intended to cover important related features such as realistic large demand size, network performance, and Quality of Service (QoS) requirements. At the same time this work focuses on network stability and analyzes the impact of network stability while implementing a green solution. Our work proposes a penalty and filtering mechanism which helps to find an optimal balance between stability and green networking. By using realistic input data, significant values of switched-off links and nodes are reached which demonstrate the effectiveness of our algorithm. The obtained result validated the importance of considering environmental factors rather than considering only energy. Results also show the trade-off between environmental and performance concerns, considering a couple of performance indicators. Moreover, it is shown that the penalty and filtering mechanism is an effective approach to avoid incoherent system and improve the stability of the system. As a whole, this conducted research and contributions reported through this manuscript stand as a valuable solution on the road to sustainable networking.

Keywords: green networking, pollution aware routing, data and control plane, environmental metrics

Titre de la thèse :

Métriques environnementales pour des réseaux de communication plus écoresponsables

(Green Metrics to Improve Sustainable Networking)

Présenté par M. Md Mohaimenul Hossain

Pour l'obtention du Doctorat de l'Université de Lorraine

Mention Automatique, Génie Informatique.

Sous la direction de :

Professeur Eric Rondeau et Professeur Thierry Divoux

1. Introduction

Le monde actuel reposant sur une économie et des activités personnelles de plus en plus digitalisées a pour conséquence que la quantité d'information circulant à travers internet ne cesse de croître d'année en année. Les architectures de communication doivent être dimensionnées en conséquence, voire surdimensionnées pour anticiper les besoins futurs ou pour pallier des pics de trafic ou encore des défaillances éventuelles. Aussi, actuellement les réseaux représentent 20% de la consommation d'énergie de la sphère numérique sachant que si cette sphère était un pays, elle sera au cinquième rang des pays en terme de consommation d'énergie. Il est donc essentiel de trouver des stratégies d'optimisation considérant à la fois cette consommation tout en maintenant la Qualité de Services exigée par les utilisateurs de l'internet. Cette thèse propose tout d'abord un premier algorithme de routage permettant de réduire l'énergie nécessaire pour alimenter le réseau en utilisant deux mécanismes : l'endormissement des nœuds et l'adaptation des débits des lignes de transmission. Le problème étant NP difficile, des algorithmes génétiques sont employés. Les résultats montrent l'intérêt de cette approche sur le réseau Européen Géant-2 et cela en se reposant sur des architectures SDN (Software Defined Network). Cependant, une optimisation basée uniquement sur l'énergie peut être une vue tronquée sur l'impact environnemental de l'internet. Dans la suite de cette thèse, la manière de produire l'électricité a été prise en compte pour analyser le routage selon deux autres indicateurs : le taux d'énergie renouvelable et le taux d'émission carbone. Les algorithmes ainsi développés sont de nouveau appliqués à Géant-2 pour illustrer l'incidence de ces nouveaux indicateurs sur le routage. Finalement, la reconfiguration du plan de contrôle du réseau doit se faire au rythme de l'évolution du trafic. Mais cela peut générer une instabilité du système de communication et donc une rupture de Qualité de Service. Aussi, des mécanismes de filtrage et de pénalité sont introduits dans les précédents algorithmes pour éliminer des

changements intempestifs de reconfiguration néfastes au maintien global de la performance du réseau.

2. Etat de l'art

La communauté scientifique traitant des réseaux a déjà investigué cette problématique appelée « green networking ». Il existe notamment de nombreuses contributions sur l'optimisation de l'acheminement des messages en prenant en compte son coût/consommation énergétique ou encore le taux d'énergie renouvelable. Certains travaux proposent pour ce faire d'utiliser des techniques d'endormissement des nœuds d'interconnexion, ou des techniques d'ajustement des débits lien en fonction de la bande passante utilisée (ALR : Adaptive Link Rate) ou encore de se positionner dans un contexte SDN pour contrôler de façon centralisée les décisions amenant à rendre plus propre les architectures de communication.

