Universitat Politècnica de Catalunya (UPC)

Castelldefels School of Telecommunications and Aerospace Engineering

Department of Network Engineering

Energy-Aware Routing Techniques for Software-Defined Networks

Author: Adriana Fernández-Fernández

Ph.D. Advisor: Cristina Cervelló-Pastor

Thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Network Engineering

Barcelona, June 2018

Focus on doing the 20% of work that will yield you 80% of the results.

Pareto Principle

Abstract

Achieving energy efficiency has recently become a key topic of networking research due to the ever-increasing power consumption and CO_2 emissions generated by large data networks. This problem is becoming even more concerning and challenging given the drastic traffic increase expected over the next few years. However, the use of efficient energy-aware strategies could overturn this situation reducing the electricity consumption of Internet data transmission networks, as well as contributing to mitigate the environmental impact of other sectors.

The existence of redundant network elements with high capacities is a common design practice in current network infrastructures in order to face suddenly failures or peak traffic flows. However, these additional resources remain either unused or barely used most of the time leading to an undesired energy waste. Therefore, putting into sleep mode (i.e. a low-power state) unused elements is an effective and widely-accepted strategy to decrease the consumption of data networks. In this context, Software-Defined Networking (SDN) can be seen as an attractive solution to achieve the long-awaited energy efficiency in current communications systems, since they allow a flexible programmability suitable for this problem.

This doctoral thesis tackles the problem of optimizing the power consumption in SDN through the design of energy-aware routing techniques that minimize the number of network elements required to satisfy an incoming traffic load. Different from existing related works, we focus on optimizing energy consumption in SDN with in-band control traffic 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 energy efficiency in SDN, this research is also intended to cover important related features such as network performance, Quality of Service (QoS) requirements and real-time operation. Accordingly, this study gives a general perspective about the use of energy efficient routing techniques, which cover integrated routing considerations for the data and control plane traffic in SDN.

By using realistic input data, significant values of switched-off links and nodes are reached, which demonstrates the great opportunity for saving energy given by our proposals. The obtained results have also validated the intrinsic trade-off between environmental and performance concerns, considering several performance indicators. These findings confirm that energy-aware routing schemes should be designed considering specific traffic requirements and performance metric bounds. Moreover, it is shown that jointly considering QoS requirements and energy awareness is an effective approach to improve, not only the power consumption, but the performance on critical parameters such as control traffic delay and blocking rate. Similarly, the proposed dynamic traffic allocation with congestion-aware rerouting is able to handle demanding traffic arrival without degrading the performance of higher priority traffic.

In general, our proposals are fine-grained, easy to implement and quite balanced and effective in their results looking for a suitable and readily deployment in real-world SDN scenarios. Therefore, the conducted research and contributions reported through this document not only add to what is known about the potential of energy-aware routing techniques, but also stand as a valuable solution on the road to a sustainable networking.

Acknowledgements

First and foremost, I want to thank my advisor Dr. Cristina Cervelló-Pastor. This work would not have been possible without her immense support, guidance and knowledge. But a bigger thanks for being much more than just my thesis director 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 advisor and mentor for my PhD.

Besides my advisor, I would like to express my special appreciation and thanks to Prof. Sebastià Sallent. I will always be grateful for the words of trust and the encouragement on both my research as well as on my career. Thanks for bringing expertise and sense of humor, even in the more tedious moments.

I would also like to express my sincere gratitude to the rest of my thesis committee: Dr. Joan Triay Marques and Dr. Xavier Costa Perez, 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 Dr. Paola Grosso from University of Amsterdam, who provided me an opportunity to join their team as a visiting researcher. Her insightful comments and stimulating discussions about my work motivated me to widen my research from various perspectives. A special thanks to the rest of the SNE group, for all the time we spent together as coworkers as well as friends. Thank you all for making these three months one of the best experiences I have had during my PhD.

I thank my fellow lab-mates for the long hours we were working together before deadlines, for the nice and relaxed working atmosphere and for all the fun we have had along the way. It was great sharing laboratory with all of you. Also I thank the friends and colleagues I have met during my PhD, for the exchanges of knowledge and skills and for those unforgettable moments we have enjoyed and shared along these years.

I recognize that this research would not have been possible without the financial support from the Ministerio de Economía y Competitividad of the Spanish Government, from which I got a PhD scholarship.

Last but by no means least, I would like to thank my family. 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.

To my life-coaches, my parents, who are the biggest inspiration in every aspect of my life both personal and professional. Although we have been physically separated most of the time during these last three years, they have been more than aware of every step I have done along this path. Thanks for your endless love and for making me feel that you are and will always be there for me. Because I owe it all to you. Many Thanks!

To my brother, for being the best big brother anyone could ever hope for. Thanks for making me so proud and for always setting the bar higher. My deepest gratitude goes to my grandparents, who have always supported me and encouraged me to pursue all my goals.

Finally, a special thanks to Leo, for sharing with me every moment of this adventure that we have lived together. It would not have been the same without you by my side. This work and achievement are also yours.

To my loving family

Contents

List of Figures

List of Tables

List of Abbreviations

AC-OFER Ant Colony Online Flow-based Energy-efficient Routing.

ADI ADequacy Index.

CDF Cumulative Distribution Function.

CERC Centralized Energy-efficient Routing Control.

DEAR Distributed Energy-Aware Routing.

DESRA Dynamic Energy Saving Routing Algorithm.

DyPAR Dynamic Power-Aware Routing.

ECMP Equal Cost Multiple Path.

EMMA Energy Monitoring and Management Application.

ESACON Energy Saving based on Algebraic CONnectivity.

ESIR Energy Saving IP Routing.

ESOL Energy Saving based on Occurrence of Links.

FRR Fast Rerouting.

GrIS Green Initial Setup.

ICT Information and Communication Technologies.

IEA International Energy Agency.

IGP-WO Interior Gateway Protocol Weight Optimization.

ILP Integer Linear Programming.

KPI Key Performance Indicator.

LB Load Balancing.

LC Least Cost.

LSP Label Switching Path.

MCMF Multi-Commodity Minimum Cost Flow.

MCP Maximum Clique Problem.

MILP Mixed-Integer Linear Programming.

MILP-EWO MILP-based algorithm for Energy-aware Weights Optimization.

MLU Maximum Link Utilization.

Mod-SPR Modified Shortest Path Routing.

MOEA Multi-Objective Evolutionary Algorithm.

MPLS Multi-Protocol Label Switching.

MSPF Multiple Paths by Shortest Path First.

NFV Network Functions Virtualization.

NMU Next Maximum Utility.

NSGA2 Non Sorting Genetic Algorithm 2.

NSP Next Shortest Path.

ONF Open Networking Foundation.

OSPF Open Shortest Path First.

QoS Quality of Service.

R-EAR Reliable Energy-Aware Routing.

ROD Routing On Demand.

SDN Software-Defined Networking.

SGH Strategic Greedy Heuristic.

SLA Service Level Agreement.

SNetCA Static Network Configuration Algorithm.

SP Shortest Path.

SPEA2 Strength Pareto Evolutionary Algorithm 2.

SPR Shortest Path Routing.

SPT Shortest Path Tree.

TCAM Ternary Content Addressable Memory.

TE Traffic Engineering.

TOB TCAM Occupation Balancing.

Chapter 1

Introduction

Over the last decade, Information and Communication Technologies (ICT) have been forced to face an exponential demand growth from an ever-increasing number of connected devices. For instance, global Internet traffic in 2021 will be equivalent to 127 times the corresponding volume observed in 2005, reaching 30 GB per capita, up from 10 GB per capita in 2016, while the number of devices connected to IP networks will triple the global population [1]. Such steady rise has implied an increasing energy usage contributing to a non-negligible and concerning impact on the environment as well as associated operational costs. Consequently, the need for energy efficient networking has become a major goal involving government and industry efforts and attracting a great deal of attention from research community.

According to [2], between 2007 and 2012 the global energy use of ICT grew at an annual rate of nearly 7% representing a higher value than the 3% yearly growth of the overall worldwide electricity consumption in the same time frame. These figures correspond with an increase of ICT's relative share in worldwide electricity consumption from about 4% in 2007 to 4.7% in 2012 [3].

Currently, this sector is responsible for about 2.4-3% of global electricity consumption and it is expected an annual increase of 20% [4]. Accordingly, the global carbon dioxide footprint of ICT equipments accounts for 2-2.5% of worldwide emissions [5], which equals the amount generated by the aviation industry [6]. Moreover, it is projected that the ICT-related CO_2 emissions will increase to 4% by 2020 [7]. Similarly, it has been forecasted that, compared to the level in 2010, the total electricity consumption of ICT will be doubled by 2022 and tripled by 2030 [8].

Among the main ICT areas, data transmission networks account for more than a third part of the total energy consumption in this sector [9]. Moreover, they presented the strongest electricity consumption growth (10.1% per year) with respect to end-user equipment (5.2% per year) and data centers (4.3% per year) [10].

A recent estimation from the International Energy Agency (IEA) [11] states that global electricity used by internet data transmission networks in 2015 amounted to around 185 TWh (i.e. 1% of total worldwide electricity demand). Moreover, considering the increasing global use of Internet (with an estimated growth rate between 30 and 40% per year), the electricity demand of data networks in 2021 can rise by over 70% to about 320 TWh. However, the use of efficient energy-aware strategies could overturn this estimation reducing the electricity consumption of Internet data transmission networks by 15% to about 160 TWh.

As a result, the reduction of power consumption in telecommunication networks is a crucial step to accomplish significant energy savings in this sector. The future direction of ICT impact on energy consumption and carbon emissions will depend on the adoption of power efficient solutions in data networks. At the same time, being ICT a support technology for many industries, increasing the energy efficiency in data networks can also substantially reduce the environmental impact of other sectors.

Given that in practice, the energy consumption of network equipment is not in proportion with their traffic load [12], the reduction of the number of active elements is an effective and widely-accepted strategy to decrease the consumption of data networks [13]. This feature can be implemented by putting into sleep mode (i.e. a low-power state) unused networks elements such as interconnection devices, line cards or port interfaces [14].

Although turning off entire interconnection devices enables greater energy savings, this possibility should be carefully considered, since it leads to resiliency concerns in case of network failures. Nevertheless, due to the link over-provisioning typically considered in the design and operation of backbone networks, substantial energy can still be saved by changing the status of network interfaces to sleep mode whenever a link is not transferring data.

Within this context, SDN, which is further described in this section, is a very well-suited architecture to perform an energy-aware routing and to manage the state of unused switch interfaces in a coordinated and centralized way. Therefore, the implementation of an energy-aware solution in the control plane is a valuable opportunity to solve the power consumption problem in data networks.

1.1 Software Defined Networks: An Overview

The optimization of power consumption could be considered as one promising field of application for SDN [15, 16]. In this networking architecture, control functions (routing decisions) are decoupled from the data plane (forwarding nodes) and delegated to a new entity called the controller.

The standardization work behind SDN concepts and technicalities is led by the Open Networking Foundation (ONF) in collaboration with major standards organizations such as ETSI, IETF, 3GPP, and IEEE. SDN provides four essential features [17]:

- Control and data planes separation;
- Logically centralized control and network view;
- Control plane and data plane connected through open interfaces;
- Use of external applications providing network programmability.

In essence, the basic idea of SDN –moving control functions from hardware into software– makes network environments more manageable. Unlike traditional networks, requiring manual and individual configuration of network devices to change the network behavior, network-wide routing and forwarding decisions can be taken in the decoupled and logically centralized (or even possibly distributed) control plane.

The logically centralized controller in SDN provides global knowledge of the network state and topology information. Moreover, it can manage network tasks and perform device programming without additional software or hardware-based intelligence in each one of the switching elements. In particular, an SDN controller can collect traffic patterns, perform path computation using its knowledge of the existing topology and push such routing decisions down to the data plane devices for execution.

On the other hand, forwarding nodes in the data plane are merely required to follow the instructions set by the controller to forward the traffic, being no longer involved in network control. Routing strategies are sent by the controller in the form of flow table rules and installed in the Ternary Content Addressable Memory (TCAM) of the forwarding elements. Additionally, flow entries can be modified or removed by the controller in order to manage the switches behavior with respect to traffic forwarding.

Consequently, the use of an underlying SDN architecture will facilitate the introduction and deployment of new applications and services, making it easier than with classical hardware-dependent implementations. Another advantage of exploiting SDN is the possibility to dynamically adapt control decisions to comply with diverse QoS requirements to handle heterogeneous application-driven networks.

The different planes and interfaces conforming the SDN architecture can be appreciated in the diagram shown in Fig. 1.1.

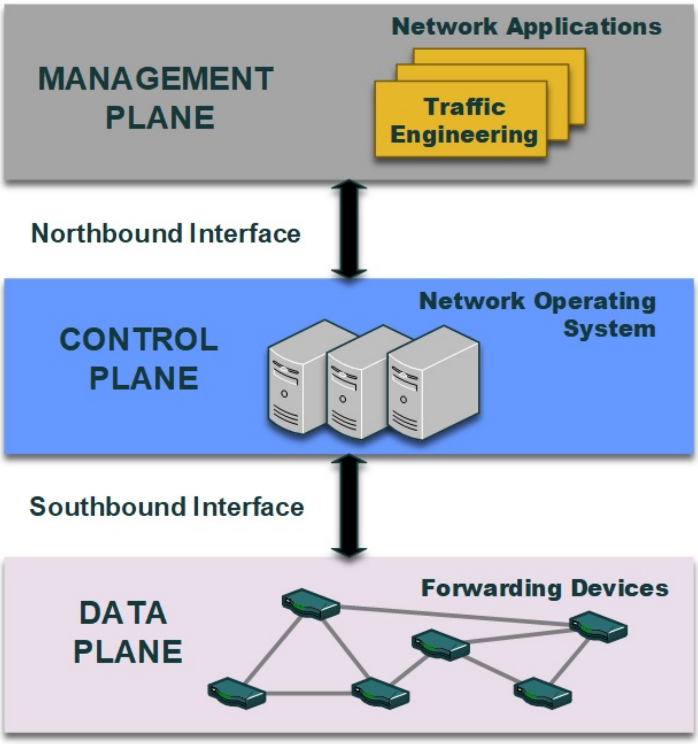
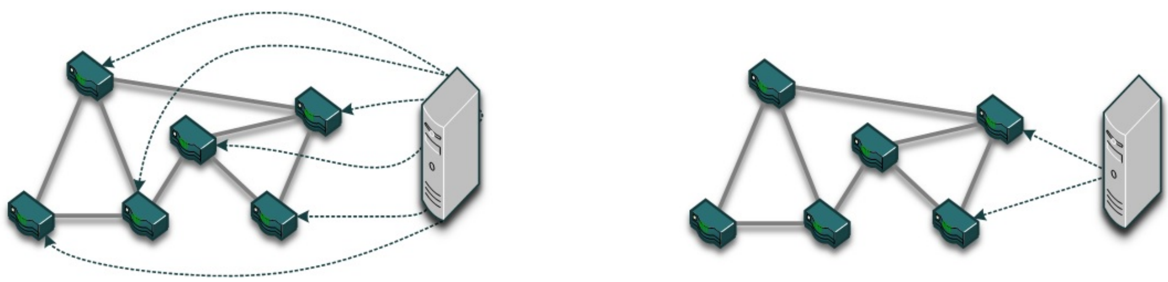


Fig. 1.1: SDN architecture.

Control plane implementations can be categorized into centralized and distributed architectures, according to the number of physical controllers deployed in the network [18]. Obviously, the use of a physically-centralized control plane consisting of a single server is a recurrent concern, since it brings scalability and reliability issues. As an alternative, the logically centralized control in SDN can be implemented with multiple physically-distributed servers in order to mitigate such issues. In this regard, two main categories of distributed SDN control architectures have been discerned based on the physical organization of SDN controllers, namely flat and hierarchical [19]. In the first approach, the network is partitioned into multiple areas, where each is handled by a single controller [20], while in the second, multiple vertical control layers are defined depending on the required services [21].

The exchange of traffic flows between switches and controllers is possible through an open southbound interface. OpenFlow [22, 23] is the first and most commonly used protocol for the southbound interface of SDN. It is already supported by commercial products of different vendors [24] and has been applied in a wide range of network environments [25]. Other examples of protocols defined for the southbound interface in SDN are ForCES [26] and OpFlex [27].



(a) Out-of-band control

(b) In-band control

Fig. 1.2: Signaling modes in SDN.

Regarding the traffic flows between control elements and forwarding nodes on the data plane, two operational modes can be identified, namely out-of-band and in-band control [18]. Out-of-band control, depicted in Fig. 1.2(a), is based on the deployment of a separate control network in addition to the one used for the data traffic. This is a common approach in data centers limited in geographical size, where CAPEX and OPEX associated with an additional control network are usually acceptable.

With in-band control, the same links are used by both data and control plane traffic, as illustrated in Fig. 1.2(b). Hence, control messages are exchanged without the need for additional infrastructure [28, 29]. Evidently, in wide-ranging carrier networks, this is a more realistic scenario since additional links dedicated to directly connect controllers and forwarding nodes are impractical and cost-inefficient.

By allowing innovation and flexibility, SDN can dramatically simplify network control procedures [30]. Specifically, the use of these programmable networks can have a significant role in reducing the energy consumption [31]. The centralized decision-making mechanism can be leveraged to perform an energy-aware routing that determines, in a coordinated way, the network elements that should be put into sleep mode and to accordingly program the forwarding nodes. Therefore, an energy-aware solution could be easily implemented in the SDN control plane.

Based on the aforementioned reasons, throughout this work we exploit the benefits of SDN to solve the problem of power consumption in data networks. Complementing the general aim of an energy-aware routing that minimizes the number of active network elements needed to route the required traffic in SDN, this research also seeks to propose several novel methods to address the following research problems with energy efficiency at its focus.

1.2 Research Problems and Objectives

As stated before, this research aims to tackle the energy consumption problem in SDN. For such purpose, we have identified four key related challenges which have naturally guided and motivated the investigation.

- **How to route, in an energy efficient way, connection requests in an SDN environment with multiple controllers.** To address this question we aim at achieving the optimal energy-aware routing in the network by selecting the number of physical resources that lead to minimum power consumption and meet a given traffic demand. We pursue to optimize energy consumption in OpenFlow networks with in-band control traffic to advance the state of the art that considers only the data plane traffic in SDN. We also seek to optimize the distribution of switches between controllers and the controllers location in terms of energy efficiency.

- **How to increase the energy saving and quantify its impact on network performance.** Considering that executing an energy efficient routing technique is only reasonable if the impact on other network performance parameters is tolerable, we need to determine the intrinsic inter-relation between both essential features. We endeavour to provide a broader analysis about the existing trade-off between power consumption and network performance, considering several performance indicators, in order to shed light on this crucial issue for communication systems nowadays.

- **How to jointly optimize QoS requirements and energy efficiency while still meeting control traffic requirements.** The conflicting nature between QoS and energy-related objectives calls for an integrated networking solution. In this context, optimization refers to a more complex joint criterion combining the traditional Traffic Engineering (TE) goals with the new energy-related ones in a common multi-objective framework in order to achieve the best compromise solution between the aforementioned goals. In this regard, control traffic requirements in terms of latency should also be included in the integrated model.

- **How to provide online energy efficient traffic allocation to real-time demands without performance degradation.** For current networking environments, which are facing less predictable traffic patterns, incoming demands need to be strategically allocated in real-time along energy efficient paths. In this way, dynamic awaking/sleeping decisions can be taken based on the arrival/departure of network traffic. In addition to minimizing the number of active elements, the routing framework should also reduce the network congestion in order to avoid associated performance degradation.

To achieve the aim of this thesis, a set of **main objectives**, closely related to the aforementioned research problems, have been defined:

1. Develop an **energy-aware routing approach in SDN** beyond the current state of the art by taking into account integrated routing considerations for data and control plane traffic, often neglected in the past.

2. Evaluate the **impact of energy-aware routing on SDN performance** considering crucial network parameters such as control traffic delay, data path latency, link utilization and TCAM occupation.

3. Compose a **multiple objective optimization solution**, jointly considering QoS requirements and energy awareness to avoid performance degradation while reducing the power consumption.

4. Design an **online energy-aware routing** to support time-variable traffic demands that dynamically employs traffic reallocation to ensure energy efficiency and congestion avoidance.

In addition, we specify a set of **secondary objectives** to provide support in attaining the main goals.

1. Identify the **existing solutions** already proposed about energy consumption in SDN, the granularity of which they are applied and **the remaining limitations in this subject**.

2. Formulate **exact mathematical programming models** and **heuristic optimization algorithms** sufficiently generic, in line with different system requirements and constraints.

3. Generalize the proposed algorithms by focusing on **in-band control traffic** to ensure compatibility with current physical topologies without a dedicated control network.

4. Compare proposed algorithms with existing related approaches in order to **validate the effectiveness and benefits of the proposed solutions**.

1.3 Contributions

For the sake of convenience, the contributions of the thesis in accordance with the proposed research problems and objectives can be summarized as follows:

I. To overcome current energy-related limitations and move to a more flexible, effective and environmentally friendly TE approach, this thesis presents as a first contribution, a set of **energy-aware routing approaches tailored to fit SDN requirements** [32–35]. To this end, exact optimization models, considering routing requirements for control and data plane communications, are formulated. In order to reduce the associated complexity of these models in large-scale topologies, heuristic algorithms are developed as well. In addition, we derive a simple and efficient algorithm to find the best controllers placement in terms of energy saving.

II. To stress the importance of energy efficiency, we propose a novel strategy aiming to reduce the number of active links. By jointly considering specific network topological properties and the use of TE techniques, a **low-complexity and energy efficient strategy** is achieved. This combined approach is **used to evaluate its impact on crucial performance metrics** [36, 37]. Being the proposal energy efficiency oriented, notable improvements in terms of energy saving can be achieved while disclosing the intrinsic trade-off between environmental and performance concerns.

III. Performance constraints, such as bounded delay for the control plane traffic and QoS requirements for data plane traffic, are crucial in the correct operation of SDN. In this context, an algorithm is proposed, which is able to tackle a **multi-objective routing scenario combining QoS requirements and energy awareness** [38]. This novel solution is implemented using a Multi-Objective Evolutionary Algorithm (MOEA) customized to our particular routing problem. In this way, data and control paths are established according to the changing traffic scenario and a significant performance improvement on critical network parameters is attained.

IV. Considering the unpredictable nature of incoming traffic in current networking environments, performing a dynamic energy-aware routing is one of the main technical challenges. For such purpose, a strategic traffic allocation is designed through an **online energy efficient and congestion aware routing** [39]. This proposal is conceived to dynamically reduce the number of active nodes and links required to manage load demanding real-time traffic patterns while avoiding the performance degradation of higher priority traffic.

1.4 Thesis Organization

In line with the thesis scope, this document is structured as shown in Fig. 1.3:

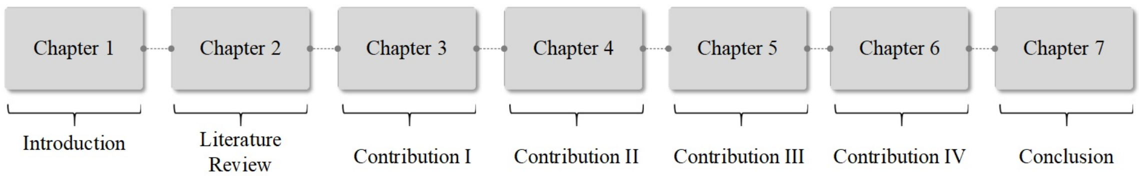


Fig. 1.3: Thesis outline.

Chapter 2 discusses the principles of the energy-aware routing problem addressed by the thesis. The chapter also presents a literature review with the relevant research work and assesses their main contributions and drawbacks. 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 by the present thesis.

Chapter 3 introduces novel energy-aware routing solutions for SDN with in-band control traffic. Starting from the initial case of having a single controller, the model is extended to support multiple controllers, either allocated in distributed domains or logically centralized. Exact mathematical formulations as well as heuristic algorithms are derived in this chapter. The problem of energy-aware controllers location is also addressed.

Chapter 4 develops a novel low-complexity heuristic algorithm relying on topology knowledge combined with TE techniques to reduce the overall power consumption. Using a link-based mathematical formulation the problem is formulated. Two heuristic algorithms are designed: a static network configuration and a dynamic energy-aware routing improving convergence times by several orders of magnitude and finding results very close to the optimal ones. Moreover, an insightful analysis is performed to evaluate the impact on several network performance metrics.

Chapter 5 describes the design of a multi-objective routing approach jointly considering QoS requirements and energy awareness, suitable for SDN environments with in-band control traffic. To achieve this, we present an optimization problem that integrates the routing requirements for data and control traffic and implement this approach using a MOEA. Performance simulations confirm the flexibility of the proposal to provide different solutions according to the specific traffic needs and validate the performance improvement on critical network parameters.

Chapter 6 proposes an online solution minimizing the number of active nodes and links required in an SDN architecture with multiple controllers and in-band control traffic. This proposal comprises of two modules: a green initial setup and a dynamic power-aware routing. Besides being compatible with SDN environments without a dedicated control network, the proposed strategy is able to handle demanding traffic arrival without degrading the performance of higher priority traffic. Simulation results show that our heuristic approach allows to obtain close-to-optimal power savings. Moreover, they validate the improvements achieved by our solution in terms of power efficiency and performance degradation avoidance.

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

Chapter 2

Background and Literature Review

This chapter is divided into four main parts, according to the research problems covered by this study. In each part, relevant research works in the literature dealing with the energy-aware routing problem related to this thesis are introduced. Finally, we extract the main open issues regarding the energy awareness property in routing strategies for SDN and propose some guidelines to overcome these existing challenges.

Energy-aware routing mechanisms were initially considered for traditional IP networks. However, the disadvantages of implementing these approaches in traditional networks have brought the attention of the networking research community to the use of SDN for minimizing power consumption. Therefore, works concerning to both networking environments are covered along this chapter.

2.1 Principles of Energy-Aware Routing

The basic idea of energy-aware routing is to manage the incoming traffic reducing the number of unused network resources, or equivalently, adjusting the offered capacity to the actual network utilization [40–42]. However, this long-awaited proportional behaviour seems incompatible with current design criteria. Typically, network infrastructures are designed with more network elements than needed to ensure redundancy in case of failures. Likewise, network links are equipped with high capacity in order to handle sudden peak traffic flows. Therefore, under light traffic loads, these additional resources are either not used or less frequently used, which leads to an undesired energy waste.

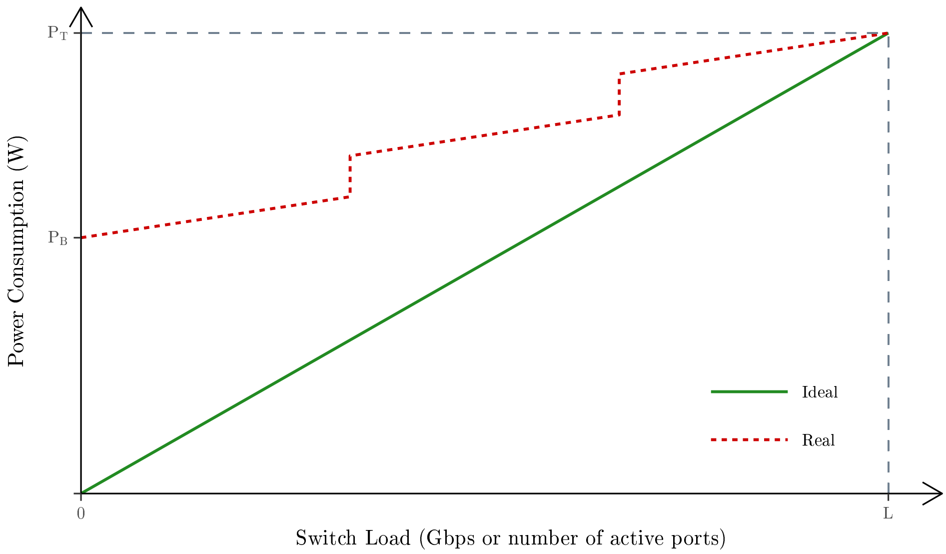


Fig. 2.1: Power model (redrawn from [12]).

To solve the problem of power consumption in data networks two well-known power management approaches have been proposed [43–45]. The first approach, also known as link rate adaptation, adjusts the speed (and capacity) of individual device's interfaces, to meet actual traffic loads and requirements. The second approach, also known as sleeping mode, attempts to concentrate the incoming traffic over the fewest amount of nodes and links in order to put the remaining unused network elements to sleep (i.e. a low-power state).

To better understand the motivations behind the two aforementioned energy management approaches, the power characterization of network equipments should be considered. In general, power consumption of networking devices is composed by a static component (due to power consumed by chassis, fans, line-cards, etc.) and a dynamic one, related to the rate of traffic flowing through their port interfaces.

Ideally, the static part –also known as the idle component– that represents the power required by an unused device, should be null. Then, in presence of an increasing traffic load, the power consumption should behave proportionally and linearly grow along with the traffic increase as shown by the line marked as Ideal in Fig. 2.1.

However, this model differs considerably from the real behaviour. In practice, whenever a device is active it consumes a fixed amount of power (P_n), irrespective of the load conditions. Additionally, this baseline power is increased by the number of active ports and the utilization of each port, which is depicted in Fig. 2.1 with the line indicated as Real.

Precisely, 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 operate. Instead of running networking devices at full capacity, which implies that they remain under-utilized most of the time, the reasoning behind this method is to vary the capacities of network interfaces as a function of the current link load. Therefore, the energy consumption of the equipment is reduced minimizing the required network link data rates in accordance with the link utilization.

In this regard, several measurement studies [12, 13] have previously proved that energy consumption of networking devices is largely independent of the amount of traffic handled by its port interfaces. In addition, reducing link data rates has a less significant effect on power consumption than putting unused network elements into energy saving modes. Explicitly, while most of the power is consumed only by turning the device on, increasing the port capacities from zero to full load represents less than 8% of total power consumption [46]. Therefore, an active port, even without carrying traffic, consumes almost full power but nearly zero power consumption is measured when it is set to the sleep mode.

Given that power consumption of network equipment is only slightly influenced by their data rate configuration and current traffic load, the sleeping mode approach is able to obtain higher energy savings than solutions based on adapting the link data rates [43, 47].

Consequently, the sleeping mode is used in this work as the power saving strategy. This means that the problem of energy consumption is tackled reducing the number of required active elements in proportion to the network load. Accordingly, we consider that the power consumed by a network node depends on the baseline power and the number of active ports, both of which represent a fixed contribution.

In the model considered in this thesis, each network element (i.e. nodes and links) can have one of two possible states: the active state at which it operates at full rate and the sleep state with zero rates. In active state, we assume that the same amount of energy is consumed by elements of the same type, thus the overall network consumption is in proportion with the number and kind of active elements. Inversely, network elements do not consume energy being in the sleep state.

Another aspect that may emerge concerning the use of the sleeping mode strategy is related to the reconfiguration times. Although turning off entire networking devices yields the major power conservation, a non-negligible amount of time is required to switch them on. According to reported measures, booting up a switch may take between 30 seconds and 5 minutes [12, 46]. Meanwhile, individual ports can be rapidly reactivated from the sleeping state in about 1-3 seconds [46, 48]. Therefore, the implementation of sleeping mode solutions is conditioned by the use of technologies allowing a quick response and low reconfiguration times. This constraint should be carefully considered by energy-aware approaches based on the sleeping mode strategy.

In this regard, it should be noted that when a node is in sleep mode it is essentially removed from the network topology, which means that it is not able to receive, process and forward traffic. Therefore, a node can only be put into sleep mode if it is not the source or destination of any incoming demand. This is the reason why our first routing strategies exclusively concentrate on minimizing only the number of active links.

Identifying the most suitable links and switches that should remain active to route incoming flows introduces an additional level of complexity in the network. In fact, the difficulty of the classical energy-aware routing problem is known to be NP-Hard, which has been proved in several research works [49–51] by reduction from different well known network optimization problems such as the fixed point-to-point connection problem [52]. This means that the consumption of resources and time complexity of optimal solutions grow exponentially with the network size, until become impractical for real-world networks. For this reason, multiple heuristic algorithms have been developed in the literature to find approximate solutions to the considered problem.

Existing energy-aware techniques based on the reduction of active network elements can be divided into two categories, namely traffic-based and topology-based solutions, according to the elements considered in the model. In this section, we analyze in more detail works that deal with each one of these two approaches.

2.1.1 Traffic-Based Solutions

Under the assumption of dealing with expected (i.e., known in advance) traffic, traffic-based solutions are routing mechanisms that aggregate traffic over a network subset in over-provisioned systems. By adopting this strategy, the number of turned on network components handling the incoming traffic is minimized. Over the last decade, traffic-based solutions have been widely studied in order to tackle the problem of power consumption.

One of the most significant works in this area is [53], in which authors propose GreenTE, an intra-domain, centralized TE mechanism that finds a set of links that can be turned off under a given traffic load or matrix. The approach is based on a Mixed-Integer Linear Programming (MILP) formulation where the traffic demands are routed through a set of previously computed k-shortest paths. Performance requirements such as Maximum Link Utilization (MLU) and network delay are considered as constraints in the problem.

In a similar way, Bianzino et al. [54] 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. To achieve this, they formulated an optimization problem for finding minimum-power network subsets assuming the existence of traffic level with known daily behaviour. Therefore, an accurate prediction of incoming traffic is required.

In [55] the problem of switching off network elements is formulated as a variant of the Multi-Commodity Minimum Cost Flow (MCMF) problem [56]. A greedy heuristic that iterates first over all the network nodes and then through the links (both sets sorted according to rules such as random, least-link, least-flow and most-power) is proposed. The authors studied all possible node/link sorting combinations.

More recently, in [57], the authors introduce a state-of-the-art study of energy efficiency strategies in SDN. This paper addresses the importance of implementing green routing methods in SDN, taking advantage of the flexibility given by dynamic configuration and centralized network view capabilities. A summary of some existing energy-aware techniques in SDN with their key properties (benefits and drawbacks) is presented, based on a four category classification namely: traffic aware, compacting TCAM, rule placement, and end host aware.

Regarding the use of partially deployed SDN, the authors of [58] face the problem of saving energy in these hybrid scenarios. For this, they formulate an optimization model which aims to find minimum power network subsets. After proving the problem is NP-hard, they propose a heuristic solution to approach the exact solution, based on the use of several groups of spanning trees to satisfy the traffic loads.

The energy efficiency in hybrid IP/SDN networks is also addressed in [59]. 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. The most appropriate set of traditional IP nodes to be upgraded to SDN-enabled switches are selected according to six different replacement methods.

Taking into account the rule space limitations of the TCAM in SDN forwarding nodes, Giroire et al. [60] propose an energy-aware routing method for a backbone network. An ILP optimization model is presented as well as an efficient heuristic, respecting capacity constraints on links and rule space constraints on routers. Performed simulations show that important savings (similar to the classical TCAM-agnostic approach) can be achieved using the proposed smart rule space allocation.

Markiewicz et al. [61] formulated an MILP model that aims to switch on a minimum amount of routers and links to carry the traffic. To solve the problem for large networks, they present a heuristic method, called Strategic Greedy Heuristic (SGH), that iteratively selects a pre-computed shortest path for each request, according to four different strategies of processing order of requests.

Similarly, two heuristic algorithms, namely Next Shortest Path (NSP) and Next Maximum Utility (NMU), are proposed in [62] to deal with the energy aware routing problem in SDN. Considering an initial Shortest Path Routing (SPR), both models seek to minimize the energy consumption of links and switches redirecting selected flows from under-utilized links to a more utilized replacement path in order to turn off the under-utilized links if all flows are redirected. The new selected path corresponds, respectively, to the next shortest path or the most-loaded path calculated after excluding the under-utilized links. Results show that, although NSP and NMU are more efficient in minimizing the average path length, more energy can be saved initializing the network with the outputs of [61] instead of using the SPR approach.

The authors of [63] provide two greedy algorithms for minimizing the power of integrated chassis and line-cards used under constraints of link utilization and packet delay. One attempts to adjust as few requests as possible while the other reroutes all requests sorted by priority in order to get better energy savings. To achieve this they considered an expanded network topology according to the connections between forwarding nodes. Specifically, routers are symbolized as star graphs, where the center represents the integrated chassis and the leaves are the line-cards. Although the proposed scheme saves an important amount of energy, it results in a highly-loaded network environment, making the network vulnerable to link failures and sudden traffic bursts.

Focusing on the use of pre-established multi-paths, an SDN-based green routing and resource management model for Multi-Protocol Label Switching (MPLS) networks is presented in [64]. In this approach, the controller, considering several pre-established Label Switching Paths (LSPs) between each ingress and egress pair, performs three main operations: path selection, load balancing and path resizing. In the first case, paths are activated or deactivated in order to reduce energy consumption. The two other functions are intended to manage resource utilization.

Instead of assuming dedicated links between the controller and SDN nodes, in [65] the authors 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. We argue that data plane traffic should not pass through network controllers, since this will represent an additional load in these devices.

2.1.2 Topology-Based Solutions

The lack of awareness of traffic conditions in typical operative networks has led to several research works that, in order to reduce the number of active links, are oriented to control the network topology. Basically, these approaches modify the existing topology considering different requirements such as the resulting connectivity. Consequently, during the selection of the sleeping components, topology-based solutions are only concerned about keeping reachable network endpoints.

In [66] the authors present an Open Shortest Path First (OSPF)-based routing mechanism that considers the topological information exchanged among routers. The proposed energy-aware routing algorithm is based on the definition of the "exportation" mechanism where a Shortest Path Tree (SPT) is shared between neighbor nodes. The routers with the highest node-degree, called "exporters", calculate the SPTs that are used to route the traffic and force the use of these paths to all their neighbors, so that the overall set of active links can be reduced.

The exportation mechanism is enhanced in [67], where the concept of "move" was introduced turning the energy saving routing problem into a formulation of the well-known Maximum Clique Problem (MCP) in an undirected weighted graph. Given that the MCP is NP-hard, a Max-Compatibility heuristic is proposed to select the maximum number of compatible moves. In this way, the selection of exporter routers is optimized leading to further energy savings.

Authors in [68] propose a routing algorithm called Energy Saving based on Algebraic CONnectivity (ESACON), using the algebraic connectivity [69] as a metric to control the resulting network topology. Based on this metric, ESACON is able to identify and switch off the network links that less affect the network connectivity, keeping this value over a given threshold. Consequently, significant energy saving is achieved while still preserving network connectivity and performance to efficiently support the incoming traffic.

Similarly, the topology-based solution reported in [70] also takes into account the algebraic connectivity as a requirement to preserve the overall network connectivity. This work also considers the edge betweenness [71] as a metric to measure the links role in the network, placing the links least frequently used as the first candidates to be pruned. However, this approach is conceived to be implemented in a distribute way into each IP router.

The work in [72] also aims to improve the energy efficiency reducing the number of active links. For this purpose, the authors propose four different versions of the Energy Saving based on Occurrence of Links (ESOL) algorithm that show the trade-off between complexity and efficiency in powering off a great number of links. The parameters used in this approach to select the network interfaces to be switched off are the occurrences of nodes and links in shortest paths, extracted from the network topology by using the classical Dijkstra algorithm.

The analysis of including QoS requirements in an energy-aware topology-based solution is discussed by the authors of [73]. Their approach, called Energy Saving IP Routing (ESIR), is also based on the concepts of SPT exportation and move but constrained to a maximum load boundary on network links in the traffic-aware scenario. However, such requirements in their study are considered of fixed values which is not practical (given the heterogeneity of flows in networks) and limits the suitability of their work for network services of varying patterns.

In order to switch off entire networking devices, the Steiner-tree-based algorithm proposed in [74] models the network topology as a graph composed of edge and core routers and their respective links. Specifically, this approach relies on the computation of a Steiner tree [75] to obtain the minimum subgraph connecting all the edge nodes. Then, the links and core nodes not involved on this subgraph are switched off to save energy. In case of routes with hop-counts above a given threshold, the original shortest path is added to the Steiner tree in order to minimize the path length along with energy savings.

The idea of using the algebraic connectivity concept to measure the importance of a link has been proposed in the literature by different works, such as [68, 70, 76]. More in detail, the authors of [76] introduce the ADequacy Index (ADI), a traffic-agnostic metric, which depends on the algebraic connectivity, for quantifying the quality of a frugal topology. Two heuristic algorithms are proposed to create pruned topologies by removing links in some order until a given ADI threshold is reached. They differ on the order followed to remove the links. The first one relies on the algebraic connectivity, while the second on the betweenness centrality.

All the previously described works mainly tend to minimize the number of active network elements in the current topology restricting the path selection to meet some specific metric bound or connectivity rate according to the energy-aware routing, without properly addressing the performance or traffic quality indicators. Moreover, their lack of awareness about the requirements of incoming connection requests can lead to performance degradations, which is highly undesired.

2.2 Energy Efficiency and Performance Concerns

While energy-aware routing solutions optimize the power efficiency finding the minimum number of active elements needed to support the concentrated traffic for all source-to-terminal requests, the performance of this remaining subnetwork in crucial parameters such as traffic delay and link utilization is more likely to be affected.

Precisely, the existing trade-off between energy saving and network performance is illustrated in the approach presented in [77]. This work presents Routing On Demand (ROD), an OSPF-based routing mechanism that considers energy saving and performance in terms of MLU. Using non-linear optimization approaches, authors formulate a problem that aims to minimize both the MLU and energy consumption of a network subject to capacity and flow conservation constraints. After considering different scenarios, two stages are identified, namely *power-sensitive* and *performance-sensitive* each of them with a particular compromise between both metrics. They prove mathematically that, for each scenario, a set of link weights exist, under which routes derived from ROD can be converted into shortest paths and realized through OSPF. These link weights are the solution of their approach.

