Thèse

Présentée pour obtenir le grade de docteur
de Télécom ParisTech
Spécialité: Informatique et Réseaux

Aruna Prem BIANZINO

Energy Aware Traffic Engineering in Wired Communication Networks

Soutenue le 4 mai 2012 devant le jury composé de:

Rapporteurs	Raffaele BOLLA	Universitá di Genova, Italie
	Ken CHRISTENSEN	University of South Florida, USA
Examinateurs	Maurice GAGNAIRE	Télécom ParisTech, France
	Isabelle GUÉRIN LASSOUS	Université Lyon I
Directeurs de thèse	Jean-Louis ROUGIER	Télécom ParisTech, France
	Marco AJMONE MARSAN	Politecnico di Torino, Italie

POLITECNICO DI TORINO
Dipartimento di Elettronica
DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE

ENERGY AWARE TRAFFIC ENGINEERING IN WIRED COMMUNICATION NETWORKS

Doctoral Dissertation of:
Aruna Prem Bianzino

Advisors:
Prof. Jean-Louis Rougier
Prof. Marco Ajmone Marsan

2012 - XXIV

TELECOM PARISTECH
Département d'Informatique et Réseaux
Doctorat en Informatique et Réseaux

POLITECNICO DI TORINO
Dipartimento di Elettronica
Dottorato in Ingegneria dell'Informazione

ENERGY AWARE TRAFFIC ENGINEERING IN WIRED COMMUNICATION NETWORKS

Author: Aruna Prem BIANZINO

Defended the 4 May 2012 in front of the committee composed of:

Referees: Prof. Raffaele BOLLA (Universitá di Genova, Italy)
Prof. Ken CHRISTENSEN (University of South Florida, USA)

Examiners: Prof. Maurice GAGNAIRE (Télécom ParisTech, France)
Prof. Isabelle GUÉRIN LASSOUS (Université Lyon I)

Advisors: Prof. Marco AJMONE MARSAN (Politecnico di Torino, Italy)
Prof. Jean-Louis ROUGIER (Télécom ParisTech, France)

To my father, who would have been silently proud.

Contents

List of Figures

List of Tables

Abbreviations

ALR: Adaptive Link Rate
ARP: Address Resolution Protocol
CPU: Central Processing Unit
DHCP: Dynamic Host Configuration Protocol
DLF: Distributed Least Flow
DMP: Distributed Most Power
DVS: Dynamic Voltage Scaling
DWDM: Dense Wavelength Division Multiplexing
GHG: Green House Gas
ICMP: Internet Control Message Protocol
ICT: Information and Communication Technologies
IEEE: Institute of Electrical and Electronics Engineers
IGP: Interior Gateway Protocol
IGP-WO: IGP Weight Opimisation
IP: Internet Protocol
IS-IS: Intermediate System To Intermediate System
ISP: Internet Service Provider
LAN: Local Area Network
LF: Least-Flow
LSA: Link States Advertisement
MP: Most-Power
MPLS: Multi Protocol Label Switching
NIC: Network Interface Card
OEO: Optical-Electronic-Optical
OSPF: Open Shortest Path First
QoS: Quality of Service
TCP: Transmission Control Protocol
TM: Traffic Matrix
TU-Game: Transferable Utility Game

Chapter 1

Introduction

The reduction of power consumption in communication networks has become a key issue for both the Internet Service Providers (ISP) and the research community. According to different studies, the power consumption of Information and Communication Technologies (ICT) varies from 2% to 10% of the worldwide power consumption [1, 2]. Moreover, the expected trends for the future predict a notably increase of the ICT power consumption, doubling its value by 2020 [1] and growing to around 30% of the worldwide electricity demand by 2030 according to business-as-usual evaluation scenarios [3]. It is therefore not surprising that researchers, manufacturers and network providers are spending significant efforts to reduce the power consumption of ICT systems from different angles.

To this extent, networking devices waste a considerable amount of power. In particular, their power consumption has always been increased in the last years, coupled with the increase of the offered performance [4]. Actually, power consumption of networking devices scales with the installed capacity, rather than the current load [5]. Thus, for an ISP the network power consumption is practically constant, unrespectively to traffic fluctuations. However, actual traffic is subject to strong day/night oscillations [6]. Thus, many devices are underutilized, especially during off-peak hours when traffic is low. This represents a clear opportunity for saving energy, since many resources (i.e., routers and links) are powered on without being fully utilized.

In this context, *resource consolidation* is a known paradigm for the reduction of the power consumption. It consists in having a carefully selected subset of network devices entering a low power state, and use the rest to transport the required amount of traffic. This is possible without disrupting the Quality of Service (QoS) offered by the network infrastructure, since communication networks are designed over the *peak foreseen traffic request*, and with *redundancy* and *over-provisioning* in mind.

In this thesis work, we present different techniques to perform *resource consolidation* in backbone IP-based networks, ranging from *centralized solutions*, where a central entity computes a global solution based on an omniscient vision of the network, to *distributed solutions*, where single nodes take independent decisions on the local power-state, based solely on local knowledge. Moreover, different technological assumptions are made, to account for different possible directions of the network devices evolutions, ranging from

the possibility to switch off linecard ports, to whole network nodes, and taking into account different power consumption profiles.

Structure of the dissertation

In the first part of the dissertation, Chapter 2, we analyse the state of the art of the *green neworking* research, and introduce the *benchmark issue*, while in the second part, Chapter 3-5, we concentrate on the proposed technical solutions. The third part, Appendix A contains additional complementary information.

Chapter 2 is an introductory chapter that analyses the current state in green networking research. The main green networking paradigms are here introduced, and for each we analyse the principal solutions, together with the research directions.

Overviewing the current green networking research, we highlighted a lack of common evaluation scenarios and metrics for the analysis of energy saving solutions, as well as of reliable energy consumption figures and common measuring methodologies. This issue is also discussed in **Chapter 2**.

In **Chapter 3** we discuss the formulation of the resource consolidation as an optimization problem, and present its solution results for real network scenarios, considering different power models, and analysing the resulting tradeoff between achievable energy saving and offered QoS.

In **Chapter 4** we analyse in more details the existing and proposed solutions to apply resource consolidation in wired networks in a centralised manner. All the solution discussed in this chapter suppose the presence of a central entity, which has a global view of the instantaneous network status, can compute the best network configuration, and can coordinate network devices to enter such configuration.

As a further step, we present in **Chapter 5** distributed solutions for resource consolidation in wired networks. Different tradeoffs are considered between the amount of information exchanged by network devices, and the solution efficiency, in terms of achievable energy savings, and resulting QoS.

Chapter 6 concludes the dissertation and contains suggestions for further work.

Finally, **Appendix A** lists the author publications.

Chapter 2

A Picture of the Green Networking Research

The reduction of energy consumption has become a key issue for industries, because of economical, environmental and marketing reasons. If this concern has a strong influence on electronics designers, the information and communication technology sector, and more specifically the networking field, is also concerned. For instance, data-centers and networking infrastructure involve high-performance and high-availability machines. They therefore rely on powerful devices, which require energy-consuming air conditioning to sustain their operation, and which are organized in a redundant architecture. As these architectures are often designed to endure peak load and degraded conditions, they are under-utilized in normal operation, leaving a large room for energy savings. In recent years, valuable efforts have indeed been dedicated to reducing unnecessary energy expenditure, which is usually nicknamed as a *greening* of the networking technologies and protocols.

In this thesis work, we focus on protocols and performance rather than on actual transport technologies: as such, we invite the reader to [7] for an overview of energy efficiency in optical networks. Similarly, as energy-related studies in wireless networks are very specific, they would require a dedicated study. This thesis work therefore focuses on *wired* networks. In wired networks, energy saving often requires a reduction in network performance or redundancy. Considering this compromise between the network performance and energy savings, determining efficient strategies to limit the network energy consumption is a real challenge. However, although the green networking field is still in its infancy, a number of interesting works have already been carried out, which are overviewed in the current chapter.

In order to evaluate the potential impact of a "green" solution on the ICT energy consumption, it is necessary to understand how the different devices and network segments contribute to the total expenditure. The Internet, for instance, can be segmented into a core network and several types of access networks. In these different segments, the equipment involved, its objectives and its expected performance and energy consumption levels differ. As such, one may reasonably expect that both the consumption figures and the possible enhancements are considerably different. In 2002, Roth et al. analyzed the energy consumption contributions of different categories of equipment in the global

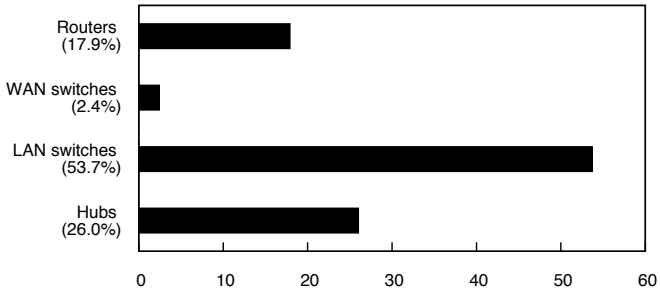

Figure 2.1: Contribution of different device types to the network energy consumption in year 2002 [8]

Internet [8]. These figures, represented in Figure 2.1, indicated that local area networks, through hubs and switches, were responsible for about 80% of the total Internet consumption at that time. In 2005, the authors of [9] estimated the relative contribution of the Network Interface Cards (NICs) and all the other network elements and concluded that the NICs were responsible for almost half of the total power consumption. More recently, studies have started reporting an increase of the consumption in the network core: for instance, in 2009 Deutsche Telekom [10] forecasted that by year 2017, the power consumption of the network core will be equal to that of the network access (the study also suggested, for the coming decade, a 12-fold increase in the power consumption of the network core, mainly due to the IP/MPLS layers). Yet, as another recent study [11] suggests that the core network consumption will instead play a minor role with respect to the other network segments, this issue needs further investigation.

Not surprisingly, everything evolves rapidly in the ICT domain, which makes the aforementioned figures and estimations quickly out-dated and possibly inaccurate. As a consequence, there is a true need for a permanent evaluation of this consumption, in order to point out and update regularly the most relevant targets for potential energy-savings. However, such an evaluation requires a collaboration of equipment manufacturers, ISPs and governments, which is clearly not an easy process. We will come back on this issue in Section 2.3.

2.1 The Green Networking Goals

The "Green Networking" may be seen in many different ways, depending on the point of view from which it is observed. From a strict **environmental point of view**, for instance, the objective of green networking is the minimization of the GHG emissions. An obvious first step in this direction is to enforce as much as possible the use of renewable energy in ICT. Yet another natural track is to design low power components, able to offer the same level of performance. However, these are not the only leads: redesigning the network architecture itself, for instance by geographical delocating network equipment towards strategic places, may yield substantial savings too for two main reasons. The first reason is related to the losses that appear when energy is transported: the closer the consumption points are to the production points, the lower this loss will be[1]. The

[1]Long-distance electricity transportation uses high-voltage lines to reduce losses: therefore, the energy losses are not directly proportional to the distance, but may rather be represented by a threshold-based

second reason is related to the cooling of electronic devices: air-cooling represents an important share of the energy expenditure in data centers and cold climates may lessen this dependency.

Geographical delocalization is also a promising approach from an **economical point of view**. The global energy market offers volatile and time-varying prices. The prices may even become negative when a production surplus appears but there is no customer demand. Energy cannot be stored efficiently, and even though consumption predictions based on historical data quite accurately trigger production units, over-production is always possible. This variability can be exploited by displacing the computation where energy has a lower cost. Going one step further, if the physical machines can be delocalized to minimize the global energy consumption, one may imagine that services too may be located at the optimal places, and that they may move when conditions vary, introducing the time dimension. Computation-intensive operations may be executed on one hemisphere or on the other so that processor (CPU) cycles follow, e.g., seasonal or day/night patterns. A huge technological challenge lies in performing such service migration without any service disruption, preserving fault-tolerance and data security.

The previous optimizations are directly function of the energy price and are not directly related to environmental considerations. The market-related issues behind this problem may lead to an optimal solution in terms of cost that is sub-optimal in terms of total energy consumption. Indeed, 100 MWh sold at a unit price of $ 120 are more expensive than 120 MWh sold at a unit price of $ 90. Thus, environmental considerations generally need a **regulatory point of view** to assist their enforcement. Regulation, which often falls into governmental duties, may push towards greening of the technology by different means (e.g., taxes on GHG emissions, diverting research funds towards energy efficiency, etc.).

Finally, from an **engineering point of view**, green networking may be better seen as a way to reduce energy required to carry out a given task while maintaining the same level of performance, which is the point of view that we will adopt in the rest of this study. Nevertheless, this point of view alone is still relevant as system efficiency from the engineering perspective still deeply relates to economical, regulatory and environmental viewpoints.

2.2 Green Strategies

Traditionally, network devices and systems are designed and dimensioned according to principles that are inherently in opposition with green networking objectives: namely, *over-provisioning* and *redundancy*. On the one hand, due to the lack of Quality of Service (QoS) support from the Internet architecture, over-provisioning is a common practice: networks are dimensioned to sustain peak hour traffic, with extra capacity to handle unexpected events. As a result, during low traffic periods, over-provisioned networks are also over-energy-consuming. Moreover, for resiliency and fault-tolerance, networks are also designed in a redundant manner. Devices are added to the infrastructure with the sole purpose of taking over the duty when another device fails, which further adds to the overall energy consumption. These objectives, radically opposed to the environmental

linear function.

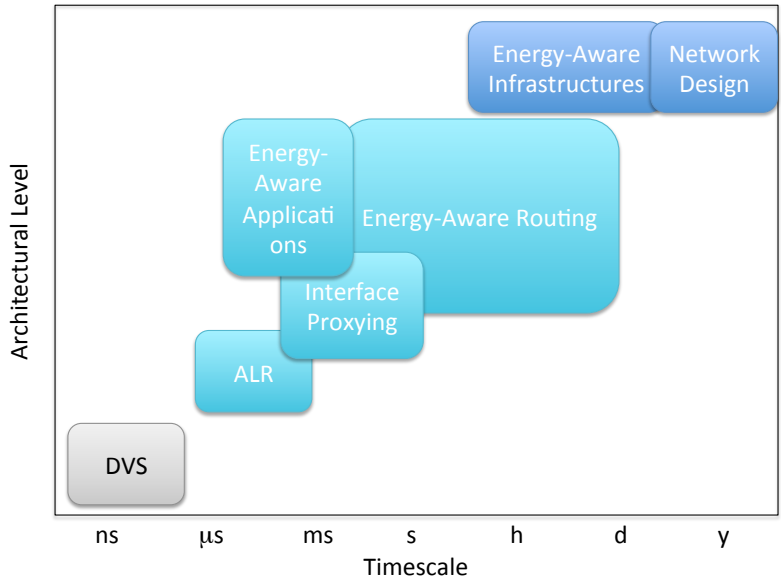

Figure 2.2: A picture of the current green networking research.

ones, make green networking an interesting and technically challenging research field. A major shift is indeed needed in networking research and development to introduce energy-awareness in the network design, without compromising either the quality of service or the network reliability.