L'originalité de nos travaux est de proposer un algorithme définissant le plan de contrôle en fonction de l'énergie consommée, ou du taux d'énergie renouvelable ou encore de l'émission carbone inhérente à la production d'électricité. Ce dernier facteur n'est pas actuellement abordé dans le domaine du routage des réseaux alors qu'il doit être considéré pour limiter le niveau de pollution de l'air et plus globalement le réchauffement/dérèglement climatique. L'algorithme proposé s'exécutera de manière centralisée à l'aide de contrôleurs SDN. Une autre originalité de notre travail est d'étudier les problèmes de stabilité de mise à jour du plan de contrôle.

3. Etude préliminaire

En guise d'étude préliminaire, pour montrer le gain potentiel de consommation d'énergie d'un cœur de réseau en agissant sur le routage des messages, nous avons étudié Géant 2. Géant 2 est le réseau à très haut débit destiné à la recherche et à l'éducation en Europe. Nous avons analysé des enregistrements de trafic réel sur Géant 2 et nous avons appliqué deux algorithmes SPF (Shared Path First) et SPFS (Shared Path First Sorted) dont l'objectif est de choisir des chemins déjà utilisés pour acheminer de nouveaux messages de manière à limiter le nombre de liens à utiliser. Ces techniques de concentration de trafic ont permis de montrer que l'on pouvait désactiver jusqu'à 41% de liens et que la bande passante pour chaque lien pouvait être réduite à 10Gb/s alors que les configurations initiales sur Géant 2 sont de 40Gb/s

4. Formalisation du problème et développement d'un algorithme pour un plan de contrôle peu énergivore.

L'objectif est d'améliorer les algorithmes SPF et SPFS en introduisant comme données paramétriques la possibilité de mettre en sommeil des nœuds et d'ajuster le débit des liens (ALR) pour ainsi adapter l'énergie en fonction de l'usage, c'est-à-dire des données à transporter.

La fonction à optimiser est la suivante, avec Π le plan de données et Γ le plan de contrôle :

$$\left(\hat{\Pi}, \hat{\Gamma}\right) = \arg \min_{\Pi, \Gamma} \int_0^t \sum_{v \in V} \delta_{\Pi}^v \left(\alpha_v + \sum_{w \in V} \delta_{\Pi}^{v,w} \beta_{v,w} (\Gamma_{v,w}) \right) t \cdot dt \quad (1)$$

Pour trouver la meilleure solution, l'algorithme doit parcourir l'ensemble des solutions pour appliquer la fonction d'optimisation (1). Le problème est NP difficile au vu du nombre de demandes qui doivent traverser une architecture de communication complexe. Nous avons donc décidé d'utiliser les Algorithmes Génétiques (AG) comme heuristique pour trouver une solution satisfaisante à notre problème. Le travail a consisté dans un premier temps à définir un chromosome permettant de coder pour chaque demande son chemin (figure 1).

Chromosome																																															
Demand 1																Demand 2																Demand 3															
L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0

Figure 1. Liens (L) utilisés pour chaque demande

Puis la seconde tâche a été de trouver les bons paramètres à utiliser pour l'AG. Un plan d'expérience a été conduit pour trouver le meilleur réglage de ces paramètres. La figure 2 montre les résultats de cette étude.

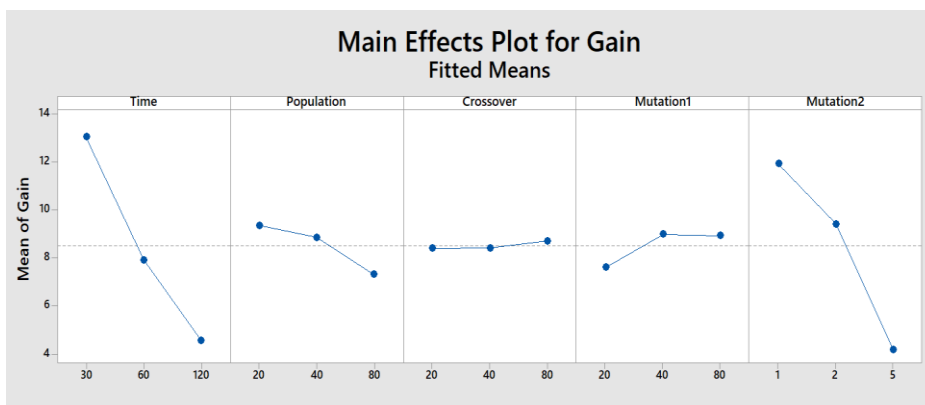


Figure 2. Réglage des paramètres de l'AG : Durée : 30s, Population : 20, Croisement : 20%, Mutation 1 : 20%, Mutation 2 : 1%