In this regard, there are several works that, in order to address performance concerns in the energy-aware routing problem, restrict the path selection to the ones that meet specific network metrics [53, 65, 78, 79].

For instance, authors in [78] propose a heuristic-based algorithm, Multiple Paths by Shortest Path First (MSPF), which aims to maximise the number of switched-off routers and cables subject to satisfying MLU and path length constraints. They consider links with bundled cables that can be switched off independently and demands are routed through one or more paths. Two versions of the algorithm are proposed, which differ in the type of network elements (links or nodes) that are first considered to be switched-off.

Similarly, the power-aware routing reported in [79] also takes into account the MLU constraint. In this work the behavior of the overall power consumption achieved under different QoS requirements is studied. Specifically, the traffic transmission on the link was modeled as a M/M/1 queue (considering a service rate equal to the link capacity) and the analysis concerning the QoS requirements was carried out by varying the MLU factor between appropriate values that guarantee a low end-to-end delay.

Negative effects on signaling overhead and service quality experienced by traffic flows may be incurred in case of recurrent configuration changes. Following this idea, in [80] authors present an energy-aware management strategy that selectively turns off network devices considering a set of multiple traffic scenarios. Two single-path routing strategies are developed (i.e. a fixed and a variable routing), which include a penalty parameter for switch state transitions and an upper bound limiting the number of state changes of each network interface. Comparisons between both models suggest that the flexibility of the variable routing is beneficial for both congestion and consumption metrics. Moreover, the effects of restricting the number of times a network interface can be switched on are discussed. In this regard, it is shown that, compared with the conventional scheme, adding the card reliability constraint does not affect significantly the energy savings and prolongs the lifetime of equipments.

A different approach is introduced in [81], where authors analyze the increase in the number of control messages as a result of implementing energy-aware algorithms in an SDN controller. In particular, they model the problem as an ILP aiming to minimize the control overhead subject to an energy constraint that limits the total energy consumption of the network. Additionally, two polynomial-time heuristic algorithms are proposed to find near-optimal solutions for the problem. Finally, the existing trade-off between energy efficiency of green routing and the generated control overhead was validated in an SDN domain with a single controller.

Despite the potential benefits of energy efficient routing, its suitability to be implemented by network operators is naturally conditioned by the impact on other performance parameters. The work in [82] investigates the scope of the impact on route lengths and on fault tolerance. To do so, they propose a heuristic approach which iteratively tries to remove the edges that are less loaded. Simulation results show that the route lengths increase only 27% in average for almost all studied topologies. Meanwhile, to achieve fault tolerance fast switching-on technologies can be used or disjoint path constraints may be added to the problem.

Undoubtedly, network fault tolerance is one of the most concerning performance parameters that limits the deployment of energy-aware solutions. The disposition to put a network element into sleep mode is determined by the capability of the network to quickly react in case of failures. Authors of [83] discuss the trade-off between energy optimization and network reliability. They studied five existing green routing algorithms to analyze the impact of these proposed approaches on two reliability measures (terminal and route reliability). This work is extended in [84], where the authors formulate the Reliable Energy-Aware Routing (R-EAR) problem, which aims to switch off as many links as possible to optimize energy consumption, while guaranteeing the MLU and the required level of terminal reliability or route reliability. To solve this problem a heuristic algorithm is also provided.

The implications of achieving different levels of resiliency to failures and robustness to traffic variations for the network energy-aware efficiency are analyzed in [85]. To do so, two schemes are proposed: (i) optimization models minimizing the energy consumption of IP networks while guaranteing survivability and robustness; and (ii) suboptimal MILP-based heuristics exploiting variants of the original exact formulations. In particular, eight different protection/robustness strategies are considered combining the different features to quantitatively analyze the trade-off between energy cost and level of protection and robustness. It is shown that significant savings are achieved even when both survivability and robustness are fully guaranteed.

On the other hand, in some networking scenarios achieving energy saving does not necessarily affect negatively the network performance or its impact is tolerable. This is validated by the work presented in [54] where the average link utilization does not increase considerably as a consequence of the proposed green routing approach. By contrast, it is noted that the establishment of a maximum load bound on links may significantly limit the applicability of energy-aware techniques. This is due to the existence of low capacity access links which remain heavily loaded even under an energy-agnostic routing approach. Thus, reducing the maximum link load may easily conduct to unfeasible solutions.

A similar analysis is conducted in [86], where authors investigate the impact of practical constraints on the performance of energy-aware routing schemes in SDN. For such purpose, the energy-aware routing problem is modeled as an ILP considering discreteness of link rates and limitation of flow rule space. Results confirm that the inclusion of these practical constraints has a major impact not only on the energy efficiency of SDN but also on routes length and links utilization.

2.3 Multi-Objective Traffic Engineering

In common energy-aware routing models, path selection is generally formulated as a single objective optimization problem with either a single metric (minimizing the number of active links) or a single function encompassing different metrics (minimizing the number of active nodes and links). However, existing trade-off between network performance and energy saving has motivated the necessity to consider QoS requirements in order to achieve multiple TE goals.

A first trivial way to reduce energy consumption and guarantee QoS is to incorporate traffic requirements into mathematical models by means of additional constraints. However, this simple approach may led to over-provisioning (i.e. routing a connection onto a path that has too many resources for it) and, consequently, to a reduction in the number of future requests that can potentially be accommodated. Moreover, several performance studies have shown that, by optimizing multiple objectives simultaneously, better solutions can be obtained [87–89]. Therefore, instead of considering only traditional single objective functions with requirement constraints for the paths computation, several works evaluate the potential and the effective applicability of multi-objective procedures, in order to define routing strategies that can guarantee low energy consumption and good performance at the same time.

For such purpose, authors in [90] tackle a multi-objective optimization problem managing the link weights so as to minimize the energy consumption (primary objective) as well as a network congestion measure (secondary objective). To do so, a MILP-based algorithm for Energy-aware Weights Optimization (MILP-EWO) is presented. This approach takes advantage of the Interior Gateway Protocol Weight Optimization (IGP-WO) algorithm [91] to modify the OSPF weights according to the considered objectives. Predicted traffic matrices are assumed and link capacities are considered in this off-line intra-domain proposal to put network elements (links and routers) into sleep mode and to guarantee low levels of network congestion. Thus, the quality of solutions heavily relies on the traffic prediction accuracy.

The approach proposed in [92] simultaneously optimizes the power saving and different QoS-related parameters in software defined data center networks, according to a pre-defined combination of software quality requirements. The authors propose four different linear programming approaches that schedule requested traffic flows on the switches considering different metrics in the objective function such as energy consumption, throughput, transition time between sleep/active mode and their combination. An evaluation decision framework is implemented to assess their proposal. However, the size of the set of path for all flows considered in their solution is only scalable for data center topologies such as Fat-Tree [93], where the number of possible paths is small and does not grow rapidly along with the network size.

The work in [94] aims to improve the energy efficiency together with the quality of transmission in software defined flexible optical networks. To do so, the authors propose a multi-domain routing and spectrum assignment algorithm that takes into account quality of transmission (in terms of bit error rate) and energy saving. These two objectives are balanced considering connection requests separated into two classes of services, for each of which one objective is optimized in the path selection.

In the same way, both metrics can be improved if two different routing approaches are implemented in the SDN controller and applied depending on the context and network operator goals. This idea is conceived in [95], where a heuristic-based algorithm, named GoGreen, is designed to compute routing paths being energy efficient and satisfying the traffic requirement in terms of bit rate. According to the traffic type (video streaming, web browsing, sensor messages), one of the aforementioned objectives is taken as the first metric and used to determine the best k paths. Then, computed paths are sorted following the second metric and the first route is selected as the most suitable solution. Simulations show the trade-off between the number of considered paths (i.e. k) and the solutions quality.

The possibility of choosing different routing algorithms is also proposed in [89] through the design of an SDN-based integrated control plane. After collecting the network energy related information and the QoS requirements, specific traffic groups are defined. Then, based on the specific user application one out of three possible routing algorithms, namely Least Cost (LC), Shortest Path (SP) and Load Balancing (LB), is selected. Specifically, the LC algorithm is assigned to the web-surfing traffic, the SP algorithm to the VoIP traffic and the LB algorithm to the IPTV traffic. In this way, the QoS level for crucial traffic type can still be maintained, while a more energy efficient routing algorithm is employed for non-crucial traffic.

On the other hand, evolutionary algorithms [96, 97] have been applied to solve single and multi-objective problems in a wide variety of contexts in SDN [98], including routing strategies oriented to achieve power efficiency. For instance, the use of a MOEA for route selection has been proposed in [99] for dynamic optical networks with a centralized software defined integrated control plane. The solution is based on considering different objective functions according to the traffic type. In particular for higher priority traffic, they improve the energy performance without degrading QoS by taking energy saving as the secondary objective. Their approach supports multiple QoS requirements in terms of network performance, such as delay and blocking rate. However, only one network topology is considered, neglecting the effect that different network scenarios may have on the solution quality.

Similarly, in [100] a multi-objective particle swarm optimization algorithm is conceived to achieve network energy saving and load balancing in SDN. In this work, the problem is formulated as a multi-objective mixed integer programming model adding QoS constrains to the basic maximum concurrent flow problem. Specifically, it guarantees that the total delay of a routing path allocated to a demand cannot exceed the maximum delay it allows. The proposed heuristic algorithm, called MOPSO, dynamically aggregates and balances the incoming traffic putting the unused switches and links into sleeping mode.

2.4 Dynamic Green Traffic Allocation

Despite being efficient approaches to reduce the energy consumption, most of the aforementioned approaches are still limited since they are off-line. Assuming accurate traffic patterns fixed and known a priori may not be appropriate for current dynamic networks in which users can join or leave the network in an unpredictable way, affecting the overall traffic. Evidently, an online input is a more realistic consideration for energy-aware solutions and allows to dynamically adapt the number of active network elements to the arriving traffic.

An energy-aware routing and traffic management solution is proposed in [101] to reduce the energy consumption, determined as the number of active Open-Flow switches in the network. For this, a low complexity algorithm is presented using, for each pair of endpoints, a pre-computed set of shortest paths to select the route that minimizes the number of switches that become active after allocating the flow. Although this proposal allows real-time operation routing flows sequentially, only low-loaded nighttime traffic is considered, failing to extensively examine the implications of more demanding scenarios.

A similar approach is conceived in [102], where a dynamic routing scheme applying traffic aggregation for each incoming flow is proposed. Instead of only considering the number of hops, in this proposal the number of active links and nodes is also used as a routing metric. Paths are only taken into account if they have sufficient residual bandwidth to assume the incoming flow. In case of congested shortest paths, the path minimizing the MLU is selected.

In order to deal with traffic change in real-time manner, a Centralized Energy-efficient Routing Control (CERC) strategy is proposed in [103]. In this strategy the centralized controller is in charge of four main functions: link status monitoring, link sleeping, link awakening, and link status forecast. In essence, during traffic idle times low-loaded links are put into sleep mode and switched back on once the traffic increases, in order to avoid network congestion. During the sleeping procedure, candidate links are selected following two criteria of link utilization: traffic amount and number of node pairs. Routing paths are calculated based on the link metrics and the use of Equal Cost Multiple Path (ECMP). Additionally, the granularity of interval time for uploading the link status information to the controller is analyzed in this paper. However, simulations only consider a synthetic topology, without evaluating the performance of such approach in real-world networks with measured traffic traces.

The authors of [104] present the design of an Energy Monitoring and Management Application (EMMA) to minimize energy consumption in SDN-based backhaul networks. They formulated this problem as a non-linear optimization model and proposed heuristic algorithms for the dynamic routing of flows and the management of the resulting link and switch activity. However, such algorithms were implemented in an SDN emulation environment with out-of-band control traffic, limiting their applicability to networks where dedicated links between the controller and forwarding nodes are deployed. Different proof-of-concept prototypes showing the applicability of EMMA in three realistic use cases (i.e. a software switch network, a mmWave mesh network and an analogue RoF domain for ground-to-train radio access) are discussed in [105].

An online flow-based approach that takes into account the dynamic arrival and departure of users in SDN-based campus networks is designed in [106]. In this paper, the authors formulate the problem of routing the new incoming flow and dynamically re-optimizing the existing flows as an ILP subject to QoS constraints (i.e. bandwidth and delay), which aims to reduce the total energy consumption in the wireless and wired parts of the network. Given the NP-hard nature of such problem, an ant colony-based heuristic, called Ant Colony Online Flow-based Energy-efficient Routing (AC-OFER), is proposed to approximate to the ILP optimal solution.

Considering the undesired consequences of recomputing the routing paths in dynamic energy saving approaches, in [107] an energy efficient routing scheme based on Fast Rerouting (FRR), namely GreenFRR, is introduced. This paper aims to reduce the routing convergence time considering the occurrence of frequent traffic changes in a network. To do so, authors first formalize the FRR-based energy efficient routing problem and prove the associated NP-hardness. Thus, heuristic algorithms are proposed, which maximize the number of sleeping links and provide available rerouting paths quickly when a routing convergence is triggered.

In [46] authors propose ElasticTree, a network-wide power manager to save energy in data centers using SDN. This solution dynamically finds the minimum set of network elements required by changing traffic loads, while satisfying performance and fault tolerance constraints. In this regard, three strategies were studied, namely Formal Model, Greedy Bin-Packing and Topology-aware Heuristic. While the first option presents scalability issues and the second saves less power, the best performance is obtained by the Topology-aware Heuristic. However, this approach is specifically conceived for FatTree networks.

Another approach about power efficiency in software defined data center networks is presented in [108]. In this work different energy-aware routing strategies, combining common routing and scheduling algorithms, are evaluated and implemented as a OpenNaaS-based prototype. However, these strategies are only applicable in data centers and are also incompatible with environments without dedicated control networks.

2.5 Open Issues

Although throughout recent years the power consumption of communication networks has been extensively treated and several solutions focused on reducing the number of active elements have been proposed, existing methods have a number of weaknesses to be applied in current and future data networks.

First, green routing methods in SDN focus on power minimization considering only the data plane traffic, neglecting thus the energy consumption associated with the required control messages exchanged between controllers and forwarding nodes. Furthermore, there is little or no work that provides energy-aware routing mechanisms in OpenFlow networks where implementing a dedicated control network is not feasible either for physical or cost-related restrictions. Obviously, this practical implementation modality of SDN (i.e. the in-band mode) is a more realistic scenario for large backbone networks, where additional links dedicated to transfer the control messages between controllers and forwarding nodes, are impractical and cost-inefficient. Different from previous works, we aim to extend the energy-aware routing performance to SDN with in-band control traffic.

Second, the impact of energy-aware routing on network performance has been corroborated and examined by several existing works. However, this challenge becomes even more critical for SDN with in-band control traffic where performance degradations will affect not only data plane communications but connections with the controller. In this regard, a further evaluation is required for the considered scenario. To do so, specific network topological properties and the use of TE techniques can be jointly exploited to provide a low-complexity energy-aware strategy in SDN with multiple controllers.

Third, few energy-aware works in SDN take QoS requirements into consideration. In addition, they do not reflect performance indicators that are crucial in the correct operation of SDN (such as control traffic delay) and therefore, should be prioritized in order to provide an effective and useful solution. Hence, there is still room for routing schemes that find the best compromise solution between environmental and quality challenges.

Finally, there are no energy-aware proposals providing real-time operation for in-band SDN. Consequently, this may limit the utilization of such approaches in real world deployments. Moreover, existing online solutions are mostly designed for low-loaded nighttime traffic, being the implication of more demanding scenarios still an open issue. Thus, a novel energy-aware routing, including dynamic routing decisions and congestion awareness, should be designed for this analysis.

Chapter 3

Tailored Solutions to the Energy-Aware Routing Problem in SDN

This chapter is based on:

- **A. Fernández-Fernández**, C. Cervelló-Pastor and L. Ochoa-Aday, "A Distributed Energy-Aware Routing Algorithm in Software-Defined Networks," in *Advances in Intelligent Systems and Computing. Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection.* Cham: Springer International Publishing, 2016, vol. 473, pp. 369–373.
- **A. Fernández-Fernández**, C. Cervelló-Pastor and L. Ochoa-Aday, "Energy-Aware Routing in Multiple Domains Software-Defined Networks," *Advances in Distributed Computing and Artificial Intelligence Journal (ADCAIJ)*, vol. 5, no. 3, pp. 13–19, Nov. 2016.
- **A. Fernández-Fernández**, C. Cervelló-Pastor and L. Ochoa-Aday, "Improved Energy-Aware Routing Algorithm in Software-Defined Networks," in *Proc. 41st IEEE Conference on Local Computer Networks (LCN'16)*, Dubai, UAE, Nov. 2016, pp. 196–199.
- **A. Fernández-Fernández**, C. Cervelló-Pastor and L. Ochoa-Aday, "Achieving Energy Efficiency: An Energy-Aware Approach in SDN," in *Proc. 59th IEEE Conference on Global Communications (GLOBECOM'16)*, Washington DC, USA, Dec. 2016, pp. 1–7.

3.1 Introduction

By exploiting the flexibility of SDN control plane, in this chapter we make use of TE techniques to optimize the overall power consumption reducing the number of links required to handle a given traffic matrix. To ensure compatibility with SDN using in-band control traffic, in this proposal control paths between controllers and switches (and between controllers) are also established. Moreover, to avoid additional traffic load in the controllers, we establish that data plane communications cannot be routed through these devices. Similarly, we consider the different variants for control plane implementation in SDN (i.e. centralized and distributed) in order to provide a wide-scope perspective. In addition to the energy-aware routing problem, the controller placement issue is also addressed in this chapter. Different from previous works [109], which focus on minimizing the control traffic delay, we propose a simple and efficient approach that aims to determine the best controller location in terms of energy saving.

The rest of this chapter is structured as follows. In Section 3.2 we explain the main considerations of our approaches together with the exact optimization models formulated for each one of the different control plane implementations in SDN. The developed heuristic algorithms are described in Section 3.3. The simulation strategies and the obtained results are presented and analyzed in Section 3.4. Finally, Section 3.5 concludes this chapter.

3.2 Exact Algorithms

The problem at hand is formulated using ILP models, where the objective is to minimize the number of links used to route a given traffic demand.

3.2.1 One Single Controller

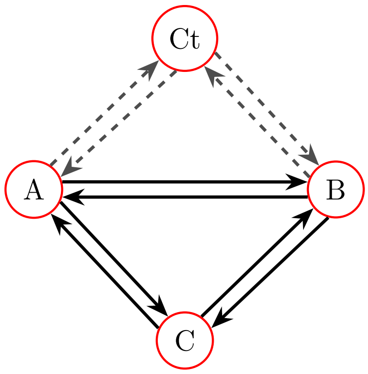
A single centralized controller is considered in this initial system model, which performs the energy-aware routing and determines the link interfaces that should be put into sleep mode.

3.2.1.1 Network Model

Given the controller location Ct, we modeled the SDN by a directed graph $G = (V, E)$ where V is the set of nodes (being $Ct \in V$) and E denotes the set of links. Each link $e \in E$ has associated its capacity, denoted by c_e. Considering D as the set of data traffic demands between any pair of forwarding nodes, let T denote the set of associated control plane traffic required. In this respect, we will use K to denote the overall set of traffic flows in the network ($D \cup T = K$). We use t_k to denote the throughput of a traffic flow $k \in K$.

A basic diagram illustrating the network model considered in this subsection is depicted in Fig. 3.1. This figure shows, at the left, a simple SDN topology with in-band control composed of 3 forwarding nodes and one centralized controller. The possible source/target pairs of data and control traffic are also included in this diagram.

In Fig. 3.1, links connected to the controller are differentiated from the rest of network links (with the color and type of line) in order to graphically identify them as inadmissible routes for the data plane traffic.



$$D = \begin{cases} A \longleftrightarrow B, \\ B \longleftrightarrow C, \\ C \longleftrightarrow A \end{cases}$$

$$T = \begin{cases} Ct \longleftrightarrow A, \\ Ct \longleftrightarrow B, \\ Ct \longleftrightarrow C \end{cases}$$

Fig. 3.1: Basic diagram of considered network model with a centralized controller.

In addition, considering P_k as the set of paths that can be used to route each $k \in K$, let $P_e^k \subset P_k$ be the subset of paths that use link $e \in E$ and $P_c^k \subset P_k$ denote the subset of paths that pass through the controller Ct.

3.2.1.2 Optimization Problem Formulation

To minimize the number of links used to route a given traffic demand, we define the following binary variables:

x_e: describes the state of a link $e \in E$.

$$x_e = \begin{cases} 1 & \text{if } e \text{ is active,} \\ 0 & \text{otherwise.} \end{cases}$$

$\gamma_{k,p}$: describes the selection of a path $p \in P_k$ to route each $k \in K$.

$$\gamma_{k,p} = \begin{cases} 1 & \text{if } p \text{ is selected to route } k, \\ 0 & \text{otherwise.} \end{cases}$$

Considering the notation of binary variables given above, the optimization model can be formulated as:

$$\text{minimize} \sum_{e \in E} x_e \tag{3.1}$$

subject to the following constraints:

$$\sum_{p \in P_k} \gamma_{k,p} = 1 \qquad \forall k \in K \tag{3.2}$$

$$\gamma_{k,p} = 0 \qquad \forall k \in D, \forall p \in P_c^k \tag{3.3}$$

$$\sum_{k \in K} \sum_{p \in P_e^k} \gamma_{k,p} t_k \leq c_e x_e \qquad \forall e \in E \tag{3.4}$$

The objective function in (3.1) minimizes the number of active links, i.e. the number of links used to route the control and data traffic.

Constraints in (3.2) ensure that only one path $p \in P_k$ is selected to route each $k \in K$. Constraints in (3.3) force that paths passing through the controller cannot be used to route data plane traffic. Constraints in (3.4) ensure that the total traffic in each active link $e \in E$ is less than its capacity c_e.

3.2.2 Multiple Controllers

In practice, the logically centralized control in SDN can be implemented with multiple distributed physical controllers, which is the scenario considered in this section.

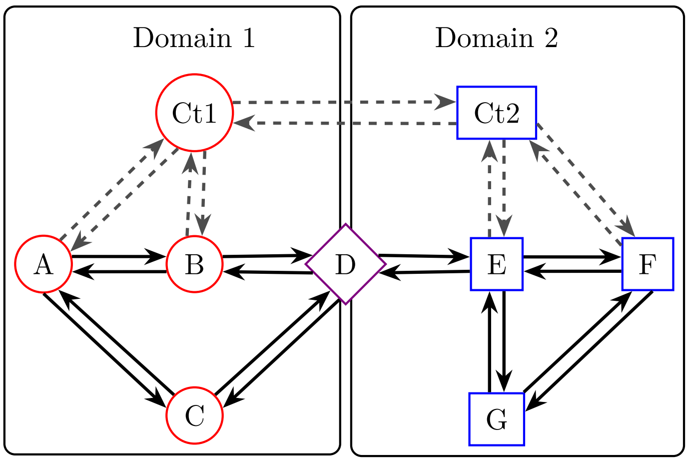
3.2.2.1 Distributed Proposal

We first consider a multi-domain SDN architecture, where each domain has a centralized controller with a number of predefined switches associated to it. We assume that each controller has a total knowledge of its domain topology and a partial knowledge of the global network topology, i.e. it has identified border nodes that it shares with each other domain. Inter-domain data traffic demands are routed in each domain using these nodes. It is also worth noting that we consider a network model with intra-domain in-band control.

3.2.2.1.1 Network Model

Each controller domain is represented by a directed graph $G = (V, E)$, where V and E denote the set of nodes and links respectively. Each link $e \in E$ has associated its capacity, denoted by c_e. The set $B = \{b_1, \ldots, b_{|B|}\} \subset V$ contains the border nodes. D_{v} and D_{w} denote the set of intra-domain traffic flows for the data and control plane, respectively. D_{u} denote the set of inter-domain data traffic demands. Therefore, while both endpoints of traffic requests in D_{v} and D_{w} are contained inside each domain, connections included in D_{u} should be established across multiple domains.

To better illustrate the network model considered in this subsection, Fig. 3.2 provides a basic diagram. This figure shows a simple SDN topology with in-band control composed of two controller domains and a border node between them (i.e. node D). In addition, the intra-domain connection requests included in D_{v} and D_{w} are exposed for each domain as well as the origin/destination pairs for the inter-domain data traffic in D_{u}.



$$D_{\mathrm{u}} = \begin{cases} A \longleftrightarrow E, \\ A \longleftrightarrow F, \\ A \longleftrightarrow G, \\ B \longleftrightarrow E, \\ B \longleftrightarrow F, \\ B \longleftrightarrow G, \\ C \longleftrightarrow E, \\ C \longleftrightarrow F, \\ C \longleftrightarrow G \end{cases}$$

Domain 1

$$D_{\mathrm{v}} = \begin{cases} A \longleftrightarrow B, \\ A \longleftrightarrow C, \\ A \longleftrightarrow D, \\ B \longleftrightarrow C, \\ B \longleftrightarrow D, \\ C \longleftrightarrow D \end{cases} \quad D_{\mathrm{w}} = \begin{cases} Ct1 \longleftrightarrow A, \\ Ct1 \longleftrightarrow B, \\ Ct1 \longleftrightarrow C, \\ Ct1 \longleftrightarrow D \end{cases}$$

Domain 2

$$D_{\mathrm{v}} = \begin{cases} E \longleftrightarrow F, \\ E \longleftrightarrow G, \\ E \longleftrightarrow D, \\ F \longleftrightarrow G, \\ F \longleftrightarrow D, \\ G \longleftrightarrow D, \end{cases} \quad D_{\mathrm{w}} = \begin{cases} Ct2 \longleftrightarrow D, \\ Ct2 \longleftrightarrow E, \\ Ct2 \longleftrightarrow F, \\ Ct2 \longleftrightarrow G \end{cases}$$

Fig. 3.2: Basic diagram of considered network model with a centralized controller.

As in Fig. 3.1, links are represented either by solid/black or dashed/grey lines. While the former will be shared by data and control traffic, the latter cannot be used to route the data plane communication.

For each $k \in D_{\text{v}}$, let t_k denote its throughput and P_k be the set of intra-domain paths that can be used to route this traffic. $P_c^k \subset P_k$ denote the set of paths that pass through the domain controller for each $k \in D_{\text{v}}$. In addition, let $P_e^k \subset P_k$ be the set of intra-domain paths between the source and target of k that use link $e \in E$. Similarly, this notation holds for the intra-domain control traffic flows (i.e. D_{w}) and the inter-domain data traffic flows (i.e. D_{u}).

3.2.2.1.2 Optimization Problem Formulation

The distributed proposal of our approach in multiple domains SDN, called Distributed Energy-Aware Routing (DEAR), can be formulated as an ILP model with two steps of optimization, using the following binary variables:

q_b^k: describes the selection of a border node b to route a traffic $k \in D_{\text{u}}$.

$$q_b^k = \begin{cases} 1 & \text{if } b \text{ is selected to route } k, \\ 0 & \text{otherwise.} \end{cases}$$

$l_{b,p}^k$: describes the selection of a path $p \in P_k$ to route a traffic $k \in D_{\text{u}}$ through border node b.

$$l_{b,p}^k = \begin{cases} 1 & \text{if } p \text{ is selected to route } k \text{ through } b, \\ 0 & \text{otherwise.} \end{cases}$$

r_p^k: describes the selection of a path $p \in P_k$ to route a traffic $k \in D_{\text{v}} \cup D_{\text{w}}$.

$$r_p^k = \begin{cases} 1 & \text{if } p \text{ is selected to route } k, \\ 0 & \text{otherwise.} \end{cases}$$

In the first step, each controller-instantiated agent individually computes the routing paths in its domain that minimize the number of links used. In this phase, performance constraints (e.g., control traffic delay and link utilization) could be included. Considering the notation of binary variables shown above, the optimization model of the first phase can be formulated as:

$$\text{minimize} \sum_{e \in E} x_e \tag{3.5}$$

subject to the following constraints:

$$\begin{aligned}
\sum_{b \in B} q_b^k &= 1 && \forall k \in D_{\mathrm{u}} && (3.6) \\
\sum_{p \in P_k} l_{b,p}^k &= q_b^k && \forall k \in D_{\mathrm{u}}, \forall b \in B && (3.7) \\
\sum_{p \in P_k} r_p^k &= 1 && \forall k \in D_{\mathrm{v}} \cup D_{\mathrm{w}} && (3.8) \\
r_p^k &= 0 && \forall k \in D_{\mathrm{v}}, \forall p \in P_c^k && (3.9) \\
l_{b,p}^k &= 0 && \forall k \in D_{\mathrm{u}}, \forall b \in B, \forall p \in P_c^k && (3.10) \\
\sum_{k \in D_{\mathrm{u}}} \sum_{p \in P_e^k} \sum_{b \in B} l_{b,p}^k t_k + \sum_{k \in D_{\mathrm{v}} \cup D_{\mathrm{w}}} \sum_{p \in P_e^k} r_p^k t_k &\leq c_e x_e && \forall e \in E && (3.11)
\end{aligned}$$

The objective function (3.5) minimizes the number of active links.

Equation (3.6) assures that exactly one border node is selected for every inter-domain data traffic demand. Equation (3.7) guarantees that exactly one path is used to route every inter-domain data traffic demand through the border node selected. Equation (3.8) ensures that exactly one path is used to route every intra-domain traffic flow for the data and control plane. Equations (3.9) and (3.10) guarantee that paths passing through the controller cannot be used to route data plane traffic. Finally, Equation (3.11) assures that the total traffic in each active link $e \in E$ is less than its capacity c_e. To do so, this constraint contains two terms; the former is meant to determine the amount of inter-domain traffic flowing on the link e while the latter does the equivalent analysis for the intra-domain (data and control) traffic.

After completing this computation, the distributed control plane agents in different SDN domains must exchange some performance metric (e.g. MLU in each domain) and the identifier of the selected border nodes to route each inter-domain data traffic demand (i.e. $q_b^k \quad \forall k \in D_{\mathrm{u}}$). The first element of this shared information is intended to be used as a comparison metric to define the domain with the best performance, which is also the one with the lowest probability to run out of capacity, while the second one allows a proper and coherent rerouting of inter-domain data traffic demands.

In the second step, the agent of the domain with the best performance (less MLU, for instance) recomputes its energy-aware routing paths using now, for each inter-domain data traffic demand, the border nodes preselected by its neighbor domains. The corresponding problem for the second step of optimization could be formulated using these received identifiers in Equation (3.7) of the model above.

3.2.2.2 Centralized Proposal

In SDN, multiple controllers are also deployed under hierarchical control structures to make network-wide decisions. In this way, network events and device programming are handled locally while routing paths are computed in a centralized way.

Taking as input of our system the controllers placement in the network topology, the model presented in this section determines the optimal distribution of switches between controllers in terms of energy efficiency, considering as well the load balance between controllers.

In addition, this solution takes into account the utilization of links and the delay of control paths. Therefore both elements are constrained in our model.

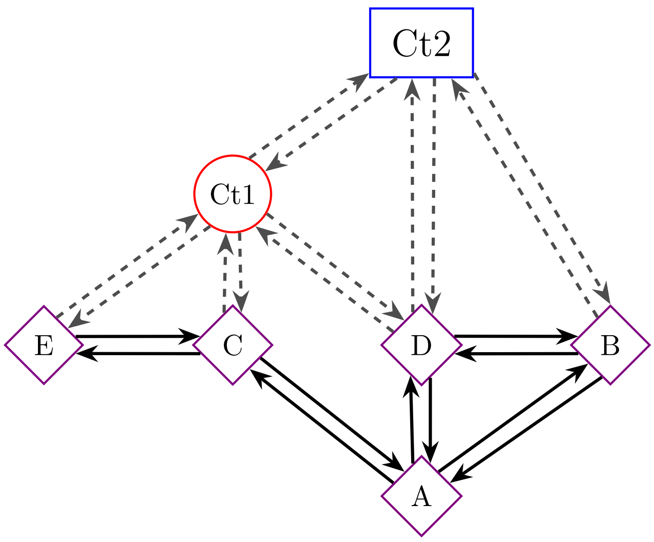
3.2.2.2.1 Network Model

The SDN is represented by a directed graph $G = (V, E, C)$, where V, E and C denote the set of nodes, links and controllers respectively, being $C \subset V$. We use c_e to denote the capacity of a link $e \in E$. We define the set of forwarding nodes as $S = \{n \mid n \in V \wedge n \notin C\}$.

Let D denote the subset of data plane communications. For the control plane, we use T to denote the subset of traffic flows between controllers and switches. In this respect, we will use K to denote the overall set of traffic flows in the network ($D + T \subset K$). Note that, in this case, the traffic flows associated with the control plane communications between controllers are also included in K. Each traffic flow $k \in K$ from source s_k to destination d_k, has associated its throughput, denoted by t_k.

In Fig. 3.3 a basic diagram representing the network model considered in this subsection is drawn. Specifically, this figure shows an in-band SDN with two controllers ($Ct1, Ct2$), assuming that $Ct2$ is on top of the hierarchy, and five switches. In this case forwarding nodes are depicted with a different color and shape than controllers to illustrate that they are not initially associated with any of the two controllers.

In addition, all possible data and control traffic endpoints included in D and T, respectively, are exposed in the figure. As in previous diagrams in Fig. 3.1 and Fig. 3.2, dashed/grey lines are used for links connected to the controllers, to indicate that they cannot be used to route the data plane traffic.



$$D = \begin{cases} A \longleftrightarrow B, \\ A \longleftrightarrow C, \\ A \longleftrightarrow D, \\ A \longleftrightarrow E, \\ B \longleftrightarrow C, \\ B \longleftrightarrow D, \\ B \longleftrightarrow E, \\ C \longleftrightarrow D, \\ C \longleftrightarrow E, \\ D \longleftrightarrow E \end{cases} \quad T = \begin{cases} Ct1 \longleftrightarrow A, \\ Ct1 \longleftrightarrow B, \\ Ct1 \longleftrightarrow C, \\ Ct1 \longleftrightarrow D, \\ Ct1 \longleftrightarrow E, \\ Ct2 \longleftrightarrow A, \\ Ct2 \longleftrightarrow B, \\ Ct2 \longleftrightarrow C, \\ Ct2 \longleftrightarrow D, \\ Ct2 \longleftrightarrow E \end{cases}$$

Fig. 3.3: Basic diagram of considered network model with a centralized controller.

In addition, let P_k be the set of paths that can be used to route each $k \in K$. Note that, in order to assure a certain delay for the subset of control plane traffic, the paths considered for these flows satisfy a maximum allowed latency bound, denoted as L_b. Let $P_e^k \subset P_k$ be the subset of paths that use link $e \in E$ and $P_c^k \subset P_k$ denote the subset of paths that pass through controller $c \in C$, for each $k \in K$.

3.2.2.2.2 Optimization Problem Formulation

To optimize the number of links used to route a given data traffic demand we use the following binary variables:

x_e: describes the state of a link $e \in E$.

$$x_e = \begin{cases} 1 & \text{if } e \text{ is active,} \\ 0 & \text{otherwise.} \end{cases}$$

$\gamma_{k,p}$: describes the selection of a path $p \in P_k$ to route each $k \in K$.

$$\gamma_{k,p} = \begin{cases} 1 & \text{if } p \text{ is selected to route } k, \\ 0 & \text{otherwise.} \end{cases}$$

$\lambda_{n,c}$: describes the association of each forwarding node $n \in S$ with a controller $c \in C$.

$$\lambda_{n,c} = \begin{cases} 1 & \text{if } n \text{ is associated with } c, \\ 0 & \text{otherwise.} \end{cases}$$

Considering the binary variable given above, the optimization model can be formulated as:

$$\text{minimize} \sum_{e \in E} x_e \tag{3.12}$$

subject to the following constraints:

$$\sum_{c \in C} \lambda_{n,c} = 1 \qquad \forall n \in S \tag{3.13}$$

$$\sum_{n \in S} \lambda_{n,c} \leq \left\lceil \frac{|S|}{|C|} \right\rceil \qquad \forall c \in C \tag{3.14}$$

$$\sum_{p \in P_k} \gamma_{k,p} = \begin{cases} \lambda_{n,c} & \forall k \in T,\ n, c \in [s_k, d_k],\ n \neq c \\ 1 & \forall k \in K \setminus T \end{cases} \tag{3.15}$$

$$\gamma_{k,p} \leq \lambda_{n,c} \qquad \begin{cases} \forall k \in T, \forall p \in P_c^k \\ \forall c \in C, n \in [s_k, t_k], n \neq c \end{cases} \tag{3.16}$$

$$\gamma_{k,p} = 0 \qquad \begin{cases} \forall k \in D, \forall p \in P_c^k \\ \forall c \in C \end{cases} \tag{3.17}$$

$$\sum_{k \in K} \sum_{p \in P_e^k} \gamma_{k,p} t_k \leq u_{max} c_e x_e \qquad \forall e \in E \tag{3.18}$$

The objective function (3.12) minimizes the number of active links, i.e. the number of links used to route the traffic.

Constraints (3.13) and (3.14) are related to controller-switch associations. The former indicates that each switch can only be associated with one controller, whereas the latter establishes the maximum number of switches that can be associated with each controller. The aim of this last set of constraints is to balance the load of switches among controllers, looking to avoid congested controllers.

Constraints (3.15)-(3.17) are related to paths selection. Specifically, constraint (3.15) ensures that only one path is selected to route each $k \in K$. Furthermore, this set of constraints guarantees for the subset of traffic flows between controllers and switches T, that each switch exchanges control messages only with its controller. Constraints (3.16) and (3.17) avoid the routing of additional traffic load through the controllers. Constraint (3.16) ensures that the control paths used for traffic flows between controllers and switches do not include any other controller that is not the source or target of the traffic. Constraint (3.17) forces that paths passing through any controller $c \in C$ cannot be used to route data plane traffic.

Finally, constraint (3.18) ensures that the total traffic in each active link $e \in E$ is less than the established MLU, denoted as u_{max}.

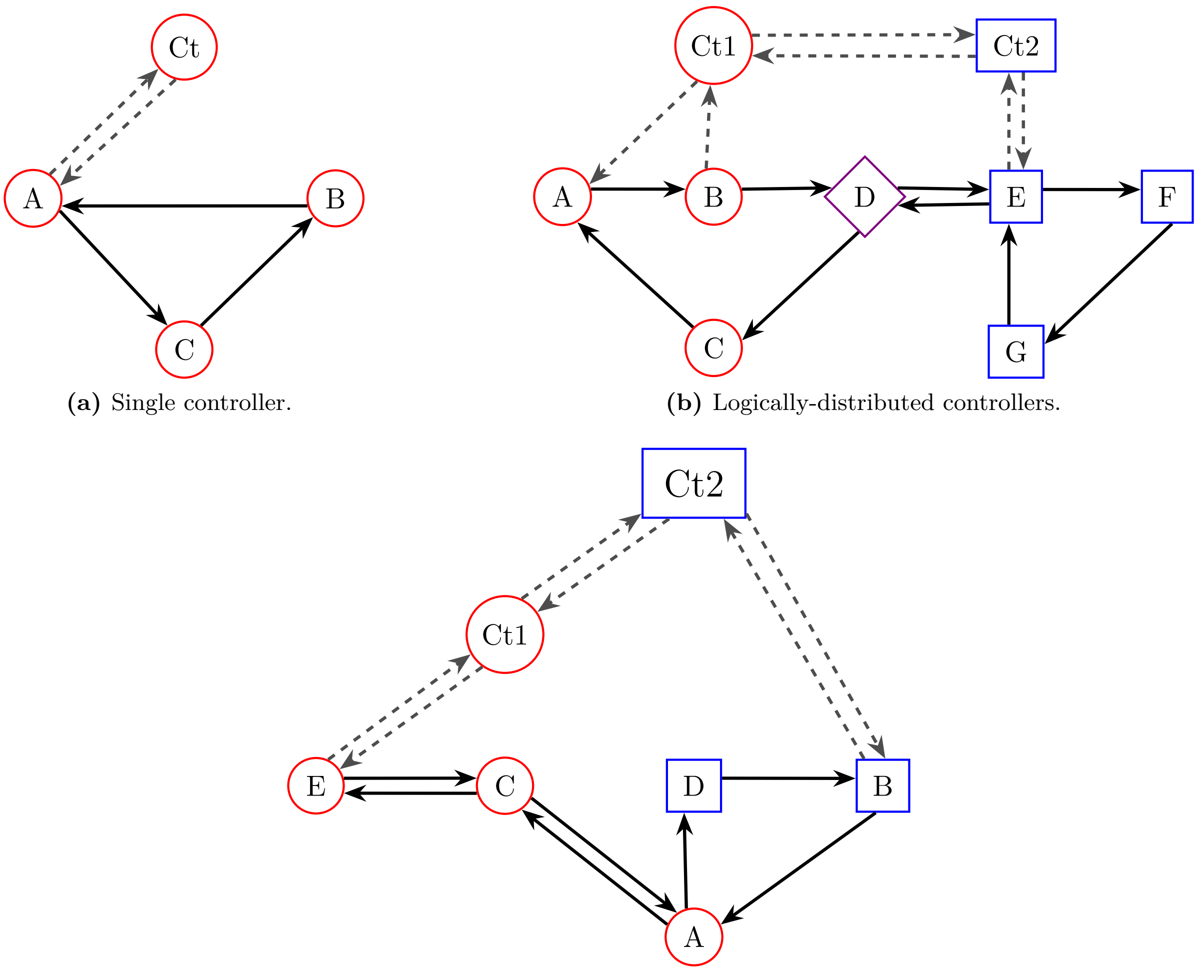
3.2.3 Model Operation Example

Taking as inputs the network graphs shown in Fig. 3.1, Fig. 3.2 and Fig. 3.3, we now illustrate an example of our models operation. In each case the corresponding ILP formulation explained above is used to determine the best paths (in terms of minimizing the number of used links). This is done assuming a one-to-one data traffic scenario together with the control traffic between each switch and its associated controller. For the sake of comparison, in Fig. 3.4 we show only the resulting active links after applying the three aforementioned models.