While, for the time being, network devices and protocol are mostly unaware of the energy they consume, a number of valuable research works have started exploring energy-awareness in fixed networks. A natural classification of the different approaches may be based on the *timescale* of the decisions involved by the green strategy. As pointed out in [12], timescales on the order of nanoseconds to microseconds apply to CPU and the instruction level, which is relevant in the computer and software architecture levels, and thus concern only individual building blocks of a single system. Timescales on the order of micro to milliseconds are instead relevant at the system layer. At these timescales, actions may be taken between consecutive packets of the same flow (inter-packets, intra-flow), possibly involving several components at the same time, but likely confined within a single system. Larger timescales, on the order of one second and above, allow instead the action to span between multiple entities, possibly involving coordination of such entities as well. Notice that timescales directly define the *architectural level* at which actions can be taken: the shorter the timescale is, the lower the layer and the less possible interaction among different components.

The main branches of the green networking research explored so far are reported in Fig. 2.2, classified by timescale and architectural level. Lower values of the y-axes correspond to solutions concerning single chips, while going up we find solutions involving whole devices, and device systems, of growing size. The main branches of green networking research are, namely, **Adaptive Link Rate (ALR)**, **interface proxying**, **energy aware applications**, and **energy aware routing**. The remaining of this section briefly overviews these four research directions, we refer the interested reader to [4,13] for a more detailed overview.

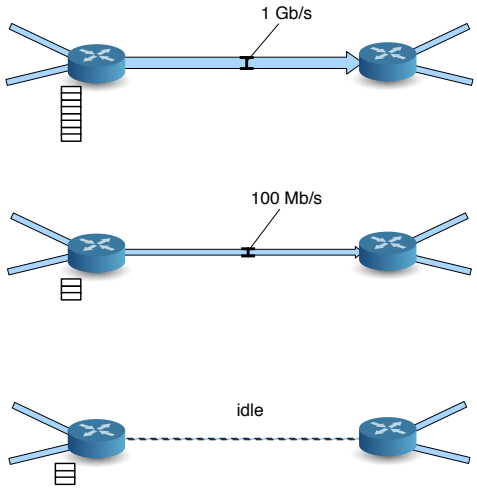

Figure 2.3: Adaptive Link Rate strategies: the rate of a 1 Gb/s link can be reduced to 100 Mb/s (rate switch, middle plot) or the link can be made idle to save energy (sleeping mode, bottom plot), depending on the adjacent routers loads.

2.2.1 Adaptive Link Rate

Most of the effort in green networking has been devoted to solutions that are referred to as ALR, up to now. These techniques are designed to reduce energy consumption in response to low utilization in an on-line manner (i.e., at run time). A considerable number of works have explored this solution, and the IEEE Energy Efficient Ethernet Task Force is now complete with the approval of IEEE Std 802.3az-2010 at the September 2010 IEEE Standards Board meeting [14].

The basic intuition behind this solution is based on the fact that the energy consumption of Ethernet links is largely independent from their utilization level [15], but depends only on the negotiated rate and working state (e.g., on/sleep). In order to have a power consumption of the network system as proportional as possible to its utilization level, it is hence possible to either (i) force links entering low power states during idle periods, or (ii) reduce the link rate during low utilization periods. The two strategies are illustrated in Fig.2.3.

A wide literature exists, that analyses the different possibilities offered by the link rate adaptation/switching strategy, and its key points: (i) which and how many low power states to use, (ii) use a sleep mode or a rate adaptation, (iii) consider different QoS versus energy saving tradeoffs. Different algorithms have been proposed to drive the rate/state change, which better adapts to different power models and technological assumptions (e.g., switching time), as well as to different traffic conditions. The proposed algorithms rage from the simple instantaneous observation of the transmission buffer state, as in [16], to more complex solutions, including timers to avoid oscillations, as in [17], or an analysis of the temporal evolution of the buffer state, as in [18]. Different algorithmic solutions are compared in [19], while practical aspects have been standardized by the IEEE Energy Efficient Ethernet Task Force [14].

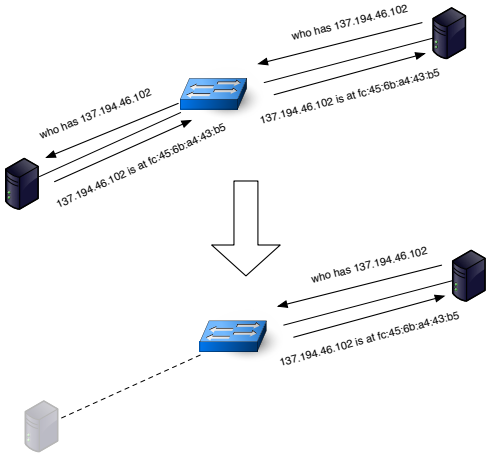

Figure 2.4: External proxying: a switch acts as a proxy for ARP traffic, allowing the target machine to sleep at least until data traffic is sent.

2.2.2 Interface Proxying

This category includes green networking solutions exploiting the presence of consistent time intervals in which access devices are unused, but would normally be forced to support network connectivity tasks (e.g., periodically sending heartbeats, receiving unnecessary broadcast traffic, etc.). In these time intervals, edge nodes can for instance enter an energy saving state, while the network connectivity tasks may be taken over by other nodes, such as proxies, momentarily faking the identity of idle devices, so that no fundamental change is required in network protocols.

With respect to the previously analysed solutions, where functionalities could be simply turned off (e.g., no transmission at all when the link is idle), in the case of end devices this is however not possible, as some functionalities need to be delegated (i.e., traffic processing is handed over to more energy efficient entities). Indeed, without proxying solutions, even though users are idle, background network traffic is nevertheless received and needs processing, thus preventing PCs from going in sleeping mode.

The idea behind *interface proxying* consists in delegating the processing of the network background traffic (a.k.a. *chatter*). Such processing may include simple filtering (e.g., in the case of unwanted broadcast/portscan traffic), simple responses (e.g., in the case of ARP, ICMP, DHCP), or even more complex tasks (e.g., in the case of P2P applications such as Gnutella or BitTorrent). Such tasks can be delegated from the energy-hungry mainboard CPUs of end devices to a number of different entities: e.g., locally to the low-energy processor onboard of the NIC of the same device, or to an external entity (in this latter case, a single proxy may be deployed for several machines in a LAN environment, or this duty may fall to the set-top-box in a residential environment). In all cases, the proxying entity will also be in charge of waking up the full system when non-trivial packets requiring further processing are received. Fig. 2.4 illustrates an example of external proxying in which a switch answers an ARP request instead of the targeted computer, allowing the end machine to remain asleep until it receives applicative traffic.

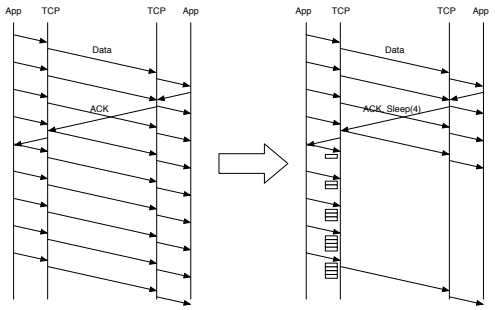

Figure 2.5: In a modified version of TCP (right), the receiver may notify its peer of its intention to sleep. During the sleep period (expressed here in number of segments for the sake of illustration), the source buffers the specified amount of data instead of directly transmitting it.

Different solutions have been explored so far, falling in the *interface proxying* category, that make it a pretty mature field. Explored solutions include software and hardware implementations performing traffic classification and proxying over currently available hardware [20], as well as commercial solutions, as the Wake on Demand feature of Mac OS X v10.6, to make shared files available also from sleeping machines [21].

2.2.3 Energy Aware Applications

Some work have been done in the direction of reducing the unnecessary energy waste in networks by modifying the network devices operating systems and applications, as well. Also in this case, the goal is to allow network equipments entering sleep states without loosing their network presence, as for *interface proxying*. Unlike interface proxying, where the objective is reached by delegating functionalities to more energy efficient devices, the idea behind *energy aware applications* is to achieve the goal by redesign specific applications and operating system modules.

An example of energy aware application is the *green* version of BitTorrent described in [22]. Here peers advertise their energy state, so that other peers may avoid waking up idle peers, preferring chunks that are available at active ones.

Acting on the operating system to reduce the network energy consumption may be instead done by a redesign of the transport-layer functionalities. In this case, the optimizations are then shared by all the applications. An example of this approach has been proposed in [23]. Here, specific signaling options are introduced in the TCP header to signal the intention to enter a sleep state to the other party, which will buffer data for the sleeping duration, without interrupting the connection. This strategy is illustrated in Fig. 2.4.

As the spectrum of the Internet applications is clearly a huge one, many directions remain to be explore to push energy awareness into applications and operating systems. A good starting point may be the translation of the green principles into good programming practices. This practice may potentially have a huge impact on the greening of networks, since is able to hit both widespread programming libraries and popular end-user applications.

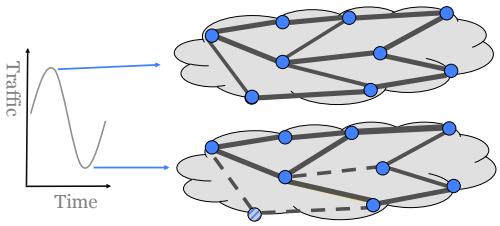

Figure 2.6: Energy aware routing: routers and links are put to sleep when the network load is low, while preserving connectivity. This technique may increase the load on some links (links are represented with different edges thicknesses on the picture, to reflect different loads) and QoS performance need to be carefully studied.

2.2.4 Energy Aware Routing

The solutions seen so far involve only local decisions, through a single device or a very small set of collaborative devices. While these techniques alone offer non-negligible energy saving, further improvement can be expected from a reasonable amount of collaboration between individual devices, sharing a wider knowledge on the system state.

Following the resource consolidation principle, energy aware routing (illustrated in Fig. 2.6), generally aims at aggregating traffic flows over a subset of the network devices and links, allowing others to enter energy saving states. These solutions should preserve connectivity and QoS, for instance by limiting the maximum utilization over any link, or ensuring a minimum level of path diversity. Flow aggregation may be achieved, for example, through a proper configuration of the routing weights. On the basis of different technological assumptions adopted, different solutions for energy aware routing aims at putting in a sleep mode both links and nodes, or only links.

The problem of energy aware routing has been faced from different points of view, starting from the original proposition in [24]. At first, centralized solutions have been studied, supposing the presence of a central unit with a network global view, and evolving from the formal definition of the corresponding optimization problem [25], to the proposition of different heuristics. Centralized solutions to the energy aware routing approach are analysed in details in Chapter 4. More recently, distributed approaches have been proposed in the direction of energy aware routing, in which usually nodes share information laveraging the existent routing protocol, and take independent decisions on the working/sleep state of the incident links. Distributed solutions for energy aware routing are discussed in details in Chapter 5.

2.2.5 Clean-Slate Approaches and Network Design

The techniques seen so far involve the injection of energy awareness into current network architectures. Few works exist in the literature that have tackled energy awareness from a global perspective, advocating the complete redesign of the network architectures. Some works, for instance, advocate a higher use of optical networks (e.g. Dense Wavelength Division Multiplexing - DWDM). It is now admitted that optical switching is much more energy efficient, while offering an extremely large capacity. At the same time, these technologies still suffer from a lack of flexibility with respect to the electronics domain (as in the optical domain no buffering is possible, which motivated optical burst switching [26]). A future challenge is probably to find efficient architectures combining

both optical transport and packet processing, when needed. For instance, [27] falls in this category by proposing to complement the Internet with a parallel virtual DWDM "super-highway" dedicated for deterministic traffic.

Finally, the problem of introducing energy-awareness into the network design process is studied in [28, 29] from an operational research point of view. In more detail, [28] introduces the energy consumption cost into the multicommodity formulation of the design problem, together with the performance and robustness constraints. A similar approach is adopted in [29], which evaluates also the tradeoff between energy consumption and network performance, highlighting the fault tolerance characteristics of the different possible working points.

2.3 The Benchmark Issue

All the solutions to reduce the network energy consumption, as well as their evaluation, strongly rely on an energy consumption model. On the one side, we find **network dimensioning** solutions, i.e., solutions facing an already defined network scenario, including an already defined set of network devices. This set of solutions selects the best working point for the network devices, in order to adapt the network energy consumption to the current load (e.g., interface proxying, or energy-aware routing). The choice of the device working points must be based on a specific energy model, describing a specific technological scenario. The evaluation of network dimensioning solutions should take into account the tradeoff between the achievable *energy saving* and the reduction of the *QoS and redundancy level*, that result form the application of a specific solution.

Often, rather different estimates are gathered, via different models, for the same power figure. This is especially true whenever large scale networks, as opposed to individual devices, are considered. Yet, we point out that obtaining power figures for real network infrastructures represents a very challenging task, due to the inconsistence of the different models, which further become quickly out-of-date. A careful sensitivity analysis of a power model may show that the uncertainty of the overall results remains substantially high, even when relying on carefully chosen and cross-verified data [30]. A community-wide effort is hence necessary to minimize such uncertainty, toward the definition of a comprehensive methodology for measuring and reporting the energy consumption of networks, and toward the creation and maintenance of a repository of energy-related figures. We contribute to this effort by profiling the end-user power consumption related to the Web browsing and Flash plug-in loading, by considering different hardware platforms, operating systems, browsers, and websites [31]

On the other side of the green networking spectrum, we find **network design** solutions, which address the design of new network infrastructure. This set of solutions selects the network technologies and the set of network devices to form the infrastructure (e.g., [27–29]). The choice of the network technology and devices is generally driven by the tradeoff between the system performance, and the corresponding energy expenditure. Despite the fact that energy efficient communication networks have already gained a considerable attention by a broad research community, a comprehensive methodology for measuring and reporting the energy consumption of the network systems looks far to be established.

The current literature proposes many heterogeneous metrics to qualify and quantify the energy savings. Due to this heterogeneity, comparison of competing solutions may be more favorable to a solution rather than to another, depending on the metrics used to express the results of the benchmark. Hence, it becomes fundamental to define a coherent framework for the evaluation of networks, which should be able to fully characterize any possible tradeoff between energy consumption and system performance. Our contribution in this direction is represented by the comparison and contrast of various energy-related metrics used in the recent literature, by means of a taxonomy definition, as well as through relevant case studies [32].

Chapter 3

Energy-Aware Routing as an Optimization Problem

Once a network has been designed (i.e., the resources that will compose it have been deployed), a periodical off-line process is applied to optimize the utilization of resources, which is usually referred to as "routing optimization". This classical process consists, in particular, in determining the paths used for each origin-destination pair or, equivalently, to ingress-egress routers in a transit network. Common optimization objective is to avoid congestion by e.g., balancing the traffic as evenly as possible on the network links, or by ensuring that maximum link utilization always remains below a given threshold. In pure IP networks, the path used by each flow is determined by the Internal Gateway Protocol (IGP), based on link administrative weights. Network dimensioning is thus handled by careful weight assignments, for instance using IGP Weight Optimization (IGP-WO) algorithms [33].

One of the most common green practices in network dimensioning consists in *resource consolidation*: this technique aims at reducing the energy consumption due to devices underutilized at a given time. Given that the traffic level in a given network approximately follows a well known daily and weekly behavior, there is an opportunity to aggregate traffic flows over a subset of the network devices, allowing other devices to be temporarily switched off. This solution shall of course preserve connectivity and QoS, e.g., by limiting the maximum utilization over any link. In other words, the required level of performance will still be guaranteed, but using an amount of resources that is dimensioned over the actual traffic demand, rather than for the peak demand. Flow aggregation may be achieved, for example, through a proper configuration of the routing weights in an IP network.