L'AG a ensuite été testé sur deux scénarii de demandes sur Géant 2 avec un premier scénario considérant beaucoup de demandes (400) et l'autre peu de demandes (25). Le tableau 1 montre que l'AG permet de réduire jusqu'à 34% la consommation d'énergie par rapport à SPF et que l'efficacité de l'AG pour minimiser la consommation d'énergie dépend du nombre de demandes. Un réseau fortement utilisé ne laisse pas de possibilité d'endormissement de nœuds ou de liens.

	Scenario 1 (400 Demands)			Scenario 2 (25 Demands)		
	Energy (MWh)	Cost(Euro)	Co2 (Tonnes)	Energy (MWh)	Cost(Euro)	Co2(Tonnes)
Current Infrastructure	2154	246016	594	2154	246016	594
SPF	2006	229053	553	1324	151217	365
GA	1987	226881	548	881	100651	243

Tableau 1. Analyse énergie/coût/Co2 sur un an pour les deux scénarii en utilisant SPF ou GA

5. Energie, Energie verte, Emission carbone : Métriques pour le green networking

Dans la section 4, la fonction d'optimisation reposait sur la consommation d'énergie permettant aussi de déterminer le coût et l'émission carbone (voir tableau 1). Dans la suite de nos travaux, le but a été d'introduire dans la fonction objectif soit le taux d'énergie verte utilisé, soit l'émission carbone émise tout en gardant la structure de l'AG. La fonction d'optimisation est donc maintenant :

$$(\hat{\Pi}, \hat{\Gamma}) = \underset{\hat{\Pi}, \hat{\Gamma}}{\operatorname{argmin}} \int_0^t \sum_{v \in V} \Phi_v * \delta_{\Pi}^v \left(\alpha_v + \sum_{w \in V} \delta_{\Pi}^{v,w} \beta_{v,w}(\Gamma_{v,w}) \right) t. dt \quad (2)$$

Le paramètre Φ a été ajouté à l'équation (1) pour représenter le critère à optimiser (Energie, Energie verte, Carbone).

(2) a été testé de nouveau sur le même scénario que précédemment (avec 25 demandes). La figure 3 montre les résultats obtenus et confirme évidemment que les solutions trouvées dépendent du critère sélectionné. Notamment, le critère carbone permet de réduire très significativement l'émission carbone et de rendre donc moins polluant le réseau.

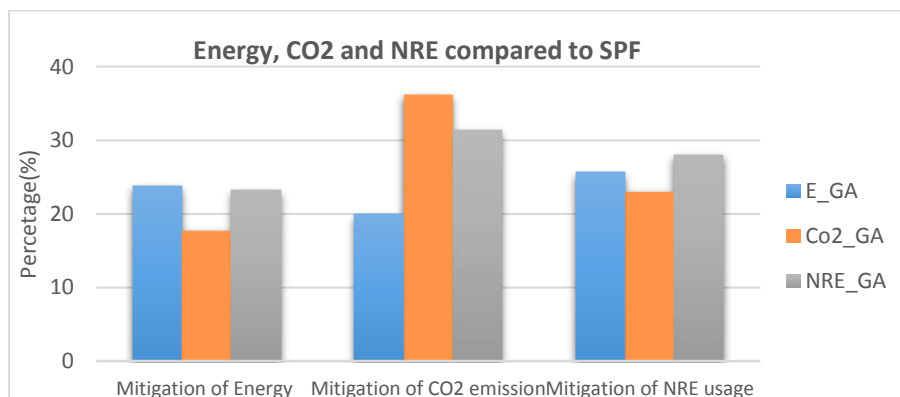


Figure 3. Comparaison des gains suivant les critères : Energie, Energie verte, Carbone.