Specifically, in Fig. 3.4(a) for the single controller case, only 5 links (out of 10 total links) are needed to establish the required data paths and associated control routes. In this way, a 50% of energy saving (in terms of number of active links) is attained.

The physically and logically distributed scenario, covered in Fig. 3.4(b), shows a reduction in power consumption of 42.3%. This result is achieved after performing the two steps optimization process described in subsection 3.2.2.1.2.

Finally, in Fig. 3.4(c) the operation of the logically-centralized approach with multiple controllers can be appreciated. In this case the traffic between controllers is also considered and the control path delay is constrained by the network diameter (according to the number of hops). In addition, the distribution of switches between controllers is depicted through colors and shapes, indicating that nodes A, C and E are associated with $Ct1$ and switches B and D are controlled by $Ct2$. As a result of applying our optimization model in this example, only 13 links (out of 22 total links) stay active to satisfy the given traffic demand considering the routing requirements established for control and data plane traffic, allowing 41% of energy saving.



(a) Single controller.

(b) Logically-distributed controllers.

(c) Logically-centralized controllers.

Fig. 3.4: Models operation example.

3.3 Heuristic Algorithms

Although the models presented above allow the attainment of optimal solutions for the power consumption problem in SDN, they become challenging to solve on large and even medium-scale topologies given the NP-Hard difficulty of the energy-aware routing problem. To overcome this issue, in this section we develop some heuristic algorithms.

3.3.1 Energy-Aware Routing

The proposed algorithm, shown in Algorithm 1, starts finding the set of admissible routes between every pair of network nodes that satisfy the required constraints for control and data plane communications (line 1). For control traffic, these paths are delay-constrained by the maximum allowed latency bound (L_b) and do not pass through any other controller that is not the source or target of the switch-controller pair. Meanwhile, possible data paths do not pass through any controller in the network. In addition, the subset of paths from each forwarding node to all the controllers in the network is stored in P_c (line 3). Using these computed control paths, in line 5, a sorted list of forwarding nodes is stored in L. This list is sorted in ascending order following two criteria:

1. the number of possible controllers to associate with,
2. the number of possible control paths.

Going through this list, the algorithm starts satisfying the most critical cases and the solution can be found with fewer iterations. Then, after setting the first element in L as the node n to be initially considered (line 6), the algorithm initializes the set of active links with a maximum value.

The main loop of the algorithm determines for each possible control path of the selected node n, the number of active links in the network after routing all data and control traffic. The configuration of paths with fewer active links is then selected in this process. Inside this loop, we will use a set of additional variables, denoted in the pseudocode as P', X', Y', U', to denote the temporal values computed at each iteration. Therefore, for each possible control path of the selected node n, these variables are initialized with the resulting values (i.e. control path, active links, controller-switch association and links utilization) derived from routing the considered

Algorithm 1 ENERGY-AWARE ROUTING

Require: $G = (V, E, C)$ network graph with controller placements, S forwarding nodes, D data traffic demands

Ensure: P data and control paths, X active links, Y controller-switch associations, U links utilization

1: $R \leftarrow$ Array of admissible routes between every pair of nodes in G
2: **for** $s \in S$ **do**
3:     $P_c[s] \leftarrow$ subset of routes $r \in R$ from s to every controller $c \in C$
4: **end for**
5: $L \leftarrow S\_Sorted$
6: $n \leftarrow$ First node in L
7: $len(X) \leftarrow \infty$
8: **repeat**
9:     **for** $p \in P_c[n]$ **do**
10:         Initialize(P', X', Y', U') by routing p
11:         **for** $s \in L - \{n\}$ **do**
12:             PATHSELECTOR$(s, None, P_c[s])$
13:         **end for**
14:         $O \leftarrow$ List of (c, s) in P'
15:         **for** $(c, s) \in O$ **do**
16:             PATHSELECTOR$(c, s, R[c, s])$
17:         **end for**
18:         **for** $(c, c) \in G$ **do**
19:             PATHSELECTOR$(c, c, R[c, c])$
20:         **end for**
21:         **for** $(s, s) \in D$ **do**
22:             PATHSELECTOR$(s, s, R[s, s])$
23:         **end for**
24:         **if** $len(X') \leq len(X)$ **then**
25:             $len(X), P, X, Y, U \leftarrow len(X'), P', X', Y', U'$
26:         **end if**
27:     **end for**
28:     **if** $len(X) = \infty$ **then**
29:         **if** $n =$ last node in L **then** break
30:         **end if**
31:         $n \leftarrow$ Next node in L
32:     **end if**
33: **until** $len(X) \neq \infty$

control path (line 10).

Four inner loops are included inside the main loop of the algorithm. The first inner loop is used to determine the path to a controller for each other forwarding node in L (line 12). Note that the possible control paths from each forwarding node are to any controller in the network to which it could be associated under the maximum allowed latency bound (L_b) considered for control paths. The path selected in this step precisely defines the controller for each forwarding node, which is stored in O (line 14). Using this switch-controller associations, the algorithm performs a second inner loop to determine now the control paths from each controller to its associated switches (line 16). Then, the two remaining loops are used to search the rest of required control paths (i.e. controller to controller in line 19) and the data paths (line 22). In the four cases, the paths selection is done using the PathSelector method, which will be further explained later on in this section.

Once all the required paths are computed, a checking process is used to evaluate the suitability of the current configuration of paths in line 24. That is, the algorithm compares the number of links required by the current iteration with the values stored in the global variables. If a smaller amount of links is found, the global variables are updated. In this way, the best values achieved after considering all control paths of node n are returned by the algorithm.

On the contrary, if after analyzing all control paths of node n, the algorithm still cannot find a feasible configuration of paths to route all control and data plane traffic flows, the main loop repeats this process for the next node stored in L. This is done until the solution is found or until all forwarding nodes are analyzed, i.e. when the algorithm breaks without a solution. Note that this last option occurs when, given a controller placement, an admissible configuration for controller-switches association could not be found or when the network has not sufficient capacity to meet the demand requirements under the established constraints.

As previously said, the paths selection is done by the PathSelector method described in Algorithm 2. This function is used to select for a pair of nodes (a, b), the best admissible route between them, denoted in the algorithm as SeP, in terms of minimizing the number of active links in the network. When this function is called for the first time, i.e. for determining the path between each forwarding node and one controller, the number of forwarding nodes already associated with the controller is considered (line 4 to 8). In this way, the controller load, in terms of managed forwarding nodes, remains balanced.

For each admissible path the number of required additional links is computed in line 9. This value is used to compare the current route with the previous ones and in case of improvement (i.e. fewer additional links are required), a new candidate solution is saved (line 13). In addition, a path can only be selected if it has sufficient bandwidth to route the demand volume, under the considered MLU constraint.

Finally, if no paths could be found, the algorithm skips to another control path of node

Algorithm 2 PATHSELECTOR$(a, b, Admisible_Paths)$

```
B ← ∞
SeP ← None
for p ∈ Admisible_Paths do
    if b = None then
        if p is related to an already loaded controller then
            continue
        end if
    end if
    off ← number of links in p that are not in X′
    if off ≤ B then
        if p has sufficient bandwidth then
            B ← off
            SeP ← p
        end if
    end if
end for
if SeP = None then
    continue to evaluate next p ∈ P_c[n]
end if
Update P′, X′, Y′, U′ by routing SeP
```

n for a new iteration. Alternatively, the algorithm returns the selected path and updates the considering variables with the values corresponding to the establishment of this new route.

3.3.2 Energy-Aware Location of a Single SDN Controller

In addition to the route scheduling used, the energy saving in a network is also impacted by the choice of the controller location. Thus, based on the proposed energy-aware routing approach, in this subsection the controller location problem is investigated. This analysis aims to define the best network nodes where to place the controllers and to yield the minimum power consumption. In particular, we evaluate the energy saving for all possible controller locations and select the one with the maximum value as the *energy-aware controller placement*.

Although the maximum energy saving is attained, if the size of the network is large, a thorough search among all locations becomes challenging to solve. Therefore, in Algorithm 3 we present a simple heuristic approach that reduces the space search to find the energy-aware controller location, considering the number of neighbor nodes and the connections between them. Since our model establishes that data plane communications cannot be routed through the network controller, locations with neighbors directly connected between them require a fewer

Algorithm 3 Controller Location

Require: $G = (V, E)$ network graph, $|C|$ number of controllers
Ensure: Controllers placement (C)

1: $Search\_Space \leftarrow$ NULL
2: $N_v \leftarrow$ Set of neighbors of node $v \in V$
3: $A \leftarrow$ Sorted list of nodes in ascending order of $|N_v|$
4: $h \leftarrow 0$ ▷ number of hops
5: **while** $|Search\_Space| < |C|$ **do**
6:     $neigh \leftarrow \infty$ ▷ number of neighbors
7:     **for** $v \in A \setminus Search\_Space$ **do**
8:         **for** $i, j \in Combinations(N_v, 2)$ **do**
9:             **if** i, j are connected through h hops and $|N_v| \leq neigh$ **then**
10:                 Add v to $Search\_Space$
11:                 $neigh \leftarrow |N_v|$
12:             **end if**
13:         **end for**
14:     **end for**
15:     increment h
16: **end while**
17: $ES \leftarrow 0$
18: **for** $c \in Combinations(Search\_Space, |C|)$ **do**
19:     $ES' \leftarrow$ Energy-Aware Routing$(C = c)$
20:     **if** $ES' > ES$ **then**
21:         $C \leftarrow c$
22:     **end if**
23: **end for**

number of links to meet the data and control traffic demands.

Based on this analysis and taking as inputs the network graph together with the amount of controllers to be placed, the algorithm initially defines the set of neighbors N_v for each node $v \in V$, which is used to create the list of nodes A sorted in an ascending node degree order (lines 2 and 3). Then, the variable h is initialised to a minimum value, which is used to indicate the values of hops between neighbors that will be incrementally considered along the algorithm operation. After performing this preparatory stage, the algorithm starts a while loop, which is meant to assure that the reduced list of possible locations ($Search\_Space$) has enough elements to place all the required controllers. Inside this loop, the temporal variable $neigh$, which is used to store the minimum value of neighbors found for a given value of h, is initialised with a maximum value (line 6).

In the next steps, the algorithm iterates over the ordered list A evaluating each network node that has not been already allocated in the set of candidate solutions (line 7). In particular, for each node $v \in A$, the algorithm then iterates over all the combinations of two nodes that can be drawn from the set of neighbors N_v (line 8). Using this combinations, it determines which are the nodes with neighbors connected through a given number of hops h (without considering the connections through v) and simultaneously selects the ones that also present the minimum $|N_v|$. In this way, the locations with neighbors connected between them, either directly or through the fewest number of intermediate elements and with fewer number of neighbor nodes (i.e. $|N_v|$), are stored in the *Search_Space* list (line 10). If more locations are still needed to place the required controllers, the value of h is incremented and the process previously described is repeated.

Once the length of the reduced search space is at least the same than $|C|$, the second part of the algorithm is preformed (lines 17 to 23). Using the list of nodes stored in *Search_Space*, the algorithm builds candidate solutions combining the selected nodes according to the amount of required controller locations (line 18). For each of these possible solutions, the algorithm determines the number of links needed using the proposed Energy-Aware Routing (Algorithm 1). The one with the greater energy saving is selected then as the most convenient controllers location.

3.3.3 Complexity Analysis

The computational complexity of the energy-aware routing presented in Algorithm 1 is determined by its main loop. In the worst case, this while loop will be executed S times, being S the number of forwarding nodes. However, it should be emphasised that as a consequence of iterating over an ordered list, in most of the cases the algorithm is able to find a solution after analyzing only the first node in L and the extreme case of executing S times will be quite uncommon. Inside this loop, the iterative process and the related complexity are directly linked to the number of connections to be established (i.e. K), which are considered along the inner loops, and the complexity introduced by the PathSelector method. Given that in Algorithm 2 a greedy search is performed, the complexity of the PathSelector method can be specified as $O(M)$, where M denotes the maximum number of admissible routes evaluated by this function. Please note that M cannot be found beforehand since it will depend of several factors such as the network topology, the number and location of controllers and the maximum allowed latency bound (L_b). Therefore, the overall algorithm complexity should be formulated based on it. Consequently, the worst run-time complexity of Algorithm 1 can be expressed as $O(SKM^2)$.

Regarding the energy-aware controller location, the complexity of the first part of Algorithm 3 (i.e. the generation of the reduced space search) is strongly dependent on the considered network topology. Considering H as the minimum number of hops between neighbors for at least C nodes, being C the amount of required controllers, and assuming a full mesh scenario as the worst case for the combinations of two nodes from the set of neighbors, this step has an upper-bound complexity of $O(HV^3)$, where V denotes the number of network nodes. Given that generating all combinations of r elements from a list of size n takes $O(n^r)$, the second part of Algorithm 3 has a complexity of $O(n^r SKM^2)$, where r and n denote the required number of controllers and the amount of nodes in *Search_Space*, respectively. Therefore, the overall algorithm complexity can be defined as $O(HV^3 + n^r SKM^2)$.

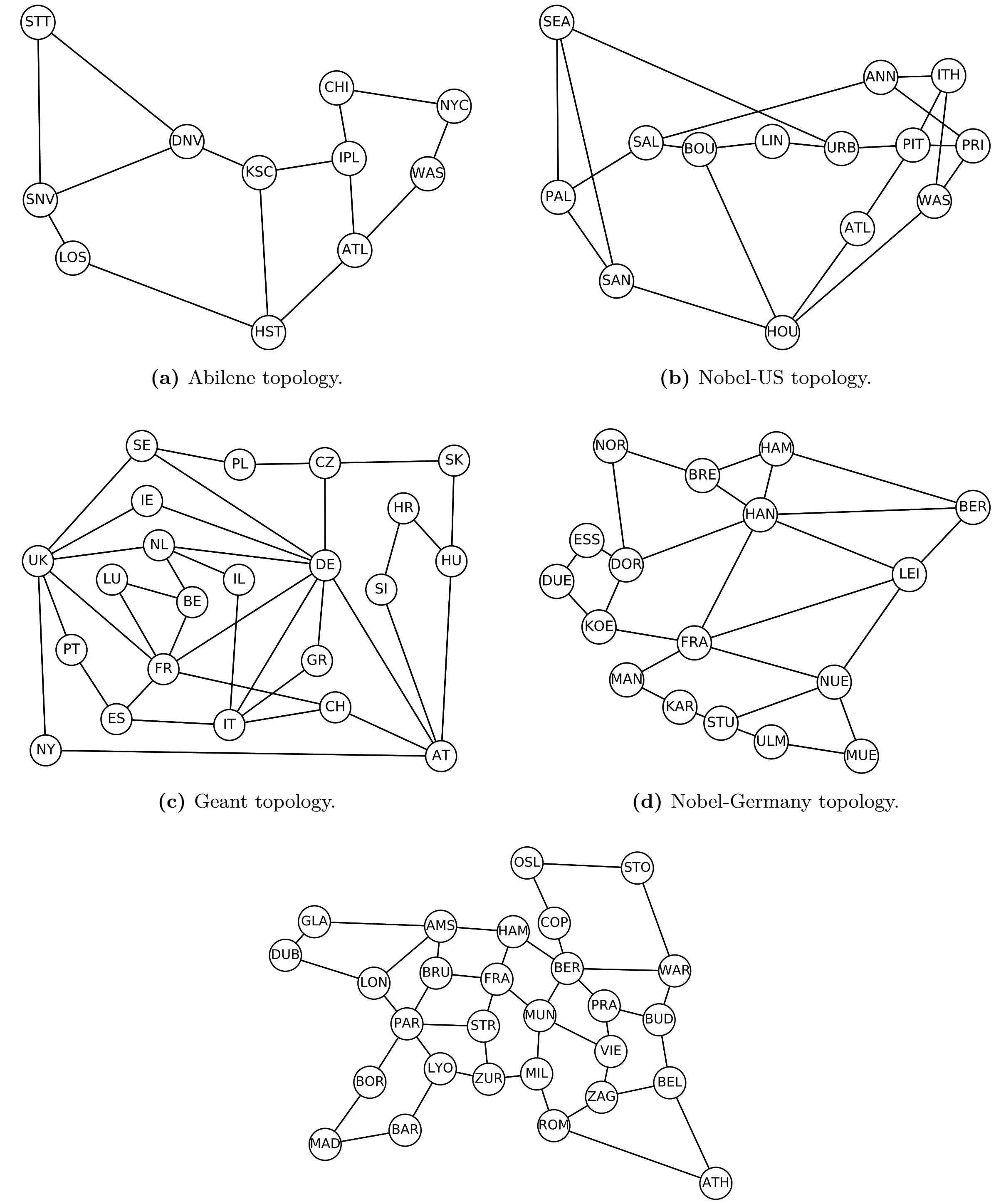
3.4 Simulations and Results

In this section we describe the evaluation of our energy-aware routing approach and analyze the results obtained. We used the linear programming solver Gurobi Optimizer [110] to assess the performance of the ILP models and the heuristic algorithms were developed using Python as programming language. All computations were carried out on a computer equipped with 3.30 GHz Intel Core i7 and 16 GB RAM.

We conducted our simulations using real-world network topologies collected from SNDlib [111], considering each router in the network as an SDN node or as a possible controller placement. The topology graphs considered along this section are shown in Fig. 3.5.

For the considered topologies, the traffic matrices have been obtained from specific files also downloaded from [111]. These files contain the directed data traffic demands between different origin/destination pairs of the considered network scenarios. To introduce the presence of control plane traffic, for each network topology we have replaced the data traffic demands from/to the node(s) acting at each instance as controller(s), by the considered control traffic rate, which is fixed to 1.7 Mbps according to the work in [112]. In case of assuming multiple controllers, the same operation is performed between each pair of nodes used as controllers.

Since the topologies used in our experiments are backbone networks, for the sake of simplicity and without loss of generality, we opted to compute the communication delay as the propagation latency. The energy savings were computed as the number of links in sleep mode over the total



(a) Abilene topology.

(b) Nobel-US topology.

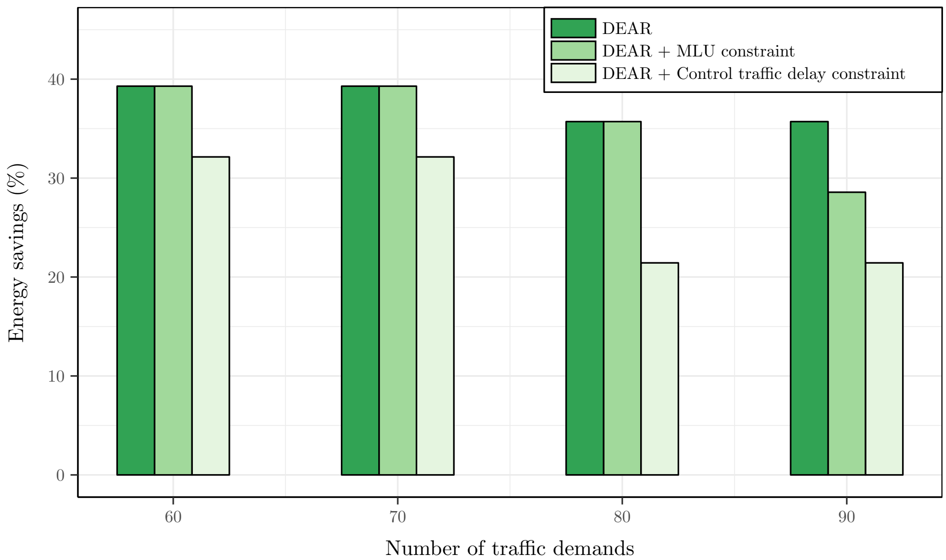
(c) Geant topology.

(d) Nobel-Germany topology.

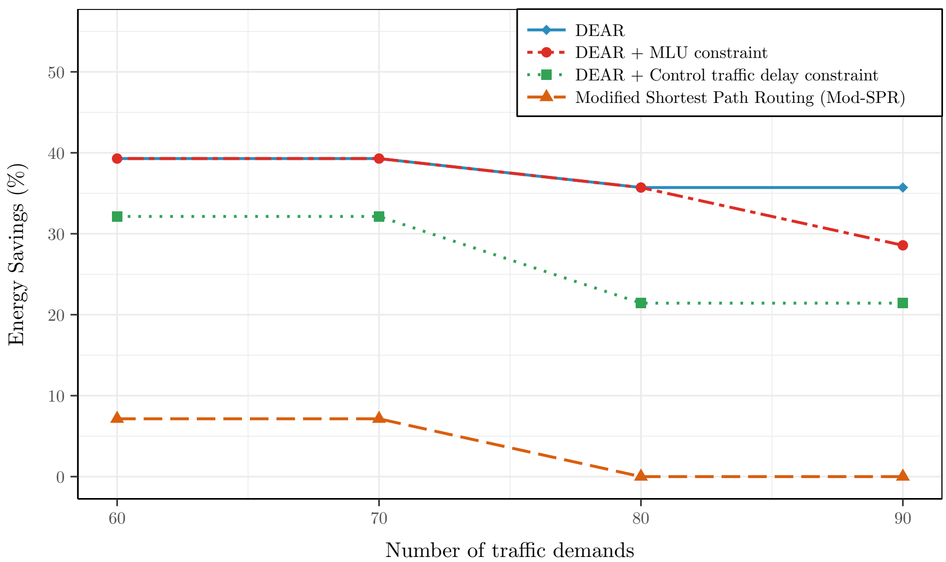
(e) Nobel-EU topology.

Fig. 3.5: Real-world network topologies used in the simulations.

amount of network links. In general, the results gathered in this section are obtained without blocking, i.e. all incoming traffic demands have been allocated for the different considered approaches.



(a) Percentage of shutdown links.



(b) DEAR vs. SPR.

Fig. 3.6: DEAR performance in the Abilene topology with two domains.

3.4.1 DEAR Performance

The evaluation of DEAR in the Abilene topology (11 nodes, 28 links) is shown in Fig. 3.6 for the case of having two controller domains. Two other versions of the algorithm with additional constraints (that is MLU and control traffic delay constraints) are also included in the figure. We used the subset of online available traffic matrices measured at 00:00 on March 1st 2004 [113].

Taking into account the geographical distribution of nodes in the Abilene topology shown in Fig. 3.5(a), we assume the two domains divided by Houston (HST) and Kansas City (KSC), which are therefore the border nodes. Then, a centralized controller is placed at each domain using the energy-aware controller location described in Algorithm 3. As a result, in the western domain the controller is placed at Seattle (STT), while Indianapolis (IPL) is the controller location for the eastern domain.

Results show that DEAR can save until near to 40% of energy consumption with low traffic. As expected, more restrictive constraints will imply less energy saving. This is due to, in order to meet the new performance requirements, a fewer number of alternate paths can be considered in the optimization. Therefore, it will be a trade-off to manage in accordance with the main objectives of each implementation.

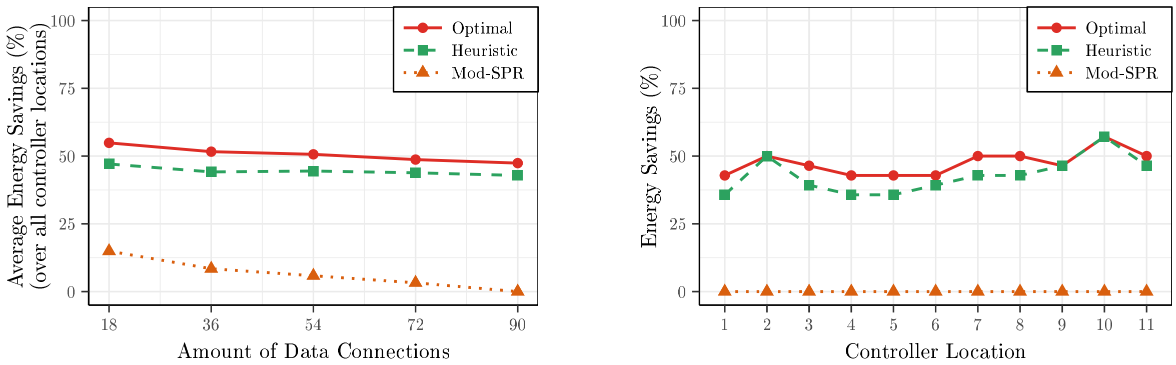
To get a sense of the energy saving values achieved by our approach, we also included the analysis of using a Modified Shortest Path Routing (Mod-SPR). Mod-SPR can be considered as a default SPR algorithm for SDN with in-band control traffic, where data plane traffic cannot be routed through any controller. We use Mod-SPR as a fair comparison in our evaluation since there is no research considering energy saving with in-band control traffic in SDN under the routing behaviour presented in our proposal.

Moreover, in all cases our distributed energy-aware routing approach outperforms the Mod-SPR in terms of energy saving. In general, DEAR achieves significant energy savings but bigger improvements over Mod-SPR are reached when the traffic grows.

3.4.2 Optimal vs. Heuristic Solutions

To evaluate the performance of our heuristic algorithms against the optimal solutions achieved by the ILP models we start considering the case of one centralized controller. To do so, besides the Abilene topology, we also include in this analysis the Nobel-US (14 nodes, 42 links) topology with its default traffic matrix provided in [111].

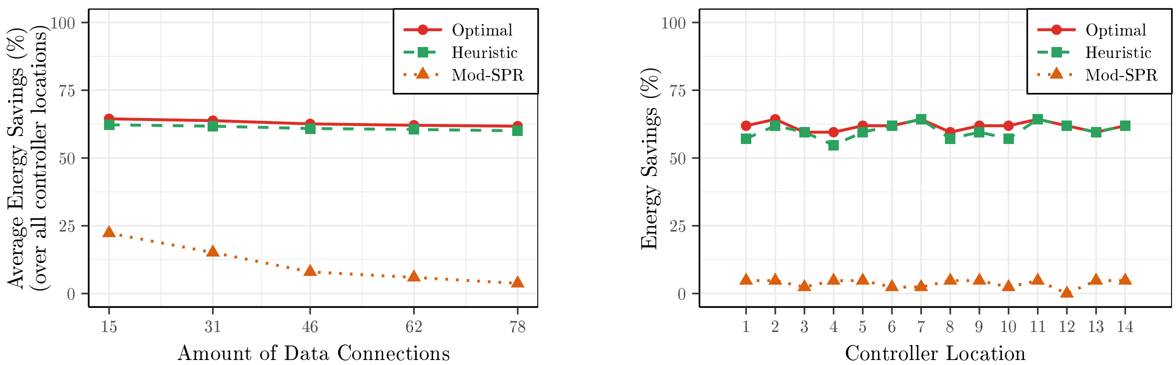
Fig. 3.7 and Fig. 3.8 show the energy savings reached by the routing models in the Abilene and Nobel-US topologies, respectively, varying: a) the amount of data traffic load and b) the controller location. In the first case we show the average values of energy savings computed after running the proposed solutions considering each network node as the controller (i.e. 11 simulations were performed in Abilene and 14 in Nobel-US). For the second case we fix the number of incoming data connections to the maximum provided value (i.e. 90 data traffic demands for Abilene and 78 for Nobel-US).



(a) Varying the amount of data traffic load.

(b) Varying the controller location.

Fig. 3.7: Energy saving comparison between optimal, heuristic and Mod-SPR models for one network controller in the Abilene topology.



(a) Varying the amount of data traffic load.

(b) Varying the controller location.

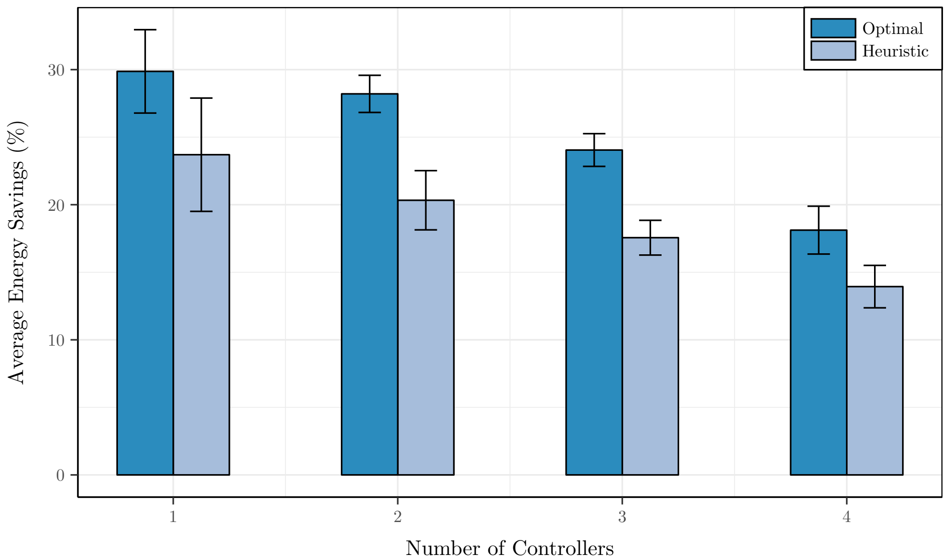
Fig. 3.8: Energy saving comparison between optimal, heuristic and Mod-SPR models for one network controller in the Nobel-US topology.

In both networks our energy-aware routing approach outperforms the Mod-SPR in terms of energy saving. Furthermore, for the case of only one network controller the heuristic algorithm accomplishes close-to-optimal average energy savings, with differences under 8%.

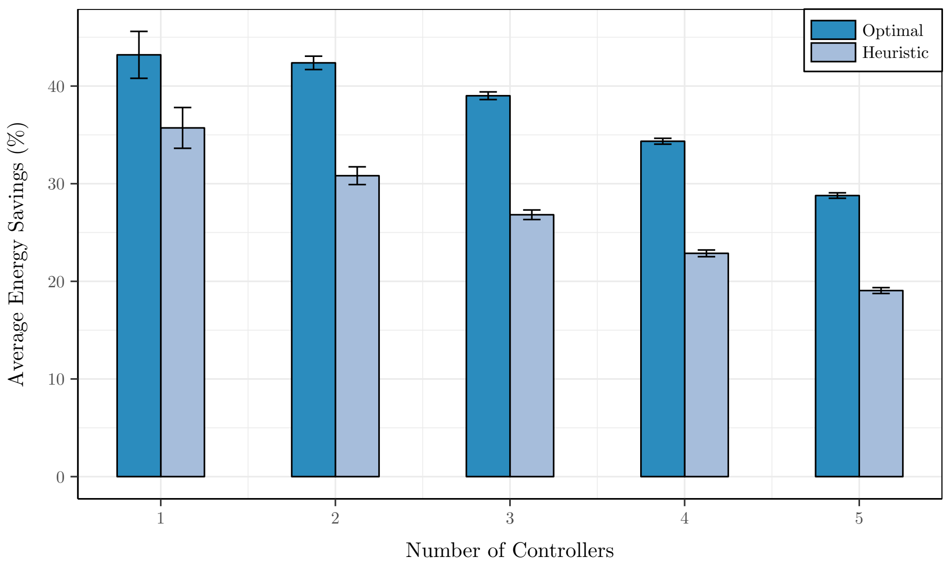
As expected, in Fig. 3.7(a) and Fig. 3.8(a) the energy saving decreases while the number of data flows grows, since new paths need to be established. The impact of controller placement on the power consumption is revealed in Fig. 3.7(b) and Fig. 3.8(b). In general, results have been determined with a 95% confidence interval not exceeding 5% of the indicated average values.

We now extend the analysis to the case of multiple logically-centralized controllers. To do so, in both networks we consider all possible controller placements (i.e. by which feasible solutions could be found) and compute the average energy savings for different numbers of controllers.

Fig. 3.9 shows the average values of energy saving and the obtained confidence intervals for a



(a) Average of shutdown links in the Abilene topology.



(b) Average of shutdown links in the Nobel-US topology.

Fig. 3.9: Average energy saving comparison between optimal and heuristic solutions for different numbers of controllers.

95% confidence level. In essence, the average energy savings reached by our optimization model are up to 30% and 43% in the Abilene and Nobel-US topologies, respectively. Moreover, our heuristic algorithm allows to obtain close-to-optimal energy savings in all cases, with differences under 13%.

As it is shown, energy savings decrease while the number of controllers grows. This behavior is expected given that in our approach data plane traffic cannot be routed through network controllers. Therefore, with the increase of network controllers a higher number of links, used to

Table 3.1: Average Execution Time (s) for different numbers of controllers on real topologies.

Topology	$\lvert C \rvert$	Simulations	Optimal	Heuristic
Abilene	1	11	0.6351166	0.02545899
	2	39	0.5631809	0.03049248
	3	60	0.4893286	0.03668678
	4	41	0.4268516	0.04307767
Nobel-US	1	14	8.068126	0.1452133
	2	89	6.307062	0.1227532
	3	318	4.995122	0.1465722
	4	711	4.150009	0.1727835
	5	927	3.445598	0.2240748
Geant	1	22	3446.204	305.5014
	2	217	2616.54	53.60402

route control traffic, cannot be used for data plane communications (i.e. links directly connected to the controllers).

Table 3.1 shows the average execution times required by the optimal model and the heuristic algorithm in the three network topologies considered in our experimental simulations. The number of simulations conducted in each case is also listed in the table for the different topologies. Although in all cases the heuristic algorithm is better in computation time, when the number of nodes and links is small, like in the Abilene and Nobel-US topologies, there is little difference between both approaches. However, as the network size grows, like in the Geant topology (22 nodes, 72 links), the processing time increases dramatically. For instance, in this topology the ILP model can take more than 3000 s to find solution while it is almost less than 300 s for the heuristic algorithm, i.e. one order of magnitude improvement.

For the Geant topology we only show the case of having one and two controllers, because beyond this limit the convergence time of solving the exact model considering all possible combinations of nodes as controllers placement, became unfeasible. This is because, although the computation time decreases as the number of controllers grows, the amount of possible combinations is considerably increased (i.e. 1273 simulations for $|\boldsymbol{C}| = 3$, 4977 simulations for $|\boldsymbol{C}| = 4$ and 13757 simulations for $|\boldsymbol{C}| = 5$).

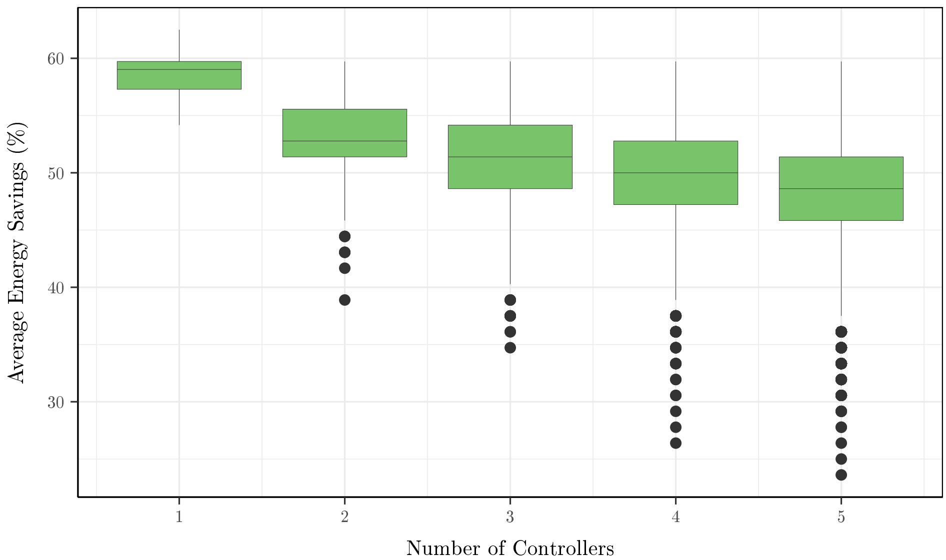


Fig. 3.10: Average energy saving in the Geant topology varying the number of controllers.

3.4.3 Performance in Large-Scale Topologies

Due to the computational complexity of the exact model showed above in networks similar or larger in size than Geant, we use our heuristic algorithm to test the energy efficiency in this topology considering the subset of online available traffic matrices measured at 00:00 on May 5th 2005. The average power saving potential of Geant for different number of controllers is then shown in Fig. 3.10.

This figure enables us to analyze the level of energy savings achieved, as well as their distributional characteristics for different amount of controllers. Note that when the number of controllers is between 2 and 5, the values of energy savings are approximately balanced around 50%. However, as the number of controllers in the network is increased, there are substantially more variation and outlier values. As previously stated, the maximum energy savings are attained when the network has a single controller.

In this network, higher energy savings than in the Abilene and Nobel-US topologies are achieved. The reason for this is that Geant has more link redundancy, therefore a higher number of alternate paths between each pair of nodes could be considered in order to reduce the number of links used in the network.

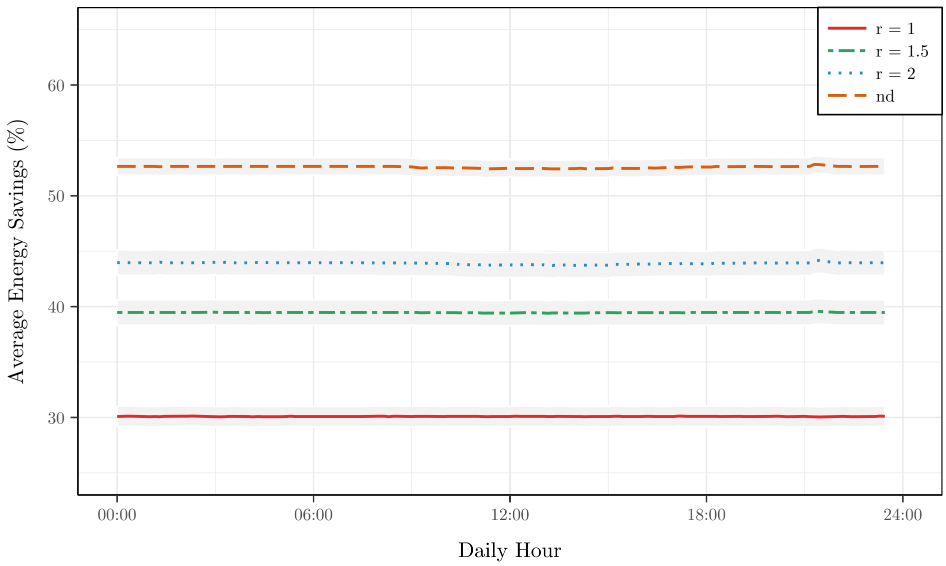


Fig. 3.11: Average energy saving in the Geant topology with two controllers during a day period varying the control paths delay bound.

3.4.4 Impact of Control Paths Delay Bound

So far, we had considered the control paths latency bounded by the network diameter, but now we analyze how the energy efficiency is affected varying this constraint. To do so, we use the daily set of traffic matrices measured every 15 minutes on May 5th 2005. Then, we ran our algorithm at each 15-min interval and collected the power savings for different values of control paths delay constraint (L_b). In this experiment we set to 100 the number of simulations conducted at each 15-min interval during this day period, each one considering a different placement of two controllers in the Geant topology. We use the notation r to denote the relation among L_b and the shortest path propagation latency for every control pair of nodes.

Fig. 3.11 shows the average energy savings using the network diameter (nd) as delay bound against three more restrictive possibilities ($r = 1, 1.5$ and 2). For instance, $r = 2$ means that every control path latency is, at most, twice that of the shortest path. In addition, error bands are included to reflect the 95% confidence interval of the obtained results. Since the traffic offered for the Geant topology is almost constant during a day period, the energy savings outlined present very few variations. As expected, less energy is saved when we use only shortest paths (i.e. $r = 1$) to route the control messages, but even then, energy savings of 30% could be achieved. This result shows that our approach enables considerable power savings without degrading the delay of control plane traffic.

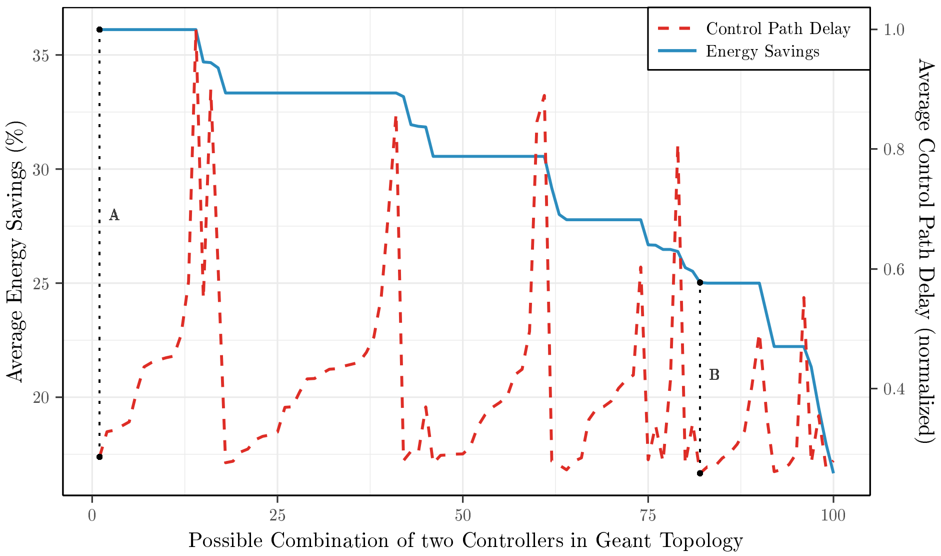


Fig. 3.12: Average energy saving and average control paths delay in the Geant topology for different controllers placement.