3.1 The Problem Formulation

The problem of resource consolidation may be formalized as an optimization problem, where the objective is the minimization of the total network energy consumption, and constraints include the classical connectivity constraints, and QoS constraints. In more details, we represent the network as a directed graph, $G = (N, L)$, with N the set of nodes modeling interconnection devices, and L the set of arcs modeling the communication links. For any network element a (node or link), we will denote by l_a its load and by c_a

its capacity, i.e., the maximum load it can support.

The objective is to find the network configuration (i.e., the working point of the nodes and links of the network) that minimizes the total network energy consumption, expressed as the sum of the consumptions of all nodes and links. The energy consumption of devices is generally considered to be composed by (i) a fix part E_{0a} (corresponding to the idle energy consumption), and (ii) a variable part $E_{va}(l_a/c_a)$, depending on the traffic load of the device. To model this, a binary variable x_a is considered to model the status of the element a ($x_a = 1$ whenever a is *on* and $x_a = 0$ otherwise). Finally, links are full duplex and they are considered entirely powered on as soon as one direction conveys traffic. Since in the above graph formulation the two directions are separately modeled, the link load is the sum of both directions loads. With this model, the network total energy consumption may be represented by the following expression (where the first sum needs to be divided by a factor 2 in order to avoid counting links twice):

$$\frac{1}{2}\sum_{(i,j)\in L}(E_{fij}((l_{ij}+l_{ji})/(c_{ij}+c_{ji}))+x_{ij}E_{0ij})+\sum_{n\in N}(E_{fn}(l_n/c_n)+x_nE_{0n}) \tag{3.1}$$

The load imposed to the network is defined by a Traffic Matrix (TM) that specifies, for every couple of ingress and egress nodes (s, d), the traffic flowing from s to d, denoted by r_{sd} hereafter. Each traffic requests from s to d is routed across the network, generating a traffic of f_{ij}^{sd} over any link (i, j). This TM defines the following set of constraints:

$$\sum_{(i,s,d)\in N^3} f_{ij}^{sd} - \sum_{(i,s,d)\in N^3} f_{ji}^{sd} = \begin{cases} r_{sd} & \forall (s,d)\in N^2, j=s \\ -r_{sd} & \forall (s,d)\in N^2, j=d \\ 0 & \forall (s,d)\in N^2, j\neq s,d \end{cases} \tag{3.2}$$

As mentioned above, to preserve QoS, no links should reach a 100% utilization, or more in general, an arbitrary value α that the network manager considers safe enough. This defines the following set of constraints:

$$\sum_{(s,d)\in N^2} f_{ij}^{sd} = l_{ij} \leq \alpha c_{ij} \quad \forall (i,j)\in L \tag{3.3}$$

We further assume node load to be directly proportional to the amount of traffic entering and leaving the node. In particular, we consider that they are equal, which adds the following constraints to our problem:

$$l_n = \sum_{(i,n)\in L} l_{in} + \sum_{(n,i)\in L} l_{ni} \quad \forall n\in N \tag{3.4}$$

Finally, we consider that a node or a link is switched off if its load is equal to zero. This allows to relate variables x_a and l_a for any element of the network through the following sets of constraints:

$$Zx_{ij} \geq l_{ij} + l_{ji} \quad \forall i, j \in L \tag{3.5}$$

$$Zx_n \geq l_n \quad \forall n \in N \tag{3.6}$$

where Z is a "big" number (i.e., greater than twice the maximum between the nodes and the links capacities), used to force the variable x_a to take the value 1 when a has a load greater than 0, and the value 0 when $l_a = 0$

Minimizing the total energy consumption (3.1) while satisfying all the constraints mentioned in this section is a mixed integer program, with binary variables (x_a) and continuous variables (l_a).

3.2 Results on Real Network Scenarios

The above introduced optimization problem has been solved for some network scenarios and energy models, in order to gather guidelines on its behavior.

First of all, the problem requires to rely on an accurate energy consumption model in order to be correctly formulated and solved. Yet, we point out that obtaining energy consumption figures for real network infrastructures represents a very challenging task, due to the inconsistence of the different models, which further become quickly out-of-date. The studied formulation (eq. 3.1) includes thus a general model, describing the device energy consumption as a function of their utilization level, and expressing it in a parametric form, making the model easily extensible to other cases.

It is generally accepted that the energy consumption of network devices grows linearly between a minimum value E_0, which corresponds to the idle state, and a maximum value M, which corresponds to the maximum utilization [34]. Furthermore, a null energy consumption is assumed when the device is in a sleep/off state[1] (i.e., when its utilization level is equal to 0). The general model for the considered devices is illustrated in Fig. 3.1 by a solid red line, and will be referred to as "idleEnergy".

Two special cases of the considered energy model are of particular interest in our analysis. In the "fully proportional" model, the parameter E_0 is equal to 0. This model represents an ideal case of fully energy aware devices, such as communication links supporting rate adaptation [17]. Nodes could also present such a behavior when their components are regulated in function of the load (e.g., Dynamic Voltage Scaling (DVS), modular switching fabrics, etc.). The fully proportional model is thus a resultant of several green technologies, which are not necessary available today, and is thus to be considered as a futuristic scenario. This model is illustrated in Fig. 3.1 by a dashed green line. On the opposite, in the "energy agnostic" model, the E_0 parameter is equal to M.

[1]Depending on the considered technology, devices may be completely switched off, or allow different sleep states. We will generically refer to any sleep/off state as *off*, but similar considerations can be done for a generic sleep state.

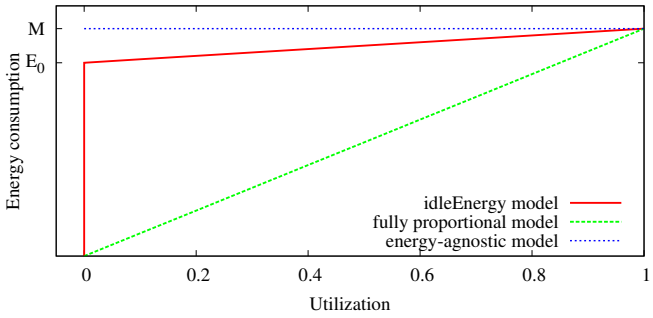

Figure 3.1: Different models for the network device energy consumption, expressed as parameterized function of the device utilization.

This case models network elements whose energy consumption is constant, independently from their load, and are never powered off (i.e., a common case today). This case is illustrated in Fig. 3.1 by a dotted blue line.

Two interesting case studies for the optimum solution of the energy-aware routing are represented by the Geant network scenario [35], and by the Italian ISP network scenario [36]. The Geant case study represents a worst case for what concerns the achievable energy saving, since in this network scenario all nodes are access nodes (i.e., source and destination of traffic requests), which can hence never be switched off. On the other hand, the Italian ISP scenario represents a well-disposed scenario for what concerns the achievable energy saving, since the network have been designed with a high degree of redundancy, and an high number of devices can be switched off without affecting the normal network working state.

In practice, topology and traffic details are publicly available for the **Geant** scenario [35]. It represents a real and fairly complex network, including 23 nodes, and 74 links (the network topology is reported in Fig. 3.2 (a)). As traffic data, we selected a subset of the available TMs, specifically 24 TMs, taken at hourly intervals between 00:30 and 23:30 of 5/5/2005. Notice that this TM set includes the complete traffic variation of a standard working day. Notice also that results are averaged over the whole day period. The aggregated traffic variation is reported in Fig. 3.2 (b). For what concerns the energy model, we tuned its parameter for this network, according to power figures available in [16, 37–39], which represent widely accepted and diffused figures available in the literature. Table 3.1 summarizes the used values for the E_0 and M parameters, where C represents the node switching capability. As the overall switching capability for nodes in the Geant topology is not available, we considered a node as being able to switch the double of the sum of the capacity of all links connected to it. This is a design conservative choice, that would allow the network manager to add a reasonable number of links without having to change the devices.

The optimization problem has been modeled using AMPL [40], and CPLEX [41] has been used for its numerical solution. Results have been obtained considering the power figures reported in Table 3.1, considering an *idleEnergy* behavior, but also, based on the same value of the M parameter, considering an *energy-agnostic* and a *fully proportional* behavior, to model both legacy energy agnostic devices, and foreseen more green devices.

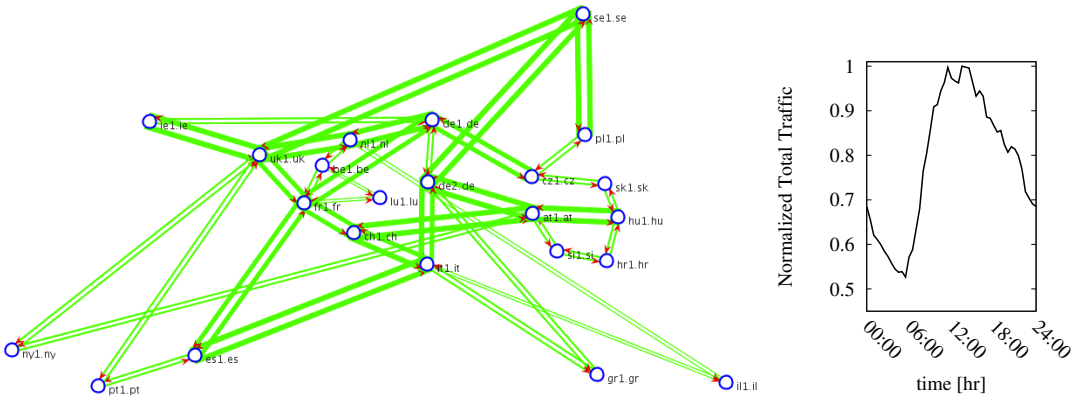

Figure 3.2: A representation of the Geant network topology used in the solution evaluation (on the left - the link thickness is proportional to the link bandwidth), and the day/night behavior of the aggregated traffic profile for the Geant network scenario (on the right).

Table 3.1: Energy consumption parameters in Watts, for the different Geant network elements.

Network element	E_0 [Watt]	M [Watt]	Ref.
Nodes	$0.85C^{3/2}$	$C^{3/2}$	[38]
(0-100] Mbps links	0.48	0.48	[39], [16]
(100-600] Mbps links	0.90	1.00	[39], [16]
(600-1000] Mbps links	1.70	2.00	[37]

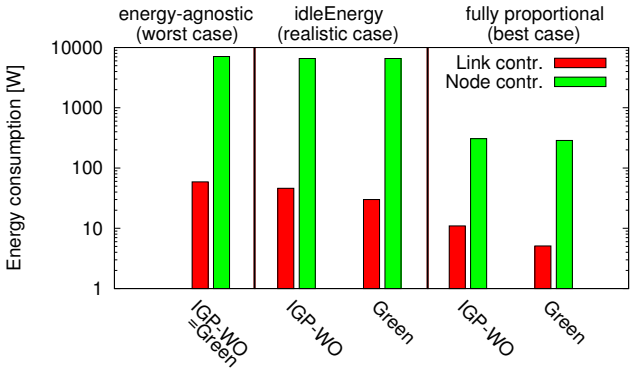

Figure 3.3: Energy consumption in Watts for the Geant network scenario, considering the IGP-Wo routing and the routing resulting from the solution of the optimization problem ("Green"), for different energy models.

As a performance metric, the percentage of energy saving has been selected, with respect to a routing configuration using the IGP-WO algorithm [42]. IGP-WO is a standard practice in operator networks. This reference scenario will be referred to as "IGP-WO routing" in the following. The total network energy consumption resulting for the three energy models, in the case of IGP-WO routing, and of optimal routing ("Green") are reported in Fig. 3.3. More detailed results are reported in [43].

In general, results obtained for the Geant network scenario with the *idleEnergy* model, show that energy saving is mainly a consequence of switching off network el-

ements, since this avoids the idle energy consumption E_0. From the values considered in Table 3.1, it is clear that the impact of the fix component E_0 on the overall energy consumption is much greater than the proportional energy component due to the device load ($M - E_0$). Moreover, it results also that the energy parameters of nodes are generally two orders of magnitude larger than the ones of links. This means that energy saving resulting from switching off links represents a small contribution to the total energy saving (even if consistent with respect to the sole link contribution, i.e., about 34% of saving). However, given the topology and traffic level, it is generally not possible to switch off nodes (every node is source and destination of traffic, in the Geant network), even if it is possible to switch off links. This is why the Geant scenario represents a worst case for energy-aware routing, lower bounding the achievable energy saving (the overall energy saving is on average around 0.2%).

In the case of the *fully proportional* model, the energy saving is a consequence of the aggregation of traffic over paths involving the most energy efficient devices, while there is no interest in switching off nodes and links, since there is no idle energy consumption ($E_0 = 0$). It results that (i) it is possible to achieve a much higher energy saving by means of energy-aware devices (presenting a fully proportional energy consumption) than with nowadays devices presenting at most a partial energy awareness, and that (ii) green routing and green technologies (such as link rate adaptation [14] and dynamic voltage and frequency scaling [44], which bring links and nodes close to a fully proportional behavior) naturally interact for enhanced saving performances (e.g., green routing brings a further 8% of energy saving, when used on the Geant scenario with fully-proportional devices, much more than what it was able to bring in the case of the *idleEnergy* model, i.e., 0.2%).

The Geant network scenario represents a limit case also for what concerns QoS considerations. If the maximum link load is bounded by the α parameter, the link load distribution (the main QoS indicator) does not change much when passing from the IGP-WO routing to the optimum energy-aware routing (the two distributions are compared in Fig. 3.4 for the 00:30 TM under the *idleEnergy* model. Qualitatively similar considerations hold for the other scenarios as well). The main difference between the two load distributions is that the "Green" solution brings a considerable number of links to a zero utilization (links that can now be turned off to save energy). As a consequence, green routing also increases the number of links with a higher utilization level, as traffic is aggregated over a subset of the network devices. In the Geant scenario, the switch off procedure only involves links which are lightly loaded when the IGP-WO routing is used (i.e., about 5% of average load). This is why such procedure does not affect much the link load distribution, even if intuitively it is acting against the common practices to guarantee QoS in network (e.g., redundancy and distribution of the charge over all the available paths).

On the other side of the scenario spectrum, we can find the **Italian ISP** [36]. The "Italian" network, in fact, presents different hierarchical level nodes, among which only the nodes belonging to the first and the last level are source and destination of traffic, while nodes belonging to other hierarchical levels can be switched off to save energy, without affecting the network working state. Moreover, all the inter-level connections are redundant, to guarantee fault tolerance, making possible to switch off redundant

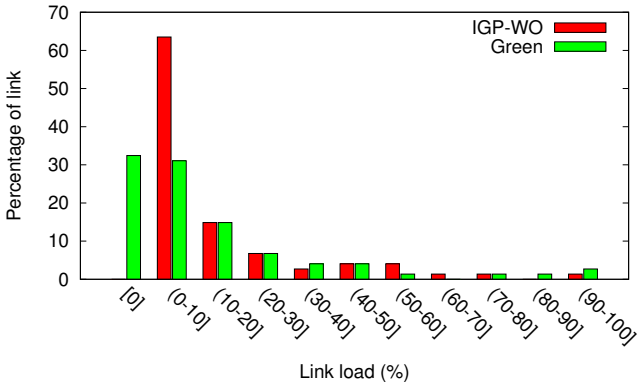

Figure 3.4: Link load distribution under the IGP-WO and optimum energy aware routing ("Green").

paths to save energy. Finally, the energy model takes into account different technologies with respect to what has been done for the Geant scenario, including optical transceivers, and regenerators along links, and resulting in a much higher contribution of links to the total power consumption. As a result, a much higher energy saving can be achieved (i.e., about 35%), even with non energy-proportional devices (see [36] for more detailed results).