6. Mécanismes pour éviter les mises à jour intempestives du plan de contrôle.

L'AG est un processus qui doit continuellement être exécuté pour prendre en compte les changements de topologie du réseau et les fluctuations du trafic de façon à trouver les meilleures solutions courantes. Si le contrôleur SDN régénère à chaque nouvelle solution un nouveau plan de contrôle, il peut en résulter une instabilité dans les phases successives de reconfiguration du réseau. Pour éviter ce problème, il est important d'analyser le bénéfice de chaque nouvelle solution obtenue par rapport à la précédente pour évaluer l'intérêt de changer le plan de contrôle. Pour ce faire, le concept de pénalité est introduit et est associé à la fonction d'optimisation :

$$\left(\hat{\Pi}, \hat{\Gamma}\right) = \arg \min_{\Pi, \Gamma} \int_0^t \sum_{v \in V} \Phi_v \times \delta_{\Pi}^v \left(\alpha_v + \sum_{w \in V} \delta_{\Pi}^{v,w} \beta_{v,w}(\Gamma_{v,w}) \right) t \cdot dt + \Omega(\Pi, \Gamma) \quad (3)$$

Dans (3), Ω représente le coût de reconfiguration et permet ainsi de savoir si la nouvelle solution a un réel intérêt par rapport au plan de contrôle actuel. Ce coût (4) peut être relatif à l'énergie, le taux d'énergie verte ou au carbone exprimé dans ε :

$$\Omega = \frac{\mathcal{U}}{|\mathcal{E}_{\Pi', \Gamma'} - \mathcal{E}_{\Pi_0, \Gamma_0}|} \quad (4)$$

7. Conclusions et perspectives

Les travaux conduits dans cette thèse ont permis de formaliser et de développer des mécanismes de mise à jour du plan de contrôle d'un réseau SDN pour acheminer les demandes en tenant en compte à la fois des exigences de qualité de services et de durabilité. Les résultats montrent que notre approche amène à une réduction importante de l'impact de l'Internet sur l'environnement. Une optimisation relative au taux d'émission carbone, souvent négligée dans la littérature, montre un intérêt indéniable par rapport aux décisions prises dans le cadre des conférences internationales sur le climat (COP : Conferences of Parties). Finalement, notre démarche propose des mécanismes de seuil pour éviter des changements non contrôlés de reconfiguration du réseau qui pourrait aboutir à une dégradation du service.

Les perspectives à ces travaux sont de trois niveaux. Le premier niveau serait d'importer des paramètres contextuels (externes) plus fins dans la fonction d'optimisation comme l'évolution dynamique de l'émission carbone et le taux d'énergie verte pour produire de l'électricité. Le coût de l'électricité pourrait aussi être un critère à considérer. Le deuxième niveau serait de tester nos algorithmes dans des environnements de communication différents comme dans les réseaux d'entreprises, les campus Universitaires, les réseaux urbains ou encore d'avoir une approche de bout en bout, c'est-à-dire en intégrant dans la fonction d'évaluation les coûts environnementaux des serveurs d'applications. Finalement, le troisième niveau consisterait à développer des algorithmes intelligents (dans une couche IDN : Intelligence Defined Network) en incluant des prédictions sur l'usage du réseau mais aussi sur les coûts environnementaux de façon à améliorer les prises de décisions lors des changements du plan de contrôle.

8. Bibliographie

- M. M. Hossain, J.P. Georges, E. Rondeau, T. Divoux, Energy, Carbon and Renewable Energy: *Candidate Metrics for Green-aware Routing?* Sensors 2019, 19 (13), 2901. 30 June 2019.

- M. M. Hossain, E. Rondeau, J.-P. Georges, T. Bastogne, *Modeling the power consumption of Ethernet Switch*, International SEEDS Conference 2015: Sustainable Ecological Engineering Design for Society, 17-18 Septembre 2015, Leeds, UK.
- E. Rondeau, JP. Georges, Md M. Hossain, H. K. Mohammad, *Is Green Networking a new workload for ICT engineers? A network topology example*, 20th IFAC World Congress, IFAC 2017, Jul 2017, Toulouse, France. 2017
- Maleki, M. Hossain, JP Georges, E. Rondeau, T. Divoux, *An SDN Perspective to Mitigate the Energy Consumption of Core Networks – GÉANT2*, 3rd International Sustainable Ecological Engineering Design for Society (SEEDS) Conference, 13-14 Sept. 2017, Leeds, UK
- Md M. Hossain, E. Rondeau, J.P. Georges, T. Divoux, *Greening The Internet By Building Sustainable Network Infrastructure using SDN*, SDNDay'18, Paris, 22 Nov. 2018