3.4.5 Impact of Controllers Placement

The controller placement, a key issue in SDN, has also a direct influence in the energy efficiency that can be achieved in network topologies. To better understand this behaviour, we use the previous 100 simulations for different placements of two controllers in the Geant topology when $r = 1$.

Fig. 3.12 shows, for the different controllers placement, the energy savings together with the average control path delay (normalized to a maximum value of unity). The x-axis enumerates the 100 possible locations of two controllers considered in our simulations. The index of this axis is according to the descending order of energy savings. The first controller pair emphasized at the left in the figure (with the letter A) presents the best performance in terms of both metrics for the considered sample. This point corresponds with the selection of United Kingdom and Czech Republic as controllers placement. We can observe a small difference in terms of average latency between this point and the one denoted as B, which in this subset of possible controller placements, achieves the minimum average control paths latency. However, the average energy saving can change in more than 10% between both points.

Although the energy saving and the control paths delay could be considered as opposing optimization aims, this solution enables the reduction of power consumption with minor impacts on delay of control plane traffic. Based on this approach the controllers can be placed

Table 3.2: Energy Saving (%) for different placement strategies on real topologies.

Topologies	**Abilene**	**Nobel-US**	**Nobel-Germany**	**Geant**	**Nobel-EU**
Nodes	11	14	17	22	28
Links	28	42	52	72	82
k-median	35.71	59.52	46.15	58.33	52.44
k-center	35.71	54.76	51.92	59.72	52.44
Algorithm 3	57.14	64.29	65.38	66.67	60.98
Ex_Search	57.14	64.29	65.38	66.67	62.20
Max_Improv	21.43	9.53	19.23	8.34	8.54

strategically in the network, considering these two metrics as requirements.

Table 3.2 shows the results of testing our energy-aware controller location approach (Algorithm 3) to place a centralized controller for 5 different network topologies. For each network, we also consider two other well-known controller placement strategies, namely k-median and k-center [109], which determine the node that minimizes the average and maximum control delay, respectively, as the controller placement. In addition, an exhaustive search among all locations (*Ex_Search*) was included to verify the maximum energy saving achieved in each topology using our energy-aware routing algorithm.

The four placement strategies use our energy-aware routing to establish the data and control paths, making them comparable models. Therefore, the difference between them relies only on the criterion to select the best controller placement. We compute the maximum improvement (Max_Improv) as the difference between the energy saving reached by our heuristic algorithm and the minimum value, achieved by the k-median or k-center methods.

As it is shown, in all cases our heuristic approach improves the energy saving achieved by the k-median and k-center strategies, with increases of around 20% of energy saving in Abilene and Nobel-Germany. Moreover, it achieves the maximum energy saving in almost all the topologies, except in Nobel-EU, where the maximum energy saving is achieved placing the controller at Munich or Brussels, locations that do not have the fewest number of neighbor nodes. Therefore, these locations do not appear in the *Search_Space* list formed by our algorithm. Even so, differences are under 1.5%.

3.5 Conclusion

In this chapter we proposed several energy-aware routing approaches minimizing the number of active links required to route the control and data plane communications for large-scale SDN with in-band control traffic. The proposed approaches comprise the different control plane implementations in SDN, i.e. single controller, multiple domains managed by a single controller and multiple controllers under network-wide routing. In the last one, performance constraints that are crucial in the correct operation of SDN, such as maximum link utilization, bounded delay for the control plane traffic and load balance between controllers, were considered. To solve the problem at hand, we have formulated optimization models that integrate the specific routing requirements for data and control traffic. Due to the time complexity of exact models in large-scale topologies, we also developed heuristic algorithms that improve the power consumption in the network with results close to the optimal responses, while reducing the computation times. In addition, we derived a simple and efficient algorithm to find the best controller placement in terms of energy saving. Based on experimental simulations using real topologies and traffic demand, we have proved that our energy-aware approaches achieve significant energy savings and outperforms the SPR with noticeable improvements. In addition, results showed that the heuristic algorithms converge much faster and can handle larger network sizes for which the exact model cannot find solutions in reasonable time. Moreover, we proved that energy consumption depends on the specific controller location, and the proposed algorithm for controller placement attains comparably good results. Using this approach, for a given traffic demand, controllers can perform energy-aware routing and determine the link interfaces that should be put into sleep mode. In this way, an energy-aware control plane could be achieved.

Chapter 4

Impact of Energy-Aware Routing on SDN Performance

This chapter is based on:

- **A. Fernández-Fernández**, C. Cervelló-Pastor and L. Ochoa-Aday, "Evaluating the Impact of Energy-Aware Routing on Software-Defined Networking Performance," in *Proc. of the XIII Jornadas de Ingeniería Telemática (JITEL'17)*, Valencia, Spain, Sep. 2017, pp. 241–248.
- **A. Fernández-Fernández**, C. Cervelló-Pastor and L. Ochoa-Aday, "Energy Efficiency and Network Performance: A Reality Check in SDN-Based 5G Systems," *Energies*, vol. 10, no. 12, pp. 2132:1–2132:27, Dec. 2017.

4.1 Introduction

Despite consistent efforts to improve the network power efficiency, energy-aware routing techniques may lead to performance degradations when QoS requirements are neglected [31, 114]. Inspired by this reality, this chapter introduces a novel energy-aware strategy which will be used to evaluate its impact on crucial performance metrics.

Instead of restricting the path selection and potential improvements in terms of energy efficiency to meet specific metric bounds –as previously done in Chapter 3– this proposal aims to get insight into energy saving capability and to quantify the existing trade-off between power consumption and several performance indicators, which is a crucial issue for communication systems nowadays.

Looking for an energy efficient and low-complexity approach, in this chapter we propose a hybrid strategy comprising topology and traffic-based decisions. Additionally, it is also fully compatible with current dynamic networking environments considering an SDN architecture with multiple controllers and in-band control traffic.

On the other hand, the complexity of considering the entire topology for the selection of the most suitable routes in the models presented in Chapter 3 can be very expensive in networks with major path redundancy. Alternatively, in this work the network topology is strategically pruned, which reduces the number of paths and the consequent computation complexity.

The rest of this chapter is structured as follows. In Section 4.2 the energy consumption optimization problem is formalized through a general link-based mathematical formulation. In Section 4.3 we explain the main features of our low-complexity energy-aware approach together with a detailed description of its two comprised modules. The simulation strategies and the obtained results are presented and analyzed in Section 4.4. Finally, Section 4.5 concludes this chapter.

4.2 Problem Statement

To formalize the energy consumption optimization problem in this Section we provide a link-based mathematical formulation intended to reduce the overall complexity of the exact approach in network topologies with large-scale path redundancy. Additionally, QoS constraints and performance metric boundaries are not considered in this model, as it aims to provide upper bounds for the energy efficiency.

4.2.1 Network Model

We consider an SDN represented by a directed graph $G = (V, E, C)$, where V, E and C denote the set of nodes, links and controllers respectively, being $C \subset V$. We use $c_{i,j}$ to denote the capacity of a link $(i, j) \in E$. We define the set of forwarding nodes as $S = \{n \mid n \in V \wedge n \notin C\}$.

Considering F as the entire set of traffic flows existing in the network between any pair of nodes, let D denote the subset of data plane communications. For the control plane, we use T to denote the subset of traffic between controllers and switches, and H to denote the subset of traffic between controllers. Accordingly, $F = D \cup T \cup H$. Each flow $f \in F$ from source s_f to destination t_f, has associated its throughput, denoted by b_f.

4.2.2 Formulation

To optimize the number of links used to route a given traffic demand matrix we develop an ILP model, using the following binary variables:

$x_{i,j}$: describes the state of a link $(i,j) \in E$.

$$x_{i,j} = \begin{cases} 1 & \text{if } (i,j) \text{ is active,} \\ 0 & \text{otherwise.} \end{cases}$$

$t^f_{i,j}$: describes the selection of a link $(i,j) \in E$ to route a flow $f \in F$.

$$t^f_{i,j} = \begin{cases} 1 & \text{if } (i,j) \text{ is selected to route } f, \\ 0 & \text{otherwise.} \end{cases}$$

$\lambda_{n,c}$: describes the association of each forwarding node $n \in S$ with a controller $c \in C$.

$$\lambda_{n,c} = \begin{cases} 1 & \text{if } n \text{ is associated with } c, \\ 0 & \text{otherwise.} \end{cases}$$

Considering the entire set of demands fixed and known in advance, all the optimal control and data paths in terms of energy efficiency can be computed jointly in a global optimization process. Given the notation of binary variables given above, the optimization model can be formulated as follows:

$$\text{minimize} \sum_{(i,j)\in E} x_{i,j} \tag{4.1}$$

subject to the following constraints:

To manage each forwarding node in the network $n \in S$, a single controller is selected.

$$\sum_{c\in C} \lambda_{n,c} = 1 \qquad \forall n \in S \tag{4.2}$$

Additionally, the number of switches associated with each controller cannot exceed the controller capacity. In this expression we use R_c to denote the computational and networking resources, in terms of number of forwarding nodes that can be supported by a controller $c \in C$.

$$\sum_{n \in S} \lambda_{n,c} \leq R_c \qquad \forall c \in C \tag{4.3}$$

To avoid additional traffic load through network controllers, data plane communications (i.e. $f \in D$) cannot be routed through these devices. Furthermore, control traffic between controllers and switches (i.e. $f \in T$) will not pass through any other controller that is not the source or target of the traffic. The same must hold true for the traffic between controllers (i.e. $f \in H$). In these constraints we use $N(i)$ to denote the set of neighbors of a node i and n_f to identify the forwarding node involved in the source/target pair of traffic flow $f \in T$.

$$\sum_{j \in N(i)} t_{i,j}^{f} \leq \begin{cases} 0 & \forall f \in D, \forall i \in C \\ \lambda_{n_f,i} & \forall f \in T, \forall i \in C \\ 0 & \forall f \in H, \forall i \in C \setminus \{s_f, t_f\} \end{cases} \tag{4.4}$$

The routing of data plane communications and control traffic exchange between controllers, follows the traditional flow conservation constraints.

$$\forall i \in V, \forall f \in D \cup H :$$
$$\sum_{j \in N(i)} t_{i,j}^{f} - \sum_{j \in N(i)} t_{j,i}^{f} = \begin{cases} 1 & \text{if } i = s_f \\ -1 & \text{if } i = t_f \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

Meanwhile, for the subset of control plane communications between controllers and switches $f \in T$, these constraints are modified to assure that each switch exchanges control messages only with its controller. Similarly, the forwarding node and controller involved in the source/target pair of traffic flow $f \in T$, are denoted with n_f and c_f, respectively.

$$\forall i \in V, \forall f \in T :$$

$$\sum_{j \in N(i)} t_{i,j}^{f} - \sum_{j \in N(i)} t_{j,i}^{f} = \begin{cases} \lambda_{n_f,c_f} & \text{if } i = s_f \\ -\lambda_{n_f,c_f} & \text{if } i = t_f \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

A link (i,j) is active if it is used by some traffic flow $f \in F$. Furthermore, the total traffic in each active link must be less than its assigned capacity.

$$\sum_{f \in F} t_{i,j}^{f} b_f \leq c_{i,j} x_{i,j} \qquad \forall (i,j) \in E \tag{4.7}$$

Using this model, the centralized controller can determine the optimal routes and set the required flow rules on each forwarding node before the traffic arrival. However, considering the high complexity of the presented optimization problem [49], the definition of heuristic solutions is needed to solve it on current real-world networks.

4.3 Heuristic Algorithms

In this section we present a novel solution for the energy efficiency problem in SDN comprising topology and traffic-based decisions. More precisely, we exploit specific network topological properties combined with the use of TE to reduce the overall power consumption.

An illustrative diagram of this strategy is shown in Fig. 4.1. The first component, denoted as Static Network Configuration Algorithm (SNetCA), is a topology-based solution intended to be statically activated at specific instances as a planned operation. On the other hand, the traffic-based module, denoted as Dynamic Energy Saving Routing Algorithm (DESRA), is activated by the arrival of each incoming traffic demand. Therefore, an accurate prediction of incoming traffic is not needed.

In essence, this approach finds the routes between network elements that minimize the number of active links used, considering that links are shared between data and control plane traffic (i.e. in-band mode). Therefore, control paths between controllers and switches (in both senses) and between controllers are also established.

Additionally, given the controllers placement in the network topology, our model determines the ideal distribution of switches between controllers in terms of energy efficiency and load

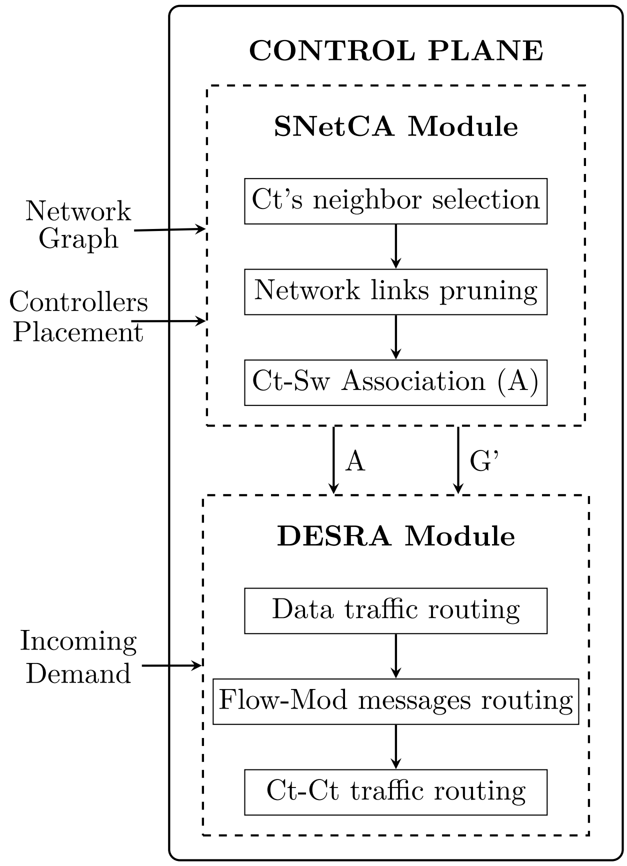


Fig. 4.1: Illustrative diagram of the proposed approach.

balancing among controllers. In our energy-aware approach, the routing of additional traffic load through controllers is avoided, i.e. admissible control paths do not pass through any other controller that is not the source or target of the traffic and data plane communications cannot be routed through these devices. The two main parts enclosed within the proposed energy-aware approach are described in more details in the following subsections.

4.3.1 Static Network Configuration Algorithm (SNetCA)

By considering the typical link redundancy of backbone networks, we design a Static Network Configuration Algorithm, denoted as SNetCA, which aims to prune as many links as possible in order to stress the importance of energy saving. Additionally, the most favorable switch-controller associations in terms of energy efficiency and load balance, are determined in this stage.

The algorithm, described in the Algorithm 4 pseudo-code, is composed of three steps:

1. selecting one of the controller's neighbors, as the node that will remain connected to it in

the outcome topology;

2. identifying the links that do not disconnect the graph to be put into sleep mode;

3. associating each node with one controller and computing the control path between them.

The input of the Algorithm 4 is the network topology with controllers placement and its outputs are a pruned network with a reduced number of links, denoted as G', an array keeping the controller-switch associations, denoted as A, and the control paths from each node to its controller, denoted as P_{sc}. Additionally, we use X to denote the set of active links $X \subseteq E$ and U to store the utilization of network links.

In the first step, the algorithm iterates over the set of network controllers in order to evaluate each one of its neighbors (lines 4-20). The selection of one neighbor node for each controller is based on the betweenness centrality, which measures the intermediary role of a node in the network. In the proposed approach, we use a simplified version of this metric considering only the shortest paths from a controller to every switch. The array B defined on line 5 is used to record the betweenness centrality computed by the algorithm for each neighbor of the considered controller.

In particular, after computing the shortest paths from one controller as single source in line 6, the algorithm determines whether a neighbor node belongs to each of these paths and increases the B_n associated with that node (lines 7-13). Note that when a controller's neighbor is another controller, the link between them is not considered as a candidate to be pruned (line 7).

For each controller the list of neighbor nodes is sorted according to the decreasing order of B_n and stored in L (line 14). This list is then used to identify the neighbor with the highest betweenness centrality, which will be the node that will remain connected to the controller in the resulting pruned topology. To do so, in line 16 the algorithm selects, for the controller considered in the current iteration, the first node in L and stores this pair in Sel_N. For the remaining nodes in L, the links between them and the controller are removed from the resulting network graph (line 18). This means that they are put into sleep mode in the original graph.

In the next step, the algorithm iterates over the set of directional links in the pruned network that do not have any controller as its extreme nodes (lines 21-28). At each iteration the algorithm attempts to increase the number of switched-off edges. A new link is removed only when the resulting graph remains being strongly connected, i.e. at least one path exists between every

Algorithm 4 SNetCA

Require: $G = (V, E, C), S$

Ensure: $G' = (V, E', C)$, A, P_{sc}

1: $N_c \leftarrow$ Set of neighbors of controller $c \in C$
2: $Sel\_N \leftarrow$ NULL ▷ Array of selected switch for each controller
3: $G' \leftarrow G$

Step 1

4: **for** $c \in C$ **do**
5: $B \leftarrow$ NULL ▷ Array of betweenness values
6: $SP_c \leftarrow$ Set of shortest paths from controller $c \in C$ to every other node $v \in V$
7: **for** $n \in N_c \setminus C$ **do**
8: $B_n = 0$
9: **for** $p \in SP_c$ **do**
10: **if** path p goes through node n **then** $B_n = B_n + 1$
11: **end if**
12: **end for**
13: **end for**
14: $L \leftarrow N_c\_Sorted$ according to decreasing order of B
15: **for** $s \in L$ **do**
16: **if** c not already in Sel_N **then** $Sel\_N = Sel\_N \cup (s, c)$
17: **end if**
18: Remove links (s, c) and (c, s) from G'
19: **end for**
20: **end for**

Step 2

21: **for** $i, j \in E'$ **do**
22: **if** $i \in C$ or $j \in C$ **then** continue
23: **end if**
24: $G'' \leftarrow G'$
25: Remove controllers $c \in C$ and link i, j from G''
26: **if** G'' remains strongly connected **then** remove link i, j from G'
27: **end if**
28: **end for**

Step 3

29: **for** $s, c \in Sel\_N$ **do**
30: **if** s is already associated with another $c' \neq c$ **then** continue
31: **end if**
32: PathSelector(s, c)
33: Update P_{sc}, A, X, U
34: **end for**
35: **for** the rest of $s \in S$ **do**
36: PathSelector(s, C)
37: Update P_{sc}, A, X, U
38: **end for**

pair of nodes in the network. To accomplish this, a temporal graph, denoted as G'', is created in line 24. This graph is used to check the required connectivity between all the forwarding nodes without passing through any controller in the network. Thus, after removing from G'' the network controllers and the considered link in line 25, the algorithm validates whether the possibility of reaching any node in the network is not affected. This being the case, the considered link is removed from the resulting graph (line 26).

The last step of the algorithm is intended to determine a control path from each forwarding node to one controller (lines 29-38). To achieve this goal, the algorithm starts evaluating the pairs of controller-switch already stored in Sel_N (line 29). For each pair, the algorithm first determines whether the considered switch is still available. If this is the case, in line 32 an admissible control path minimizing the number of active links is computed for this switch-controller pair using the method PathSelector described in Algorithm 5, which will be further explained below. As stated previously, admissible control paths do not pass through any other controller that is not the source or target of the traffic.

The remaining forwarding nodes are then considered by the algorithm in line 35. Note that in this case the algorithm takes into account the control paths to all controllers in the network (line 36). Precisely, the path computed by the PathSelector in this step defines the controller for the rest of forwarding nodes. After computing the selected paths for the two aforementioned loops, the different involved sets (P_{sc}, A, X, U) are updated according to the performed routing decisions (line 33 and 37).

Using this initial control plane configuration, switches send to the controller requests through packet_in messages when a new traffic flow arrives, as well as statistics and failure notifications. Consequently, there is an initial set of active links in the network before the ingress of traffic flows as well as some link utilization.

The PathSelector method, first introduced in Section 3.3.1, performs the energy-aware path selection. In essence, this function is used to select the best admissible route between a pair of nodes, in terms of minimizing the number of active links in the network.

The key idea of this function is to perform a low-complexity greedy evaluation between all the admissible paths to select the most suitable route in terms of energy-efficiency, while guaranteeing a balanced load of switches between controllers and the capacity constraint of links. Since this method works over the pruned network with a reduced number of links, (i.e. G'), the considered set of admissible paths is significantly smaller than in the original topology and the solution can be found with fewer iterations.

Algorithm 5 PathSelector(a, b)

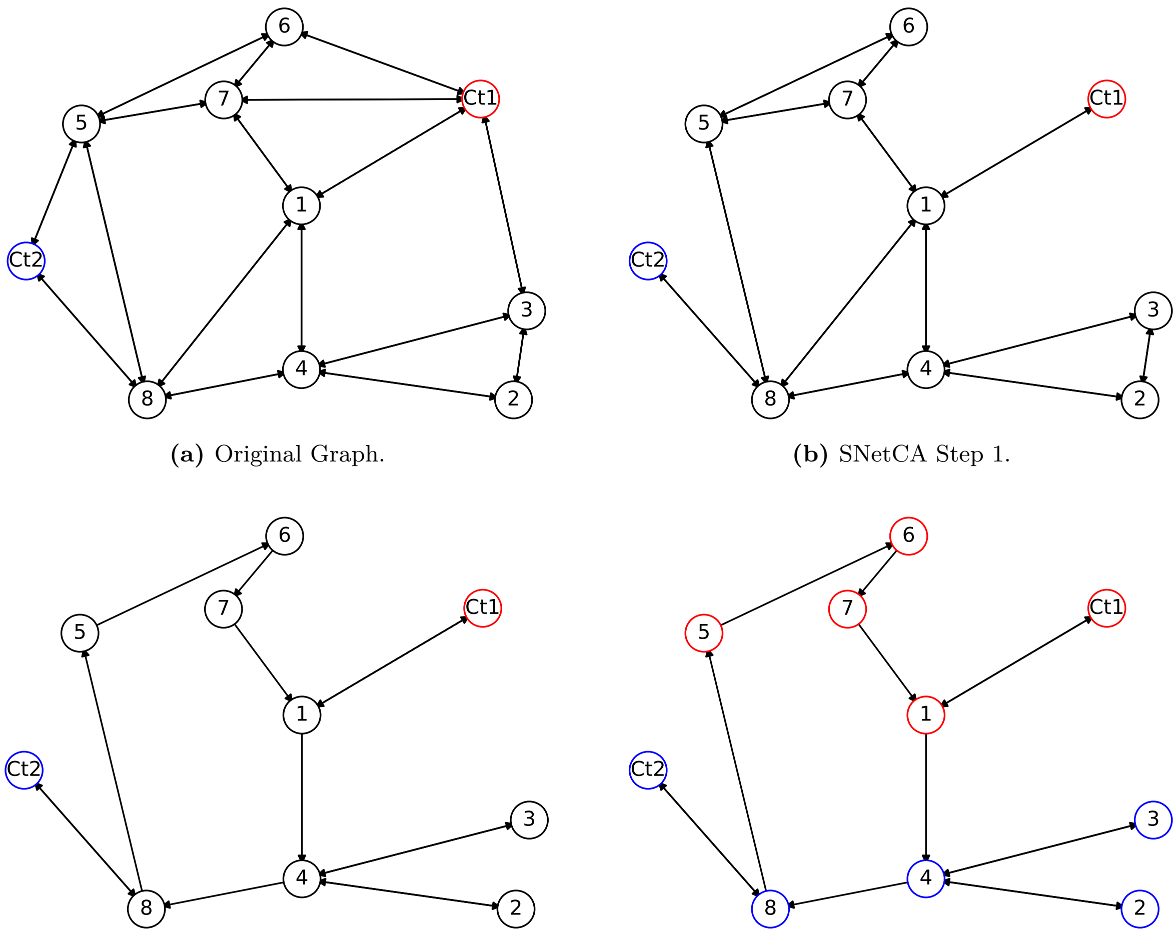
1: $L \leftarrow \infty$
2: $SeP \leftarrow None$
3: **for** $p \in$ Get_All_Admissible_Paths(G', a, b) **do**
4:     **if** $b = C$ and p is related to an already loaded controller **then**
5:         continue
6:     **end if**
7:     *off* $\leftarrow$ number of links in p that are not in X
8:     **if** *off* $< L$ and p has sufficient bandwidth **then**
9:         $SeP \leftarrow p$
10:         $L \leftarrow$ *off*
11:     **end if**
12: **end for**

When this function is called for determining the path between a forwarding node and a non-specific controller (i.e. using the set of controllers C as the traffic destination in the line 36 of Algorithm 4), the controller load, in terms of managed forwarding nodes, is analyzed in line 4. In this way, each path related to a controller with a number of already associated forwarding nodes equal to the given threshold, is not further considered.

In general, for each admissible path in G' between node a and b the algorithm computes the number of links required by this path that are not included yet in the set of active links X (line 7). This number, which represents the amount of additional links to be activated if the incoming flow is allocated over the considered path, is stored in the variable *off*. Then, each time we go through line 8, the algorithm checks whether exists an admissible path with an *off* value smaller than the one already selected during one of the earlier passages through this step. If the considered path is more energy efficient (i.e. requires the activation of fewer links) than the current recorded path, that path is replaced with this new one in SeP (line 9) and the variable L is populated with its *off* value to keep the record of the smaller amount of required links (line 10). In addition, the path only can be selected if it has sufficient link capacity to route the required traffic volume (line 8).

To better illustrate the operation of our static approach proposed in Algorithm 4, Fig. 4.2 provides a detailed example which goes across each one of the three steps explained above.

In Fig. 4.2(a) we show the original network with 8 forwarding nodes, 2 controllers ($Ct1$, $Ct2$) and 34 directed links. Then, after executing the first step of SNetCA, only one neighbor node remains connected to each controller as can be corroborated in Fig. 4.2(b). This step implied a



(a) Original Graph.

(b) SNetCA Step 1.

(c) SNetCA Step 2.

(d) SNetCA Step 3.

Fig. 4.2: Example of SNetCA operation.

reduction of 8 links with respect to the original graph.

The second step, exemplified in Fig. 4.2(c), identifies the links that do not affect the data plane connectivity between forwarding nodes and removes all these links from the considered graph. Consequently, the number of links in sleep mode is considerably increased by this step. Specifically, only 15 links remains active in the resulting pruned network, which represents around 56% of energy saving (in terms of number of active links).

Finally, the controller-switch associations determined by the step 3 are shown in Fig. 4.2(c). In this figure, the distribution of switches between controllers is depicted through colors, indicating that exactly 4 nodes are associated with each controller. In this way, the load of switches is evenly distributed among controllers. This step also establishes a control path between each forwarding node and its controller.

Algorithm 6 DESRA

Require: G', A, d incoming traffic request
Ensure: P_{ss}, P_{cs}, P_{cc} data and control paths, X active links, U links utilization

1: $Ct_1 \leftarrow A[s_d]$
2: $p = \textsc{PathSelector}(s_d, t_d)$
3: Update P_{ss}, X, U
4: **for** $n \in p$ **do**
5:     $Ct \leftarrow A[n]$
6:     $\textsc{PathSelector}(Ct, n)$
7:     Update P_{cs}, X, U
8:     **if** $Ct_1 \neq Ct$ **then**
9:         $\textsc{PathSelector}(Ct1, Ct)$
10:         Update P_{cc}, X, U
11:     **end if**
12: **end for**

4.3.2 Dynamic Energy Saving Routing Algorithm (DESRA)

When a new traffic demand arrives, a routing request is sent from the input node to its associated controller using the path between both devices previously computed during the static network configuration phase. Based on its global knowledge of the network topology, this controller calculates the required data path minimizing the number of links that need to be activated for this connection request and creates the flow forwarding rules.

The proposed dynamic energy-aware routing is shown in Algorithm 6. For an incoming demand d from source s_d to destination t_d, the algorithm starts storing in Ct_1 the controller associated with the source node (line 1). This controller is the main responsible of managing this traffic request. Using the PathSelector method in line 2, the most favorable admissible data path in terms of energy consumption is computed. This is done considering that admissible paths do not pass through any controller in the network.

Next, a loop is used to establish the required control plane communications for each node along this path (lines 4-12). Inside this loop the algorithm starts identifying the controller associated with each node in the data path (line 5). Then, in line 6, a control path is computed from the identified controller to the considered node. These paths are needed to set the required flow forwarding rules in each switch using the flow_mod messages.

Given the multidomain scenario considered, the nodes traversed by the data traffic may be associated with different controllers. When a node is not associated with Ct_1 (lines 8-11), an additional control message is sent from this controller to the other, in order to inform the second controller of the flow forwarding rule that needs to be installed in one of its managed nodes. To do so, in line 9, a control path is also computed between both controllers.

In each case, after computing the corresponding data or control path, the variables involved with such routing (i.e. established paths, active links and links utilization) are updated (lines 3, 7 and 10).

4.3.3 Complexity Analysis

Considering that the computation of the shortest paths from each single controller is done in $O(V \log V)$ using the Dijkstra algorithm, the selection of one controller's neighbor in the fist step of Algorithm 4 has a worst run-time complexity equal to $O(V \log V + V^2 + S \log S + S)$, where V is the number of total network devices and S is the number of forwarding nodes. Given that this operation is performed C times, being C the number of controllers, the complexity of step 1 becomes $O(CV^2)$.

Pruning as many links as possible without disconnecting the network graph during the second step has a complexity equal to $O(L(V+L))$, where L is the number of links which are not directly connected to any controller and $O(V + L)$ refers to the connectivity checking process [115].

The complexity of the last step is determined by $O(SM)$, where M indicates the worst-case complexity of Algorithm 5, i.e. the maximum number of admissible paths between a pair of nodes in the pruned graph. Although M cannot be found beforehand since it will depend of several factors such as the path redundancy of the network topology and the number and location of controllers, it should be noted that after pruning the network topology, the number of admissible paths and the consequent computation complexity of this method are significantly reduced with respect to the original graph. Therefore, the SNetCA complexity can be expressed as $O(CV^2 + L^2)$.

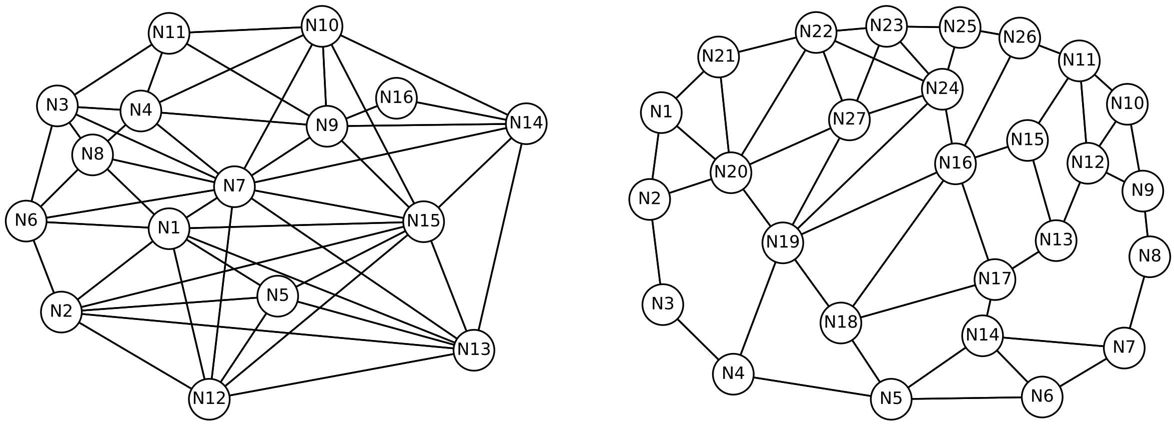
The complexity of the proposed dynamic routing solution, mixing together both Algorithms 5 and 6, is $O(SM^2)$ since the maximum length of a data path is precisely the number of forwarding nodes given the routing restrictions avoiding data traffic through network controllers. Given that M is usually a small number after pruning the network topology and does not grow rapidly along with the network size, it is reasonable to run the algorithm upon each flow request.

Based on the previous analysis, the overall algorithm complexity of the solution presented in this chapter, considering the static and dynamic parts, can be formulated as $O(CV^2+L^2+SM^2)$.

4.4 Simulations and Results

In this section we describe the evaluation of our energy-aware approach and analyze the achieved results. We used the linear programming solver Gurobi Optimizer [110] to assess the performance of the ILP model. Meanwhile, the proposed control framework described in Section 4.3 was implemented using the programming language Python to develop the heuristic algorithms. All computations were carried out on a computer equipped with 3.30 GHz Intel Core i7 and 16 GB RAM.

We conducted our simulations using real-world network topologies and traffic demands collected from SNDlib [111], considering each router in the network as an SDN node or as a possible controller placement. Since the topologies used in our experiments are backbone networks, for the sake of simplicity and without loss of generality, we opted to compute the communications delay as the propagation latency. Specifically, we use three of the most link-redundant network topologies existing in SNDlib in order to assess the effectiveness of the proposed scheme. The mentioned topologies are: New York, Geant and Norway. In Fig. 4.3 we show the network graphs of New York and Norway (for the Geant topology please see Section 3.4).



(a) New York topology. **(b)** Norway topology.

Fig. 4.3: Real-world network topologies used in the simulations.

In addition, the main characteristics of the three considered topologies are summarized in Table 4.1. This table presents the topological properties together with a general description of the provided traffic load for each studied topology. Specifically, for New York and Norway we use the default traffic matrices provided, while for Geant we select the subset of available traffic matrices measured at 00:00 on May 5th 2005.

Table 4.1: Main characteristics of considered topologies.

Topology	$\lvert V \rvert$	$\lvert E \rvert$	$\lvert D \rvert$	**Mean node degree**	**Traffic demand (Mbps)**	
					Total	**Mean**
New York	16	98	240	6.12	1774.0	7.39
Geant	22	72	430	3.27	42565.53	98.99
Norway	27	102	702	3.78	5348.0	7.62

As we are considering an in-band scenario, after placing the controllers in each simulation instance, we have deleted the data traffic demands from/to them. Being the network load an important parameter that impacts the efficiency of any energy-aware solution, we also compute the average network load ρ, defined as follows, where the traffic flowing on each link was obtained routing each traffic demand using the shortest path.

$$\rho = \frac{\sum\limits_{(i,j)\in E} \sum\limits_{f\in D} t_{i,j}^{f} b_f}{\sum\limits_{(i,j)\in E} x_{i,j} c_{i,j}} \tag{4.8}$$

Results from this early analysis confirm that the real scenarios considered present a low network traffic load. More in detail, the average network loads of the three network topologies are less than 0.05, being New York links particularly lightly loaded (i.e. $\rho < 0.01$). This behavior is typical in real backbone topologies where capacity planning strategies aim to ensure that core links are always significantly over-provisioned relative to the offered average load. This is done as an attempt to avoid congestion in case of peak load and to allow the fulfillment of Service Level Agreement (SLA) requirements. Intuitively, low loaded networks are suitable scenarios for deploying energy-aware solutions that concentrate traffic and turn off unused network links. Therefore, this analysis suggests that substantial energy savings are possible. Moreover, we can deduce that potential energy benefits will be more limited by the required connectivity and topological properties than by the network traffic load. For the control traffic we assume an average rate of 1.7 Mbps [112].

Considering a homogeneous scenario, where all controllers have the same computational and networking capabilities, in our simulations we set the maximum number of forwarding nodes that can be associated with each controller as follows. In this way, switches are evenly distributed and the load of switches is balanced among controllers.

$$R_c = \left\lceil \frac{|S|}{|C|} \right\rceil \qquad \forall c \in C \tag{4.9}$$

To analyze the performance of our energy-aware approach five groups of evaluations are presented varying the number of controllers in the considered topologies. In the first part, performance of our heuristic algorithms with respect to the optimal model are analyzed in order to fix an upper bound of the power saving capabilities of proposed solutions. In the second part, the potential of SNetCA to prune a network topology is investigated. Then, we compare our solution with the four strategies of another energy-aware method proposed in the literature. Next, we analyze the impact of our model on crucial network performance metrics, such as latency, link utilization and TCAM occupation. Finally, we provide an initial analysis exploring the existing trade-off between energy savings and network resilience.

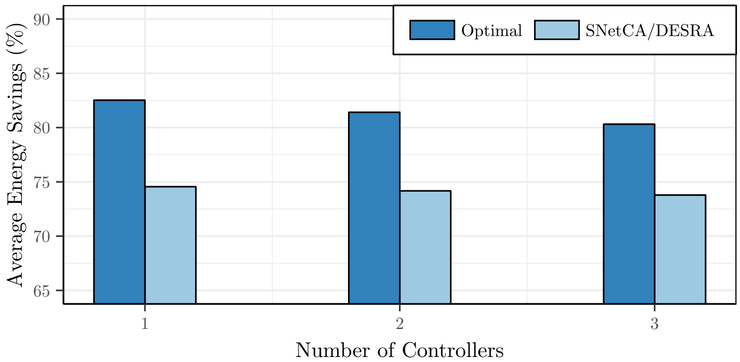
In general, the results gathered in this section are obtained without blocking, i.e. all incoming traffic demands have been allocated for the different considered approaches.

4.4.1 Optimal vs. Heuristic Solutions

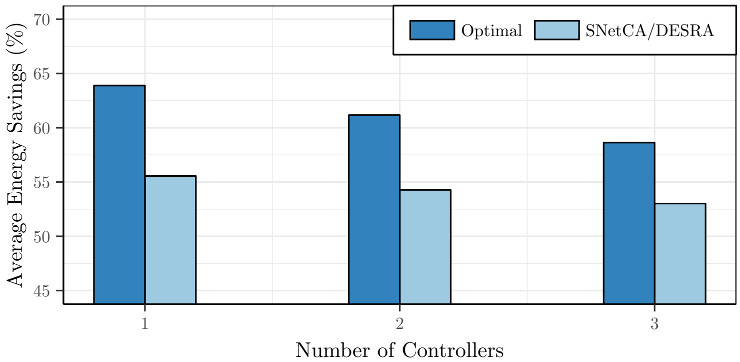
To assess the performance of proposed heuristic algorithms against the optimal solution achieved by the ILP model, we show their behaviors in Fig. 4.4 using the considered topologies.

We compute the average energy savings considering all admissible controller placements for different numbers of controllers. Note that a controller placement is admissible when the assumptions established in this proposal to avoid the routing of additional traffic load through network controllers can be kept (i.e. the network graph without any controller is strongly connected). Energy savings were computed as the number of links in sleep mode over the total amount of network links.

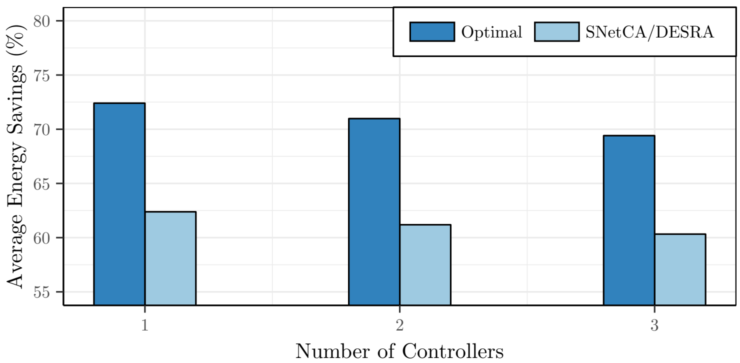
As shown in Fig. 4.4, the energy savings reached by the optimization model are up to 64%, 72% and 83% in the Geant, Norway and New York topologies, respectively. On the other hand the heuristic strategy (denoted in the figure as SNetCA/DESRA) allows to obtain close-to-optimal energy savings, with differences under 10% in all cases. In general, results have been determined with a 95% confidence interval not exceeding 1.1% of the indicated average values.



(a) New York topology.



(b) Geant topology.



(c) Norway topology.

Fig. 4.4: Average energy saving comparison between optimal and heuristic solutions for different numbers of controllers.

We can also see that energy savings slightly decrease while the number of controllers grows. This behavior is expected given that in our approach data plane traffic cannot be routed through network controllers. Therefore, with the increase of network controllers a higher number of links, used to route control traffic, cannot be used for data plane communications (i.e. links directly connected to the controllers). A similar decreasing behavior can be noticed in the gap between optimal and heuristic results since, as the amount of network controllers grows, a fewer number

Table 4.2: Average execution time (s) for different numbers of controllers on real topologies.

Topology	$\|C\|$	Simulations	Optimal	SNetCA	DESRA
New York	1	16	165.25394	0.06042855	0.7634286
	2	119	155.11156	0.05261856	0.6301443
	3	546	73.87162	0.04025800	0.5103288
Geant	1	22	111.58066	0.05155555	2.4344444
	2	217	77.18424	0.04857143	1.9748800
	3	1273	43.45415	0.04672522	1.5586752
Norway	1	27	9748.63320	0.09865218	8.9823043
	2	349	4828.40740	0.09627403	7.2662776
	3	1000	2246.13205	0.08228750	6.4095375

of feasible solutions can be considered by the linear solver.