3.3 Possible Formulation Extensions

The above presented formulation for the energy-aware routing considers a specific technological scenario, in which communication links can be switched off, as well as whole routers. Note that whole routers are switched off only if all the incident links are switched off. Some technological scenarios may present different constraints. For instance, considering currently deployed nodes, they present long switching on/off times, requiring eventually to act on a multi-hour optimization base, in order to minimize the number of reconfigurations across the day. Furthermore, in case of faults, off devices have to be switched back on, in order to accommodate the extra traffic previously carried by the fault device. In case of too long switching on times, network management policies may not allow switching off whole routers, but only communication links.

On the other hand, when device architecture allows acting on a more fine granularity, by independently switching off single parts, the model can easily be extended to account for that possibility. For instance, it may be possible to switch off single cables in a bundle, as considered in [45], or linecards for which all ports are off, or switching fabrics for which all the connected line cards are off, etc.

We choose a model which was complete enough to account for all network devices, modular enough to be easily extendable, and, at the same time, simple enough to be solvable in reasonable amount of time for average size network scenarios. Starting by the presented model, extensions to include different architectures or policies, are straightforward.

Chapter 4

Device Criticality and Heuristics for the Energy-Aware Routing

The optimization of the energy-aware routing problem, as presented in Chapter 3, is a mixed integer program, presenting both integer and linear variables, and thus NP-hard to be solved. As such, the time required for its solution computation rapidly increases, becoming infeasible for large topologies. This remains true even when considering ad-hoc formulation modifications, which take advantage of particular network characteristics, as shown in [36]. Moreover, in the solution of the optimization problem, the choice of the set of devices to be switched off to save energy is driven only by the energy costs, and does not take into account the "criticality" of devices in the network scenario.

To overcame the complexity of finding an optimal solution to the energy-aware routing when considering large networks, some heuristics have been studied and evaluated. In particular, the proposed heuristics are based on successive switch off attempts for network devices, based on a specific *ranking*: every network device is considered in turn, and switched off if its absence in the current network configuration does not affect the normal network working state [25, 36, 46]. Examples of evaluated device rankings are:

- Most Power (MP - switching off devices from the most power-hungry to least one). The intuition behind this heuristics is to maximize the impact of the resource consolidation process on the energy saving [36].
- Least Flow (LF - switching off devices starting from the one which would be least loaded in an energy agnostic configuration). The intuition behind this heuristics is to minimize the impact of the resource consolidation process on the traffic routing [36]. This heuristics has been further refined in [46], by recomputing the link load after each switch off attempt, instead of considering only the link load of the energy-agnostic configuration along the whole resource consolidation process.
- Opt-edge (OE). This heuristics exploits specific characteristics on the network topology in exam, namely the multi-home connection of edge nodes, by grouping paths to/from edge nodes, on a subset of the higher level nodes, and switching off the others [25].
- Least Link (LL - switching off nodes from the one with the lowest degree, and the corresponding adjacent links). This heuristics leverage on the intuition that nodes

with lower degree are easier to be switched off without affecting the network functionalities, and that switching off a single high degree node can deny to more than one other node to be switched off without affecting the normal network working state [25].

- random (R). This ranking is usually used as lower bound in the performance evaluation.

Some of the proposed heuristics are specifics for network scenarios in which it is possible to switch off entire nodes (e.g., LL), others can be adapted for scenarios with different technological constraints (e.g., the MP policy may be used to sort both links, and nodes). In the case in which both nodes and links can be switched off, it has been shown that higher energy savings can be achieved by first performing switch off attempts on nodes, and then on the residual links, as otherwise specific link configurations may then avoid nodes to be switched off [36]. For instance, in the case of the MP policy, a first switch off attempt is performed on nodes, following the MP ranking on nodes, and, then, a switch off attempt is performed on the residual links, following the MP ranking on links. In this way, mixed strategies may be defined, following two different rankings on nodes and links. Detailed results are presented in [36], analysing the different resulting trade-offs between energy saving and QoS.

4.1 A Criticality-Driven Device Ranking: the G-Game

All the previously considered rankings select the set of devices that can be safely switched off on the basis of their power consumption (e.g., MP), or of their working state in an energy-agnostic network configuration (e.g., LF). We believe that the resource consolidation process should be driven by an accurate evaluation of the criticality of devices in the specific network scenario. However, there is no satisfactory definition of the criticality of nodes in a network. Classical indexes rank routers based either on the sole *topological* aspects, such as *betweenness centrality*, *degree*, *closeness*, *eigenvector*, or on the sole *traffic* load, such as LF.

The Game theory represents a powerful tool to define a criticality index that accounts for both aspects at the same time: modeling the resource consolidation problem as a cooperative Transferable Utility Game (TU-Game), the *Shapley value* of each node indicates how much the node contributes in the traffic delivery process, and how its absence would affect the network on "average" (i.e., over all possible network configurations). This game, (the *Green-Game*, or G-Game for short, from now on) takes as its only inputs the *network topology*, i.e., the set of links and devices, and the *TM*, i.e., the amount of traffic routed by the network between each pair of devices. The Shapley value on the G-Game defines a joint topology-aware and traffic-aware ranking of the network devices, that can profitably be used to drive the resource consolidation process.

4.1.1 The G-Game Definition

A communication network can be represented as a graph $G = (N, E)$. N is the set of vertices, whose elements, $i \in N$, represent the interconnection nodes (routers, switches, etc.). E is the set of edges, whose elements $e = \{i, j\} \in E$, represent the communication

links existing between pairs of nodes $i, j \in N$. We denote by n the cardinality of N (i.e., $n = |N|$). Between any two nodes $i, j \in N$, data may be transported along one or several *paths*. A path is an ordered sequence of vertices $p_{i,j} = (i = i_0, i_1, i_2, \ldots, i_{k-1}, i_k = j)$. A path that does not contain twice the same node, $\forall i_a, i_b \in p_{i,j}, i_a \neq i_b$, is called an *acyclic* (or *loop-less*) path.

Communication networks are dimensioned based on measurements and estimates of the volume of data they have to support under realistic conditions. Various scenarios such as daytime traffic, nighttime traffic, etc. may be considered. Each scenario is characterized by a *traffic matrix*, $T = (t_{i,j})_{i,j \in N}$, in which an element $t_{i,j}$ represents the volume of traffic entering the network through node i and exiting through node j. We denote by v the total traffic load that the network has to route, with respect to a given TM T: $v(N) = \sum_{i,j \in N} t_{i,j}$.

Let us consider an arbitrary subgraph of G, $G_S = (S, E_S)$ formed by the nodes $S \subseteq N$ and by the corresponding subset of edges $E_S = \{\{i, j\} : i, j \in S\} \subseteq E$. The amount of traffic that G_S can effectively transport, with respect to T, is denoted by $v(S) = \sum_{i,j \in S} t_{i,j} \mathbb{1}G_S(i, j)$, where $\mathbb{1}G_S(i, j) = 1$ whenever i and j are connected in G_S (i.e., there exist a path in G_S from i to j) and zero otherwise. By convention $v(\emptyset) = 0$.

Let us denote by $\mathcal{P}(N)$ the set of parts (i.e., subsets) of N. As N is a finite set of elements and as v is a function of $\mathcal{P}(N)$ into $\mathbb{R}$, the couple (N, v) defines a *coalitional game*, precisely the *G-Game*. A group of nodes (players), $C \subseteq N$ is called a *coalition* and the value $v(C)$ is called the *worth* of the coalition C, while v is called the *characteristic function* of the game. The problem of determining which network elements can be safely switched off without disrupting the network can be modeled as the search for a coalition with the same worth as the full network, but with a reduced size.

In other words, given any TM, we need to identify the most important nodes in the network: in case the problem is modeled as a coalitional game, the solution is represented by Shapley value. The Shapley value averages the marginal contribution of each node over many possible scenarios, which makes it perfectly suited to find a good tradeoff between saving energy and preserving QoS.

Let us denote by Σ_N the set of permutations over N: $\Sigma_N = \{\sigma : N \to N : \sigma \text{ is a bijection}\}$. We also denote by $B[i, \sigma]$ the set of nodes that appear *before* node i with respect to permutation σ, including i itself: $B[i, \sigma] = \{j \in N \text{ s.t. } \sigma^{-1}(j) \leq \sigma^{-1}(i)\}$. $B(i, \sigma)$ is similarly defined as the set of nodes that appear *before* node i with respect to permutation σ, excluding i: $B(i, \sigma) = \{j \in N \text{ s.t. } \sigma^{-1}(j) < \sigma^{-1}(i)\}$. The *marginal value* of node $i \in N$, with respect to the order σ is defined as:

$$m_i^\sigma = v(B[i, \sigma]) - v(B(i, \sigma)) .$$

Intuitively, the marginal value of a node according to an order represents its importance in maintaining the network performance when nodes are switched off (or fail), one by one, following the order σ. The *Shapley Value* ϕ_i of node $i \in N$ is defined as the average of the marginal values associated to i for all possible permutations of N:

$$\phi_i = \sum_{\sigma \in \Sigma_N} \frac{m_i^\sigma}{n!} . \tag{4.1}$$

ϕ_i defines a ranking on the nodes, which appears particularly relevant for our problem. For each node i, ϕ_i increases with the number of coalitions that i participates to and with the importance of i in each coalition. The Shapley value takes indeed into account the number of primary and backup paths each node lays on, reflecting the position of the node in the topology in a similar way to centrality measures. Sec. 4.1.3 provides a comparison with other classical centrality indexes. Exploring every path, the Shapley value grants higher values to nodes whose removal would disconnect the graph, or to nodes belonging to small sets whose presence in the network is essential for traffic delivery. Another important advantage of this approach is that this ranking takes into account the characteristic function v, defined as the volume of traffic transported by a coalition. In other words, the higher the value ϕ_i for a device i is, the higher its contribution to traffic routing on average over all coalitions will be. For more insight on the Shapley value, we refer the interested reader to [47].

4.1.2 On the Efficient Computation of the Shapley Value

The computation of Shapley value according to (4.1) is computationally expensive, as it requires considering all the $n!$ potential permutations of N. However, any coalitional game can be decomposed as a linear combination of *unanimity games* [47]. This decomposition provides a less expensive method to calculate the Shapley value. For a set of players N, an unanimity game, (N, u_R), is defined over a subset of nodes $R \subseteq N$ by its characteristic function, u_R, which associates to any subset $C \subseteq N$ a boolean value: $u_R(C) = 1$ if and only if $R \subseteq C$, $u_R(C) = 0$ otherwise. By convention, $u_R(\emptyset) = 0$. Any coalitional game (N, v) admits a unique decomposition in unanimity games over $\mathcal{P}(N)$:

$$v = \sum_{C \in \mathcal{P}(N)} \lambda_C \, u_C, \text{ with } \lambda_C(v) \in \mathbb{R}, \forall C \in \mathcal{P}(N)\,, \tag{4.2}$$

where λ_C are called the Harsanyi dividends [48], that are defined recursively by:

$$\lambda_C = \sum_{B \subset C} (-1)^{|C|-|B|} \, v(T)\,. \tag{4.3}$$

The Shapley value of a node $i \in N$ is fully determined by these dividends, considering all the subsets of N in which i appears:

$$\phi_i = \sum_{C \in \mathcal{P}(N); i \in C} \frac{\lambda_C}{|C|}\,. \tag{4.4}$$

The complexity of this computation is $O(3^n)$, considering that this expression requires at most a computation of all the 2^n Harsanyi dividends. Each $\lambda_C(v)$ computation requires to enumerate all the subsets B included in C (i.e., $2^{|C|}$ sets). Ordering the sets C by increasing cardinality, we can thus see that the total complexity for computing all the dividends can be expressed as: $\sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n$. Even though 3^n is asymptotically lower than $n!$, the algorithm complexity remains exponential.

Fortunately, a further simplification is introduced in [49]: as the Shapley value reflects the importance of a node in the routing process, we do not need to consider the whole

Figure 4.1: Toy examples illustrating the Shapley value computation. On the left graph, two acyclic paths exist between i and j, one augmenting the other. On the right graph, alternative paths exist between i and j. Dark arrows represent the traffic request from node i to node j.

$\mathcal{P}(N)$, but only the elements that represent valid paths in which the node participates. In addition, "augmented" paths shall not be considered. Let us consider two paths, P and Q between i and j, such that $Q = P \cup R$. Q is an "augmented" path, since $P \subseteq Q$. For example, let us consider paths $P = (i, A, B, j)$ and $Q = (i, A, C, B, j)$ in Fig. 4.1 (left). Nodes in $R = Q \setminus P$ (i.e., C in the example) do not provide any alternative when a node in P is switched off. Therefore, they should not increase their score for participating in path Q. Note that cyclic paths are special cases of augmented paths, meaning that only acyclic paths are of interest.

More formally, let us now denote by $\mathcal{M}_E(\{i,j\})$ the set of all acyclic paths between i and j in G, and let K_{ij} denote the cardinality of this set. For each path p, we denote by $\pi(p)$ the unordered set of nodes composing p. For instance, $\pi((A,B)) = \pi((B,A)) = \{A,B\}$. Let us also denote by $\mathcal{P}_k(\mathcal{M}_E(\{i,j\}))$ the set composed by all the combinations of the union of k paths in $\mathcal{M}_E(\{i,j\})$. Let us extend the π notation to a set of paths, by posing $\pi(p) = \pi(p_1) \cup \pi(p_2) \cup \ldots \cup \pi(p_k)$ for a path $p = \{p_1, p_2, \ldots, p_k\}$. The following expression defines the graph-restricted game [50], by introducing a characteristic function for a unanimity game that removes the influence of augmented paths:

$$u_{i,j} = \sum_{k=1}^{K_{ij}} \left(\sum_{p \in \mathcal{P}_k(\mathcal{M}_E(\{i,j\})} (-1)^{k+1} \, u_{\pi(p)} \right) . \tag{4.5}$$

To better understand the rationale behind (4.5), let us consider the toy-case example of Fig. 4.1 (left). First, the set of acyclic paths is composed of $K_{i,j} = 2$ elements: $\mathcal{M}_E(\{i,j\}) = \{p_1 = (i,A,B,j), p_2 = (i,A,C,B,j)\}$. Applying the previous formula, we may express $u_{i,j}$ as:

$$\begin{aligned} u_{i,j} &= u_{\pi(p_1)} + u_{\pi(p_2)} - u_{\pi(\{p_1,p_2\})} \\ &= u_{\{i,A,B,j\}} + u_{\{i,A,B,C,j\}} - u_{\{i,A,B,C,j\}} \\ &= u_{\{i,A,B,j\}} \, . \end{aligned}$$

We may thus neglect the augmented paths and restrict our computations on the set $\mathcal{M}^*_E(\{i,j\}) = \{P \in \mathcal{M}_E(\{i,j\}) : \nexists Q \in \mathcal{M}_E(\{i,j\}) \; Q \subset P\}$. For a given path $p \in \mathcal{M}^*_E(\{i,j\})$, the value of $u_{\pi(p)}$ is equal to 1 for every subset of nodes part of this path, leading to a Shapley value increase proportionally to $t_{i,j}$ and inversely proportionally to the path length. For a path p and a node h, let us define $\mathbb{1}h(p) = 1$ if node h belongs to

p, and $\mathbb{1}h(p) = 0$ otherwise. Denoting by K^*_{ij} the cardinality of $\mathcal{M}^*_E\{i,j\}$, and by $\phi(i,j)$ the Shapley value of the unanimity game $u_{i,j}$, the Shapley value granted to a node h is thus $\phi_h = \sum_{i,j} \phi_h(i,j)$, with

$$\phi_h(i,j) = t_{i,j} \sum_{k=1}^{K^*_{ij}} \left(\sum_{p \in \mathcal{P}_k\left(\mathcal{M}^*_E\{i,j\}\right)} \frac{(-1)^{k+1}}{|\pi(p)|} \mathbb{1}h(p) \right) . \tag{4.6}$$

For the sake of illustration, let us consider the example depicted on Fig. 4.1 (right). If we consider that the traffic matrix only has one non-null element, say $t_{i,j} = 1$, the resulting Shapley value $\phi = (\phi_i, \phi_A, \phi_B, \phi_C, \phi_j)$ is:

$$\begin{aligned} \phi &= \left(\left(\frac{1}{3}, \frac{1}{3}, 0, 0, \frac{1}{3} \right) + \left(\frac{1}{4}, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \right) - \left(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5} \right) \right) \\ &= \left(\frac{23}{60}, \frac{8}{60}, \frac{3}{60}, \frac{3}{60}, \frac{23}{60} \right) . \end{aligned}$$

These values show that the traffic source and destination, i and j, are the most critical nodes, as their Shapley value is maximal. Then comes A, which lies on the shortest path from i to j, and finally B and C are granted the smallest values, as they represent a longer, backup path.