Table 4.2 shows the average execution times required by the optimal model and the heuristic algorithms in the three network topologies considered in our experimental simulations. These values were computed using a 10% trimmed mean in order to reduce the effect of outliers on the central tendency. While the SNetCA column shows the execution times required to prune the network, the other two columns contain how many seconds are spent by each routing approach to compute all the required control and data paths according to the incoming traffic. As it is shown, in both approaches computation times mostly tend to decrease while the number of controllers grows. This is due to the fact that these energy-aware approaches avoid the routing of additional traffic load through network controllers. Therefore, with the increase of network controllers a fewer number of alternate paths between each pair of nodes need to be considered in the simulation.

Although in all cases the proposed strategy outperforms the optimal model in terms of computation time, a higher improvement is achieved as the network size grows. For instance, in the Norway topology the processing times required by the optimal model increase dramatically. The ILP model in this topology can take more than two hours on average to find a solution, which is a great limitation in current networking environments. Meanwhile, it is always less than 10 s for the heuristic strategy, i.e. almost a three order of magnitude improvement. This comparison validates the improvements achieved by the heuristic proposal in terms of computation time and clearly justifies its necessity.

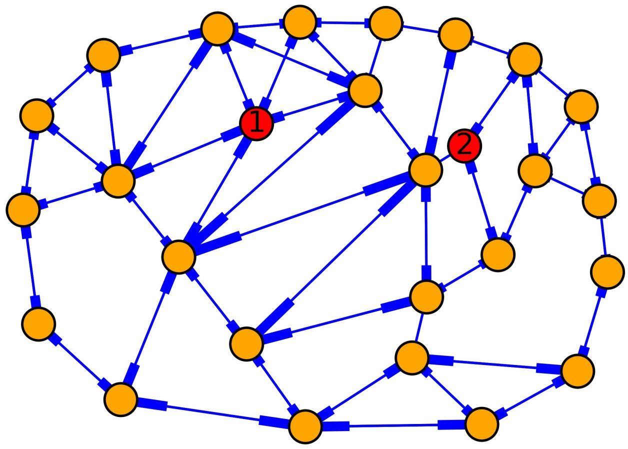
The number of simulations conducted in each case is also listed in Table 4.2 for the different topologies. This value represents the combinations of nodes considered as controllers placement for a given amount of controllers. A combination of nodes is an admissible controller placement as long as feasible solutions can be found under the required routing restrictions avoiding the routing of additional traffic through network controllers. For the Norway topology all possible combinations of 3 nodes as controllers were not considered in this subsection because the overall convergence time of solving the exact model for such amount of simulations (a total of 2844) became unfeasible. Instead, for the comparison with the optimal solutions we limit the analysis of this topology when $|C| = 3$ to 1000 simulations as shown in Table 4.2. In the following subsections the total amount of possible combinations are considered when referring to the evaluation of the heuristic approach.

4.4.2 SNetCA Performance

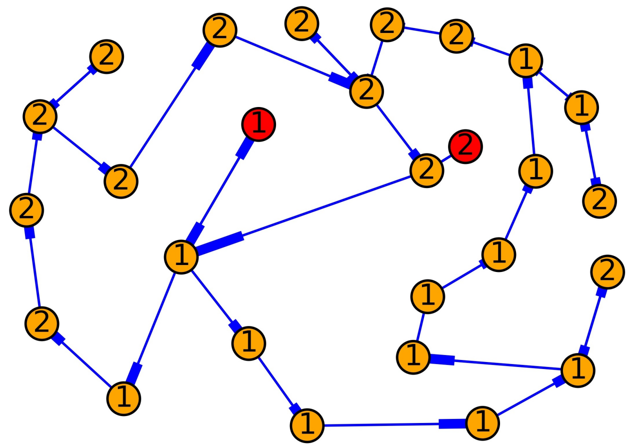
In order to evaluate the effectiveness of the proposed topology-based module, Fig. 4.5 shows an example of the performance of SNetCA on the Norway topology, considering two network controllers placed at nodes denoted as 1 and 2 and emphasized with a different color in the figure. The distribution of switches between controllers is depicted through the use of labels in each node, indicating the controller number to which the node is associated. Here we focus our attention on the Norway topology, but similar results have been obtained for the two remaining topologies considered.

A comparison between the original network and the resulting graph illustrated in Fig. 4.5(a) and Fig. 4.5(b) respectively, shows a difference of 67 edges, which represents more than 65% of total network links. These links are pruned by our algorithm guaranteeing that the resulting graph remains being strongly connected and avoiding additional traffic load through network controllers.

Additionally, as a result of applying SNetCA on the Norway topology, switches are distributed between controllers minimizing the number of required active links and ensuring a balanced load among controllers. For instance, 12 switches are associated with controller 1 while the remaining 13 are managed by controller 2. Regarding the average path length, in the pruned topology, an increase of 4.30 ms or, equivalently, 5 hops with respect to the original graph is incurred as a consequence of putting the links into sleep mode.



(a) Original Norway graph.



(b) Resulting Norway graph.

Fig. 4.5: Performance of SNetCA on the Norway topology.

To provide a more general perspective, Fig. 4.6 shows, for the three considered topologies, the average number of links pruned by SNetCA, which contributes directly with the energy efficiency achieved by this proposal. In this analysis we also consider all the admissible placements of 1 to 3 controllers.

As it is shown, an important number of links is pruned in all the topologies considered, but the highest energy savings are achieved in the New York topology. The reason for this is that New York has much more link redundancy than Geant and Norway. Therefore a higher number of links can be pruned while guaranteeing the network connectivity. In general, the more redundant the network, the higher number of links can be put to sleep mode applying this strategy.

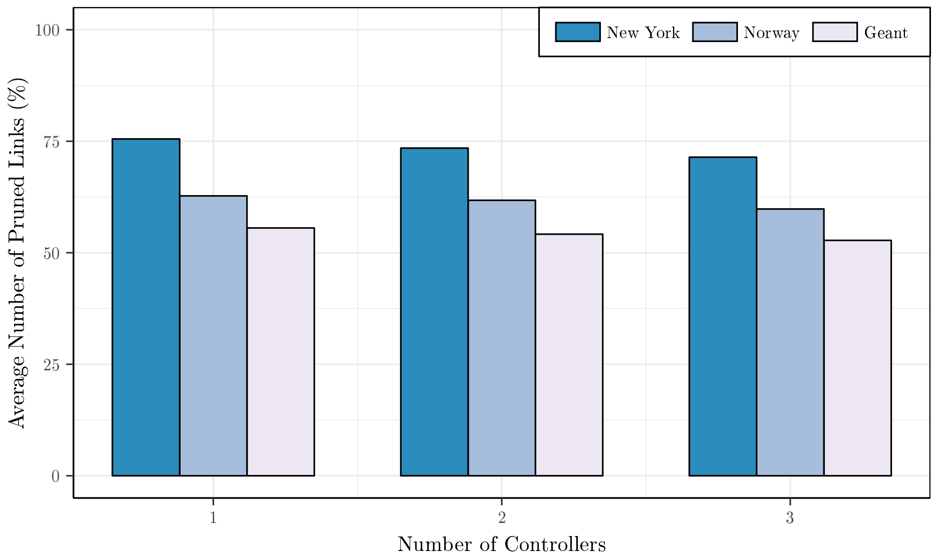


Fig. 4.6: Average number of pruned links in the three topologies varying the number of controllers.

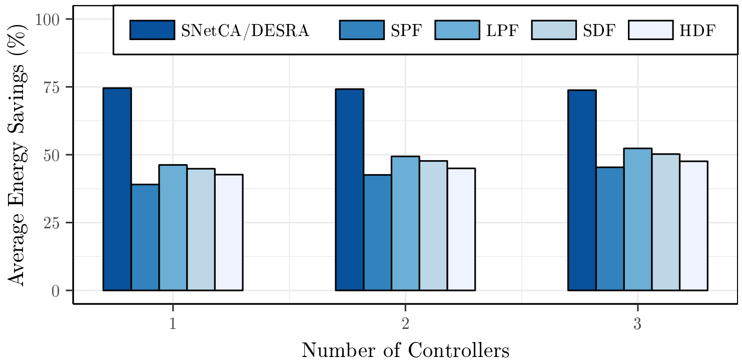
4.4.3 Assessment of Energy Saving Potential

To assess the suitability of the proposed solution in terms of energy efficiency we compare its performance with another related work existing in the state of the art. SGH [61] is an energy-aware routing solution based on knowledge of the incoming traffic requests. This proposal selects, among a certain number of pre-calculated shortest paths, the most suitable one in terms of energy savings to allocate each traffic demand, as long as it has enough capacity. To do so, traffic demands are ordered according to the following four different strategies:

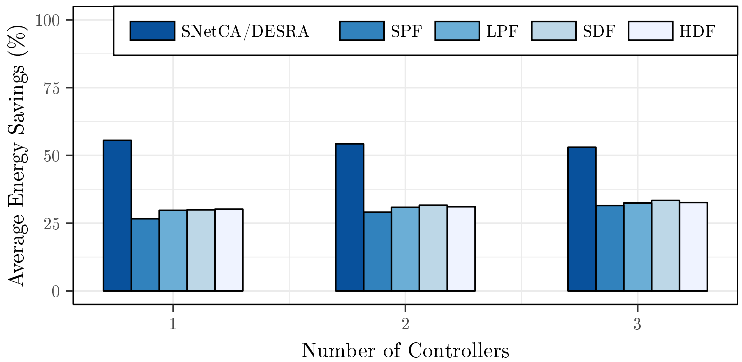
- Node pairs with shortest shortest path first (SPF)
- Node pairs with longest shortest path first (LPF)
- Node pairs with smallest demand first (SDF)
- Node pairs with highest demand first (HDF)

Fig. 4.7 shows the average performance in terms of energy savings of the two-module based strategy SNetCA/DESRA with respect to the four different versions of SGH in the three real topologies analyzed for different amount of controllers.

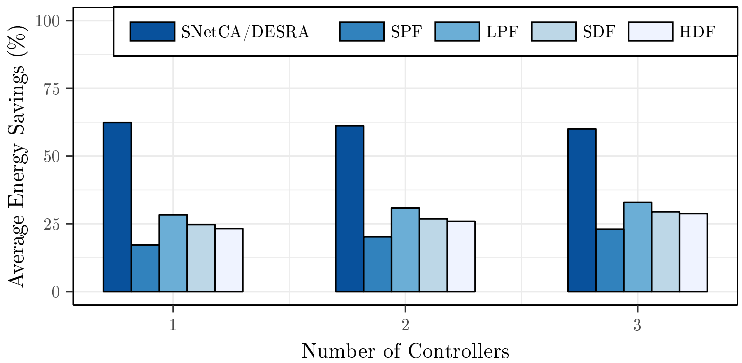
As we are considering an in-band SDN with multiple controllers, the traffic matrix provided to SGH includes, together with the data demands, a control traffic flow for each pair of associated controller-switch and for each pair of controllers in the network. In this way, required control



(a) New York topology.



(b) Geant topology.



(c) Norway topology.

Fig. 4.7: Average energy saving comparison between proposed strategy and SGH solutions for different numbers of controllers.

plane paths are also established by SGH. This is done taking into account the distribution of switches between controllers obtained from SNetCA.

Additionally, the routing restrictions established in this proposal to avoid additional traffic load through network controllers are considered in the computation of the pre-calculated shortest paths. On the other hand, given that DESRA is an online routing strategy, the connection requests are sequentially allocated as they appear in the considered traffic matrix.

In Fig. 4.7a, Fig. 4.7b and Fig. 4.7c we can see that SNetCA/DESRA performs better than the remaining energy-aware algorithms. While similar results are obtained by the four different versions of SGH, the proposed strategy achieves notable improvements in the three considered topologies. For instance, differences up to 35%, 29% and 45% of energy savings are reached in the New York, Geant and Norway topologies, respectively. In general, results have been determined with a 95% confidence interval not exceeding 2.2% of the indicated average values.

The significant differences obtained in terms of energy savings are mostly due to the operation of SNetCA before the traffic arrival, which is able to prune a great number of links without affecting the network capabilities to manage the incoming requests. Additionally, the routing decisions performed by SGH are limited to a predefined number of pre-computed shortest paths and fail to extensively exploit the energy saving potential of each topology. Therefore, even without a prior arrangement of demands based on an accurate knowledge about the incoming traffic, the proposed scheme puts to sleep mode a higher percentage of links, being able to save substantially more energy.

4.4.4 Impact on Network Performance

It is to be emphasized that in our energy-aware approach QoS constraints and performance metric boundaries are not taken into account. This is not a limitation but a choice; since we intend to measure the impact of our proposal on the network performance metrics as a trade-off with the energy saving improvements. In fact, we are presenting an effective and easy to implement green routing mechanism that emphasizes the importance of energy efficiency in the operation of current data networks.

In order to assess the impact of our energy-aware approach on the network performance, we adapt two well-known state-of-the-art routing algorithms: SPR and LB, for their use in the considered in-band SDN environment. Additionally, being the rule space a significant issue of concern in SDN, we include in this analysis an algorithm balancing the number of rules installed in each forwarding node, denoted here as TCAM Occupation Balancing (TOB). In general, these algorithms are greedy heuristics that follow the procedure described in Algorithm 7.

More precisely, these algorithms will evaluate every candidate admissible path and find the one prioritizing some performance metric such as: traffic latency, link utilization or TCAM occupation. According to the approach used the selected path will be:

Algorithm 7 GREEDY_BASELINES

```
Require: G, A, D
for d ∈ D do
    p_d = Find_Best_Data_Path
    Update network metrics
    for s ∈ p_d do
        if s has no control path already established in A then
            p_c = Find_Best_Control_Path
            Update network metrics
        end if
    end for
end for
```

SPR The one minimizing the propagation latency

LB The one minimizing the maximum link utilization

TOB The one minimizing the maximum TCAM occupation

It is clear that, for each metric, the corresponding baseline algorithm will have better performance than our energy-aware approach. However, the purpose of this evaluation is to use these algorithms as a reference point to illustrate the energy-aware solution impact on network performance. All of them follow the assumptions established in this proposal to avoid the routing of additional traffic load through network controllers. Similarly, SNetCA is still used to determine the distribution of switches between controllers. Although we may focus our attention on some specific network for the different performance metrics, the general conclusions that will be derived are independent of the specific topology and hold whichever network is examined.

4.4.4.1 Traffic Latency

In a first set of simulations, we analyze how the data and control paths latency is affected by routing decisions. To evaluate the impact of our algorithm on control path delay, we collect, for each traffic demand, the lengths of its associated control paths and the corresponding shortest paths.

Fig. 4.8, Fig. 4.9 and Fig. 4.10 show this behavior for the three studied topologies considering all possible placements for different amount of controllers.

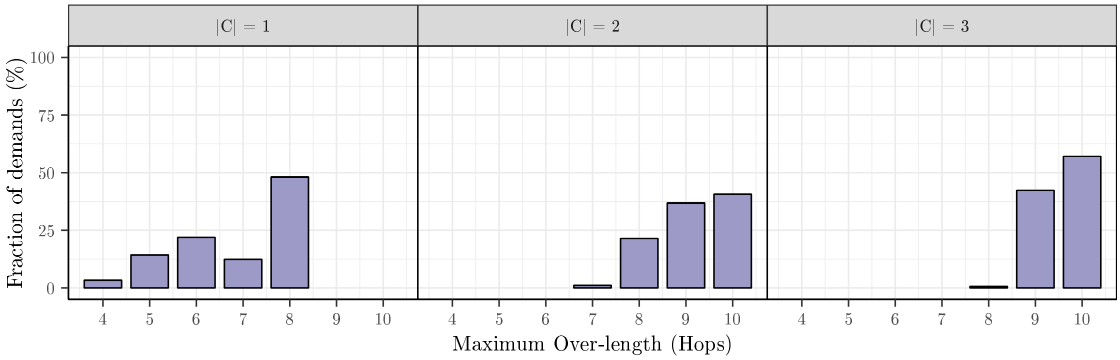


Fig. 4.8: Distribution of maximum control traffic over-length in the New York topology.

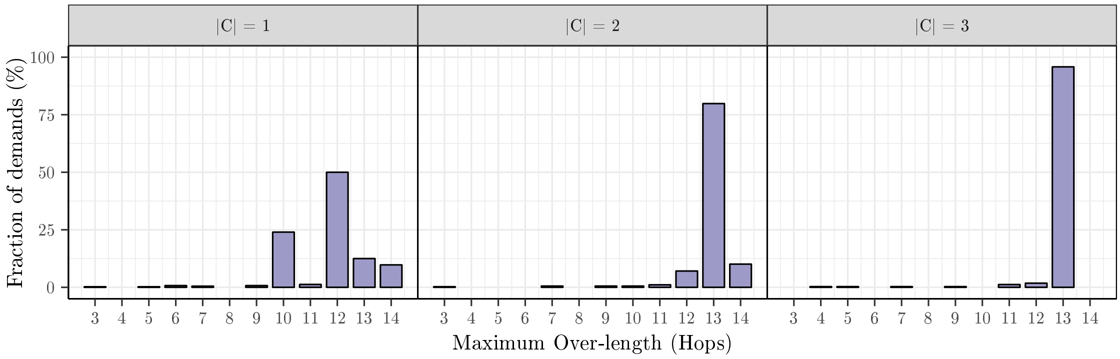


Fig. 4.9: Distribution of maximum control traffic over-length in the Geant topology.

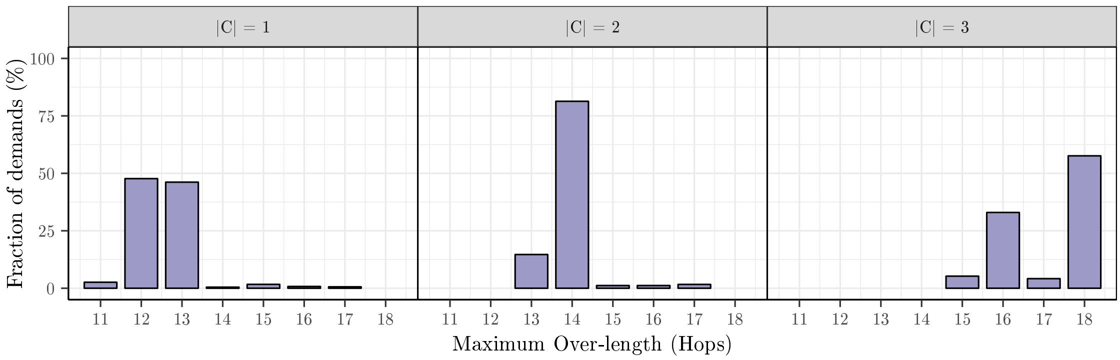
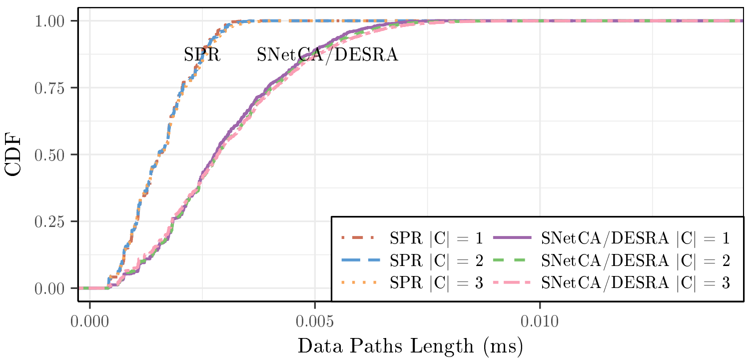
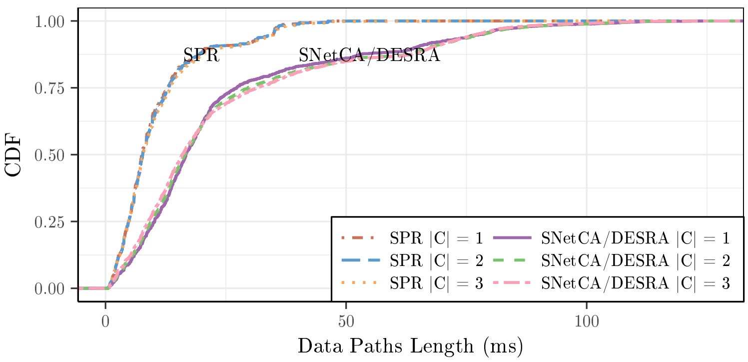


Fig. 4.10: Distribution of maximum control traffic over-length in the Norway topology.

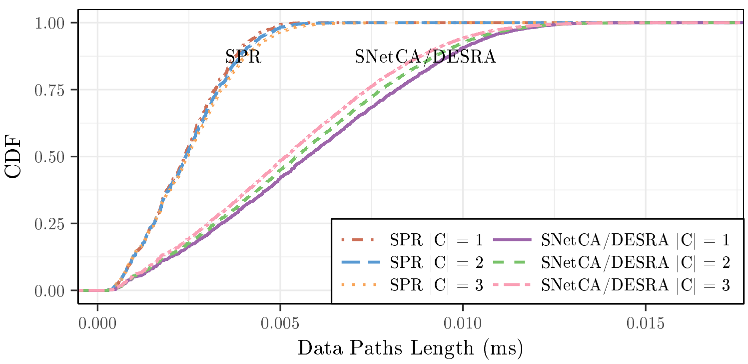
The notation Maximum Over-length is used to denote the maximum difference (in number of hops) between the length of the routing solution and the shortest path. For instance, when this value is equal to 0, it means that every control traffic is routed using exactly the shortest path. As it is shown, in all cases when the number of controllers grows, the control traffic is routed using a larger number of hops for a higher fraction of demands. Being Norway the largest



(a) New York topology.



(b) Geant topology.



(c) Norway topology.

Fig. 4.11: Cumulative distribution function of data paths latency varying the amount of controllers.

one in terms of network size (number of nodes and links), control paths in this topology are increased by a higher number of hops.

To take a closer look at the data plane, we draw in Fig. 4.11 the Cumulative Distribution Function (CDF) of data paths latency for the three topologies considering all possible locations of one to three controllers. As shown in Fig. 4.11(a), Fig. 4.11(b) and Fig. 4.11(c), the CDFs of data paths latency for different amount of controllers are quite similar. However, we can see that under the energy-aware routing, data path delay is affected since larger data paths are used in order to minimize the number of active links. For instance, in Fig. 4.11(b), only 87% of data paths exhibit delays lower than 50 ms, meanwhile all control paths in the SPR case are under this value. This performance degradation is less critical in the two other topologies, which are deployed in smaller geographic areas compared to Geant. In general, the larger the network (in terms of geographic length), the more increase in latency is incurred.

Despite the presented latency degradations with respect to SPR, the solution performance observed in Fig. 4.11(a) and Fig. 4.11(c) are suitable for supporting latency critical services in 5G networks demanding end-to-end delays lower than 10 ms [116], such as robotics and telepresence, virtual reality, health care, among others. Likewise, less demanding applications and use case scenarios to be addressed in 5G networks, such as intelligent transport systems and smart grid, with latency requirements up to 100 ms [116], could be conceived and deployed in Geant-like environments.

These latency degradations confirm that, according to traffic requirements, specific performance bounds may be required. In addition, even when the latency degradations are acceptable, longer average path lengths impact the resilience of the traffic flows, since a greater number of network elements (nodes and links) gets involved in the routing increasing the probability of incurring network failures. To mitigate these negative implications and assure the suitability of the proposed algorithm for delay-critical services, we include in this analysis the evaluation of a delay-constrained version. To do so, we now restrict the number of links that can be initially pruned by SNetCA. Specifically, network links belonging to the shortest path between any pair of nodes are not removed during the static network configuration phase. In this way, the shortest path will always be available if it is needed for allocating the incoming traffic when executing the DESRA module.

Fig. 4.12 and Fig. 4.13 show the performance of the delay-constrained version considering data and control paths latency bounded by the factor r. This latency threshold is used to denote the relation in between the delay requirement and the shortest path propagation latency for every established path. For instance, $r = 2$ means that every path latency is, at most, twice that of the shortest path.

On the one hand, as shown in Fig. 4.12, under the delay-constrained approach the control traffic can be routed incurring in smaller over-lengths with respect to the performance-agnostic

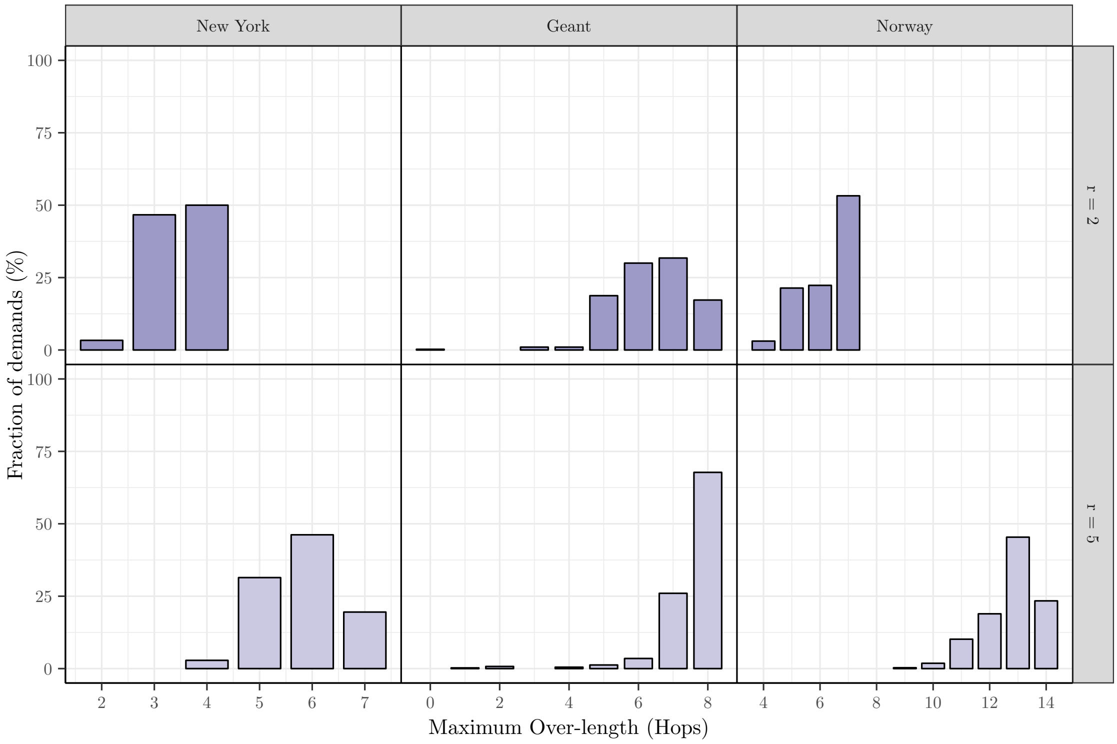
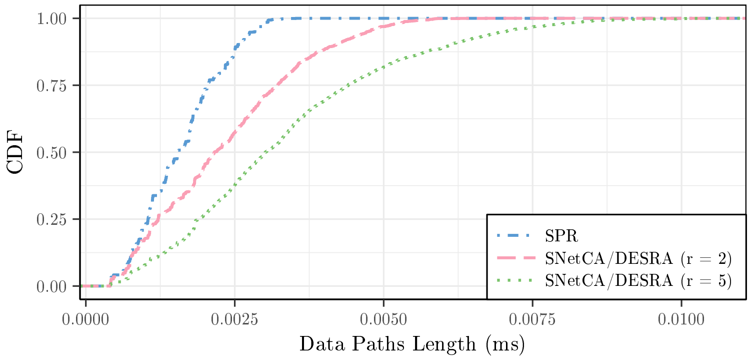


Fig. 4.12: Distribution of maximum control traffic over-length for 1 controller under delay constraints.

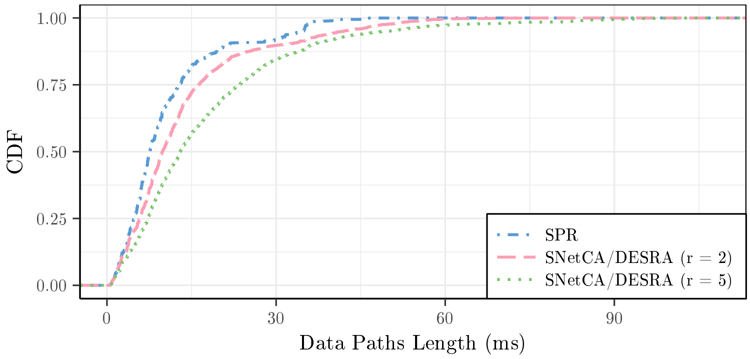
approach. Higher differences in number of hops between $r = 2$ and $r = 5$ can be seen in New York and Norway than in Geant. In this topology, although the distribution of demands reflects the expected difference between both approaches, the maximum number of hops is not particularly increased. This is due to the existence of longer links (i.e. links with higher propagation delays) in Geant with respect to the other two topologies. Therefore, in the Geant topology, an increase in the allowed path latency in terms of delay, for instance from $r = 2$ to $r = 5$, can be equivalent to a small increase in terms of hops (i.e. to just a few additional long links).

On the other hand, Fig. 4.13 shows that data paths delays are less compromised when more restrictive possibilities in terms of latency ($r = 2$ and 5) are considered. In general, as more restrictive is the latency bound used, a better performance in terms of delay can be achieved.

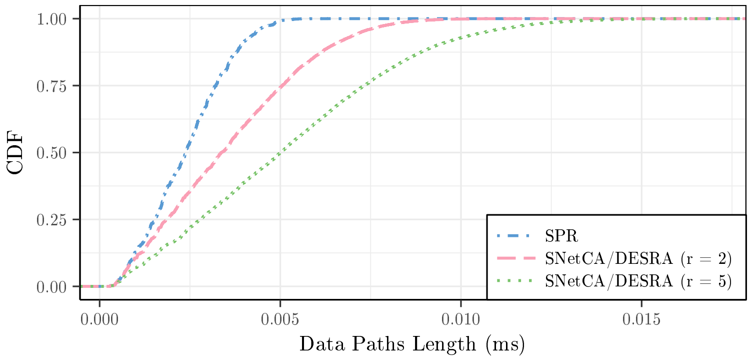
As expected, considering a latency restriction during paths selection will negatively impact the potential improvements in terms of energy efficiency. To validate this affirmation we re-compute the energy saving under the delay-constrained approach using $r = 2$. Optimal values were obtained after adding to the ILP model presented in Section 4.2 the following constraint, where $d_{i,j}$ and L_f denote the link propagation delay and the traffic maximum latency bound,



(a) New York topology.



(b) Geant topology.



(c) Norway topology.

Fig. 4.13: Cumulative distribution function of data paths latency for 1 controller under delay constraints.

respectively.

$$\sum_{(i,j)\in E} t^f_{i,j} d_{i,j} \leq L_f \qquad \forall f \in F \tag{4.10}$$

The energy savings achieved by the delay-constrained version for $r = 2$ are shown in Table 4.3. In this table optimal and heuristic values are depicted as well as the energy saving differences

Table 4.3: Energy savings with paths delay constrained to $r = 2$.

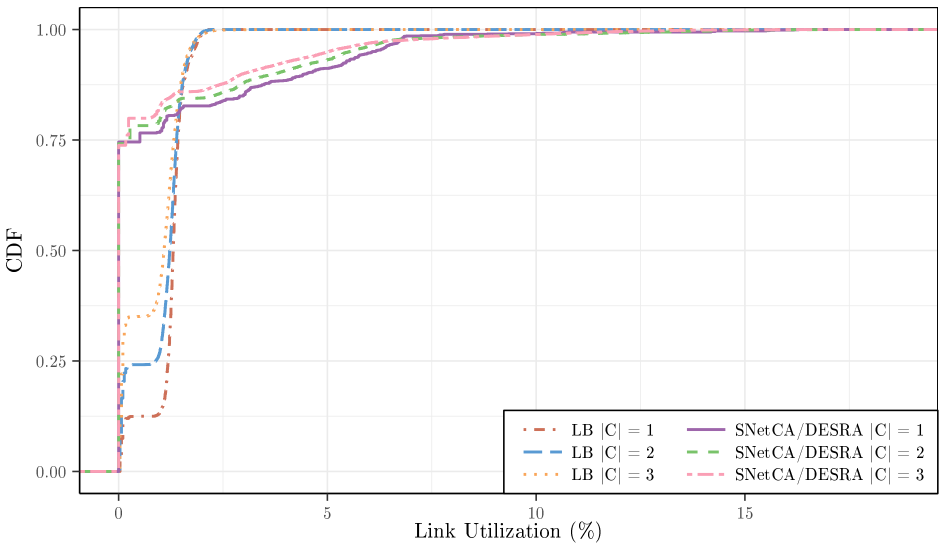
Topology	$\|C\|$	**Optimal**	**Delay-constrained heuristic**	**Difference**
New York	1	58.16327	39.92347	34.63010
	2	61.40456	41.11645	33.05608
	3	62.37759	42.04418	31.74105
Geant	1	29.67172	17.67677	37.87879
	2	33.81976	18.65719	35.61828
	3	34.72549	19.85249	33.16422
Norway	1	30.93682	16.95715	45.42484
	2	35.87280	17.98141	43.20742
	3	37.58308	18.48646	41.84001

between the performance-agnostic heuristic results shown in Fig. 4.4 and the delay-constrained version. Interestingly, we can also see that, unlike the values observed in Fig. 4.4, as the amount of controllers is increased greater energy savings are achieved. Evidently, for a given amount of nodes an increase in the number of network controllers is directly related with a reduction in the number of data paths that need to be allocated by the proposed solution. The effect of this relation is less significant under the performance-agnostic approach as data path lengths can be considerably increased. However, in this case, the data paths established by the DESRA module are more likely to require the activation of additional links in order to meet the considered delay restriction. Thus, it can be concluded that, under the delay-constrained version, the number of data demands to be allocated has a higher impact on the amount of links required over the initially pruned topology and hence, on the final energy efficiency.

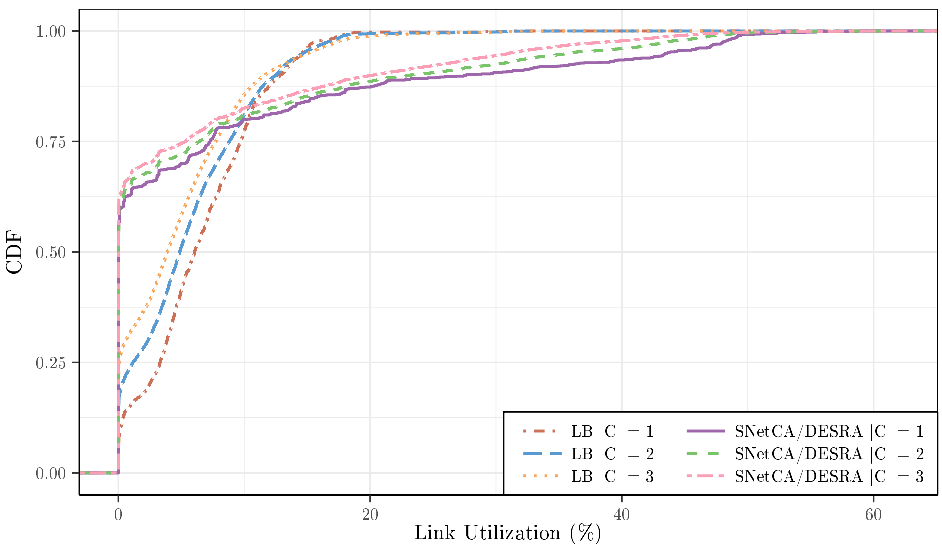
4.4.4.2 Links Utilization

The selection of routing paths minimizing the energy consumption has a direct influence in the traffic load of all the network links. To better showcase this situation, we use the New York and Geant topologies and the LB algorithm. Fig. 4.14 provides the CDF of link utilization under both algorithms considering all possible locations of one to three controllers in both topologies.

As expected, the fairness of traffic distribution is altered by the energy-aware routing, since under this approach traffic is concentrated in a fewer number of links. Therefore, there is a subset of active links that is more overloaded than the others. For instance, in Fig. 4.14(b) the



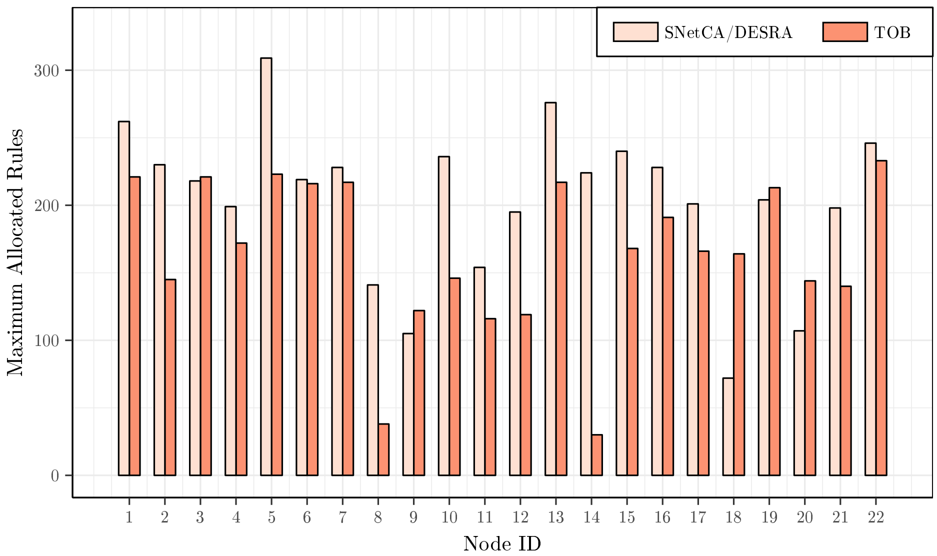
(a) New York topology.



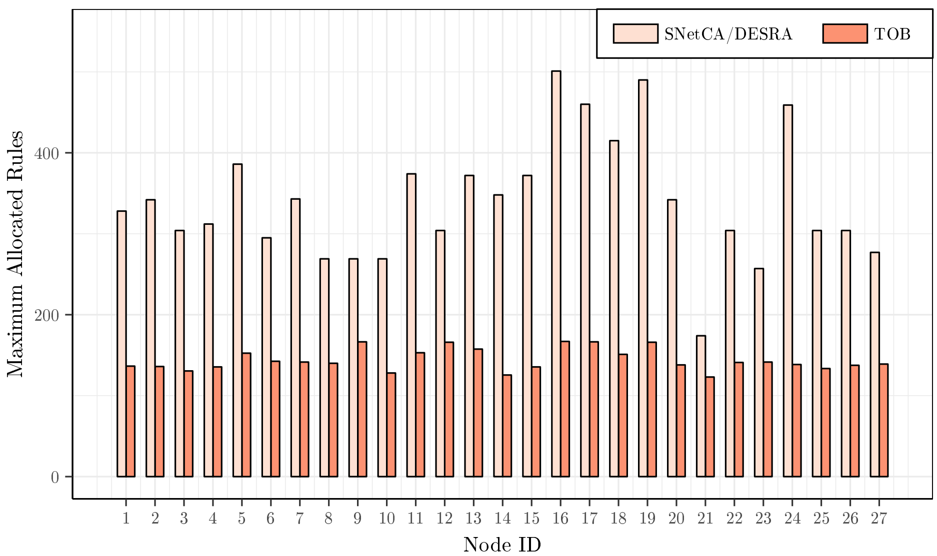
(b) Geant topology.

Fig. 4.14: Cumulative distribution function of link utilization varying the amount of controllers.

link utilization of some links in Geant is increased to more than twice the value achieved by the LB algorithm. Nevertheless, even in the more loaded cases the link utilization in this topology is under 60%. A less concerning situation can be observed in Fig. 4.14(a) since the given traffic load in the New York topology is very low.



(a) Geant topology.



(b) Norway topology.

Fig. 4.15: Average TCAM occupation with $|C| = 2$.

4.4.4.3 TCAM Occupation

Intuitively, an energy-aware routing would affect the allocation of flow rules, which is a practical constraint in OpenFLow-based network devices, given that traffic flows are redirected to minimize the number of active links. In Fig. 4.15, we evaluate the impact of our approach on TCAM occupation with respect to the TOB algorithm using the Geant and Norway topologies and all possible locations of two network controllers.

As expected, the number of installed rules is raised by the energy-aware routing. For instance, this increase is observed in all network devices in the Norway topology and in 17 out of 22 nodes in the Geant topology, being in some cases more than twice the value obtained by the TOB algorithm. This behavior is related to the fact that the path length increases considerably in proportion with the amount of links pruned by the proposed solution. As the path length is increased, a higher number of flow rules is required to allocate each single flow. Furthermore, the SNetCA/DESRA performance in both topologies is still physically acceptable considering that a routing table can support around a few thousands of rules [60].

4.4.4.4 Energy Savings

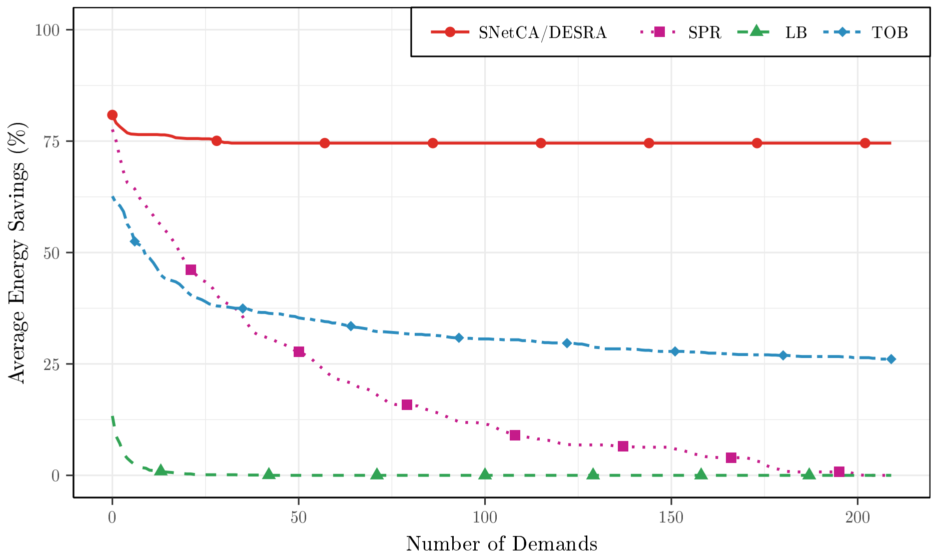
To get a sense of the other side of the trade-off between energy efficiency and network performance, Fig. 4.16 shows the average energy performance of all the considered routing models in the New York and Norway topologies for the case of one centralized controller in the network. Results have been determined with a 95% confidence interval not exceeding 5% of the indicated average values.

As expected, in all cases energy savings decrease while the number of demands grows, since new paths need to be established to accommodated such traffic. The flat tendency of energy savings achieved by the power-aware solution despite the increase in allocated demands is possible given the low network load discussed at the beginning of this section. Moreover, the proposed strategy greatly outperforms SPR, LB and TOB in terms of energy saving. In general, SNetCA/DESRA is able to achieve significant energy savings but bigger improvements with respect to the other approaches are reached when the traffic grows.

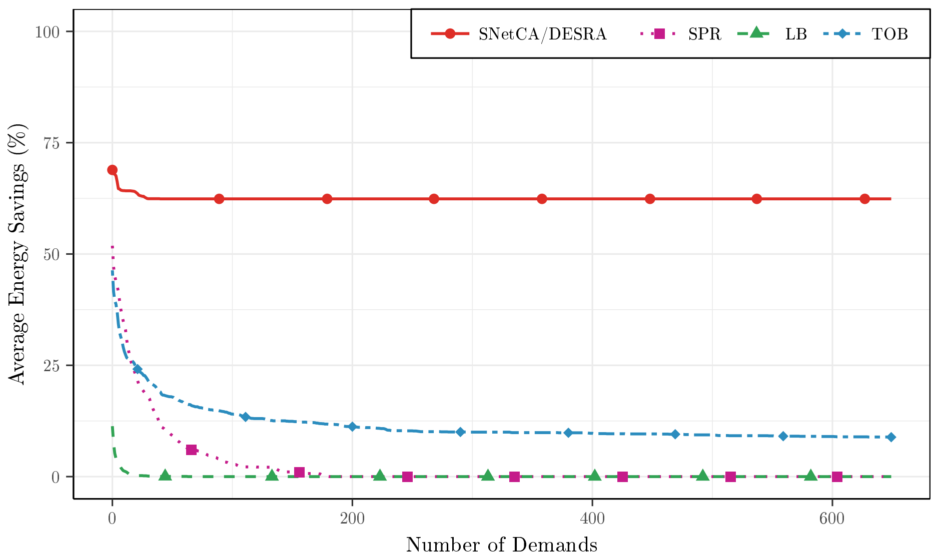
4.4.5 Resilience Considerations

While SNetCA allows important gains in terms of energy efficiency by pruning as many links as possible and leaving available only the minimum number of links needed to support the incoming traffic, the remaining subnetwork is more vulnerable to resource failures and sudden traffic bursts. To ensure an adequate network reliability while keeping low energy consumption, resilience constraints should be taken into account.

Given that the largest Laplacian eigenvalue of a graph (referred here as λ_{max}) is a widely accepted metric to assess network robustness with respect to link and node removals [117], we use it to control the resulting resilience after applying SNetCA. Several papers about graph



(a) New York topology.



(b) Norway topology.

Fig. 4.16: Average energy saving with $|C| = 1$.

theory [118, 119] sustain that a network is more resilient the higher the largest eigenvalue of its Laplacian matrix. In general, these networks are more robust since they have small diameters, higher numbers of nodes and link disjoint paths and are likely to expand faster.

In order to provide an initial investigation exploring the existing trade-off between energy savings and network reliability, in this analysis we evaluate a resilience-constrained version of SNetCA (i.e. SNetCA'), which enables to limit the admissible λ_{max} reduction rate due to the link removal process. Additionally, to improve the redundancy for the control paths the amount

Table 4.4: Average performance of SNetCA' under connectivity and resilience constraints.

C_{con}	δ	New York		Geant		Norway	
		ES (%)	λ_{max}	ES (%)	λ_{max}	ES (%)	λ_{max}
1	0	54.0816	11.4983	29.7979	9.51619	51.6703	8.05608
	1	74.5535	10.9656	55.5555	7.60123	62.3819	6.76403
2	0	52.0408	11.6290	27.0202	9.64202	49.7095	8.09439
	1	72.3214	11.1583	52.7146	7.78362	60.3485	6.84699
3	0	50.2721	11.7208	31.0185	9.60844	47.8823	8.10417
	1	70.1360	11.3623	49.0740	8.04641	58.2352	6.90745
4	0	48.2312	11.8481	37.2685	9.46297	46.9079	8.08731
	1	67.9591	11.4991	46.9907	7.76580	55.7315	6.96638
5	0	46.3556	12.0095	33.3333	9.47861	45.7516	8.06783
	1	65.7434	11.6203	44.1666	7.98822	54.2483	7.04646
6	0	44.8979	11.9304	27.3148	9.42488	44.1176	8.14669
	1	63.7755	11.5188	42.5925	8.02495	51.4705	7.57228

of neighbors that will remain connected to the controller is relaxed using different bounds (from one to the controller degree).

Table 4.4 gathers some of the obtained results, in terms of energy savings and λ_{max}, after applying SNetCA' in the three considered topologies with one centralized controller. The presented results validate the impact of varying these two criteria (i.e. the controller connectivity and the λ_{max} reduction rate) on energy savings and network robustness.

While the controller connectivity (denoted as C_{con}) is bounded by a number of neighbors between one and the controller degree, the allowed λ_{max} reduction rate (denoted as δ) was normalized using the following expression:

$$\delta = 1 - \frac{\lambda_{max}(\text{SNetCA'})}{\lambda_{max}(\text{Original})} \tag{4.11}$$

Accordingly, δ is able to adjust the reduction of the λ_{max} derived from the loss of link redundancy. Specifically, values of δ closer to 0 are only possible if the λ_{max} achieved after applying SNetCA' is similar to the value of this metric in the original graph.

As shown in the table, when $C_{con} = 1$ and $\delta = 1$, the resilience-constrained version of

SNetCA behaves exactly as the resilience-agnostic one. However, less energy can be saved as more restrictive values of C_{con} and δ are imposed, since each of these elements determines that a fewer number of links could be put into sleep mode by the pruning function of SNetCA.

Inversely, when no resilience degradation is allowed (i.e., $\delta = 0$), higher values of λ_{max} are obtained. More importantly, we can observe that considerable energy savings can still be reached by the proposed heuristic strategy, while ensuring the original network reliability.

Although higher values of λ_{max} are expected when the bound of nodes that will remain connected to the controller is increased, this is not always the behavior that can be appreciated in the table for the three network scenarios considered. The reason for this is that every network node, which is considered in one particular instance of the simulation as the controller, does not have the same node degree. Thus, for each considered possibility of C_{con}, different amounts of nodes are part of the presented average values.

We stress that the impact of switching off networks links on SDN reliability is even more critical for networks with in-band control traffic where any links/node failure will affect not only data plane communications but the connection with the controller. Hence, a further analysis about this crucial issue will be done in follow-on work.

4.5 Conclusion

In this chapter we proposed an energy-aware strategy that minimizes the number of active links required to route the incoming traffic suitable for SDN environments with in-band control traffic and multiple controllers. To achieve such goal, we first provided a link-based formulation of the optimization problem, integrating the routing requirements for data and control traffic. Given the overall complexity of the exact model in large-scale topologies, a heuristic hybrid approach is conceived, comprising two algorithms: a static network configuration and a dynamic energy-aware routing. In this way, the number of links to be considered in the paths' computation is significantly reduced by the first component, which contributes to decreasing the computation times. Based on experimental simulations using real-world topologies and traffic matrices, energy savings between around 50% and 80% are reached by the proposed energy-aware approaches. In addition, the heuristic strategy attains results very close to the optimal values, converges much faster and can handle larger network sizes for which the exact model fails to find solutions in reasonable time. Besides the notable improvements in terms of achieved energy saving, an insightful analysis was presented to evaluate the impact on network performance. In this regard, extensive simulations validate that crucial network parameters such as control traffic delay, data path latency, link utilization and TCAM occupation are affected by the performance-agnostic energy-aware model. Therefore, this proposal stands as a valuable proposal for designing routing schemes suitable for current control planes, since it discloses and manages the intrinsic trade-off between environmental and performance concerns.

Chapter 5

Multi-Objective Routing Combining QoS and Energy Awareness

This chapter is based on:

- **A. Fernández-Fernández**, C. Cervelló-Pastor and L. Ochoa-Aday, "A Multi-Objective Routing Strategy for QoS and Energy Awareness in Software-Defined Networks," *IEEE Communications Letters*, vol. 21, no. 11, pp. 2416–2419, Nov. 2017.

5.1 Introduction

An effective energy management, as well as an enhanced user experience, are essential design goals to fulfill the requirements of current and future communications systems for heterogeneous applications and services. However, some of these objectives may be in conflict, and specific strategies must be developed. Precisely, the existing trade-off between energy efficiency and network performance, which was established in the previous chapter, is now further addressed.

In this chapter we derive a multi-objective routing approach in order to limit the implications of energy-aware routing on QoS. This approach enables the reduction of power consumption without service degradation yet considering the routing for data and control plane traffic in SDN. This factor is of key importance in order to jointly optimize QoS requirements for the data plane traffic and energy efficiency while still meeting control traffic requirements. More precisely, given a current network topology and the controller location, the proposed algorithm will find how data traffic demands and associated control traffic should be routed such that the energy consumption, the control traffic delay and the blocking rate are minimized. The results indicate that the proposed solution can significantly improve these metrics, in accordance with the traffic type.

The rest of this chapter is structured as follows. In Section 5.2 a complete mathematical formulation of the optimization problem is presented. In Section 5.3 we design a Multi-Objective Evolutionary Algorithm and explain its main features. The simulation strategies and the obtained results are presented and analyzed in Section 5.4. Finally, Section 5.5 concludes this chapter.

5.2 Problem Statement

In this Section we provide a mathematical formulation to describe the considered multi-objective optimization problem. In essence, two objective functions are defined in order to simultaneously optimize the energy efficiency and the performance of control and data plane communications while routing a given traffic demand. The former objective refers to the number of used links, while the second integrates QoS requirements and control traffic delay.

5.2.1 Network Model

We consider an SDN modeled as a directed graph $G = (V, E, C)$, where V, E and C denote the set of nodes, links and controllers respectively, being $C \subset V$. We use c_e to denote the capacity of a link $e \in E$. We define the set of forwarding nodes as $S = \{n \mid n \in V \wedge n \notin C\}$.

Each incoming demand k, has associated its QoS requirements imposed by the SLA, in terms of bandwidth and latency, denoted by b_k and l_k, respectively. With respect to the control plane, the incoming demand k has associated several traffic flows between controller and switch pairs, which are denoted by the set T. Consequently, we will use F_k to refer to the overall set of traffic flows generated in the network due to the demand k (i.e. $k + T \subseteq F_k$). For each flow $f \in F_k$ we use s_f and d_f to denote its source and destination, respectively.

Let P_f be the set of paths that can be used to route each flow $f \in F_k$, being P_k the notation used to identify the subset of paths corresponding to demand k. In addition, let $P_e^f \subset P_f$ be the subset of paths that use link $e \in E$, for each $f \in F_k$ and $P_c^k \subset P_k$ denote the subset of data paths that pass through controller $c \in C$, which will not be used to route data plane communications. We use b_p and l_p to denote the minimum bandwidth and total latency of a path $p \in P_f$, respectively.

5.2.2 Formulation

The proposed multi-objective optimization problem can be formulated using the following decision variables:

x_e: describes the state of a link $e \in E$.

$$x_e = \begin{cases} 1 & \text{if } e \text{ is active,} \\ 0 & \text{otherwise.} \end{cases}$$

$\gamma_{f,p}$: describes the selection of a path $p \in P_f$ to route a traffic flow $f \in F_k$.

$$\gamma_{f,p} = \begin{cases} 1 & \text{if } p \text{ is selected to route } f, \\ 0 & \text{otherwise.} \end{cases}$$

λ_i: describes the use of a node $i \in S$ by the selected data path.

$$\lambda_i = \begin{cases} 1 & \text{if } i \text{ belongs to the selected data path,} \\ 0 & \text{otherwise.} \end{cases}$$

$w_{f,p}$: describes the cost of using a path $p \in P_f$ to route a traffic flow $f \in F_k$.

The cost of using a path to route a traffic flow is defined to evaluate the suitability of each available route. This variable is in accordance with the traffic type (i.e. data or control) and based on a best-fit scheme. Consequently, the two objective functions are defined as follows:

Minimize the number of active links in the network:

$$\text{minimize} \sum_{e \in E} x_e \tag{5.1}$$

Minimize the total cost of routing the incoming demand:

$$\text{minimize} \sum_{f \in F_k} \sum_{p \in P_f} \gamma_{f,p} w_{f,p} \tag{5.2}$$

subject to the following constraints:

To route the incoming traffic demand k, a single data path is selected in Equation (5.3a). Afterward, using Equation (5.3b), control messages are sent from controllers only to those switches belonging to the selected data path.

$$\sum_{p \in P_f} \gamma_{f,p} = \begin{cases} 1 & f = k \qquad (5.3a) \\ \lambda_{d_f} & \forall f \in T,\, d_f \in S \qquad (5.3b) \end{cases}$$

Equation (5.4) establishes that paths passing through any controller $c \in C$ cannot be used to route a data plane demand k.

$$\gamma_{k,p} = 0 \qquad \forall p \in P_c^k, \forall c \in C \tag{5.4}$$

A node $i \in S$ belongs to the data path selected to route the incoming demand k if there is traffic in any of its incoming or outgoing edges, being $N(i)$ in Equation (5.5) the set of neighbors of i.

$$\lambda_i \geq \frac{1}{2} \sum_{j \in N(i)} \left(\sum_{\substack{p \in P_k \\ (i,j) \in p}} \gamma_{k,p} + \sum_{\substack{p \in P_k \\ (j,i) \in p}} \gamma_{k,p} \right) \qquad \forall i \in S \tag{5.5}$$

The cost of using a data path to route the incoming traffic demand k depends on the gap between the requested amount of resource quality and the available ones as established in Equation (5.6a). Likewise, the cost of using a control path is related to its latency, being l_f^{sp} in Equation (5.6b) the shortest path delay of a control traffic flow $f \in T$.

$$w_{f,p} = \begin{cases} \frac{1}{2} \left[W(x) \left(\frac{b_p - b_k}{b_p} \right) + W(x) \left(\frac{l_k - l_p}{l_k} \right) \right] & f = k,\, \forall p \in P_k \qquad (5.6a) \\ \frac{l_p - l_f^{sp}}{l_p} & \forall f \in T,\, \forall p \in P_f \qquad (5.6b) \end{cases}$$

To ensure the SLA fulfillment, data paths that do not meet QoS requirements, are penalized with an infinite cost using the validation function $W(x)$ defined in Equation (5.7).

$$W(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \infty, & \text{otherwise} \end{cases} \tag{5.7}$$

Equation (5.8) ensures that a link e is active if it is used by some path $p \in P_f$ of some traffic flow $f \in F_k$

$$x_e \geq \gamma_{f,p} \qquad \forall f \in F_k, \forall p \in P_e^f, \forall e \in E \tag{5.8}$$

Finally, Equation (5.9) assures that the total traffic in each active link $e \in E$ is less than the link capacity.

$$\sum_{f \in F_k} \sum_{p \in P_e^f} \gamma_{f,p} b_f \leq c_e x_e \qquad \forall e \in E \tag{5.9}$$

Using this model, the centralized controller can determine the optimal routes and set the required flow rules on each forwarding node before the traffic arrival. Then, when a new demand enters the system, it is carried over the single path previously computed.

However, the use of traditional mathematical programming methods (such as linear combination of weights, goal programming and ϵ-constraint methods) to solve a multi-objective optimization problem has a number of important drawbacks [120]. For instance, they will require a precise definition of weights, desired targets or objective function bounds, being the quality of the obtained results dependent on the selection done. These values are not usually known a priori and finding the correct ones is not easy and will require some extra computational effort. An additional problem with these techniques is that they may not yield a non-dominated solution when the Pareto front is concave or discontinuous, which certainly limits their applicability.

Therefore, in the next section an effective routing scheme is proposed for the presented multi-objective optimization problem based on the use of evolutionary algorithms.

5.3 Multi-Objective Evolutionary Algorithm

Evolutionary algorithms have been extensively used in multi-objective problems involving conflicting objectives and intractable large and highly complex search spaces. These mechanisms are particularly suitable to solve multi-objective optimization problems since they deal simultaneously with a set of possible solutions (the so-called population). Commonly known as MOEAs [96], these strategies simulate the process of natural evolution using a class of stochastic optimization methods.

Different to traditional mathematical programming techniques, evolutionary algorithms can explore solutions over the entire search space and are less susceptible to the shape or continuity of the Pareto front. A striking property is the fact that even with an incorrect initial parameter setting, these population-based metaheuristics are robust enough to provide fairly good results [121]. Moreover, many of these techniques make use of the elitism concept to reduce the computation time. Additionally, they can be implemented using a parallel approach (multiple threads and processes in order to achieve parallelism inherent in current multi-core CPUs), which allows potential benefits in terms of speed and quality of the obtained approximations [122].

The goal of MOEA is to find or approximate a group of trade-off solutions called the Pareto-optimal solution set. A solution is Pareto-optimal when no improvement can be made on one objective value without degrading any other. Therefore, multiple optimal solutions can be found in a single run of the algorithm, instead of having to perform a series of separate simulation runs [123]. Then, a further processing is needed to select a preferred solution from this set based on user-defined criteria.

The particular MOEA proposed in this work is based on the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [124] which is an enhanced version of its predecessor SPEA [125]. This mechanism, characterized by few configuration parameters, rapid converging speed, good robustness and orderly-distributed solution sets, stands out as one of the most representative and currently used MOEA [126–129].

More in detail, SPEA2 implements a Pareto-based fitness assignment strategy with a nearest neighbor density estimation to determine the relevance of each individual in the current population. At each generation, the non-dominated solutions found are kept in a secondary population (B), which is the outcome of the algorithm once the terminal condition is reached. The general methodology of SPEA2 exploited in this approach is shown in Fig. 5.1. For a more in-depth description, please see [124].

The motivations behind this choice arise from the fact that SPEA2 is regarded as one of the most efficient elitist MOEA. The fitness assignment strategy employed to select a solution at each iteration, based on the number of solutions that dominate it, the number of solutions that are dominated by it, and the distance to the k-th nearest neighbor, allows to maintain a well-spread Pareto front while impeding local optima. Additionally, the use of a secondary population with a constant size and the enhanced archive truncation procedure guarantees an adjustable elitism

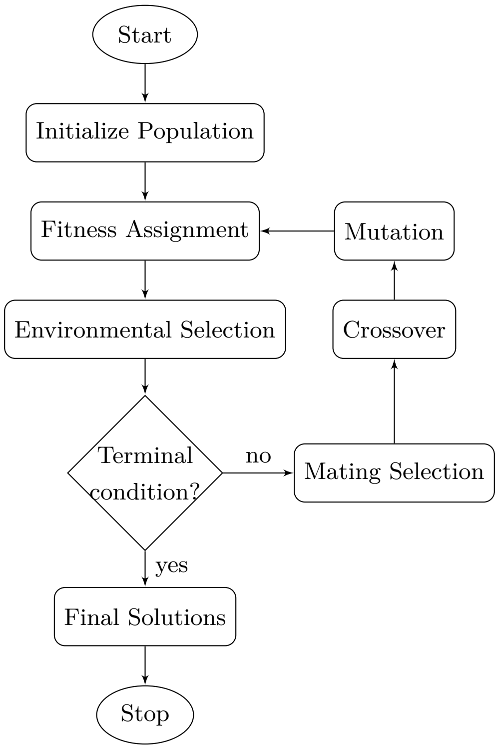


Fig. 5.1: SPEA2 general procedure.

scheme able to preserve non-dominated solutions and prevent the removal of boundary solutions. Finally, it has been widely applied to several multi-objective problems in both industrial and academic fields with prominent results compared to similar approaches [130–133].

The following outlines the specific considerations done to customize these basic concepts to our particular routing problem.

5.3.1 Design Issues

The key idea of this proposal is to fully take advantage of the high control flexibility given by the dynamic configuration capabilities and centralized network view of SDN.

The initially required control plane configuration (i.e. controller-switches association and node to controller control paths) is determined off-line adapting the integer linear problem previously presented in Section 3.2.2.2. Using this initial control plane configuration, switches send to the controller packet_in requests when a new traffic flow arrives, as well as statistics and failure notifications. Consequently, there is an initial set of active links in the network before the ingress of traffic flows.

When a new demand arrives, a routing request is sent to the controller, which calculates the data and remaining control paths (from it to its associated switches), based on the proposed multi-objective approach. The computed paths are established then, through the required flow forwarding rules.

5.3.1.1 Chromosome Structure

In the proposed scheme, a candidate solution or chromosome represents a selected set of n paths. The first $n-1$ routes identify the control paths used by the controller to install the flow forwarding rules in each node traversed by the data traffic. The last gene in the solution is the data path for the requested connection, being $n-1$ precisely the number of nodes along this path.

5.3.1.2 Initial Population

The set of solutions that form the initial population (A) of an evolutionary algorithm is usually obtained by random generation. In this instance, a hybrid approach is considered. Data paths are selected randomly from the set of admissible paths, i.e. the ones that satisfy the QoS requirements and do not pass through the network controller. By contrast, control paths are selected considering the routes used previously by other demands, if they still have enough capacity. Hence, the traffic is likely to be concentrated on the same links, as long as it can be accommodated, reducing the number of active links. In case that no path has been already established between the controller and a node, control paths are selected randomly from the overall set of feasible routes.

5.3.1.3 Evaluation

To evaluate the suitability of each individual in the population, two objective functions were defined according to the conflicting goals considered in this multi-objective routing strategy.

The first objective function supports performance requirements for control and data plane communications using a best-fit scheme. This behaviour is modeled in Equation (5.10) considering an individual i. The first term of this expression deals with the latency of the control paths associated to the demand k, while the second term is related with the QoS requirements of such incoming data traffic. Specifically, using this equation, a lower cost is assigned to individuals whose control paths (p_c) have the closest delays to the lowest ones (sp_c) and data path (p_d)

best satisfies the SLA thresholds. Consequently, two important routing performance metrics are improved, namely the control traffic delay is reduced (first term) and the number of future requests that can potentially be accommodated is increased (second term). In order to avoid dominant effect, the possible values of the different parameters are normalized into the interval (0, 1).

$$Q_i = \sum_{p_c \in i} \frac{l_{p_c} - l_{sp_c}}{l_{p_c}} + \frac{1}{2}\left(\frac{b_{p_d} - b_k}{b_{p_d}} + \frac{l_k - l_{p_d}}{l_k}\right) \tag{5.10}$$

The second objective, related to energy awareness, aims to minimize the number of links that need to be activated when a connection request arrives. Although the most convenient paths for the incoming demands are computed as they arrive, the existing routing should be taking into account in order to reduce the number of active links. Therefore, control paths required for the incoming traffic are selected considering the routes used previously by other demands, if they still have enough capacity. Considering L_i as the set of links used to route an individual i and J the record of links currently active in the network, Equation (5.11) determines the amount of additional links required for the incoming demand.

$$R_i = |L_i - J| \tag{5.11}$$

5.3.1.4 Crossover

According to a crossover rate (c_r), two solutions from the mating pool (M) are selected randomly to be parents. We apply a single-point approach, where a random common node from both data paths (apart from the source and the destination nodes) is selected as the crossover point. Two different data paths are generated by swapping the first part and second part of both parents. Then, a loop detection process (similar to the one explained in [99]) is applied to ensure the validity of resulting data paths. Finally, the control path for each node is taken from the corresponding parent and added to form the two children.

5.3.1.5 Mutation

Likewise, based on a mutation rate (m_r), a random node is selected from an individual data path. A new solution is then generated considering this node as traffic source. Using this new solution, the original data path is modified from the selected node to the destination node. After applying the loop detection process, a new solution is created adding to the resulting data path, the corresponding control routes.

5.3.1.6 Selected solutions

To select a preferred solution we use a service-differentiation approach, i.e. demands have been separated into two classes with different QoS requirements. Incoming traffic belonging to the class 1 (QoS_sensitive) is routed using the solution with the best performance in the first objective function. Otherwise, for demands under the class 2 (Best_effort) the solution that minimizes energy consumption is selected for transmission.

5.3.2 Complexity Analysis

The computational complexity of the proposed algorithm is dictated by the respective complexities of the fitness assignment and the environmental selection [134]. The fitness assignment step has a worst run-time complexity equal to $O(N^2 \log N + N^2 + N^2)$, where $N = A + B$, which can be expressed as $O(N^2 \log N)$, determined by the density estimation process. Meanwhile, the complexity of the environmental selection is dominated by the truncation procedure, which has a worst run-time of $O(N^3)$. However, this feature can be implemented in such a way that its run-time is reduced to $O(N^2 \log N)$, since farther neighbors are only considered when they are actually used and not in advance. Therefore, the overall time complexity after a number of T generations is $O(TN^2 \log N)$.

Given the good performance of SPEA2 when the population size and the number of iterations are small [130], reasonable computation times can be expected. Moreover, the overall complexity of this approach does not grow rapidly along with the network size avoiding thus scalability issues.

5.4 Simulations and Results

5.4.1 Simulation Setup

The proposed model was tested using a real-world network topology collected from the online available database SNDlib [111], considering one node as the controller (according to the well known minimum k-median model [109]) and the remaining nodes as SDN switches. All computations were carried out on a computer equipped with 3.30 GHz Intel Core i7 and 16 GB RAM.

The evolutionary parameters were empirically selected after performing several trials to confirm their suitability in maintaining a good trade-off between quality of solutions and required running time. These values are also commonly used in other related works which employ MOEAs (see [89]). Specifically, to implement the SPEA2 algorithm we use the following parameters:

- Primary population size (A): 20
- Secondary population size (B): 10
- Mating pool size (M): 5
- Crossover rate (c_r): 0.9
- Mutation rate (m_r): 0.1

Additionally, in order to reduce computation times, two stopping criteria are considered. The MOEA terminates either when 100 generations were explored or when there was no improvement after 30 consecutive generations.

Results have been determined with a 95% confidence interval not exceeding 4% of the indicated average values, estimated by running our algorithm 30 times with different prime number seeds on each scenario instance. This value for the number of conducted runs is also widely used by several related works on the field [89, 124, 135].

Three scenarios were considered according to the $\alpha \in [0, 1]$ value, which defines the proportion of demands that belongs to the class 1, being demands randomly allocated in one of the two classes. For the control traffic we assume an average rate of 1.7 Mbps [112].

The average running time of the algorithm, implemented in Python, takes less than 3 s. This result is perfectly suitable for the considered use case of traffic provisioning on large-scale transport SDN. Traditional transport networks can lead to over dimensioning the network and to a non-optimal use of resources. On the other hand, on large-scale transport or carrier SDN the control layer builds up a view of the topology, and a global view of network resources is attained. Thus, control applications can be constructed to optimize traffic flows over the network and automate the provisioning of the network. Moreover, on many of this kind of

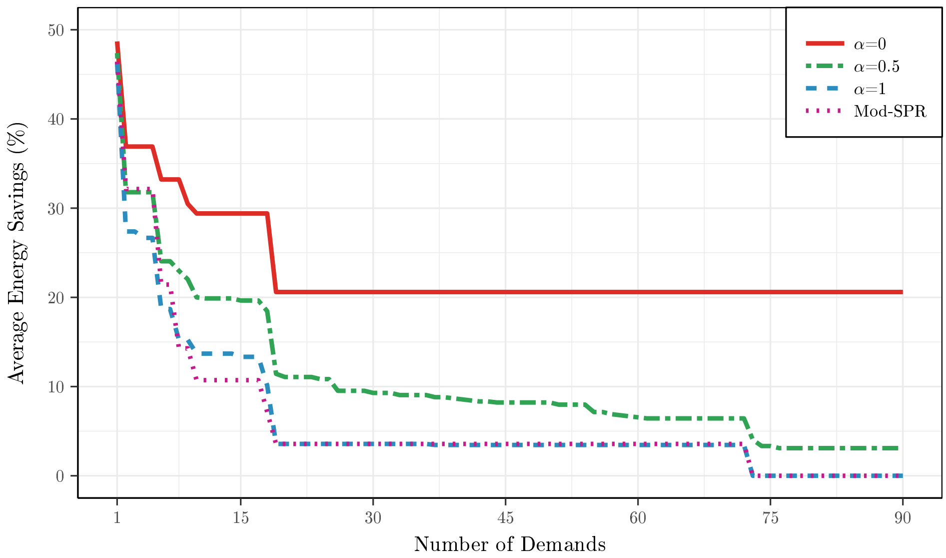


Fig. 5.2: Average energy saving in Abilene with the reported traffic matrices.

networks demands are characterized by large bandwidths (due to traffic aggregation) and long durations, which are also predictable. As a consequence, the path networking has a time scale that allows using this type of algorithms, to provide an efficient flow scheduling.

To the best of our knowledge, there are no similar multi-objective strategies involving QoS requirements and energy awareness in SDN with in-band control traffic for comparing the performance of our approach. Therefore, we use a modified version of the SPR (Mod-SPR). Mod-SPR can be considered as a default SPR algorithm for SDN with in-band control traffic and no data plane communications through any controller. The motivations behind this choice arise from the fact that SPR is a standard routing approach still widely used in current operative networks, and no proposal should be considered unless it can outperform this traditional benchmark.

5.4.2 Energy Efficiency and Control Path Delay Trade-off

Fig. 5.2 shows the average energy savings using the Abilene topology (11 nodes, 28 links) with the reported traffic matrices (measured at 00:00 on March 1st 2004) and the number of nodes as the required delay (in terms of hops). The energy savings were computed as the number of links in sleep mode over the total amount of network links.

As expected, the energy saving decreases while the number of demands grows, since new paths need to be established for the incoming traffic. It is also observed that less energy is saved when the amount of class 1 demands grows, due to the reduction of traffic routed through the

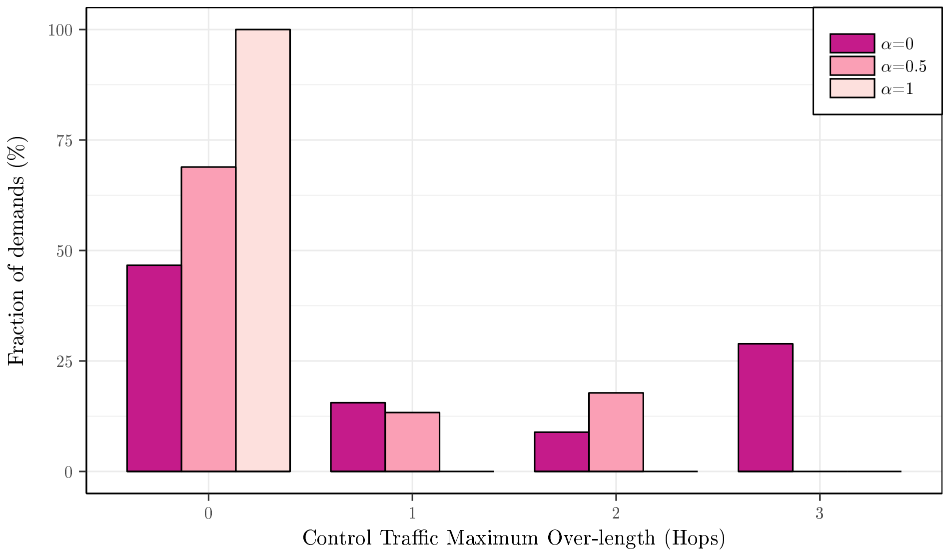


Fig. 5.3: Control traffic over-length in Abilene with the reported traffic matrices.

paths minimizing the number of active links in the network. Nevertheless, even in the more demanding scenario ($\alpha = 1$) the achieved energy saving values are no worse than the ones obtained with Mod-SPR.

Given the conflicting nature of the considered objectives, the best energy efficiency is achieved while sacrificing performance on QoS-related parameters. To examine this trade-off, in Fig. 5.3 we show the impact of our algorithm on control path delay, a crucial performance metric in SDN. For each allocated data traffic demand, we collect the length (in number of hops) of all its associated control paths and their corresponding shortest paths. The notation Maximum Over-length, shown in the figure, is used to denote the maximum difference between these two values for each data traffic demand. For instance, when this value is equal to 0, it means that every control traffic required by the considered data demand is routed using exactly the shortest path. As it is shown, in scenarios with larger amount of QoS_sensitive demands, the control traffic is routed using a fewer number of hops for a higher fraction of demands, avoiding thus performance degradations. Specifically, when $\alpha = 1$ all the control traffic required for every data demand is allocated without incurring any additional delay with respect to the shortest paths. On the contrary, when traffic is all of class 2 ($\alpha = 0$) the control path delay is affected in order to minimize the number of active links, being for some cases (around the 25% of data demands) 3 hops longer than the corresponding minimum path delay. In the remaining scenario, where demands are equally distributed between both classes ($\alpha = 0.5$), we can see that up to 2 hops

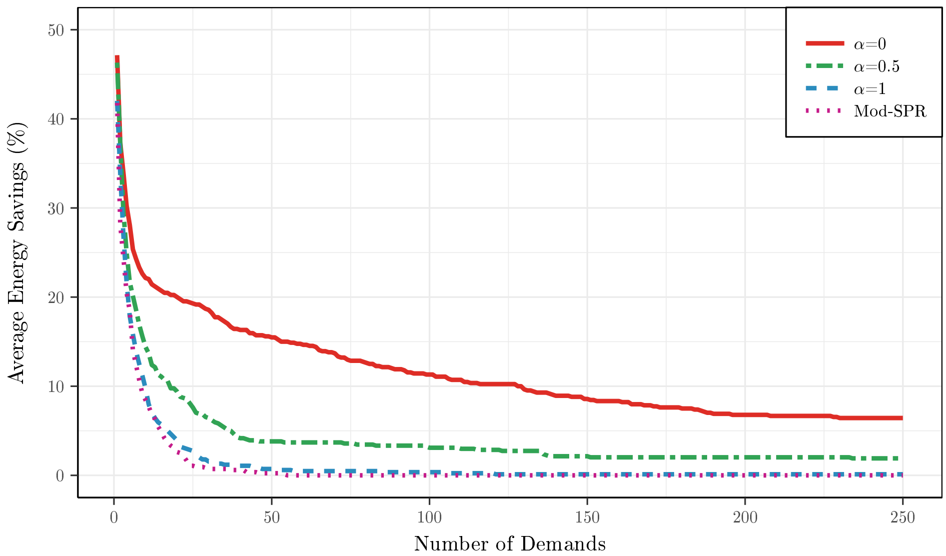


Fig. 5.4: Average energy saving in Abilene with generated traffic demands.

can be added to a control path, yet for more than 60% of demands all the associated control traffic is allocated over the shortest path.

5.4.3 Blocking Rate

To further evaluate the implications of a more demanding traffic load in the energy efficiency and the number of accepted requests, 250 demands were generated with random sources and destinations. In order to get a higher amount of incoming requests with lower bandwidth requirements, we assume an exponentially distributed traffic rate with mean value of 100 Mbps. A random delay ranging from the shortest path length to x times this value (where x is the total number of switches) is used to consider a wider range of different delay requirements.

On the one hand, it can be confirmed from Fig. 5.4 that a higher volume of traffic will imply the use of more active links and thus, less energy can be saved. In the three scenarios, the increasing load limits the possibilities of aggregating the traffic and new paths are required in order to avoid the blocking. However, some savings can still be achieved when not all the incoming traffic belongs to the class 1.

On the other hand, Fig. 5.5 shows the performance of the considered approaches in terms of blocking rate. We can see in this figure that, in all cases, the proposed routing algorithm significantly outperforms the Mod-SPR in terms of accepted demands. This result is mainly because, while Mod-SPR evaluates only the shortest path and if it is saturated blocks the traffic

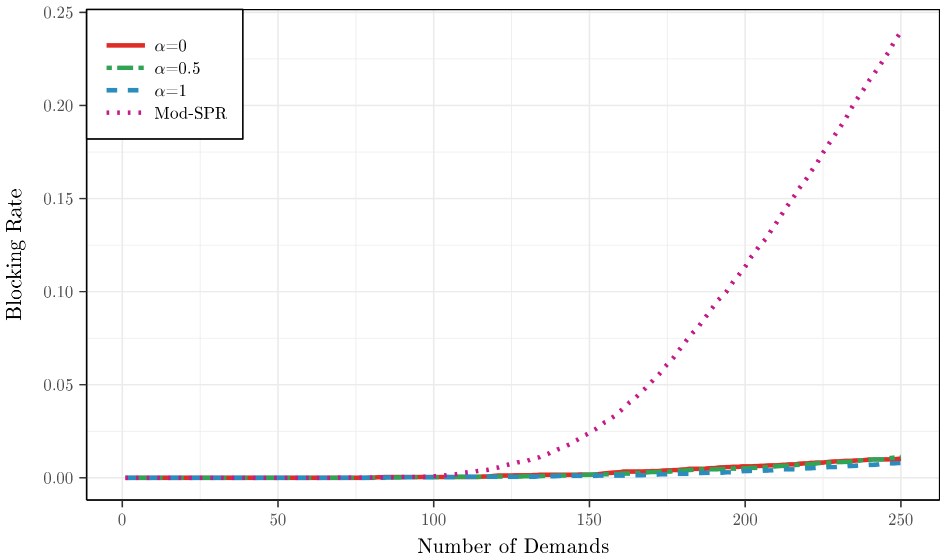


Fig. 5.5: Average blocking rate in Abilene with generated traffic demands.

demand, the proposed evolutionary algorithm finds the most suitable paths across the network taking into account the considered objective functions. Furthermore, the best-fit scheme used in the first objective function also contributes to mitigate the blocking since it accommodates the traffic trying to leave more resources available for future demands. Although, following this reasoning, a lower number of blocked demands is expected when a higher number of demands uses the paths prioritizing the first objective function, it can be appreciated in Fig. 5.5 that the performance of our algorithm in this topology is almost the same in terms of accepted demands for the three scenarios considered. This is caused by the topological characteristics of the Abilene network, where the blocking rate depends heavily on the low redundancy of admissible data paths.

5.5 Conclusion

In this chapter, we have proposed a multi-objective routing approach jointly considering QoS requirements and energy awareness, suitable for SDN environments with in-band control traffic. To achieve this, we have formulated an optimization problem that integrates the routing requirements for data and control traffic and implemented this approach using a MOEA based on SPEA2. In this way, multiple objectives are jointly optimized achieving the best compromise solution between crucial performance goals. Besides of being an effective routing scheme, the most significant added value of this proposal is the flexibility of driving different solutions according to the changing traffic scenario. Extensive simulations using a real topology with static and randomly generated traffic matrices validate the performance improvement on critical network parameters such as energy efficiency, control traffic delay and blocking rate. The proposed strategy is a valuable and preliminary approach for designing routing schemes for SDN control planes aware of multiple objectives.

Chapter 6

Dynamic Performance and Congestion Aware Energy Efficient Routing

This chapter is based on:

- **A. Fernández-Fernández**, C. Cervelló-Pastor, L. Ochoa-Aday and P. Grosso, "An Online Power-Aware Routing in SDN with Congestion-Avoidance Traffic Reallocation," in *Proc. of the 17th IFIP-TC6 International Networking Conference (NETWORKING'18)*, Zurich, Switzerland, May 2018, to be published.

6.1 Introduction

In previous chapters we assume that the entire set of traffic demands is fixed and known in advance, thus all the routes (for data and associated control traffic) are computed jointly in a global optimization process. Although this assumption is completely suitable for planning purposes and operative transport networks with predictable incoming traffic, it can be a limitation for more dynamic networking environments. Inspired by this reality, the work in this chapter aims to adapt previous heuristic approaches to support time-variable traffic demands while performing online rearrangements of previously allocated paths.

In essence, the approach presented in this chapter builds on the work in Chapter 4 combining a control plane configuration with a dynamic routing for an SDN architecture with multiple controllers and in-band control traffic. However, such approach is leveraged to dynamically reduce the number of active nodes and links required to manage the incoming traffic in real-time.

Furthermore, instead of restricting the potential of power-aware solutions to low-loaded environments, a novel strategy is proposed to manage more demanding environments. To do so, a congestion-aware rerouting is also included to efficiently manage currently active resources while avoiding the performance degradation of higher priority demands.

The rest of this chapter is structured as follows. The energy consumption optimization problem is formalized in Section 6.2 through a link-based mathematical formulation, which extends previously presented models to support that forwarding nodes can be put into sleep mode. In Section 6.3 we explain the main features of our heuristic power-aware approach together with a detailed description of its two comprised modules. The simulation strategies and the obtained results are presented and analyzed in Section 6.4. Finally, Section 6.5 concludes this chapter.

6.2 Problem Statement

To formalize the power consumption optimization problem in SDN, in this section we present its mathematical formulation. The proposed model computes in a global optimization process all the optimal control and data paths in terms of power efficiency. To that end, similar to previously presented models, the incoming traffic demands are considered fixed and known in advance. Although this assumption is not suitable for dynamic scenarios, the purpose of this model is to provide optimal bounds for the energy efficiency that can be achieved at any particular time for a given traffic load flowing through the network.