Computing the Shapley value using (4.6) is still computationally intensive when considering a realistic, and hence complex, network scenario. Specific heuristics have been designed and evaluated in [51], which considerably reduce the Shapley value computational complexity, focusing only on paths relevant for the network operation, but still resulting in an accurate computation of the Shapley values. These heuristics leverage on the use of the taboo search [52], and on the limitation of the maximum path length, based on the intuition that very long paths (i.e., greater than the network diameter) are rarely used in real networks, and that path contributions to the Shapley value are inversely proportional to the path lengths (as shown, e.g., in (4.6)).

4.1.3 Results on Real Network Scenarios

Once the Shapley value is computed for every node in the network, it can be used as ranking to determine in which order switch off attempts are executed on nodes (i.e., from the less to the most critical). In this section, we evaluate the tradeoff between QoS and energy savings, comparing the proposed method with other classical node ranking schemes.

To provide a relevant evaluation, we take special care in building a realistic scenario. As far as the network is concerned, we consider the reference topology of an ISP participating in the TIGER2 project, and the corresponding traffic matrix. This network, depicted in Fig. 4.2, represents a portion of the ISP access/metropolitan network segment. The light-shaded nodes (1 to 8) are access nodes, source and destination of traffic

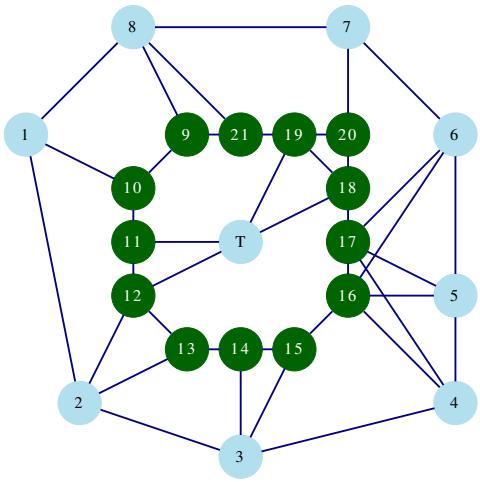

Figure 4.2: The TIGER2 reference topology.

requests, and can not be switched off. The dark nodes (9 to 21) are transit nodes, performing only traffic transport, and can be switched off. Node T is the traffic collection point, providing access to the core network and the big Internet, with whom nodes typically exchange the majority of the traffic.

We adopt the node power consumption model proposed by [38], widely accepted in the literature, and already used for the performance evaluation of the optimal energy-aware routing, in Chapter 3. The power consumption P_i (in Watts) of a node, is related to its switching capability C_i (in Mb/s) according to $P_i = C_i^{2/3}$. Again, since the node switching capability has not been disclosed for the considered scenario, we consider that a node is able to switch twice the capacity of its entire set of connected links. As, in this technological scenario, nodes are responsible for the majority of the network power consumption [43], we focus on methods to switch-off *nodes* and neglect the energy that might be further saved by switching off links (i.e., network interfaces).

The G-Game vs. Other Possible Criticality Rankings

The "criticality" of nodes in a network can be evaluated relatively easily based on the sole topology, or on the sole volume of traffic routed by each node. For what concerns topology based rankings, the most widely used ones are based either on the connectivity of each node (Degree centrality [53]), on the number of shortest paths passing through each node (Betweenness centrality [54]), on the average distance between each couple of nodes (Closeness centrality [55]), or on the importance of nodes neighbors (Eigenvector centrality [56]). For what concerns the amount of routed traffic, we will refer to the LF ranking, proposed by [25], and already described at the beginning of this Chapter.

The above indexes either consider the topology or the traffic, but not both. The Shapley value resulting from the G-Game, instead, takes into account (i) the traffic expressed by the traffic matrix and (ii) the importance of the node in the routing process. The node importance in the topology is evaluated in the G-Game by taking into account the number of paths a node lies on, similarly to the betweenness centrality. However, unlike betweenness centrality, the Shapley value takes into account failure scenarios by

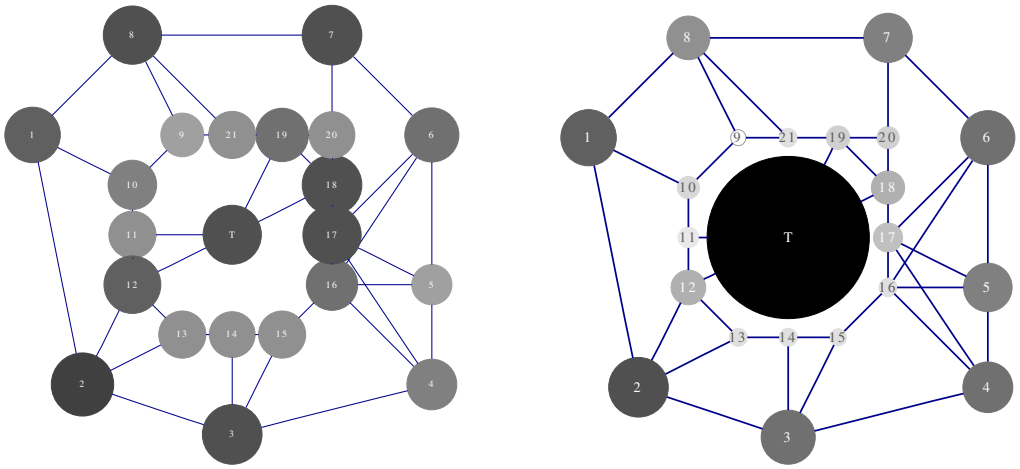

Figure 4.3: Node criticality when considering only the network topology in the G-Game, i.e., G-Game U-TM (left), and when considering also the real traffic matrix, i.e., the full G-Game (right).

considering not only the shortest paths, but also longer paths that can provide alternate paths in degraded scenarios. Note that a scenario in which a node has been switched off to save energy, or the same node fails, are equivalent from the point of view of the routing, and of the G-Game.

All the above listed criticality indexes have been evaluated on the reference network scenario. We also compare two different versions of the Shapley value: (i) a simplified index that reflects only the network topology, considering the G-Game with a uniform traffic matrix, referred to as G-Game U-TM hereafter; (ii) the full G-Game earlier defined, that considers the actual traffic matrix. Fig. 4.3 offers a graphical representation of the difference between the G-Game and the G-Game U-TM: in this representation, the size and color of a node represent its criticality in the considered game (the bigger and the darker the node, the higher its criticality). As expected, the collection point T has the largest worth in the G-Game due the amount of traffic transiting to/from the big Internet, whereas transit nodes $i \in [9, 21]$ have a lower worth as they are interchangeable. As long as the traffic matrix is satisfied, there is no preference among transit nodes.

Recall that, to switch off nodes, we are only interested in the order of criticality among nodes, rather than in the evaluation of the precise values of node criticality. Therefore, to compare the different rankings we compute the Paerson correlation coefficients between every pair of rankings. Results are summarized in Tab. 4.1, where coefficients range from -1 to 1: a value close to 1 reflects a direct correlation (i.e., same order), a value close to -1 reflects an inverse correlation (i.e., inverse orders), and a value close to 0 reflects the absence of correlation. From these results, four families of rankings appear: LF and Shapley value produce singular rankings (i.e., that are not correlated with any other). Most topology-related rankings (Betweenness, Closeness, G-Game U-TM) are similar (correlation ~ 0.9) and are evaluated only through the G-Game U-TM hereafter. Degree and Eigenvalues also form a distinct family which is omitted below as resulting less pertinent, and performing poorly.

Table 4.1: Correlation coefficients between the rankings defined by different criteria.

	G-Game (U-TM)	Betweenness	Degree	Closeness	Eigenvector	G-Game	LF
G-Game (U-TM)	1.00						
Betweenness	0.97	1.00					
Degree	0.46	0.53	1.00				
Closeness	0.87	0.91	0.62	1.00			
Eigenvector	-0.01	0.08	0.73	0.18	1.00		
G-Game	0.41	0.43	0.25	0.51	-0.02	1.00	
LF	0.43	0.49	0.48	0.60	0.19	0.56	1.00

Energy Savings vs. QoS

Energy saving capability is evaluated with respect to the energy-agnostic configuration, in which all nodes are powered on (referred to as "Baseline" configuration). We focus on three different node rankings: (i) the one obtained by the full G-Game, (ii) the one obtained by the G-Game U-TM, based only on topology, and representative of the "topology-related" ranking family, and the (iii) LF ranking, based only on the TM. The resulting orders of nodes are reported in Tab. 4.2.

Table 4.2: List of the network nodes, ordered (left to right) from the least to the most critical one, according to different criticality rankings. Underlined values identify nodes that can be switched off such that the network remains able to carry the traffic matrix. Bold values identify nodes considered in the resource consolidation process.

Ranking	Node ID
G-Game	**9** **15** 13 14 **16** *21* 11 10 20 *19* 17 18 12 8 5 7 4 3 6 1 2 T
G-Game (U-TM)	5 **9** **14** **20** 21 15 *13* 11 10 4 16 19 6 1 12 17 8 T 7 3 18 2
LF	**9** **15** 8 7 5 4 **21** 20 2 3 1 6 14 11 10 *19* *16* 13 12 17 18 T

To evaluate the pertinence of the different rankings, we select a set of nodes that can be switched off by scanning the list sorted by increasing criticality (i.e., safest first). The algorithm examines each node in turn, by checking whether its removal, in addition to nodes previously turned off, would prevent the network from routing the whole TM (by means of a linear program). Nodes that can be switched off, for the different considered orders, are underlined in Tab. 4.2. Notice that nodes that can be switched off are less critical in the G-Game ranking with respect to the LF ranking: hence, they are found earlier during the list scan.

Indeed, the energy saving objective shall affect neither the offered QoS, nor the network robustness. Yet, the greedy switch-off approaches considered so far tend to leave little space to redundancy, and even less means to *control* the redundancy level. An alternative option to control redundancy is to stop the process when reaching a preconfigured target maximum number of switched off nodes, selected by scanning the whole list if necessary.

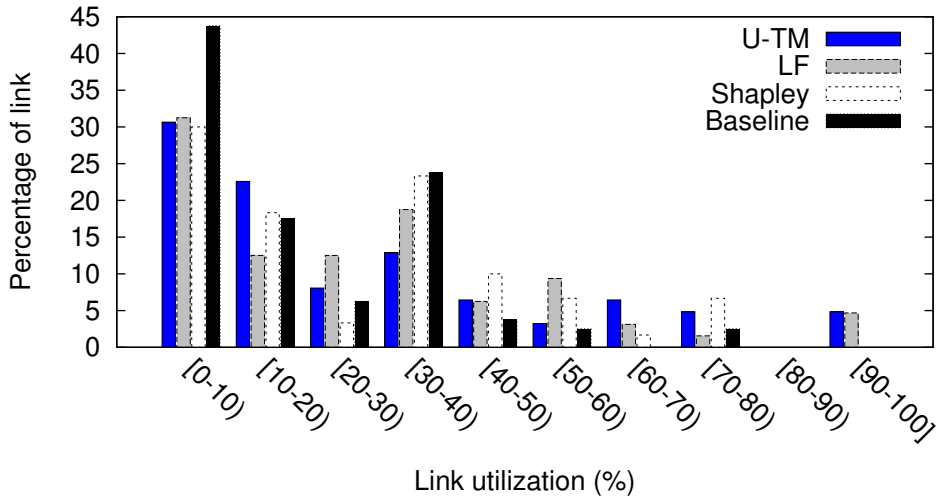

Figure 4.4: Distribution of the link utilization, considering different ranks and in the Baseline configuration.

To evaluate the impact of this strategy on the reference network, let us fix a limit of $N_{off} = 3$ off nodes, so that at most 25% of the transit nodes can be switched off at the same time; the nodes selected by each strategy are those highlighted in boldface in Tab. 4.2. After a network configuration is selected (i.e., after N_{off} nodes are switched off), we compute the link load by routing the traffic matrix on the resulting topology: in more detail, we use TOTEM [57] to perform an optimization of the routing weights (using the IGP-WO algorithm [42]) and route the traffic enabling Equal Cost Multi Path (ECMP). It follows that we are able to not only evaluate topological properties, but also to precisely measure the load on individual links. This is an important point, as the distribution of the link utilization is a very relevant Traffic Engineering (TE) indicator for carriers.

The resulting energy saving is reported in Tab. 4.3, together with the average path length $\bar{l}$; we also report a weighted average path length $\bar{l}_{TM}$, where paths are weighted by the amount of traffic they transport over the TM. The increase of the average path length is a logical consequence of switching off some nodes. Notice that the average path length is minimal for the G-Game, and reduces with respect to the baseline configuration. To get an intuition on how the average path length may decrease by switching off nodes, let consider again the toy case of Fig. 4.1 (right), and suppose for the sake of illustration that traffic shall be shared evenly on paths (i,A,j), and (i,B,C,j), resulting in $\bar{l} = \bar{l}_{TM} = 2.5$ hops. The resource consolidation process may disable one of the two paths, bringing either to an *increase* (i.e., only the (i,B,C,j) path is available, $\bar{l} = \bar{l}_{TM} = 3$ hops), or to a *decrease* (i.e., only the (i,A,j) path is available, $\bar{l} = \bar{l}_{TM} = 2$ hops) in the average path length, depending on which nodes are switched off.

Finally, Fig. 4.4 reports link utilization distributions for the different rankings when $N_{off} = 3$ and the baseline configuration (i.e., $N_{off} = 0$). Notice that the G-Game yield to excellent performances, as the link distribution is roughly equivalent to the one of the baseline configuration, where no node is switched off. Especially, maximum link utilization does not increase under G-Game, with respect to the "all-on" network

Table 4.3: Variation of the average path length (in number of hops) and achievable energy saving, considering different criticality rankings.

Ranking	Off Nodes	Avg. path length $\bar{l}$	$\bar{l}_{TM}$	Energy Saving (%)
G-Game	9, 15, 16	2.45	2.99	17.05
G-Game (U-TM)	9, 14, 20	2.92	3.40	13.43
LF	9, 15, 21	2.64	3.25	16.27
Baseline	None	2.64	3.13	0.0

configuration: this means that energy saving is obtained without compromising the expected QoS. Conversely, some links reach an utilization higher than 90% for the U-TM and LF strategies. The LF strategy results in worse link distribution since it passes through longer alternate paths (i.e., considers only routing paths as in the baseline configuration and ignores fault cases), while the worse QoS results of the U-TM strategy are due to its traffic unawareness (i.e., it takes into account only the topology). In contrast, G-Game explicitly considers existing paths for different node combinations, which means that it explores configurations where some nodes are excluded (i.e., which is precisely what happens when nodes are switched off in the resource consolidation process).

Sensitivity Analysis to Traffic Matrix Variation

To gather consistent results, we consider further variations of the original scenario. The original TM presents high centralization, in the sense that most of the traffic has node T as source or destination. We therefore "smooth" the TM in a controlled fashion, keeping constant the overall traffic volume and number r of traffic demands. In more details, let $x_{s,d}$ denote a traffic request in the original TM, for a given source node s, and destination node d. In the case of a smoothed traffic matrix, every traffic request between access nodes pairs is then equal to:

$$\overline{x} = \frac{1}{r} \sum_{(s,d) \in N} x_{s,d}$$

We can now define a smoothing parameter $p \in [0, 1]$ to tune the traffic between the original traffic matrix ($p = 0$) and the smoothed traffic matrix ($p = 1$). In any intermediate scenario individuated by a given value of p, the elements in the traffic matrix(p) are set to:

$$x_{s,d}(p) = x_{s,d} - p(x_{s,d} - \overline{x})$$

We consider again the case $N_{off} = 3$, and evaluate values of p ranging from 0 to 1 by steps of 0.1. Fig. 4.5 reports, for every value of p the average and maximum link load, for the baseline and G-Game configurations. As we can see, sensitivity to TM variations is limited, with similar maximum link utilization in both cases (i.e., between 60 and 80%). Also the energy savings achieved by the G-Game only minimally vary over the same TM range (i.e., varying between 16.3% for $p = 0$, to 17.1% for $p = 1$. Results are detailed in [51]).

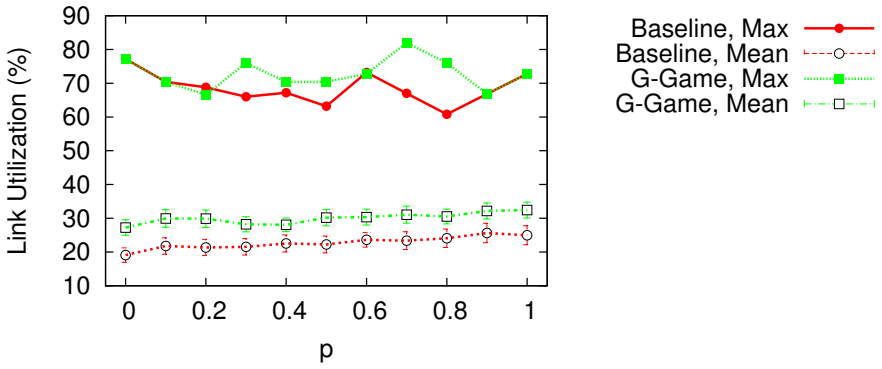

Figure 4.5: Variation of the link utilization distribution, for scenarios ranging from the original TM ($p = 0$) to the smoothed TM ($p = 1$).

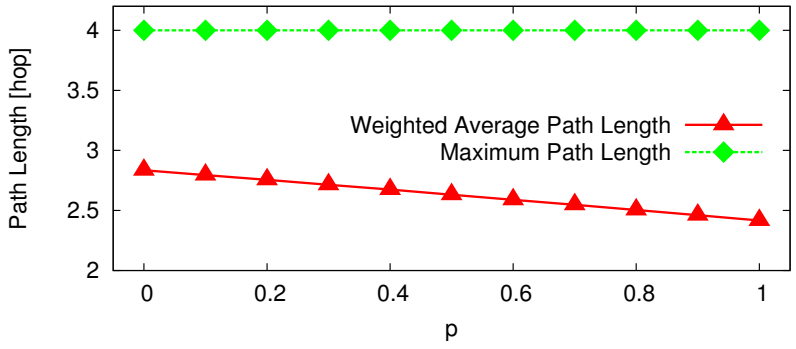

Figure 4.6: Variation of the maximum and weighted average path length, for scenarios ranging from the original TM ($p = 0$) to the smoothed TM ($p = 1$).

Fig. 4.6 reports maximum and weighted average path length for the same set of scenarios. We see that, as the TM smoothness increases, average path length decreases, as nodes tend to exchange more traffic with neighbors. On the other hand, the maximum path length remains constant, and equal to the network diameter (which further confirms the soundness of maximum path length L bound in Shapley value computation).

4.2 Evaluation of the Criticality of Links in Networks: the L-Game

The criticality of nodes in a network scenario may be accurately evaluated by means of the Shapley value, using the procedure described in Sec. 4.1. In a similar way, a criticality ranking may be defined among links in a network scenario, to be used to drive the resource consolidation process, when the devices we intend to switch off are communication links.

4.2.1 The L-Game Definition

Similarly to the G-Game, in order to compute the link criticality, the L-Game models the network as a coalitional TU-Game. The main difference lies in the fact that the players of

the L-Game are the communication links, and not the network nodes. Correspondingly, partitions of the links, i.e., subsets of links that are active while the others are asleep, can be seen as *coalitions*. The *worth* of a coalition is defined as the amount of traffic that can be accommodated by the corresponding network configuration. Links are ranked by the average worth they bring to the different network configurations, by means of the Shapley value.

Again, the benefits of considering the Shapley value as a measure of the importance of links in the network are twofold. (i) First, both traffic and topology aspects are considered, unlike in classical centrality/criticality measurements. In fact, some links can be quite "critical" for the connectivity of the network but some of these critical links might be more important than others considering the amount of traffic that they transport. If a purely topological point of view makes sense for various applications (social networks etc.), we believe that it is not appropriate from a traffic engineering perspective. (ii) Then, the Shapley value is defined as an average value over all possible sub-coalitions. In other words, the algorithm does not consider a single best path (with respect to some static link weights) but many possible alternative paths. The alternate paths may be used to better balance the load in the network, for instance, or in case of failure/switching off of some network devices. All these properties indicate that the Shapley value seems perfectly suited to reflect the tradeoff between saving energy and preserving QoS, in the resource consolidation process.

Using the graph notation already introduced in Section 4.1.1, for any subset $S \subseteq E$, we can define a reduced graph $G_S = \langle N_S, S \rangle$ corresponding to the links of S and where the vertices correspond to either the origin or the destinations of these links. The cardinality of a set S is denoted by $|S|$.

Given a path $p = (i_0, i_1, \ldots, i_k)$ in G, we say that a link $e = \{i, j\}$ *belongs to* p if and only if $\{i = i_s, i_{s+1} = j\}$ for some $s \in \{0, ..., k-1\}$ and we denote by $S(p)$ the set of links which belong to the path p, i.e., $S(p) = \{e \in E : e \text{ belongs to } p\}$.

For what concerns the Shapley value formulation, we use here again the formulation based on the so-called Harsanyi dividends (denoted λ_S for any $S \subseteq E$), as interesting in terms of computation. The Harsanyi dividends are computed recursively as $\lambda_S = v(S) - \sum_{K \subset S} \lambda_K$, being $v(S)$ the worth of the coalition S, as defined in Section 4.1.1. They can be interpreted as the marginal surplus of a coalition (i.e., S) with respect to the sum of the individual worths of smaller coalitions (i.e., $K \subset S$). The Shapley value of a link $e \in E$ is then defined as:

$$\phi_e(v) = \sum_{S \subseteq E : e \in S} \frac{\lambda_S}{|S|} \,. \tag{4.7}$$

4.2.2 On the Efficient Computation of the Link Shapley Value

Let us consider two links e_1 and e_2 that are not adjacent, i.e., $e_1 \cap e_2 = \emptyset$. Then, it is clear that $v(\{e_1\} \cup \{e_2\}) = v(\{e_1\}) + v(\{e_2\})$ and thus $\lambda_{\{e_1\} \cup \{e_2\}} = 0$. As a consequence, in order to compute the Shapley value of a link e, we don't need to consider all the possible coalitions $S \subset E$ but only the elements that represent *valid paths* (i.e., coalitions where adjacency brings "worth") in which the link participates.

Let us also denote by $\mathcal{P}_k(\mathcal{M}_E(\{i,j\}))$ the set composed by all the k-combinations of paths from the set $\mathcal{M}_E(\{i,j\})$, i.e., subsets of $\mathcal{M}_E(\{i,j\})$ composed of exactly k elements. Following the approach introduced in Section 4.1, it can be proved that the Shapley value granted to a link $e \in E$ is $\phi_e(v) = \sum_{i,j \in N} \phi_e(i,j)$, with

$$\phi_e(i,j) = t_{i,j} \sum_{k=1}^{|\mathcal{M}_E(\{i,j\})|} \left(\sum_{P \in \mathcal{P}_k(\mathcal{M}_G\{i,j\})} \frac{(-1)^{k+1}}{|\pi(P)|} \mathbb{1}e(P) \right) \tag{4.8}$$

where $\pi(P) = \bigcup_{p \in P} S(p)$ and $\mathbb{1}e(P) = 1$ if link $e \in \pi(P)$, and $\mathbb{1}e(P) = 0$ otherwise. Note that, again, only acyclic paths are considered in the computation.

As an example of computation using relation (4.8), consider a graph with two paths between i and j, namely $p = (i, A, j)$ and $q = (i, B, C, j)$ (as in Fig. 4.1 (right)). Moreover, suppose that $t_{i,j} = 1$. Then, by relation (4.8) we have that the resulting Shapley values $\phi = (\{i,B\}, \{B,C\}, \{C,j\}, \{i,A\}, \{A,j\})$ are:

$$\begin{aligned} \phi(i,j) &= \left(\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0 \right) + \left(0, 0, 0, \frac{1}{2}, \frac{1}{2} \right) - \left(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5} \right) \right) \\ &= \left(\frac{4}{30}, \frac{4}{30}, \frac{4}{30}, \frac{9}{30}, \frac{9}{30} \right). \end{aligned}$$

These values show that the links composing the shortest path (i.e., {i,A}, and {A,j}) are the most critical ones, as their Shapley value is maximal, while links {i,B}, {B,C}, and {C,j} are granted the smallest values, as they form a longer backup path.

As previously discussed for the case of the G-Game, for the practical computation of the Shapley values, the use of Eq. (4.8) is still computationally intensive considering complex network scenarios. Again, the number of considered paths may be largely reduced, without affecting the precision of the Shapley value computation, and making it tractable even for more complex network scenarios, as introduced in Section 4.1. (i) First of all, it is possible to limit the maximum path length, avoiding to account for very long paths (e.g., longer than the network diameter), which are unlikely used in operational network, and which contribution to the Shapley value are small (i.e., the contribution of each path to the Shapley value is inversely proportional to its length, as it may be seen in Eq. (4.7)). (ii) Secondly, the routers may be considered as hierarchically organized (e.g., access/core), and the routing inside the network complies to the so-called *valley-free* property (using an analogy to the BGP routing in the internet [58]): i.e., traffic travels up the hierarchy once and then down the hierarchy once, but does not go back and forth between the hierarchical levels. This rule avoids traffic to transit via "access" routers. In the FT network depicted in Fig. 4.7, for instance, this rule will insure that access nodes (8,9,..., 12) are not used to forward traffic between core routers 24 and 25.

As usual, once the device ranking has been defined, the resource consolidation process is run. In practice, the algorithm consists in progressively considering all the network links, one by one, for de-activation, following the Shapley ranking. It will be referred to as “L-Game” from now on. When a switch off attempt is executed for a link, the resulting network is analysed, all previously examined links remaining in the proper on/off state. If the resulting network remains able to accommodate the currently required traffic, with a maximum link load lower than a pre-defined threshold (θ), the link is switched

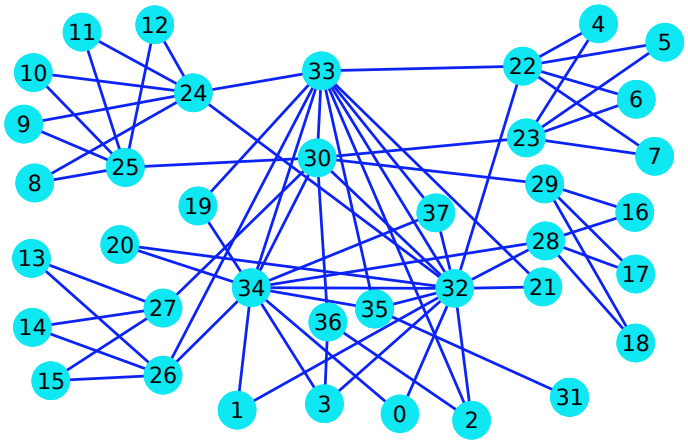

Figure 4.7: Network topology of the FT scenario.

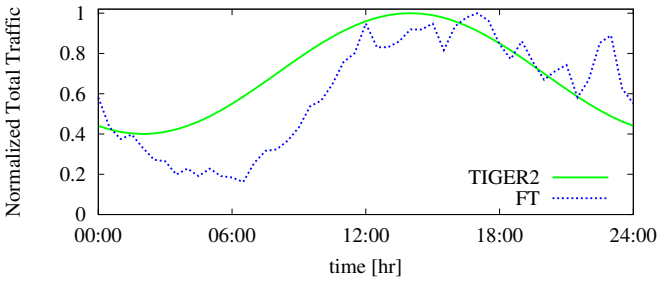

Figure 4.8: Variation of the total traffic load versus time, normalized to the peak total traffic, for the TIGER2 and FT scenarios.

off, otherwise it is left in working state. The process then iterates until all the network links have been examined.

4.2.3 Results on Real Network Scenarios

We compare the Shapley-based ranking obtained thanks to the L-Game, with two other link rankings, introduced in [36], and already described at the beginning of this Chapter: MP and LF. To provide a relevant evaluation of the proposed algorithm, it has been tested over two real network scenarios, namely **TIGER2** and **FT**.

The TIGER2 scenario corresponds to an access/metro segment of a traditional ISP network. The network, whose topology is represented on Fig. 4.2, has already been used in the evaluation of the G-Game, in Section 4.1.3. For this scenario, a single traffic matrix was provided by the ISP, to which we applied a standard night/day trend [6]. The resulting normalized total traffic versus time in the network is reported in Fig. 4.8 by the green solid line. Fig. 4.9 represents the criticality granted by the three considered rankings to the network links (namely, L-Game, LF, and MP)[1].

[1]Note that in the case of the L-Game and LF rankings, the first link subject to a switch off attempt is the one with lower ranking, while for the MP ranking the first link subject to a switch off attempt is the one with higher ranking.