In this case, to optimize the overall power consumption the number of active network elements (links and nodes) is minimized. Therefore, this model leverages preliminary work presented in Sections 3.2 and 4.2 supporting that forwarding nodes are put into sleep mode.

Being a general formulation, multiple controllers as well as SDN with in-band mode are also supported by this proposal. Given the controllers placement, this model also determines the optimal distribution of switches between controllers in terms of power efficiency and load balancing.

6.2.1 Network Model

In the proposed scheme the network topology can be modeled as a graph $G = (V, E, C)$, where V, E and C denote the set of nodes, links and controllers respectively. We define the set of forwarding nodes as $S = \{n \mid n \in V \wedge n \notin C\}$. Note that network devices can only fulfill one role, i.e. controller or forwarding node. We use $c_{i,j}$ to denote the capacity of a link $(i, j) \in E$.

Considering F as the entire set of traffic flowing through the network between any pair of nodes, let D denote the subset corresponding to data plane communications. For the control plane, we use T to denote the subset of traffic between controllers and switches, and H to denote the subset of control traffic between controllers. Each flow $f \in F$ from source s_f to destination t_f, has associated a throughput, denoted by d_f.

6.2.2 Formulation

To formulate such optimization problem, the required variables, objective functions and constraints are defined as follows:

$x_{i,j}$: describes the state of a link $(i, j) \in E$.

$$x_{i,j} = \begin{cases} 1 & \text{if } (i,j) \text{ is active,} \\ 0 & \text{otherwise.} \end{cases}$$

y_v: describes the state of a node $v \in V$.

$$y_v = \begin{cases} 1 & \text{if } v \text{ is active,} \\ 0 & \text{otherwise.} \end{cases}$$

$t^f_{i,j}$: describes the selection of a link $(i, j) \in E$ to route a flow $f \in F$.

$$t^f_{i,j} = \begin{cases} 1 & \text{if } (i,j) \text{ is selected to route } f, \\ 0 & \text{otherwise.} \end{cases}$$

$\lambda_{n,c}$: describes the association of each forwarding node $n \in S$ with a controller $c \in C$.

$$\lambda_{n,c} = \begin{cases} 1 & \text{if } n \text{ is associated with } c, \\ 0 & \text{otherwise.} \end{cases}$$

The objective function of our model seeks to reduce the overall power consumption considering the number of active nodes and links in the network. Consequently, both elements are integrated in the following expression, where P_p and P_n denote the power consumption of a port and a node, respectively.

$$\text{minimize} \quad 2P_p \sum_{(i,j)\in E} x_{i,j} + P_n \sum_{v \in V} y_v \tag{6.1}$$

A single controller must be selected to manage each active forwarding node in the network.

$$\sum_{c \in C} \lambda_{n,c} = y_n \qquad \forall n \in S \tag{6.2}$$

With the objective to avoid congested controllers, we set the maximum number of forwarding nodes that can be associated with each controller. In this way, active switches are evenly distributed and the load is balanced among controllers.

$$\sum_{n \in S} \lambda_{n,c} \leq \left\lceil \frac{\sum_{n \in S} y_n}{|C|} \right\rceil \qquad \forall c \in C \tag{6.3}$$

A node $v \in V$ is active if there is traffic in any of its incoming or outgoing edges, being $N(v)$ the set of neighbors of v.

$$y_v \geq \frac{1}{2\,|F|} \sum_{f \in F} \left(\sum_{u \in N(v)} t^f_{u,v} + \sum_{u \in N(v)} t^f_{v,u} \right) \qquad \forall v \in V \tag{6.4}$$

To avoid additional traffic load through network controllers, data plane communications (i.e. $f \in D$) cannot be routed through these devices. Furthermore, control traffic between controllers and switches (i.e. $f \in T$) will not pass through any other controller that is not the source or target of the traffic. The same must hold true for the traffic between controllers (i.e. $f \in H$).

In these constraints we use $N(c)$ to denote the set of neighbors of a controller $c \in C$ and v_f

to identify the forwarding node involved in the source/target pair of traffic flow $f \in T$.

$$\sum_{n \in N(c)} t^f_{n,c} \leq \begin{cases} 0 & \forall f \in D, \forall c \in C \\ \lambda_{v_f,c} & \forall f \in T, \forall c \in C \\ 0 & \forall f \in H, \forall c \in C \setminus \{s_f, t_f\} \end{cases} \tag{6.5}$$

The routing of data plane communications and control traffic exchange between controllers, follows the traditional flow conservation constraints.

$$\forall v \in V, \forall f \in D \cup H : \tag{6.6}$$

$$\sum_{u \in N(v)} t^f_{v,u} - \sum_{u \in N(v)} t^f_{u,v} = \begin{cases} 1 & \text{if } v = s_f \\ -1 & \text{if } v = t_f \\ 0 & \text{otherwise} \end{cases}$$

Meanwhile, for the subset of traffic between controllers and switches $f \in T$, these constraints are modified to assure that only active switches exchange control messages with its controller. Similarly, the forwarding node and the controller involved in the source/target pair of traffic flow $f \in T$, are denoted with v_f and c_f, respectively.

$$\forall v \in V, \forall f \in T : \tag{6.7}$$

$$\sum_{u \in N(v)} t^f_{v,u} - \sum_{u \in N(v)} t^f_{u,v} = \begin{cases} y_v \lambda_{v_f,c_f} & \text{if } v = s_f \\ -y_v \lambda_{v_f,c_f} & \text{if } v = t_f \\ 0 & \text{otherwise} \end{cases}$$

Finally, a link (i, j) is active if it is used by some traffic flow $f \in F$. Furthermore, the total traffic in each active link must be less than its assigned capacity.

$$\sum_{f \in F} t^f_{i,j} d_f \leq c_{i,j} x_{i,j} \qquad \forall (i, j) \in E \tag{6.8}$$

To compute all the routes (i.e. for data and associated control traffic) using this global optimization model, the entire set of traffic demands need to be fixed and known in advance. Considering this as a limitation for current dynamic networking environments, in the next section we propose a new approach to support time-variable traffic requirements.

6.3 Heuristic Algorithms

The key idea of this proposal is to fully take advantage of the high control flexibility given by the dynamic configuration capabilities and centralized network view of SDN without needing an accurate prediction of incoming traffic. In order to allow that nodes are put into sleep mode we assume network topologies with forwarding nodes divided into two categories: edge nodes (N), which are connected to some traffic source/target and transit nodes (I), which merely route other nodes traffic.

6.3.1 Green Initial Setup (GrIS)

An initial control plane configuration, previous to the data traffic arrival, is required in order to support the in-band mode in SDN. This control plane setup is intended to establish the required paths between switches and controllers, as well as between controllers. In this way, when new traffic flows arrive, switches can send to the controller routing requests through packet_in messages. To do so, in this section we propose an off-line solution denoted as Green Initial Setup (GrIS). This component will be statically activated at specific time instances as a planned operation.

The proposed strategy, shown in Algorithm 8, takes as inputs the original network topology G with the controller placements and the set of forwarding nodes (i.e., C and S), the subsets of edge and intermediate nodes (i.e., N and I) and the control traffic requirements R^c. The outputs are a reduced graph with the initially active network elements $G^A = (V^A, E^A, C)$, an array keeping the controller-switch associations A and the initially required control paths P_c.

In the first step, the algorithm attempts to reduce the number of initially activated nodes using the Net_Pruning function, shown in Algorithm 9. This method aims to remove as many devices as possible, considering as candidates the set of intermediate forwarding nodes I. These transit nodes are meant to be core nodes which will not generate or receive data traffic.

For each node inside the set I of intermediate nodes, the function computes its betweenness centrality (B_n), as a measure of its intermediary role in the network (lines 4-13). In the proposed approach, we use a simplified version of this metric considering only the shortest paths from each

Algorithm 8 GrIS Pseudocode

Require: G, R^c, N, I
Ensure: $G^A = (V^A, E^A, C), P_c, A, U$

```
1:  G′ ← NET_PRUNING(G, I)
2:  O ← Get_All_Admissible_Control_Paths(G′, R^c)
3:  N′ ← N sorted by nodes priority criteria
4:  f ← First node in N′
5:  |V^A|, |E^A| ← ∞
6:  repeat
7:      for p ∈ O[f] do
8:          Initialize (V^A′, E^A′, P′_c, A′, U′) routing p
9:          for intermediate node u ∈ p not in N do
10:             Power_Aware_Path_Selection(O[u])
11:             Update (V^A′, E^A′, P′_c, A′, U′)
12:         end for
13:         for edge node n ∈ N′ \ f do
14:             r = Power_Aware_Path_Selection(O[n])
15:             Update (V^A′, E^A′, P′_c, A′, U′)
16:             for intermediate node v added to V^A by r do
17:                 Power_Aware_Path_Selection(O[v])
18:                 Update (V^A′, E^A′, P′_c, A′, U′)
19:             end for
20:         end for
21:         for (c1, c2) ∈ G′ do
22:             Power_Aware_Path_Selection(O[c1, c2])
23:             Update (V^A′, E^A′, P′_c, A′, U′)
24:         end for
25:         if |V^A′| ≤ |V^A| ∧ |E^A′| ≤ |E^A| then
26:             V^A, E^A, P_c, A, U ← V^A′, E^A′, P′_c, A′, U′
27:         end if
28:     end for
29:     if |V^A| = ∞ ∨ |E^A| = ∞ then
30:         if f = last node in N′ then break
31:         end if
32:         f ← Next node in N′
33:     end if
34: until |V^A| ≠ ∞ ∧ |E^A| ≠ ∞
```

controller to every other node in the network (line 7). In particular, after computing the shortest paths from each controller as single source, the B_n associated with a node n is increased for each path containing the node (line 9). Using these values, in line 14, transit nodes are sorted and stored in the list I'. At each iteration of this list the function attempts to increase the number of switched-off nodes (lines 15-23). A new node is removed only when in the resulting graph forwarding nodes remain being reachable by network controllers (lines 17-18), i.e. at least one

Algorithm 9 NET_PRUNING

Require: G, I
Ensure: $G' = (V, E', C)$
$G' \leftarrow G$
$Z \leftarrow$ NULL ▷ Removed nodes
$B \leftarrow$ NULL ▷ Array of betweenness values
for $n \in I$ **do**
  $B_n = 0$
  for $c \in C$ **do**
    $SP_c \leftarrow$ Set of shortest paths from controller $c \in C$ to every other node $v \in V$
    for $p \in SP_c$ **do**
      if path p goes through node n **then** $B_n = B_n + 1$
      end if
    end for
  end for
end for
$I' \leftarrow I$ sorted according to increasing order of B
for $n \in I'$ **do**
  Remove from G' node n
  if nodes are still reachable by controllers in G' **then**
    Save n in Z
  else
    $G' \leftarrow G$
    Remove from G' nodes in Z
  end if
end for

path exists between every controller-switch pair in the network.

To accomplish this, a temporal graph, denoted as G', is created. This graph is used to check the required connectivity between controllers and forwarding nodes. After validating that the possibility of reaching every node in the network from any controller is not affected, the considered node is removed from the resulting graph. This means that these nodes together with their links are put into sleep mode in the original graph.

After pruning the network, the GrIS algorithm uses the reduced graph G' to find the overall set of admissible control paths which satisfy control traffic requirements R^c (line 2 in Algorithm 8). As previously stated, these paths do not pass through any other controller that is not the source or target of the traffic. Using these computed control paths, a sorted list of ingress and egress forwarding nodes is stored in N' (line 3). This list is sorted in ascending order following two priority criteria:

1. the number of possible controllers to associate with,

2. the number of possible control paths.

Going through this list, the algorithm starts satisfying the most critical cases and the solution can be found with fewer iterations. The main loop of the Algorithm 8 (lines 6-34) determines for each possible control path of the selected node f, the number of active elements in the network after computing all control routes. The configuration of paths with fewer active elements is then selected in this process.

Inside this loop the algorithm first determines the paths between controllers and forwarding nodes (lines 13-20). Note that, for each forwarding node x, $O[x]$ contains admissible control paths to each controller available in the network (lines 7, 10, 14 and 17). Precisely, paths selected in this step define one controller for each forwarding node. Additionally, any time a path between a switch and a controller is computed, intermediate nodes belonging to the control path but not yet in the set of currently active nodes V^A are analyzed by the algorithm (lines 9-12 and 16-19). Note that these nodes are the transit nodes that remained in the resulting graph after pruning the network. Since they are used to route some traffic, a control path is also established between them and some controller. After determining all switch-controller associations, the algorithm searches the paths between controllers (lines 21-24).

In general, the power-aware path selected for every control pair is the best route between them in terms of minimizing the number of active elements in the network as long as it has sufficient link capacity to route the traffic volume, under the considered MLU constraint. Additionally, during the selection of one control path between each forwarding node and one controller, the number of forwarding nodes already attached to the controllers is considered in order to keep a balanced load. Since the set of admissible paths obtained from the pruned network with a reduced number of elements is significantly smaller than in the original topology, the solution can be found with fewer iterations.

If after analyzing all control paths of node f, the algorithm cannot find a feasible configuration of paths to route all control and data plane traffic, the main loop repeats this process for the next node stored in N' (line 32). This is done until the solution is found or until all forwarding nodes are analyzed, i.e. when the algorithm breaks without a solution (line 30). Note that this last option occurs when, given a controllers placement, an admissible configuration for controller-switches association could not be found or when the network has not sufficient capacity to meet the control traffic requirements under established constraints.

6.3.2 Dynamic Power-Aware Routing (DyPAR)

When a new traffic demand arrives, a routing request is sent from the input node to its associated controller using the previously computed path between both devices. Based on its global knowledge of the network topology, this controller calculates the required data path minimizing the number of elements that need to be activated for this connection request and creates the flow forwarding rules. The proposed dynamic power-aware routing, denoted as Dynamic Power-Aware Routing (DyPAR) and shown in Algorithm 10, performs in essence three crucial functions:

1. Power-aware data and control path selection;
2. Performance-aware data path selection;
3. Congestion-aware traffic reallocation.

For each incoming demand d, the algorithm starts trying to get the set of admissible data paths across the current active topology (line 1). This is done considering that admissible data paths do not pass through any controller in the network. If several paths were found, the one with the highest remaining available link capacity is selected (line 3). In this way, the number of future requests that can potentially be accommodated over the currently active paths is increased. Then, traffic is allocated and links utilization are updated (line 4).

On the other hand, if no admissible data path was found to route the incoming traffic across the currently active topology, the original network graph is then considered by the algorithm (line 6). Since the now determined candidate routes will imply the use of additional network elements, the most favorable admissible data path in terms of power consumption, i.e. the one minimizing the number of active network elements, is selected to carry the demand (line 8).

After updating the active topology and links utilization in line 9, a loop is used to establish the required control plane communications for each added node along the data path (lines 10-17). In the same way, the algorithm first considers the currently active topology to set the required control path with some network controller (line 11) and the entire network in case of failing the initial attempt (line 13).

Algorithm 10 DyPAR Pseudocode

Require: G, G^A, P_c, A, U, d
1: $P_d \leftarrow$ Get_Admissible_Paths(G^A, d)
2: **if** $P_d \neq Null$ **then**
3:     $p_d \leftarrow$ Lest loaded path in P_d
4:     Update U after routing p_d
5: **else**
6:     $P_d \leftarrow$ Get_Admissible_Paths(G, d)
7:     **if** $P_d \neq Null$ **then**
8:         $p_d \leftarrow$ Power_Aware_Path_Selection(P_d)
9:         Update G^A, U after routing p_d
10:         **for** node n added to G^A by p_d **do**
11:             $P_c \leftarrow$ Get_Admissible_Paths(G^A, n, C)
12:             **if** $P_c = Null$ **then**
13:                 $P_c \leftarrow$ Get_Admissible_Paths(G, n, C)
14:             **end if**
15:             $p_c \leftarrow$ Power_Aware_Path_Selection(P_c)
16:             Update G^A, U, A after routing p_c
17:         **end for**
18:     **else**
19:         $P_d \leftarrow$ Get_All_Paths(G, d)
20:         $p_d \leftarrow$ Performance_Aware_Path_Selection(P_d)
21:         Update U, T after routing p_d
22:     **end if**
23:     $BL \leftarrow$ Link with maximum load
24:     $F \leftarrow$ Demands established through BL
25:     Congestion_Aware_Rerouting(G^A, F, BL, U)
26: **end if**

In case of incoming traffic rates exceeding the remaining available network capacity, the algorithm considers all data paths in the original network without taking into account the capacity restrictions, but keeping that data plane traffic cannot be routed through network controllers (line 19). Then, the algorithm performs a data path selection based on reducing the performance degradation incurred. More in detail, the algorithm in line 20 selects the data path inside this group whose congested links are less used by previously established demands. The reason is that, to allow the new traffic flow, the capacity remaining on those links, after allocating the QoS sensitive demands and control traffic, will be fairly divided between the involved lower-priority demands. Rates beyond this resulting throughput will be reduced and traffic will be handled on a "best-effort" basis. In this way, the proposed algorithm can efficiently handle bursty traffic and accommodate rates that exceed the remaining available capacity without affecting the QoS sensitive traffic if the network is not heavily loaded.

Algorithm 11 CONGESTION_AWARE_REROUTING

Require: G^A, F, BL, U
1: $Current\_MaxU \leftarrow U[BL]$
2: $G'' \leftarrow G^A$
3: Remove BL from G''
4: $F' \leftarrow F$ sorted by decreasing order of flow rate
5: **for** established demand f in F' **do**
6: $\quad P \leftarrow$ Get_Admissible_Paths(G'', f)
7: $\quad p \leftarrow$ Congestion_Avoidance_Path_Selection(P)
8: $\quad MaxU_p \leftarrow$ Maximum link utilization in p
9: $\quad$ **if** $p \neq None \wedge MaxU_p < Current\_MaxU$ **then**
10: $\qquad$ Reroute f and associated control traffic
11: $\qquad$ Update U and $Current\_MaxU$
12: $\quad$ **end if**
13: **end for**

Every time a new network element is added to the active topology, the algorithm tries to alleviate the congestion level on the network. To accomplish this, after identifying the bottleneck link (line 23) and the group of traffic flows going through this link (line 24), a CONGESTION_AWARE_REROUTING is performed (line 25). This function, described in Algorithm 11, starts creating in line 2 a temporal graph G'' where the most loaded link is removed (line 3). Additionally, currently established demands sharing the most loaded link are sorted in decreasing order of rate requirements with the aim of reducing the congestion after rerouting the fewer number of connections (line 4). In order to avoid frequent reallocations of a traffic flow and mitigate related negative implications, a time threshold can be easily included to select only demands that have been allocated long enough over the current path.

Using the residual graph a new set of admissible paths is obtained for each involved traffic flow (line 6). Then, the function looks for a path with lower load values trying to leave more resources available for future demands (line 7). A traffic flow is reallocated only when a feasible path is found and the MLU in the network is reduced (line 9). At the same time, the required control paths are updated (line 10).

Since the proposed approach is conceived for dynamic traffic environments, the set of established demands will be constantly checked. For those connection requests whose holding time have expired, the algorithm performs a demand removal, which means that their assigned paths are released and resources occupied by these routes become available again. Consequently, network elements used only by completed traffic demands will be then put into sleep mode.

6.3.3 Complexity Analysis

To derive the computational complexity of the green initial setup presented in Algorithm 8 we should first consider the NET_PRUNING function. This function, shown in Algorithm 9, performs two iterative operations over the set of I transit nodes. Considering that, using the Dijkstra algorithm, the single source shortest paths are computed for each controller in $O(V \log V)$, and that $O(V + E)$ time is required by the connectivity checking process, being V and E the amount of network nodes and links, respectively, the NET_PRUNING function has a complexity equal to $O(I(CV \log V + E))$.

Additionally, the runtime of Algorithm 8 is also due to its main loop. In the worst case, this while loop will be executed N times, being N the number of forwarding nodes which are endpoints of traffic demands. However, it should be emphasised that as a consequence of iterating over an ordered list, in most of the cases the algorithm is able to find a solution after analyzing the first node and this extreme case will be quite uncommon. Inside this loop, the iterative process and the related complexity are directly linked to the maximum number of admissible paths between any pair of nodes and the number of control plane connections to be initially established, which are considered along the inner loops. It is easy to conclude that the maximum number of admissible paths, denoted here as M, cannot be found beforehand since it will depend of several factors such as the network topology and the number and location of controllers. Therefore, the overall algorithm complexity should be formulated based on it. With respect to the number of control plane connections, they can be estimated by the upper bound $O(S + H)$, where S denotes the total number of forwarding nodes and H refers to the connections between controllers. Thus, the complexity of the main loop of Algorithm 8 can be expressed as $O(NM^2(S + H))$.

Regarding the dynamic power-aware routing proposed in Algorithm 10, the worst run-time complexity is imposed by data demands requiring, in addition to the generation of the data path, the activation of new forwarding nodes and the corresponding computation of a control path between them and the controllers. Since simple paths can be found in $O(V + E)$ time, using a modified depth-first search (poner la ref), and taking M as the maximum number of admissible paths, the algorithm complexity can be defined as $O(M(I+D)(V+E))$, where I and D denote, respectively, the upper bounds of transit nodes to be activated and data demands to be rerouted. Precisely, in the previous expression the $O(MD(V+E))$ component is result of applying the Congestion_Aware_Rerouting function described in Algorithm 11.

6.4 Simulations and Results

In this section we describe the evaluation of our power-aware approach and analyze the achieved results. We used the linear programming solver Gurobi Optimizer [110] to assess the performance of the ILP model. Meanwhile, the proposed control framework previously described was implemented using the programming language Python to develop the heuristic algorithms. All computations were carried out on a computer equipped with 3.30 GHz Intel Core i7 and 16 GB RAM.

6.4.1 Simulation Scenario

6.4.1.1 Network Topology

Similar to previous chapters, we conducted our simulations using real-world network topologies collected from SNDlib [111], considering each router in the network as an SDN node or as a controller placement. Specifically, we use three topologies of different sizes in order to assess the effectiveness of the proposed scheme in small, medium and large scale networks. The mentioned networks are: Nobel-US ($|V| = 14$; $|E| = 21$), Geant ($|V| = 22$; $|E| = 36$) and Cost266 ($|V| = 37$; $|E| = 57$). The topology of Cost266 can be seen in Fig. 6.1.

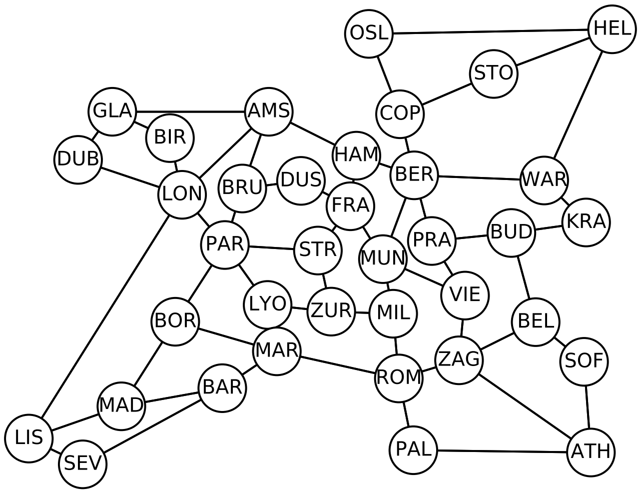


Fig. 6.1: Network topology of Cost266 used in the simulations.

To allow the possibility of putting network nodes into sleep mode, different scenarios were considered varying T, which represents the percentage of forwarding nodes that will not generate or receive traffic. According to this value, for each network topology we have selected as transit nodes the devices with the highest degree centrality as in [54, 79].

6.4.1.2 Controllers Placement

Being the controller placements out of the scope of this work we assume as preferred locations the ones minimizing the worst-case mean latencies. More precisely, we compute the mean propagation latency between each pair of nodes and associate each admissible location with the maximum average value involving it. Then, according to the number of controllers considered for each simulation instance, we place the controllers at node locations with smaller associated latency values. Note that a controller placement is admissible when the assumptions established in this proposal to avoid the routing of additional traffic load through network controllers can be kept (i.e. the network graph without any controller remains being strongly connected).

6.4.1.3 Traffic Patterns

Apart of the real static traffic matrices obtained from the topologies database in [111], we also performed some tests considering a dynamic scenario where connection requests arrive with exponentially distributed inter-arrival and holding times, taking different mean values from the sets [0.2, 1, 5] and [100, 150, 200], respectively. Accordingly, a traffic demand is generated between each pair of edge nodes (i.e. network devices which do not act as controllers or transit nodes).

Additionally, we evaluated the power savings and performance degradations considering increasing loads. To accomplish this, for each network topology we considered every pair of edge nodes with an initial random assigned data rate and computed the required data paths according to the SPR. We then identified the most loaded link from which we derived a scaling factor. Lastly, the initially assigned values were multiplied by this scaling factor to obtain the corresponding data rates for each incoming demand (see [55]). This was done considering different values of over-provisioning factor (α) to further evaluate the implications of varying traffic load. For the control traffic we assume an average rate of 1.7 Mbps [112].

6.4.1.4 Power Values

Based on the power consumption behavior of data networks explained in Section 2.1, we characterize the power consumption of a forwarding node according to the 3:1 idle:active ratio given in [46]. This proportion, obtained from measurements on real switches, assigns 3W of power for each idle port of a switch and 1W extra when the port is active. Thus, power consumption P_n of an idle forwarding node n can be computed as $3D(n)$ where $D(n)$ denotes the node degree and $P_p = 1$W. Null power consumption is assumed when the node is put into sleep mode.

6.4.2 Optimal vs. Heuristic Solutions

To assess the suitability of the proposed solution we start evaluating the performance of the heuristic algorithms against the optimal ILP model, using the Nobel-US and Geant topologies. In this particular case, we use the traffic matrices provided in [111] assuming a static scenario where demands are allocated during the entire simulation (i.e. their holding times are set to infinite and no demand removal is performed by the dynamic approach). On the other hand, given that DyPAR is an online routing strategy, the connection requests are sequentially allocated as they appear in the considered traffic matrix. This comparison is illustrated in Fig. 6.2 for different amount of controllers placement and percentage of transit nodes.

Power savings are computed using the following expression:

$$\frac{Overall\_Pw - Pw\_X}{Overall\_Pw} \tag{6.9}$$

where Pw_X refers to the power consumption achieved by the considered approach (i.e. Optimal or DyPAR) and $Overall\_Pw$ can be determined as:

$$Overall\_Pw = \sum_{n \in V} P_n + 2P_p\,|E| \tag{6.10}$$

In Fig. 6.2 power savings of up to 35% can be reached by our optimization model in both topologies. Moreover, the heuristic approach allows to obtain close-to-optimal power savings with differences under 4% (Nobel-US) and 8% (Geant). It is also observed in both networks that lower savings are achieved when the percentage of transit nodes decreases from 50% to 10%. This behavior is expected given that a reduction in the percentage of considered transit

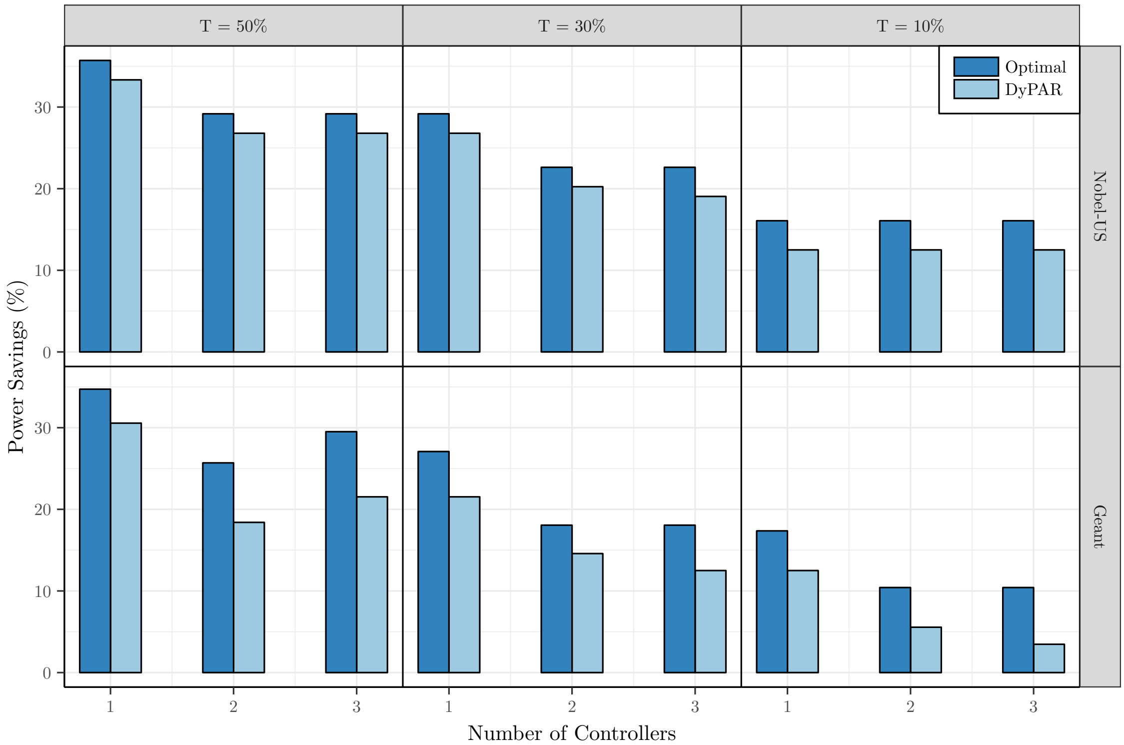


Fig. 6.2: Power saving in the Nobel-US topology as a function of controllers amount, varying the percentage of transit nodes (T).

nodes means that a smaller number of forwarding nodes can be put into sleep mode, which yield the major contribution to the attained power savings. Additionally, fewer transit nodes imply that a higher number of demands are handled, thus more paths need to be established to accommodate such traffic.

On the other hand, an increase in the number of controllers can also cause in some cases a reduction in the power savings. The reason is that in our approach data plane traffic cannot be routed through network controllers. When the number of network controllers grows, a higher number of links used to route control traffic (i.e. links directly connected to the controllers), cannot be used for data plane communications. Therefore, more links are needed to route the data and control traffic.

6.4.3 Assessment of Power Saving Potential

Due to the computational complexity of the exact model in networks similar in size or larger than Geant (see 4.4.1 for similar running time values), in what follows we use our heuristic algorithms. This is done taking into account a dynamic scenario with connection requests generated following the procedure previously explained. Several test were conducted and average values have been determined with a margin error less than 5.5% in the three considered networks, estimated by running our algorithm 10 times with different prime number seeds on each traffic configuration instance.

In terms of average running time of the algorithms, the off-line GrIS module requires around 39 ms (Nobel-US), 0.25 s (Geant) and 283 s (Cost266). Meanwhile, the DyPAR algorithm takes always less than 6.4 ms (Nobel-US), 16.5 ms (Geant) and 282.6 ms (Cost266), for all the considered traffic patterns.

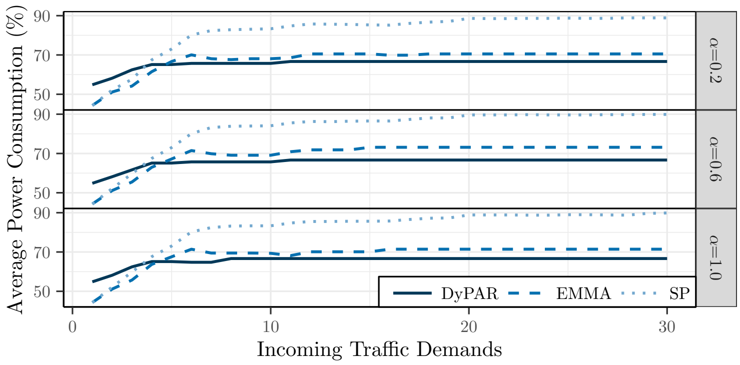
In addition, in order to evaluate the benefits of our proposal we compare the performance of the proposed algorithms with other two existing energy-aware routing approaches presented in related works [101] and [104], referred to here as SP and EMMA, respectively. As we are considering an in-band SDN, required control plane communications will be also established by these two approaches. At the same time, shortest paths used by SP and EMMA are computed holding restrictions established to avoid additional traffic load through the network controller (i.e. data traffic cannot be routed through this device). On the other hand, we set the time threshold for demands reallocation (half of connection expected duration) and the number of transit nodes ($T = 50\%$) as in [104] for the three algorithms used in this comparison.

Given the lack of support in SP and EMMA for network environments with multiple controllers we only consider the case of having one centralized network controller. However, the derived conclusions are general and a similar behavior is expected in case of having multiple controllers.

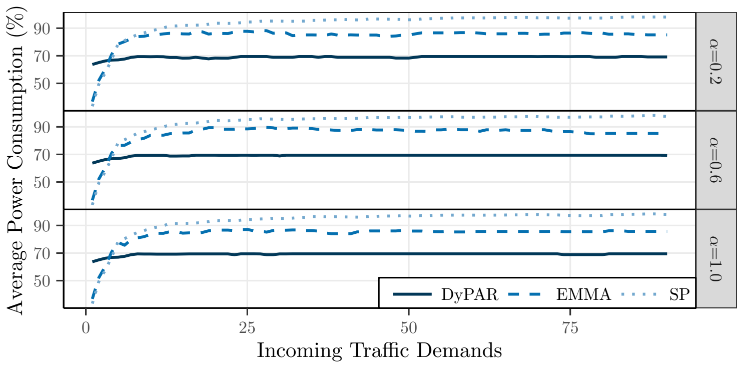
Fig. 6.3 shows the power consumption achieved by the three algorithms considering different topological scenarios and over-provisioning factor (α). These results correspond with an average arrival time of 0.2 demands/s and a mean holding time of 100 s, but similar values have been obtained for all the considered traffic patterns.

Given the initial control plane configuration performed by the GrIS module, in the three considered topologies the other two methods exhibit a better behaviour at the beginning of simulations. However, after allocating few demands more power can be saved by our approach.

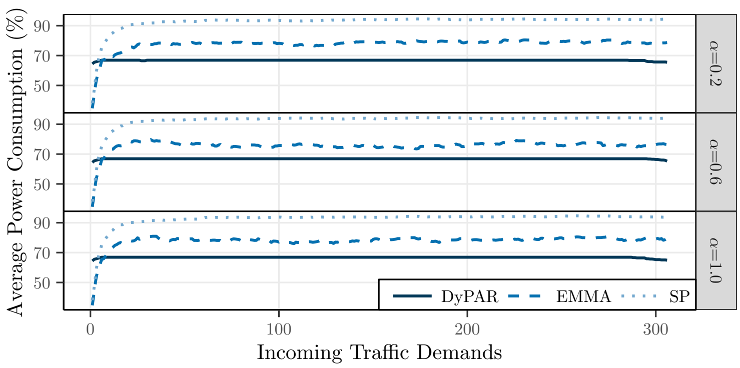
As it is shown, in terms of consumed power, DyPAR outperforms SP in all cases and it is generally better (in some cases just slightly better) than EMMA. For instance, after routing all incoming traffic, DyPAR attains power consumption reductions of up to 26.5% and 19.4% with



(a) Nobel-US topology.



(b) Geant topology.



(c) Cost266 topology.

Fig. 6.3: Power consumption in the three topologies with one controller as a function of traffic arrival, varying the over-provisioning factor (α).

respect to SP and EMMA, respectively. The reason is that SP only uses pre-computed shortest paths to allocate the incoming traffic, while EMMA also performs a power-aware rerouting any time the active topology changes in order to find better paths for already allocated flows.

On the other hand, power improvements achieved by our proposal are consequence of the combined GrIS/DyPAR operation where a minimum network subset is initially activated and new network elements (nodes and links) are only added when the incoming demand cannot be allocated on the currently active topology.

6.4.4 Performance Degradation Avoidance

These power savings are only valid if the performance of QoS sensitive demands is not compromised. Moreover, to avoid overloaded networks a capacity reserve is typically set. So far, we had not considered this capacity margin, but now we analyze how the number of allocated demands is impacted when facing a more demanding traffic pattern and in presence of a MLU constraint. In this evaluation we set the average arrival time to 5 demands/s and the mean holding time to 200 s, while keeping the over-provisioning level equal to 1, since this represents the most demanding of the considered traffic patterns for the heuristics and the most critical from the performance degradation perspective.

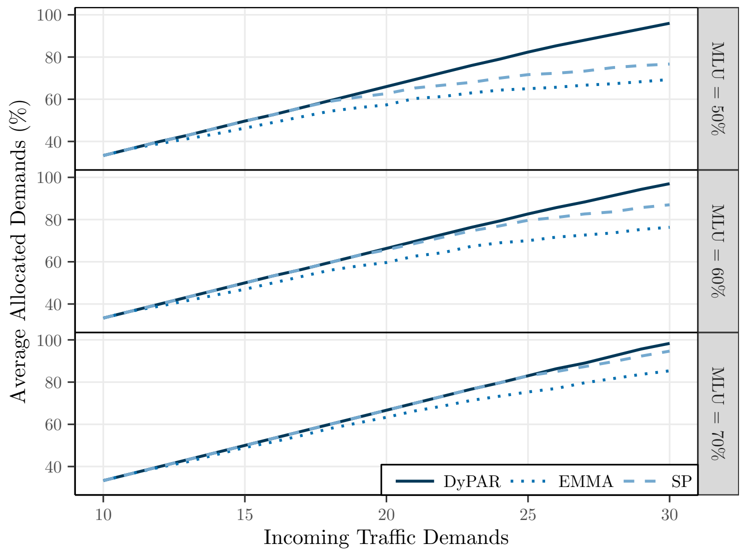
Fig. 6.4 shows the percentage of demands that can be allocated by DyPAR, EMMA and SP in Nobel-US and Geant using different values of MLU.

As it is shown, DyPAR is able to reduce the blocking rate with respect to the other two approaches as a result of the CONGESTION_AWARE_REROUTING performed by this solution. In particular, while only negligible blocking rates are attained by our approach (less than 1.2%), up to 7 and 12 demands are blocked by SP and EMMA, respectively. SP performs better than EMMA given that in case of having more than one candidate route this algorithm selects the one leaving more available link capacity.

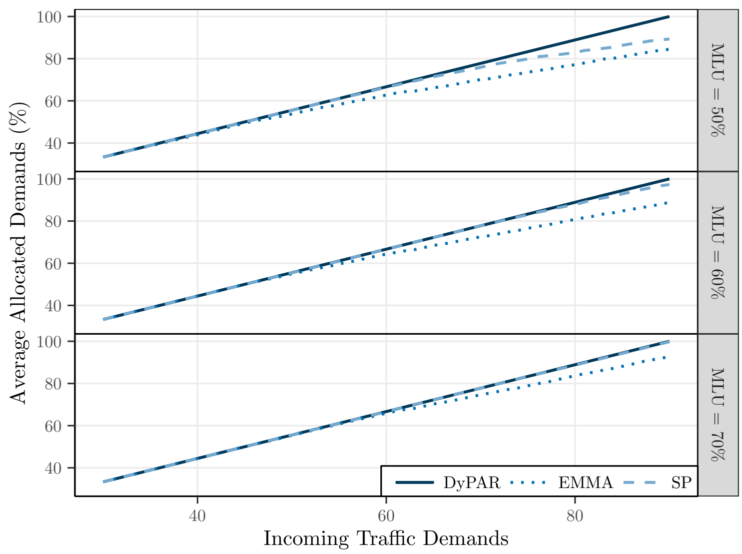
Intuitively, the capacity to successfully allocate the incoming traffic will not only be a result of the performed routing, since it is also related to the considered topology. In network topologies with more path redundancy a higher number of requests can potentially be accommodated. This difference can be noticed between Nobel-US and Geant, where an increase of allocated demands of up to 26.7% and 15.6% can be reached, respectively. Cost266 is not shown in Fig. 6.4, since a complete routing was always achieved in this topology by the three compared algorithms under the considered traffic patterns and MLU levels.

6.5 Conclusion

In this chapter we proposed a power-aware strategy that reduces the number of active nodes and links used to handle the incoming traffic suitable for SDN environments with in-band control



(a) Nobel-US topology.



(b) Geant topology.

Fig. 6.4: Number of allocated demands with one controller as a function of traffic arrival, varying the MLU.

traffic and multiple controllers. To achieve such goal, we first provided a link-based formulation of the optimization problem, integrating the routing requirements for data and control traffic. For large-scale topologies a heuristic approach is conceived combining a static control plane configuration with a dynamic power-aware routing. Besides being compatible with SDN environments without a dedicated control network, this strategy is able to handle demanding traffic arrival without degrading the performance of higher priority traffic. Through simulations using real-world topologies, we have validated that our heuristic approach allows to obtain close-to-optimal power savings, with differences under 8%. Furthermore, our proposal achieves better results in terms of power consumption and number of allocated demands than two existing related algorithms. For instance, after routing all incoming traffic, a reduction of power consumption of up to 26.5% and an increase of allocated demands of up to 26.7% can be reached by our solution.

Chapter 7

Conclusion and Perspectives

Reducing energy consumption of modern communication networks has been recognized as an ever-increasingly important need, given the noteworthy use of electrical energy by the ICT sector. To face the negative environmental implications of the rising penetration of Internet-based services in our daily lives, society and industry, we reckon that programable networks, and especially SDN, can play a very important role. 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. Given that energy efficiency in a telecommunication system relies mainly on the power consumption of its individual components, a coordinated routing and management of the whole network configuration that puts into sleep mode the redundant network devices, is an effective strategy to reduce energy concerns. Adequate solutions to this challenge are needed in order to provide the green support currently required for networks operation.