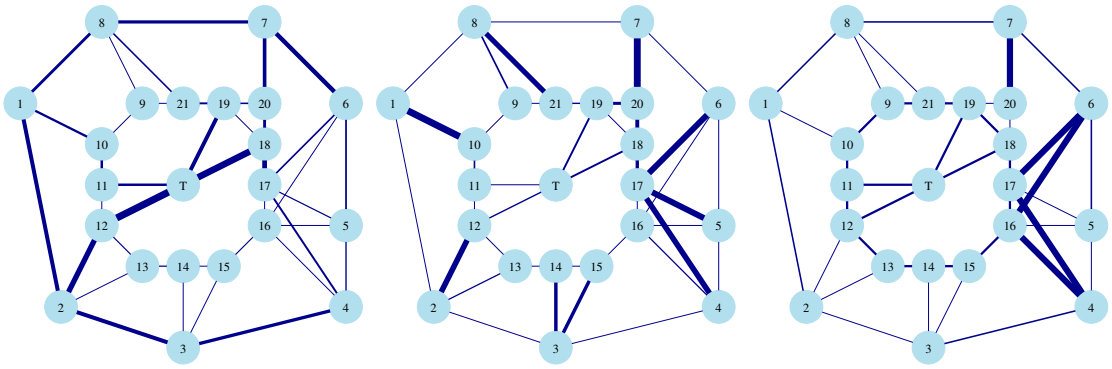

Figure 4.9: Graphical view of the different rankings for the TIGER2 network scenario, for the peak hour traffic: the thicker a link, the higher its importance is with respect to the considered rank (from left to right: L-Game, LF, MP).

Table 4.4: Main characteristics of the reference scenarios.

Parameter	Symbol	TIGER2	FT
Maximum Link Load [%]	θ	75	50
Number of nodes	$\|N\|$	22	38
Number of Links	$\|E\|$	40	72
Average link length [km]	$E[l_{ij}]$	43	278

The FT scenario represents an actual backbone IP network of France Telecom, whose topology is composed by 38 nodes and 72 bidirectional links, as reported in Fig. 4.7. For this scenario, the link capacities and lengths are provided. Finally, the operator disclosed the amount of traffic exchanged among each node pair as foreseen in year 2020 under a conservative assumption of user and traffic growth. A set of 48 TMs has been provided by the operator, representing the complete night/day variation of a typical working day. The total normalized variation of traffic is reported in Fig. 4.8 by the blue dotted line.

Tab. 4.4 resumes the main characteristics of the two reference scenarios. Traffic is routed on the minimum cost paths, considering routing weights inversely proportional to the link capacities. The performance statistics (such as link loads etc.) are averaged over all the TMs (i.e., a 24 hours period). A TM is considered every 30 minutes, over which period, no fast traffic fluctuations are considered, since aggregated traffic flows are evaluated.

In order to evaluate the network energy consumption, we use here the energy model already used in [36, 59], as it is representative for a technological scenario including optical links, along which the signal is eventually regenerated by OEO regenerators. Original power figures have been gathered from measurements performed by an ISP [36]. In this model, each 10 $Gbps$ interface consumes 50 W and amplifiers placed every 70 km consume 1 kW each, for every 10 $Gbps$ of link capacity.

Energy Savings vs. QoS

As performance metrics, we use the (i) *energy saving* on the one side, computed with respect to the configuration in which all links are powered on all time, and the (ii) average

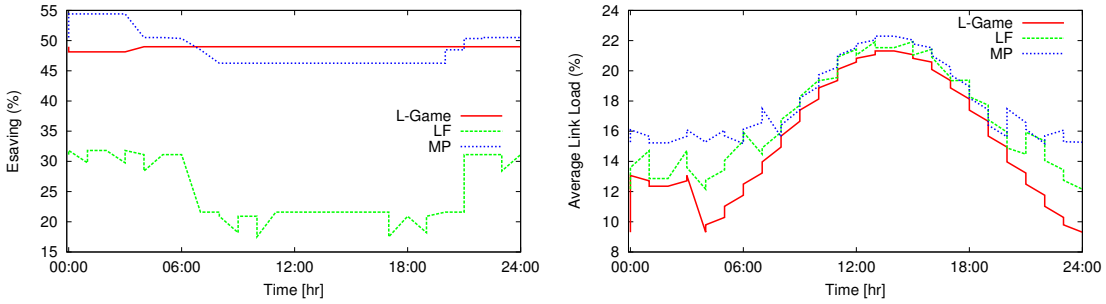

Figure 4.10: Achievable energy saving (left), and corresponding average link load (right), for the TIGER2 network scenario, for different link rankings.

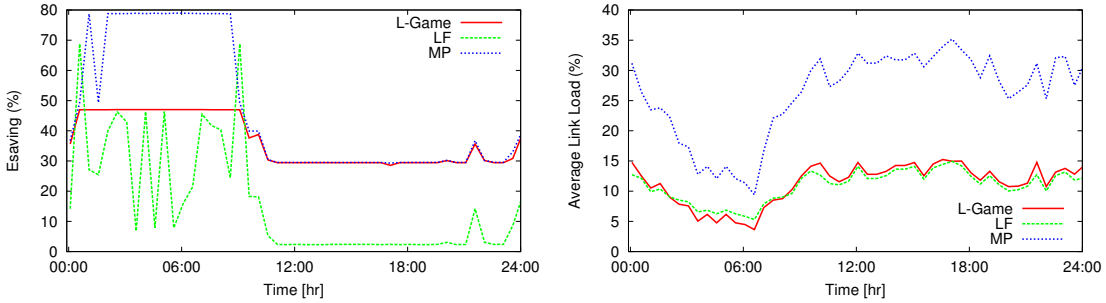

Figure 4.11: Achievable energy saving (left), and corresponding average link load (right), for the FT network scenario, for different link rankings.

link load on the other side, as indicator of the QoS offered by the network. A maximum link load, denoted by θ hereafter, is imposed, as common practice in ISP networks, as minimum QoS guarantee.

We simulated the effect of the resource consolidation algorithm presented above on both scenarios. An ad-hoc simulator in Python language has been developed for the algorithm evaluation. The achievable energy saving when considering the three different rankings, on the one-day simulation period, is reported in Fig. 4.10 (left) for the TIGER2 scenario, and in Fig. 4.11 (left) for the FT scenario. The corresponding average link loads are reported in Fig. 4.10 (right) for the TIGER2 scenario, and in Fig. 4.11 (right) for the FT scenario.

For both scenarios, two main time zones may be identified: a *day* time zone and *night* time zone. The *day* time zone is characterized by higher average link loads, representing the main limitation to the resource consolidation. In the *night* time zone, link loads are much lower, and the network connectivity represents the main limitation to resource consolidation. Of course, the possible energy savings that can be achieved are higher during the *night* time zone. This consideration is true for all the three considered rankings, the difference among which lies in the different trade off they are able to achieve between QoS and energy saving.

As already suggested in [36], MP is able to achieve an energy saving higher than LF, at the price of higher average link load, as it tries to put asleep the most power-hungry links (regardless of their criticality). The L-Game ranking is able to achieve an energy

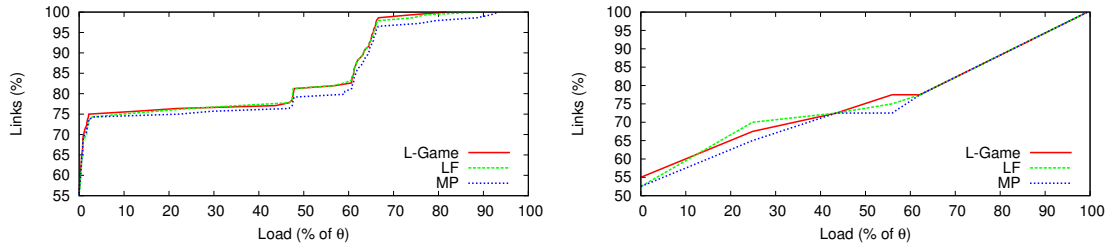

Figure 4.12: Distribution of the link loads for the FT network scenario (left) and the TIGER2 network cenario (right), for different link rankings, for the peak traffic request.

saving comparable to the one obtained by the MP ranking, but using a smaller set of network states[2] with respect to MP (i.e., 2 against 5, in the TIGER2 scenario, and 11 against 16 in the FT scenario), and requiring a smaller number of network reconfiguration (i.e., 2 against 7, in the TIGER2 scenario, and 24 against 35 in the FT scenario, per day).[3] At the same time, the L-Game ranking guarantees the lower link load among the three considered rankings, and the better link load distribution. Fig. 4.12 (left) reports the link load distribution for the FT scenario, while Fig. 4.12 (right) reports the link load distribution for the TIGER2 network scenario. The reported link load distribution are computed for the peak traffic request. As the link load is limited by the threshold θ, link load is normalized with respect to θ.

The L-Game ranking is able to achieve a better QoS performances (with respect to link loads) than the MP one, as MP drives the resource consolidation process keeping into account only the energy cost of links, but not the network topology, nor the traffic flowing on it. At the same time, the LF ranking requires more frequent network reconfigurations than MP and L-Game, as it is based on the link load, which frequently and strongly varies in time, and as the LF ranking only takes into account the amount of carried traffic, in an agnostic configuration, while the L-Game ranking accounts also for the position of links in the network, and for all possible network states.

[2]We refer here to a *network state* as an on/off configuration of the network links: two network states are different if at least a link changes its on/off state passing from one to the other. A transition between two different network states is referred to as a *network reconfiguration*.

[3]Frequently turning on and off links may affect the network routing, since each reconfiguration requires a new convergence transient.

Chapter 5

Distributed Solutions for Energy-Aware Routing

All the previously analysed solutions for the energy-aware routing are completely centralized, and require the perfect knowledge of the traffic matrix, i.e., the amount of traffic sent from each node to each other node, at each given time. These assumptions limit the applicability and deployment of centralized solutions to specific cases, considering current and foreseen network technologies. On the bases of these considerations, distributed solutions have been developed to solve the energy-aware routing problem.

In particular, Ho and Cheung proposed a solution to switch off core routers in backbone networks [60]. This solution is partially distributed to nodes, but still requires at least the presence of a control node, which has the full knowledge of each device status and solves contentions. A fully distributed solution to automatically switch off communication links has been proposed in [61], which by the way does not take into account the traffic flowing in the network. Another distributed solution has been proposed in [62], where entire routers can be switched off when the traffic is low. In the proposed solution, decisions on which links to switch off are taken on the basis of the device ID (i.e., random), and not on the basis of the current device load, moreover, a strict synchronization among nodes is required. All the previously proposed distributed solutions leverage on the OSPF routing protocol [63] to distribute network state information among nodes.

In this chapter, we describe two fully distributed solutions to automatically adapt the network power consumption to the current traffic load. Differently from other works, our solutions do not assume the presence of a central controller, nor the knowledge of the current traffic matrix, nor a strict synchronization among nodes. Extensive simulations that consider real network scenarios have been executed, showing that the proposed algorithms are able to achieve energy savings comparable to the ones of centralized solutions, with little overhead, and while guaranteeing QoS constraints.

5.1 GRiDA: a Green Distributed Algorithm

The GReen Distributed Algorithm (GRiDA), represents a distributed on-line approach to the energy-aware routing problem. It has been designed to automatically adapt the

energy consumption of IP-based networks to their current load, by dynamically switching off/on links, in a fully distributed fashion. In particular, nodes take independent, asynchronous, decisions on the on/off state of incident links, on the basis of the *load* of the links, and of the *learning*, based on past decisions.

The distributed nature of the solution allows it to: (i) limit the amount of shared information, (ii) avoid explicit coordination among nodes, and (iii) reduce the problem complexity. Moreover, nodes are *not* assumed to know the traffic requests to which the network is subject. The algorithm leverages on the use of learning to reduce the number of link reconfigurations, and hence to ease the routing protocol convergence. Moreover, GRiDA is able to react both to traffic variations and to link/node failures, while being able to achieve considerable energy savings, comparable to the ones achieved by the previously analysed centralized solutions.

5.1.1 The Algorithm Description

In order to adapt the network capacity to the current traffic demand, the GRiDA algorithm (i) switches off links whenever they are underutilized, and whenever their absence in the network does not affect the network functionalities, and (ii) switches on idle links when capacity is required to guarantee a proper reaction to faults and changes in the traffic demand. The process of link switching off/on is fully decentralized to the single nodes, which take local decisions at random intervals, without any coordination or synchronization among them. The solution results thus to be more robust and simpler to implement with respect to the centralized approaches proposed so far.

Local decisions are assumed to be based only on the knowledge of the current load and power consumption of node-incident links, and on the knowledge of the current network topology, assured by a link-state routing algorithm, e.g., OSPF or IS-IS. In GRiDA, Link-State Advertisement (LSA) messages distribute information about the current network topology, augmented by information about eventual congestion in the network, i.e., link load overcoming a threshold (ϕ), or presence of disconnected source/destination pairs. LSAs are delivered to nodes at fixed time intervals (Δ_{LSA}), selected by the network administrator, according to OSPF or IS-IS configuration.

The network infrastructure is represented as a di-graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. Vertices represent network nodes, while edges represent network links, being $N = |V|$ and $L = |E|$ the number of nodes and links respectively.

The Node Choice

A decision of a node n corresponds to entering a specific node *configuration* $K^{(n)} \in \mathcal{K}^{(n)}$, where $\mathcal{K}^{(n)}$ is the set of all possible configurations for node n; a configuration $K^{(n)}$ is a combination of on/off states for incident links. More formally, given a node n, of degree $d^{(n)}$, and an ordered list of the incident links (in lexicographical order), a configuration is the vector $(k_1^{(n)}, \ldots, k_d^{(n)})$ of the configurations of the $d^{(n)}$ incident links. The configuration $k_l^{(n)}$ of a link l is a binary variable indicating the state of the link: $k_l^{(n)} = 0$ if the link is powered off, and $k_l^{(n)} = 1$ if the link is powered on. Therefore

$|\mathcal{K}_n| = 2^{d^{(n)}}$.

The *status* S_n of a node n is the vector $(s_1^{(n)}, \ldots, s_d^{(n)})$ of the status associated to all the $d^{(n)}$ links incident to n. For each link l the status $s_l^{(n)}$ may assume 2 possible values, defined on the bases of the link load ($\rho^{(l)}$): *off*, i.e., the link is powered off or powered on but not used ($\rho^{(l)} = 0$), or *normal*, i.e., the link is used ($\rho^{(l)} > 0$).

A *utility* function is defined as: $U(K^{(n)}, S^{(n)}) = c(K^{(n)}) + p(K^{(n)}, S^{(n)})$, where $c(K^{(n)})$ is the power consumption of node n computed as the sum of the power consumed by the link in on state in configuration $K^{(n)}$, normalized to the sum of the power of all incident links (i.e., $c(K^{(n)}) = \frac{\sum_{l=1}^{d^{(n)}} k_l^{(n)} c^{(l)}}{\sum_{l=1}^{d^{(n)}} c^{(l)}}$, where $c^{(l)}$ is the power consumption of link l, see Sec. 5.1.2 for details), and $p(K^{(n)}, S^{(n)})$ is a *penalty* associated to the configuration on the basis of the current *status* and the *learning*. Since the same procedure is applied to all nodes, from now on we get rid of the index n for ease of notation.