Throughout this thesis we have addressed some of the most challenging issues regarding the design of energy-aware routing solutions reducing the number of active network elements in SDN with in-band control traffic. It should be emphasized that all the contributions of this research stand as novel and original proposals which were designed from scratch. In other words, we do not adapt or extend existing mechanisms for their conception. This thesis, focused on green routing, has proposed several methods to increase the energy efficiency of wide-range SDN-based carrier networks. We here briefly describe the main outcomes of this research in order to assess the fulfillment of research objectives. In addition, we discuss the identified limitations of the research and propose possible directions in which our work can be extended.

7.1 Research Outcomes

The first research question posed by this thesis *–how to route, in an energy efficient way, connection requests in an SDN environment with multiple controllers–* was answered in Chapter 3. A key observation identified by this thesis is that existing energy-aware routing approaches in SDN do not consider the in-band mode and neglect the energy consumption associated with the control plane communications. Thus, this work contributes to the literature by providing efficient routing solutions to the energy consumption problem in SDN environments without a dedicated control network. The proposed approaches cover the different variants for control plane implementation in SDN (i.e. centralized and distributed). In addition, other particularly important issues of SDN, such as controllers placement and distribution of switches between controllers, were also considered in our proposals.

In Chapter 4 we address our second research objective *–how to increase the energy saving and quantify its impact on network performance.* For such purpose, we focused on the use of topology knowledge available at the SDN controller followed by TE decisions for improving the energy efficiency and reducing the solution complexity. This approach, unlike existing related works, was not based on restricting the traffic aggregation according to specific performance thresholds such as link utilization or path latency. In this way, we were able to more deeply examine the implications of our energy-aware routing solution on several network performance metrics.

The closely related third question *–how to jointly optimize QoS requirements and energy efficiency while still meeting control traffic requirements–* was discussed in Chapter 5. Despite being a more challenging scheme, the proposed multi-objective approach enables the reduction of power consumption without performance degradation. Moreover, the proposed energy-aware routing gives a more fine-grained approach since it manages integrated routing considerations for data and control plane traffic in SDN, such as QoS requirements and traffic delay, respectively. By exploiting the use of this multi-objective framework, a flexible and configurable routing decision process can be achieved, suitable to support sophisticated strategies based on the traffic type.

Finally, in Chapter 6 we tackle our last research problem *–how to provide online energy efficient traffic allocation to real-time demands without performance degradation.* In order to overcome the lack of energy-aware proposals providing real-time operation for in-band SDN, we provide a power-aware control plane configuration combined with a dynamic routing strategy for generic SDN architectures (i.e. compatible with multiple controllers and in-band control traffic). Our approach is also able to handle more demanding traffic patterns while avoiding the network congestion and thus reducing the performance degradation of higher priority traffic. It is designed to be suitable for real-time network control and management as well as effective in providing important energy savings together with low blocking rates, leading to an efficient usage of the network's resources.

In general, solutions proposed along this document hold the potential to deliver substantial energy savings while considering important features for their deployment in real-world networks. Extensive simulation experiments, conducted on several real-world network topologies, demonstrate the viability and efficacy of proposed approaches. These findings are of special interest to improve the power efficiency of large backbone networks using the SDN architecture without a dedicated control infrastructure.

7.2 Room for Improvement

Some of the contributions presented in this manuscript have still room for further improvement. For instance, given the lack of comparable approaches in the literature, the heuristic algorithms for energy-aware routing and controller placement proposed in Chapter 3 have been compared against general baselines (i.e. Mod-SPR, k-median and k-center), which are not optimized in terms of energy efficiency. In this regard, future work can be devoted to adapt existing energy-aware solutions in the state of the art to also support the considered scenarios of SDN with in-band control traffic and no data plane traffic through the controllers.

Considering the impact of energy-aware routing, based on the reduction of active network elements, on SDN reliability reported in Chapter 4, further research might well be conducted on the inclusion of restoration mechanisms in order to improve the fault tolerance capacity of our models. In this way, control and data planes failures can be quickly handled while maintaining active the minimum number of links.

Regarding the Multi-Objective Evolutionary Algorithm presented in Chapter 5, another interesting future work would be to obtain exact results for this bi-objective approach. While an exact mathematical formulation was provided to define the considered problem, this ILP model was not used as a benchmark to compare the quality of solutions achieved by our algorithm due to the complexity of the system and the limitations of traditional methods to solve multi-objective optimization problems. However, for some limited cases, optimal bounds could be investigated as a follow-on task. Additionally, the use of another MOEA to implement this model (e.g. Non Sorting Genetic Algorithm 2 (NSGA2)) can be performed as future work in order to compare results obtained from both algorithms.

In general, the most evident limitation of our entire work concerns the lack of practical implementation of the proposed approaches. Implementing the proposed methods as northbound applications running on top of commercial SDN controllers, would allow to test their suitability in a real network environment and empirically validate the power consumption reduction.

7.3 Road Ahead

The adoption of 5G networks, expected by 2020, will allow handling more traffic in dense environments, providing higher data rates and reduced end-to-end latency [136]. In this scenario, achieving energy efficiency becomes even more concerning and challenging. Nevertheless, a strongly related Key Performance Indicator (KPI) to be addressed by the 5G generation is precisely the energy efficiency [137]. Accordingly, an effective energy management, as well as an enhanced network performance, are essential design goals to fulfill the requirements of future 5G systems for heterogeneous applications and services.

Energy consumption in 5G system is currently attracting a great deal of attention from networking researchers and several papers have been proposed with solutions enabling significant energy efficiency gains in the mobile networks division [138–140]. Although 5G is mostly perceived as wireless access by the user, different communication facilities and users need to be connected through backhaul networks –either using optical fiber networks or radio links– and backbone networks. Apart from being key enablers for a successful deployment of this complex architecture, these network segments are of paramount importance to reduce the energy consumption of 5G systems.

SDN is expected to play a major role in 5G systems in order to provide a more intelligent utilization of the underlying transport networks [141]. Moreover, this technology, together with

Network Functions Virtualization (NFV), can provide the required tools to support network slicing in order to accommodate simultaneously the wide range of demanded services over a common infrastructure [142–144].

Within 5G networks, more flexible network control and management strategies will be required in order to provide a higher degree of adaptability for new applications and network services. In that direction, the contributions of this thesis can be exploited as a first step to move towards an energy efficient paradigm in 5G networking using SDN and NFV technologies.

Publications

International Journals

- A. Fernández-Fernández, C. Cervelló-Pastor and L. Ochoa-Aday, "Energy-Aware Routing in Multiple Domains Software-Defined Networks," *Advances in Distributed Computing and Artificial Intelligence Journal (ADCAIJ)*, vol. 5, no. 3, pp. 13–19, Nov. 2016.
- L. Ochoa-Aday, C. Cervelló-Pastor and A. Fernández-Fernández, "Discovering the Network Topology: An Efficient Approach for SDNs," *Advances in Distributed Computing and Artificial Intelligence Journal (ADCAIJ)*, vol. 5, no. 2, pp. 101–108, Nov. 2016.
- A. Fernández-Fernández, C. Cervelló-Pastor and L. Ochoa-Aday, "A Multi-Objective Routing Strategy for QoS and Energy Awareness in Software-Defined Networks," *IEEE Communications Letters*, vol. 21, no. 11, pp. 2416–2419, Nov. 2017.
- A. Fernández-Fernández, C. Cervelló-Pastor and L. Ochoa-Aday, "Energy Efficiency and Network Performance: A Reality Check in SDN-Based 5G Systems," *Energies*, vol. 10, no. 12, pp. 2132:1–2132:27, Dec. 2017.
- L. Ochoa-Aday, C. Cervelló-Pastor and A. Fernández-Fernández, "Self-healing Topology Discovery Protocol for Software Defined Networks," *IEEE Communications Letters*, vol. 22, no. 5, pp. 1070–1073, May 2018.
- L. Ochoa-Aday, C. Cervelló-Pastor, A. Fernández-Fernández and P. Grosso, "An Online Algorithm for Dynamic NFV Placement in Cloud-Based Autonomous Response Networks," *Symmetry*, vol. 10, no. 5, pp. 163:1–163:18, May 2018.

International Conferences

- A. Fernández-Fernández, C. Cervelló-Pastor and L. Ochoa-Aday, "A Distributed Energy-Aware Routing Algorithm in Software-Defined Networks," in *Advances in Intelligent Systems and Computing. Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection*. Springer International Publishing, 2016, vol. 473, pp. 369–373.
- L. Ochoa-Aday, C. Cervelló-Pastor and A. Fernández-Fernández, "A Distributed Algorithm for Topology Discovery in Software-Defined Networks," in *Advances in Intelligent Systems and Computing. Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection*. Springer International Publishing, 2016, vol. 473, pp. 363–367.
- A. Fernández-Fernández, C. Cervelló-Pastor and L. Ochoa-Aday, "Improved Energy-Aware Routing Algorithm in Software-Defined Networks," in *Proc. 41st IEEE Conference on Local Computer Networks (LCN'16)*, Dubai, UAE, Nov. 2016, pp. 196–199.
- A. Fernández-Fernández, C. Cervelló-Pastor and L. Ochoa-Aday, "Achieving Energy Efficiency: An Energy-Aware Approach in SDN," in *Proc. 59th IEEE Conference on Global Communications (GLOBECOM'16)*, Washington DC, USA, Dec. 2016, pp. 1–7.
- A. Fernández-Fernández, C. Cervelló-Pastor, L. Ochoa-Aday and P. Grosso, "An Online Power-Aware Routing in SDN with Congestion-Avoidance Traffic Reallocation," in *Proc. of the 17th IFIP-TC6 International Networking Conference (NETWORKING'18)*, Zurich, Switzerland, May 2018, to be published.

National Conference

- A. Fernández-Fernández, C. Cervelló-Pastor and L. Ochoa-Aday, "Evaluating the Impact of Energy-Aware Routing on Software-Defined Networking Performance," in *Proc. of the XIII Jornadas de Ingeniería Telemática (JITEL'17)*, Valencia, Spain, Sep. 2017, pp. 241–248.

Technical Report

- L. Ochoa-Aday, C. Cervelló-Pastor and A. Fernández-Fernández, "Current Trends of Topology Discovery in OpenFlow-based Software Defined Networks," Scientific Report, Sep. 2015. [Online]. Available: `http://hdl.handle.net/2117/77672`.

References

[1] "Cisco Visual Networking Index: Forecast and Methodology, 2016–2021," White Paper, Cisco, Sep. 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html

[2] W. Van Heddeghem, S. Lambert, B. Lannoo, D. Colle, M. Pickavet, and P. Demeester, "Trends in Worldwide ICT Electricity Consumption from 2007 to 2012," *Computer Communications*, vol. 50, pp. 64–76, 2014.

[3] E. Gelenbe and Y. Caseau, "The Impact of Information Technology on Energy Consumption and Carbon Emissions," *ACM Ubiquity*, pp. 1–15, Jun. 2015.

[4] J. Shuja, R. W. Ahmad, A. Gani, A. I. Abdalla Ahmed, A. Siddiqa, K. Nisar, S. U. Khan, and A. Y. Zomaya, "Greening Emerging IT Technologies: Techniques and Practices," *Journal of Internet Services and Applications*, vol. 8, no. 1, pp. 9:1–9:11, Jul. 2017.

[5] X. Wang, A. V. Vasilakos, M. Chen, Y. Liu, and T. T. Kwon, "A Survey of Green Mobile Networks: Opportunities and Challenges," *Mobile Networks and Applications*, vol. 17, no. 1, pp. 4–20, Feb. 2012.

[6] B. Alena and G. Libor, "Green ICT Adoption Survey Focused on ICT Lifecycle from the Consumers Perspective (SMEs)," *Journal of Competitiveness*, vol. 4, no. 4, pp. 109–122, 2012.

[7] 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*. Cham: Springer International Publishing, 2018, pp. 283–313.

[8] Z. Andreopoulou, E. Stiakakis, and M. Vlachopoulou, "Green ICT Applications Towards the Achievement of Sustainable Development," in *E-Innovation for Sustainable Development of Rural Resources During Global Economic Crisis*. Hershey: IGI Global, 2013.

[9] B. Lannoo, S. Lambert, W. Van Heddeghem, M. Pickavet, F. Kuipers, G. Koutitas, H. Niavis, A. Satsiou, M. Beck, A. Fischer *et al.*, “Overview of ICT Energy Consumption (D8.1),” European Network of Excellence in Internet Science, EINS Consortium Report FP7–2888021, Feb. 2013.

[10] B. Aebischer and L. M. Hilty, “The Energy Demand of ICT: A Historical Perspective and Current Methodological Challenges,” in *ICT Innovations for Sustainability*, ser. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2015, vol. 310, ch. 4, pp. 71–103.

[11] “Digitalization & Energy,” International Energy Agency, Nov. 2017. [Online]. Available: http://www.iea.org/publications/freepublications/publication/digitalization-and-energy.html

[12] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, “A Power Benchmarking Framework for Network Devices,” in *Proc. of the 8th IFIP-TC6 International Networking Conference (NETWORKING’09)*, 2009, pp. 795–808.

[13] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, “Power Awareness in Network Design and Routing,” in *Proc. of the IEEE Conference on Computer Communications (INFOCOM’08)*, Apr. 2008, pp. 457–465.

[14] M. Gupta and S. Singh, “Greening of the Internet,” in *Proc. of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (ACM SIGCOMM’03)*, 2003, pp. 19–26.

[15] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

[16] T. Bakhshi, “State of the Art and Recent Research Advances in Software Defined Networking,” *Wireless Communications and Mobile Computing*, Jan. 2017.

[17] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, “Are We Ready for SDN? Implementation Challenges for Software-Defined Networks,” *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, Jul. 2013.

[18] W. Braun and M. Menth, “Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices,” *Future Internet*, vol. 6, no. 2, pp. 302–336, 2014.

[19] F. Bannour, S. Souihi, and A. Mellouk, “Distributed SDN Control: Survey, Taxonomy and Challenges,” *IEEE Communications Surveys and Tutorials*, vol. 20, no. 1, pp. 333–354, Firstquarter 2018.

[20] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: Towards an Open, Distributed SDN OS," in *Proc. of the Third Workshop on Hot Topics in Software Defined Networking (HotSDN'14)*, 2014, pp. 1–6.

[21] A. Kumar, S. Jain, U. Naik, A. Raghuraman, N. Kasinadhuni, E. C. Zermeno, C. S. Gunn, J. Ai, B. Carlin, M. Amarandei-Stavila, M. Robin, A. Siganporia, S. Stuart, and A. Vahdat, "BwE: Flexible, Hierarchical Bandwidth Allocation for WAN Distributed Computing," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 1–14, Aug. 2015.

[22] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[23] A. Lara, A. Kolasani, and B. Ramamurthy, "Network Innovation using OpenFlow: A Survey," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 493–512, First 2014.

[24] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 3, pp. 1617–1634, Third 2014.

[25] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a Globally-deployed Software Defined WAN," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, Aug. 2013.

[26] D. Avri, H. S. Jamal, H. Robert, K. Hormuzd, W. Weiming, D. Ligang, G. Ram, and H. Joel, "Forwarding and Control Element Separation (ForCES) Protocol Specification," RFC 5810 (proposed standard), 2010.

[27] "OpFlex: An Open Source Approach," White Paper, Cisco, Mar. 2014. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731304.html

[28] L. Schiff, S. Schmid, and P. Kuznetsov, "In-Band Synchronization for Distributed SDN Control Planes," *ACM SIGCOMM Computer Communication Review*, vol. 46, no. 1, pp. 37–43, Jan. 2016.

[29] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "Automatic Bootstrapping of OpenFlow Networks," in *Proc. of the IEEE International Symposium on Local and Metropolitan Area Networks (LAN-MAN'13)*, Apr. 2013, pp. 1–6.

[30] H. Kim and N. Feamster, "Improving Network Management with Software Defined Networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, Feb. 2013.

[31] M. F. Tuysuz, Z. K. Ankarali, and D. Gzpek, “A Survey on Energy Efficiency in Software Defined Networks,” *Computer Networks*, vol. 113, pp. 188–204, Feb. 2017.

[32] A. Fernández-Fernández, C. Cervelló-Pastor, and L. Ochoa-Aday, “A Distributed Energy-Aware Routing Algorithm in Software-Defined Networks,” in *Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection*, ser. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2016, vol. 473, pp. 369–373.

[33] ——, “Energy-Aware Routing in Multiple Domains Software-Defined Networks,” *Advances in Distributed Computing and Artificial Intelligence Journal (ADCAIJ)*, vol. 5, no. 3, pp. 13–19, Nov. 2016.

[34] ——, “Improved Energy-Aware Routing Algorithm in Software-Defined Networks,” in *Proc. of the IEEE 41st Conference on Local Computer Networks (LCN’16)*, Nov. 2016, pp. 196–199.

[35] ——, “Achieving Energy Efficiency: An Energy-Aware Approach in SDN,” in *Proc. of the IEEE Global Communications Conference (GLOBECOM’16)*, Dec. 2016, pp. 1–7.

[36] ——, “Evaluating the Impact of Energy-Aware Routing on Software-Defined Networking Performance,” in *Proc. of the XIII Jornadas de Ingeniería Telemática (JITEL’17)*, Sep. 2017, pp. 241—-248.

[37] ——, “Energy Efficiency and Network Performance: A Reality Check in SDN-Based 5G Systems,” *Energies*, vol. 10, no. 12, Dec. 2017.

[38] ——, “A multi-objective routing strategy for qos and energy awareness in software-defined networks,” *IEEE Communications Letters*, vol. 21, no. 11, pp. 2416–2419, Nov. 2017.

[39] A. Fernández-Fernández, C. Cervelló-Pastor, L. Ochoa-Aday, and P. Grosso, “An Online Power-Aware Routing in SDN with Congestion-Avoidance Traffic Reallocation,” in *Proc. of the 17th IFIP-TC6 International Networking Conference (NETWORKING’18)*, May 2018.

[40] F. Idzikowski, L. Chiaraviglio, A. Cianfrani, J. L. Vizcaíno, M. Polverini, and Y. Ye, “A Survey on Energy-Aware Design and Operation of Core Networks,” *IEEE Communications Surveys and Tutorials*, vol. 18, no. 2, pp. 1453–1499, Secondquarter 2016.

[41] A.-C. Orgerie, M. D. d. Assuncao, and L. Lefevre, “A Survey on Techniques for Improving the Energy Efficiency of Large-scale Distributed Systems,” *ACM Computing Surveys*, vol. 46, no. 4, pp. 47:1–47:31, Mar. 2014.

[42] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, “Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures,” *IEEE Communications Surveys and Tutorials*, vol. 13, no. 2, pp. 223–244, 2011.

[43] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, “Reducing Network Energy Consumption via Sleeping and Rate-Adaptation,” in *Proc. of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI’08)*, 2008, pp. 323–336.

[44] F. Dabaghi, Z. Movahedi, and R. Langar, “A Survey on Green Routing Protocols Using Sleep-scheduling in Wired Networks,” *Journal of Network and Computer Applications*, vol. 77, pp. 106–122, Jan. 2017.

[45] K. Bilal, S. U. Khan, S. A. Madani, K. Hayat, M. I. Khan, N. Min-Allah, J. Kolodziej, L. Wang, S. Zeadally, and D. Chen, “A Survey on Green Communications Using Adaptive Link Rate,” *Cluster Computing*, vol. 16, no. 3, pp. 575–589, 2013.

[46] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, “ElasticTree: Saving Energy in Data Center Networks,” in *Proc. of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI’10)*, 2010.

[47] L. Chiaraviglio, D. Ciullo, M. Mellia, and M. Meo, “Modeling Sleep Mode Gains in Energy-Aware Networks,” *Computer Networks*, vol. 57, no. 15, pp. 3051–3066, 2013.

[48] R. Hays, “Active/Idle Toggling with 0BASE-x for Energy Efficient Ethernet,” presentation to the IEEE 802.3az Task Force, Nov. 2007. [Online]. Available: http://www.ieee802.org/3/az/public/nov07/hays_1_1107.pdf

[49] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy, “Minimizing Routing Energy Consumption: from Theoretical to Practical Results,” INRIA, Research Report RR-7234, Jun. 2010.

[50] L. Wang, A. Fernández Anta, F. Zhang, C. Hou, and Z. Liu, “Energy-Efficient Network Routing with Discrete Cost Functions,” in *Theory and Applications of Models of Computation.* Berlin, Heidelberg: Springer, 2012, pp. 307–318.

[51] Q. Li, M. Xu, Y. Yang, L. Gao, Y. Cui, and J. Wu, “Safe and Practical Energy-Efficient Detour Routing in IP Networks,” *IEEE/ACM Transactions on Networking*, vol. 22, no. 6, pp. 1925–1937, Dec. 2014.

[52] C.-L. Li, S. T. McCormick, and D. Simchi-Levi, “The Point-to-Point Delivery and Connection Problems: Complexity and Algorithms,” *Discrete Applied Mathematics*, vol. 36, no. 3, pp. 267–292, 1992.

[53] M. Zhang, C. Yi, B. Liu, and B. Zhang, “GreenTE: Power-Aware Traffic Engineering,” in *Proc. of the 18th IEEE International Conference on Network Protocols (ICNP’10)*, Oct. 2010, pp. 21–30.

[54] A. P. Bianzino, C. Chaudet, F. Larroca, D. Rossi, and J. L. Rougier, “Energy-Aware Routing: A Reality Check,” in *Proc. of the IEEE Global Communications Conference (GLOBECOM’10)*, Dec. 2010, pp. 1422–1427.

[55] L. Chiaraviglio, M. Mellia, and F. Neri, “Minimizing ISP Network Energy Cost: Formulation and Solutions,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 463–476, 2012.

[56] I. Ghamlouche, T. G. Crainic, and M. Gendreau, “Cycle-Based Neighbourhoods for Fixed-Charge Capacitated Multicommodity Network Design,” *Operations Research*, vol. 51, no. 4, pp. 655–667, Jul. 2003.

[57] B. G. Assefa and O. Ozkasap, “State-of-the-art Energy Efficiency Approaches in Software Defined Networking,” in *Proc. of the International Symposium on Advances in Software Defined Networks (SoftNetworking’15)*, Apr. 2015.

[58] H. Wang, Y. Li, D. Jin, P. Hui, and J. Wu, “Saving Energy in Partially Deployed Software Defined Networks,” *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1578–1592, 2016.

[59] J. Galán-Jiménez, “Exploiting the Control Power of SDN During the Transition from IP to SDN Networks,” *International Journal of Communication Systems*, 2018.

[60] F. Giroire, J. Moulierac, and T. K. Phan, “Optimizing Rule Placement in Software-Defined Networks for Energy-Aware Routing,” in *Proc. of the IEEE Global Communications Conference (GLOBECOM’14)*, Dec. 2014, pp. 2523–2529.

[61] A. Markiewicz, P. N. Tran, and A. Timm-Giel, “Energy Consumption Optimization for Software Defined Networks Considering Dynamic Traffic,” in *Proc. of the IEEE 3rd International Conference on Cloud Networking (CloudNet’14)*, Oct. 2014, pp. 155–160.

[62] B. G. Assefa and O. Ozkasap, “Link Utility and Traffic Aware Energy Saving in Software Defined Networks,” in *Proc. of the IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom’17)*, Jun. 2017, pp. 1–5.

[63] R. Wang, Z. Jiang, S. Gao, W. Yang, Y. Xia, and M. Zhu, “Energy-Aware Routing Algorithms in Software-Defined Networks,” in *Proc. of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM’14)*, Jun. 2014, pp. 1–6.

[64] M. R. Celenlioglu, S. B. Goger, and H. A. Mantar, “An SDN-based Energy-Aware Routing Model for Intra-domain Networks,” in *Proc. of the 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM’14)*, Sep. 2014, pp. 61–66.

[65] A. Ruiz-Rivera, K. W. Chin, and S. Soh, "GreCo: An Energy Aware Controller Association Algorithm for Software Defined Networks," *IEEE Communications Letters*, vol. 19, no. 4, pp. 541–544, 2015.

[66] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini, in *Proc. of the IEEE Conference on Computer Communications (INFOCOM'10)*.

[67] A. Cianfrani, V. Eramo, M. Listanti, and M. Polverini, "An OSPF enhancement for energy saving in IP networks," in *Proc. of the IEEE Conference on Computer Communications Workshops (INFOCOM'11 WKSHPS)*, Apr. 2011, pp. 325–330.

[68] F. Cuomo, A. Abbagnale, A. Cianfrani, and M. Polverini, "Keeping the Connectivity and Saving the Energy in the Internet," in *Proc. of the IEEE Conference on Computer Communications Workshops (INFOCOM'11 WKSHPS)*, Apr. 2011, pp. 319–324.

[69] M. Fiedler, "Algebraic Connectivity of Graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[70] F. Cuomo, A. Cianfrani, M. Polverini, and D. Mangione, "Network Pruning for Energy Saving in the Internet," *Computer Networks*, vol. 56, no. 10, pp. 2355–2367, Jul. 2012.

[71] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, Mar. 1977.

[72] F. Cuomo, A. Abbagnale, and S. Papagna, "ESOL: Energy Saving in the Internet Based on Occurrence of Links in Routing Paths," in *Proc. of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'14)*, Jun. 2011, pp. 1–6.

[73] A. Cianfrani, V. Eramo, M. Listanti, M. Polverini, and A. V. Vasilakos, "An OSPF-Integrated Routing Strategy for QoS-Aware Energy Saving in IP Backbone Networks," *IEEE Transactions on Network and Service Management*, vol. 9, no. 3, pp. 254–267, Sep. 2012.

[74] H. Matsuura, "Energy-Saving Routing Algorithm Using Steiner Tree," in *Proc. of the IFIP/IEEE International Symposium on Integrated Network Management (IM'13)*, May 2013, pp. 378–386.

[75] ——, "Proposal of New Steiner Tree Algorithm Applied to P2MP Traffic Engineering," in *Proc. of the IEEE Global Communications Conference (GLOBECOM'12)*, Dec. 2012, pp. 2720–2726.

[76] M. Hussein, G. Jakllari, and B. Paillassa, "Frugal Topologies for Saving Energy in IP Networks," in *Proc. of the IEEE 40th Conference on Local Computer Networks (LCN'15)*, Oct. 2015, pp. 303–311.

[77] M. Shen, H. Liu, K. Xu, N. Wang, and Y. Zhong, "Routing On Demand: Toward the Energy-Aware Traffic Engineering with OSPF," in *Proc. of the 11th IFIP-TC6 International Networking Conference (NETWORKING'12)*, May 2012, pp. 232–246.

[78] G. Lin, S. Soh, K.-W. Chin, and M. Lazarescu, "Power-Aware Routing in Networks with Quality of Services Constraints," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 1, pp. 122–135, 2016.

[79] R. G. Garroppo, S. Giordano, G. Nencioni, and M. G. Scutellà, "Power-Aware Routing and Network Design with Bundled Links: Solutions and Analysis," *Journal of Computer Networks and Communications*, 2013.

[80] B. Addis, A. Capone, G. Carello, L. G. Gianoli, and B. Sansò, "Energy Management Through Optimized Routing and Device Powering for Greener Communication Networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 313–325, Feb. 2014.

[81] H. Alizadeh, D. Gozupek, S. M. Buhari, and A. Yayimli, "Minimizing Signaling Cost in Green Routing for Software Defined Networks," in *Proc. of the IEEE Symposium on Computers and Communication (ISCC'15)*, Jul. 2015, pp. 391–396.

[82] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy, "Minimizing Routing Energy Consumption: From Theoretical to Practical Results," in *Proc. of the IEEE/ACM International Conference on Green Computing and Communications (GreenCom'10)*, Dec. 2010, pp. 252–259.

[83] G. Lin, S. Soh, M. Lazarescu, and K. W. Chin, "On the Effects of Energy-Aware Traffic Engineering on Routing Reliability," in *Proc. of the 19th Asia-Pacific Conference on Communications (APCC'13)*, Aug. 2013, pp. 69–74.

[84] G. Lin, S. Soh, and K.-W. Chin, "Energy-Aware Traffic Engineering With Reliability Constraint," *Computer Communications*, vol. 57, pp. 115–128, 2015.

[85] B. Addis, A. Capone, G. Carello, L. Gianoli, and B. Sansò, "On the Energy Cost of Robustness and Resiliency in IP Networks," *Computer Networks*, vol. 75, pp. 239–259, 2014.

[86] M. K. Awad, G. Neama, and Y. Rafique, "The Impact of Practical Network Constraints on the Performance of Energy-Aware Routing Schemes," in *Proc. of the IEEE International Conference on Service Operations And Logistics, And Informatics (SOLI'15)*, Nov. 2015, pp. 77–81.

[87] J. Crichigno, W. Shu, and M. Y. Wu, "Throughput Optimization and Traffic Engineering in WDM Networks Considering Multiple Metrics," in *Proc. of the IEEE International Conference on Communications (ICC'10)*, May 2010, pp. 1–6.

[88] J. Crichigno, N. Ghani, J. Khoury, W. Shu, and M. Y. Wu, "Dynamic Routing Optimization in WDM Networks," in *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM'10)*, Dec. 2010, pp. 1–5.

[89] J. Wang, Y. Yan, and L. Dittmann, "Design of Energy Efficient Optical Networks with Software Enabled Integrated Control Plane," *IET Networks*, vol. 4, pp. 30–36, Jan. 2015.

[90] E. Amaldi, A. Capone, and L. Gianoli, "Energy-aware IP Traffic Engineering with Shortest Path Routing," *Computer Networks*, vol. 57, no. 6, pp. 1503–1517, 2013.

[91] B. Fortz and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," in *Proc. of the IEEE Conference on Computer Communications (INFOCOM'00)*, vol. 2, 2000, pp. 519–528.

[92] F. A. Moghaddam and P. Grosso, "Linear Programming Approaches for Power Savings in Software-Defined Networks," in *Proc. of the IEEE International Conference on Network Softwarization (NetSoft'16)*, Jun. 2016, pp. 83–87.

[93] M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable, Commodity Data Center Network Architecture," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 63–74, Aug. 2008.

[94] J. Zhao, Q. Yao, D. Ren, W. Li, and W. Zhao, "A Multi-Domain Control Scheme for Diffserv QoS and Energy Saving Consideration in Software-Defined Flexible Optical Networks," *Optics Communications*, vol. 341, pp. 178–187, 2015.

[95] S. B. H. Said and A. Petrescu, "Energy-Efficient Routing in SDN-Based Access Networks," in *Greening Video Distribution Networks*. Cham: Springer International Publishing, 2018, pp. 159–175.

[96] C. A. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York, NY, USA: Springer, 2007, vol. 5.

[97] D. Simon, *Evolutionary Optimization Algorithms*. Hoboken, New Jersey: John Wiley & Sons, 2013.

[98] L. Lingxia, V. C. Leung, and L. Chin-Feng, "Evolutionary Algorithms in Software Defined Networks: Techniques, Applications, and Issues," *ZTE Communications*, vol. 15, no. 3, pp. 20–36, Aug. 2017.

[99] J. Wang, X. Chen, C. Phillips, and Y. Yan, "Energy Efficiency with QoS Control in Dynamic Optical Networks with SDN Enabled Integrated Control Plane," *Computer Networks*, vol. 78, pp. 57–67, Feb. 2015.

[100] R. Zhu, H. Wang, Y. Gao, S. Yi, and F. Zhu, "Energy Saving and Load Balancing for SDN Based on Multi-objective Particle Swarm Optimization," in *Algorithms and Architectures for Parallel Processing*. Cham: Springer International Publishing, 2015, pp. 176–189.

[101] B. Özbek, Y. Aydoğmuş, A. Ulaş, B. Gorkemli, and K. Ulusoy, "Energy Aware Routing and Traffic Management for Software Defined Networks," in *Proc. of the IEEE International Conference on Network Softwarization (NetSoft'16)*, Jun. 2016, pp. 73–77.

[102] S. Oda, D. Nobayashi, Y. Fukuda, and T. Ikenaga, "Flow-based Routing Schemes for Minimizing Network Energy Consumption using OpenFlow," in *Proc. of the Fourth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies (ENERGY'14)*, Apr. 2014, pp. 69–72.

[103] X. Zhang and H. Wang, "Centralized Energy-Efficient Routing Control for Greener Internet," in *Proc. of the 8th International Conference on Communications and Networking in China (CHINA-COM'13)*, Aug. 2013, pp. 594–598.

[104] S. S. Tadesse, C. Casetti, C. F. Chiasserini, and G. Landi, "Energy-Efficient Traffic Allocation in SDN-basec Backhaul Networks: Theory and Implementation," in *Proc. of the IEEE Conference on Consumer Communications & Networking (CCNC'17)*, Jan. 2017, pp. 209–215.

[105] O. I. Abdullaziz, M. Capitani, C. E. Casetti, C. F. Chiasserini, S. B. Chundrigar, G. Landi, X. Li, F. Moscatelli, K. Sakaguchi, and S. T. Talat, "Energy Monitoring and Management in 5G Integrated Fronthaul and Backhaul," in *Proc. of the European Conference on Networks and Communications (EuCNC'17)*, Jun. 2017, pp. 1–6.

[106] A. Amokrane, R. Langar, R. Boutaba, and G. Pujolle, "Flow-Based Management For Energy Efficient Campus Networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 4, pp. 565–579, Dec. 2015.

[107] Y. Yang, M. Xu, and Q. Li, "Towards Fast Rerouting-based Energy Efficient Routing," *Computer Networks*, vol. 70, pp. 1–15, 2014.

[108] H. Zhu, X. Liao, C. de Laat, and P. Grosso, "Joint Flow Routing-Scheduling for Energy Efficient Software Defined Data Center Networks: A Prototype of Energy-Aware Network Management Platform," *Journal of Network and Computer Applications*, vol. 63, pp. 110–124, 2016.

[109] B. Heller, R. Sherwood, and N. McKeown, "The Controller Placement Problem," in *Proc. of the First Workshop on Hot Topics in Software Defined Networking (HotSDN'12)*, 2012, pp. 7–12.

[110] Gurobi Optimizer (Version 7.5). Gurobi Optimization. [Online]. Available: http://www.gurobi.com/

[111] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0-Survivable Network Design Library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.

[112] J. Li, J.-H. Yoo, and J. W.-K. Hong, “Dynamic Control Plane Management for Software-Defined Networks,” *International Journal of Network Management*, vol. 26, no. 2, pp. 111–130, 2016.

[113] Zhang, Y. (2004) Abilene TM. [Online]. Available: http://www.cs.utexas.edu/~yzhang/research/AbileneTM/

[114] R. Bolla, R. Bruschi, A. Carrega, and F. Davoli, “Green Network Technologies and the Art of Trading-Off,” in *Proc. of the IEEE Conference on Computer Communications Workshops (INFOCOM’11 WKSHPS)*, Apr. 2011, pp. 301–306.

[115] E. Nuutila and E. Soisalon-soininen, “On Finding the Strongly Connected Components in a Directed Graph,” *Information Processing Letters*, vol. 49, 1994.

[116] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, “A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions,” *ArXiv e-prints*, Aug. 2017.

[117] C. Li, H. Wang, and P. Van Mieghem, “Degree and Principal Eigenvectors in Complex Networks,” in *Proc. of the 11th IFIP-TC6 International Networking Conference (NETWORKING’12).*

[118] P. Mahadevan, D. Krioukov, M. Fomenkov, X. Dimitropoulos, k. c. Claffy, and A. Vahdat, “The Internet AS-level Topology: Three Data Sources and One Definitive Metric,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 17–26, Jan. 2006.

[119] M. Brede and B. J. de Vries, “Networks that Optimize a Trade-off Between Efficiency and Dynamical Resilience,” *Physics Letters A*, vol. 373, no. 43, pp. 3910–3914, 2009.

[120] G. Chiandussi, M. Codegone, S. Ferrero, and F. E. Varesio, “Comparison of Multi-objective Optimization Methodologies for Engineering Applications,” *Computers & Mathematics with Applications*, vol. 63, no. 5, pp. 912–942, Mar. 2012.

[121] H. P. Schwefel, “Advantages (and Disadvantages) of Evolutionary Computation over Other Approaches,” in *Evolutionary Computation 1: Basic Algorithms and Operators.* Bristol, U.K.: Institute of Physics Publishing, 2000, ch. 3, pp. 20–22.

[122] R. Bouaziz, L. Lemarchand, F. Singhoff, B. Zalila, and M. Jmaiel, “Efficient Parallel Multi-Objective Otimization for Real-Time Systems Software Design Exploration,” in *Proc. of the International Symposium on Rapid System Prototyping (RSP’16)*, Oct. 2016, pp. 1–7.

[123] K. Deb, “Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction,” in *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing.* London: Springer, 2011, pp. 3–34.

[124] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization," in *Proc. of the International Conference on Evolutionary Methods for Design Optimisation and Control with Application to Industrial Problems*, 2002, pp. 95–100.

[125] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[126] F. Zhao, W. Lei, W. Ma, Y. Liu, and C. Zhang, "An Improved SPEA2 Algorithm with Adaptive Selection of Evolutionary Operators Scheme for Multiobjective Optimization Problems," *Mathematical Problems in Engineering*, pp. 67–74, 2016.

[127] H. H. Maheta and V. K. Dabhi, "An Improved SPEA2 Multi Objective Algorithm with Non Dominated Elitism and Generational Crossover," in *Proc. of the International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT'14)*, Feb. 2014, pp. 75–82.

[128] W. Sheng, Y. Liu, X. Meng, and T. Zhang, "An Improved Strength Pareto Evolutionary Algorithm 2 with Application to the Optimization of Distributed Generations," *Computers and Mathematics with Applications*, vol. 64, no. 5, pp. 944–955, 2012.

[129] Z. Li, Y. Li, and D. Xianzhong, "Improved Strength Pareto Evolutionary Algorithm with Local Search Strategies for Optimal Reactive Power Flow," *Information Technology Journal*, vol. 9, no. 4, pp. 749–757, 2010.

[130] E. Chołodowicz and P. Orłowski, "Comparison of SPEA2 and NSGA-II Applied to Automatic Inventory Control System Using Hypervolume Indicator," *Studies in Informatics and Control*, vol. 26, no. 1, pp. 67–74, 2017.

[131] R. Gharari, N. Poursalehi, M. Abbasi, and M. Aghaie, "Implementation of Strength Pareto Evolutionary Algorithm II in the Multiobjective Burnable Poison Placement Optimization of KWU Pressurized Water Reactor," *Nuclear Engineering and Technology*, vol. 48, no. 5, pp. 1126–1139, 2016.

[132] H. Jiang, J. Chen, and T. Liu, "Multi-Objective Design of an FBG Sensor Network Using an Improved Strength Pareto Evolutionary Algorithm," *Sensors and Actuators: A. Physical*, vol. 220, pp. 230–236, 2014.

[133] B. Gadhvi, V. Savsani, and V. Patel, "Multi-Objective Optimization of Vehicle Passive Suspension System Using NSGA-II, SPEA2 and PESA-II," *Procedia Technology*, vol. 23, pp. 361–368, 2016.

[134] E. Zitzler, M. Laumanns, and S. Bleuler, "A Tutorial on Evolutionary Multiobjective Optimization," in *Metaheuristics for Multiobjective Optimisation.* Berlin, Heidelberg: Springer, 2004, pp. 3–37.

[135] M. Kim, T. Hiroyasu, M. Miki, and S. Watanabe, "SPEA2+: Improving the Performance of the Strength Pareto Evolutionary Algorithm 2," in *Proc. of the 8th International Conference on Parallel Problem Solving from Nature (PPSN'04)*, 2004, pp. 742–751.

[136] M. Agiwal, A. Roy, and N. Saxena, "Next Generation 5G Wireless Networks: A Comprehensive Survey," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 3, pp. 1617–1655, thirdquarter 2016.

[137] "Cognitive Network Management for 5G," White Paper, 5GPPP Network Management & Quality of Service Working Group, Mar. 2017. [Online]. Available: https://5g-ppp.eu/white-papers/

[138] A. Bousia, E. Kartsakli, A. Antonopoulos, L. Alonso, and C. Verikoukis, "Multiobjective Auction-Based Switching-Off Scheme in Heterogeneous Networks: To Bid or Not to Bid?" *IEEE Transactions on Vehicular Technology*, vol. 65, no. 11, pp. 9168–9180, Nov. 2016.

[139] M. Oikonomakou, A. Antonopoulos, L. Alonso, and C. Verikoukis, "Evaluating Cost Allocation Imposed by Cooperative Switching Off in Multi-Operator Shared HetNets," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11 352–11 365, Dec 2017.

[140] S. S. Soliman and B. Song, "Fifth Generation (5G) Cellular and the Network for Tomorrow: Cognitive and Cooperative Approach for Energy Savings," *Journal of Network and Computer Applications*, vol. 85, pp. 84–93, 2017.

[141] S. Sun, L. Gong, B. Rong, and K. Lu, "An Intelligent SDN Framework for 5G Heterogeneous Networks," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 142–147, Nov. 2015.

[142] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, May 2017.

[143] M. Vincenzi, A. Antonopoulos, E. Kartsakli, J. Vardakas, L. Alonso, and C. Verikoukis, "Multi-Tenant Slicing for Spectrum Management on the Road to 5G," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 118–125, Oct. 2017.

[144] A. Antonopoulos, E. Kartsakli, C. Perillo, and C. Verikoukis, "Shedding Light on the Internet: Stakeholders and Network Neutrality," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 216–223, 2017.