For a single node, the problem turns into selecting the best configuration that minimizes the utility function, while guaranteeing the global system to work properly. This problem can be solved by the support of the *Q-learning* technique [64], as the node choice is a function of the current state of the same node, and each possible choice is associated to an estimated utility function, updated by learning. Hence, node decisions in normal network working state correspond to the K minimizing $U(K, S)$. To ensure *fast reaction* to faults and sudden traffic changes, three safety mechanisms have been introduced:

- a *connectivity check* is performed on the network topology resulting from the chosen configuration, through a breadth-first search. This means that GRiDA always ensures full connectivity among nodes (and it guarantees that at least one link per node is always powered on). If a choice would lead to a network disconnection, it is not applied and its penalty is updated with the additive factor β as if a violation occurs (detailed in the following of this Section);
- if a choice taken in a non-congested network state is followed by a congestion advertised by an LSA, the choice is regretted, i.e., the node returns to the previous configuration, and the penalty corresponding to that choice is updated with an additive factor β;
- in a congestion network state, a node which is taking a decision will automatically select the *all-on* state. This choice is not subject to the regret mechanism dependent on the following received LSA.

The Penalty Evolution

The values of $p(K, S)$ are updated step-by-step, on the basis of the *learning*: if the decision of entering configuration K when in status S is followed by an LSA reporting a network critical state, the cost associated to that choice (i.e., $p(K, S)$) is incremented by an additive factor β (≥ 0):

$$p(K, S) = p(K, S) + \beta \tag{5.1}$$

Note that the power consumption is normalized when considered in the utility function, so that β has the same impact on each node, independently from its power consumption

```
Node Choice
Input: K^old, S
Output: K, K^old, S^old
  S^old = S
  if lastLSA == OK:
    K* = min_K U(K,S)
    for J in 𝒦:
      p(J,S) = p(J,S) · δ
    if (connectivity_check(K*) == OK):
      K = K*
      if K ≠ K^old:
        to_be_checked = TRUE
    else
        p(K*,S) = p(K*,S) + β
  else:
    K = all_on configuration
```

Alg. 1: The pseudo-code of the *node choice* event.

absolute value. Note also that the penalty is increased by β when the connectivity check fails.

If a decision is taken in a state S and no violations have been reported by the previous LSA, the penalty associated to choices in state S (i.e., $p(*, S)$) are decremented by a multiplicative factor $\delta \in [0, 1]$:

$$p(J, S) = p(J, S) \cdot \delta \quad \forall J \in \mathcal{K}_n \tag{5.2}$$

The pseudo code resuming the node decision process is reported in Alg. 1, where S is the current state of the node.

Intuitively, (5.1) penalizes choices which likely caused violations of connectivity or capacity constraints; (5.2) pushes nodes toward the *exploration* of all the possible choices by reducing the effect of the accumulated memory, since the factor δ is applied to all $p(*, S)$.

The pseudo code describing the procedure executed by nodes at LSA arrivals, and the corresponding penalty updates for choices which brought to violations, is reported in Alg. 2, where K is the current node configuration, K^{old} is the node configuration before the last choice, S^{old} is the node status at the time the last choice has been taken, and p is the penalty state of the node.

Algorithm Initialization

In order to speed up convergence, the cost function $p(K, S)$ is properly initialized. The intuition is to discriminate between (i) switching off an unloaded link, or (ii) switching off a link which is carrying traffic (which can be less safe for the network functionality). In addition, we need to avoid multiple attempts of radical switching off choices during convergence by further penalizing configurations with an higher number of off links and link loads larger than zero.

```
LSA Arrival
Input: K, K^old, S^old, p
Output: K, p
  if to_be_checked == TRUE:
    if LSA != OK:
      p(K, S^old) = p(K, S^old) + β
      K = K^old
    to_be_checked = FALSE
```

Alg. 2: The pseudo-code of the *LSA arrival* event.

More formally, an initial penalty function $\theta_l(k_l, s_l)$ is associated to each link l in each possible status $s_l \in S$ entering each possible configuration $k_l \in K$:

$$\theta_l(k_l, s_l) = \begin{cases} 0 & s_l = \text{off} \ \vee k_l = 1 \\ 1/d & else \end{cases} \tag{5.3}$$

The $\frac{1}{d}$ factor is a normalization over the node degree.

Then, the penalty $p(K, S)$ is initialized to $\sum_{l \in n} \theta_l(k_l, s_l)$. The procedure is repeated for all nodes $n \in V$, and for all configurations $K \in \mathcal{K}$ and all status $S \in \mathcal{S}$.

The Solution Complexity

Making a choice at a given node implies the following 3 steps: (i) access to the penalties corresponding to the current status, (ii) computation of the utility corresponding to all the possible configurations from the current state, and (iii) executing a connectivity check.

First, we analyse the *time complexity* of our solution. The first step implies, in the worst case, to find the correct memory entry among all the 2^d possible state, which can be done through a binary search tree in a time $O(\log 2^d) = O(d)$. The computation of the utility function is simply the sum of the penalty and of the energy cost of the d incident links, which should be computed for all the 2^d possible configurations, resulting in a time $O(d2^d)$. Finally, the connectivity check on the chosen configuration results in a time $O(N + L) = O(N + dN) = O(dN)$, considering a breadth-first search. Summing the contribution of the 3 steps, the time complexity of the solution results $O(d2^d + dN)$, scaling linearly with network size N, and exponentially with node degree d. As it has been shown in [53], the node degree is actually limited in real network scenarios, thus it does not represent a critical issue.

For what concerns the solution *space complexity*, instead, a node needs to store in the worst case, for each possible status, a penalty for each possible configuration, resulting in a matrix of $2^d \times 2^d = 4^d$ memory entries. Actually, simulation results show that less than 10% of the entries are visited on average (see Sec. 5.1.2), and hence just a minimal amount of memory is required. Thus, rather than storing the entire matrix, compact structures may be adopted to reduce the size of the matrix.[1]

[1]Notice that also the initial penalty matrix θ can be efficiently stored in a compact format since it is only based on the set of available configurations and states.

5.1.2 Results on Real Network Scenarios

To provide a relevant evaluation of the described algorithm, we tested it over 3 different scenarios, ranging from a metropolitan segment network to a European-wide network.

Scenario Description

TIGER2: The first testing scenario is an access/metropolitan segment of a traditional telecom operator network [51]. It has already been used for the evaluation of centralized solutions in Chapter 4. The network topology is reported in Fig. 4.2.

For this scenario, an actual traffic matrix has been provided. The maximum link utilization is guaranteed to be smaller than 70% ($\phi = 0.7$), and 47 traffic matrices have been generated applying the sinusoidal traffic profile described in [65], and represented in Fig. 5.1 (left) by the green dashed line labeled "TIGER2".[2]

Geant: We consider the actual Geant Network [35], whose topology is reported in Fig. 3.2. It has been already used in the performance evaluation of the optimal solution, in Chapter 3. For this network topology, actual traffic matrices are publicly available, among which we selected the 48 traffic matrices of 05/05/2005 (a typical working day). The corresponding variation in terms of total traffic load is reported in Fig. 5.1 (left) by the red continuous line.

Italian ISP: Finally, we considered a topology inspired by the national network of an ISP (see [65] for details). This scenario is referred to as "Italian ISP", and its network topology is reported in Fig. 5.1 (right). It is a hierarchical network composed of 373 nodes, organized in 5 levels: core, backbone, metro, access and Internet nodes. The *core* level is composed by few nodes densely interconnected by high-capacity links, and offering connectivity to the Internet by means of a peering node. Going down in the hierarchical levels, the number of nodes increases, and the link capacity decreases.

The access nodes and the Internet peering node are sources and destinations of traffic. The traffic requests for this topology have been generated following a measured traffic profile (reported in Fig. 5.1 (left) by the blue dotted line), as described in [65].[2]

The Power Model: We are interested in the power consumption related to links which includes the power consumption of the router linecards and of the amplifiers/regenerators along the link. The power model introduced in [65] is used here, as representative of current actual devices. We consider ports consuming $c^{nic} = 50\,\mathrm{W}$ for each $B^{ref} = 10\,\mathrm{Gbps}$ of link capacity, and amplifiers consuming $c^{a} = 1\,\mathrm{kW}$ for each $B^{ref} = 10\,\mathrm{Gbps}$ of link capacity, with an amplifier every $m^{a} = 70\,\mathrm{km}$. Therefore, the power consumption $c^{(l)}$ of a link l, with capacity $B^{(l)}$ and length $m^{(l)}$, results in: $c^{(l)} = \lceil \frac{B^{(l)}}{B^{ref}} \rceil (\lfloor \frac{m^{(l)}}{m^{a}} \rfloor c^{a} + 2c^{nic})$.

The Traffic Model: In our simulations, traffic requests are constant over fixed time intervals Δ_{TM}, after which a new traffic matrix is considered. Traffic is expected to change on moderate time scale, so that $\Delta_{TM} = 30$' or higher (aggregated traffic flows are considered). The traffic matrices have been obtained from direct traffic measurements

[2] This is a national ISP network, where all nodes are in the same timezone.

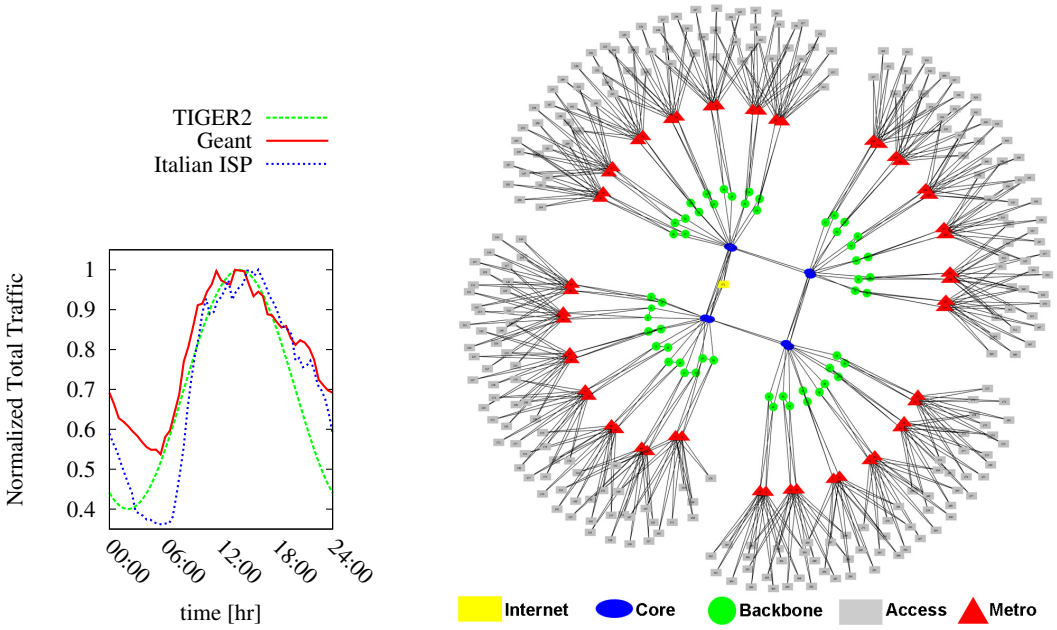

Figure 5.1: Variation of the total traffic load versus time, normalized to the peak total traffic, for the 3 simulation scenarios (left), and the Italian ISP network topology (right).

where available; otherwise, they are computed starting from a single measured traffic matrix and imposing a synthetic traffic profile.

Parameter Setting

A new TM is considered every time interval Δ_{TM}. A randomly selected node is waken up to take a decision every random interval Δ_c, uniformly distributed between the LSA interval Δ_{LSA} and $\Delta_{c,Max}$. Time intervals must be chosen in order to have, on the one hand, at least one LSA occurrence between two consecutive decisions, and on the other hand, a significant number of decisions per node to allow algorithm convergence. Note that LSA timings are compliant with current OSPF specifications [63]. On average, a single node takes a decision every $\Delta_c \times N$, where N is the number of nodes in the network.

Values for the parameters in the different simulation scenarios are summarized in Tab. 5.1. The number of nodes for the *Italian ISP* network reflects the fact that core, backbone, and metro nodes are running the GRiDA algorithm, while access and Internet nodes are not running the GRiDA algorithm. Indeed, access and Internet nodes in the *Italian ISP* network are not connected among them, hence, all the links in the network are still considered also when GRiDA is not run on those nodes.

Performance Evaluation

We implement GRiDA on a custom event-based simulator written in python and C languages. Node choices, LSA arrivals and traffic matrix changes are the possible events. Moreover, network statistics including link load, node configurations and power consumption are stored in a log. Unless otherwise specified, we simulate a time period of one week, by repeating the set of traffic matrices. In the following, we first analyze

Table 5.1: Simulation parameters for the 3 simulation scenarios.

Parameter	TIGER2	Geant	Italian ISP
Δ_{LSA} [s]	5	5	5
Δ_{TM} [min]	30	30	48
$\Delta_{c,Max}$ [s]	25	25	50
N	22	23	112 + 261
δ	1.0	0.999	0.9
β	50	50	1
ε	50	50	50
ϕ	0.7	0.7	0.5
Choices / Node / Traffic Matrix	5.5	5.2	0.9

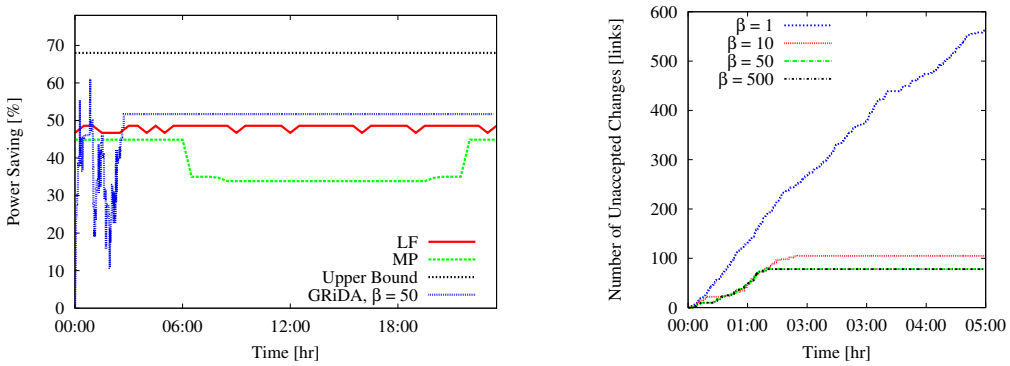

Figure 5.2: TIGER2 network: (left) Power saving versus time, considering different algorithms, (right) cumulative number of unaccepted changes for different β.

the transient behavior of the algorithm on the different scenarios. As a second step we consider the sensitivity of average performance metrics to parameter settings.

Transient Analysis

We start by evaluating the performance of GRiDA on the *TIGER2* scenario. We set $\delta = 1$ for testing the convergence of the algorithm. We then compare the power saving of GRiDA against the upper bound obtained solving the optimal problem formalized in Chapter 3 for the off-peak traffic, and the centralized LF and MP heuristics described in Chapter 4. A perfect knowledge of the TM is assumed for the optimal problem and centralized heuristics.

Fig.5.2 (left) reports the power-saving versus time of GRiDA, LF, MP and the upper bound. It reports the power saving computed as the percentage of saved power with respect to a configuration in which all links are powered on. Since the LF and MP heuristics are centralized and require the knowledge of the TM, we run them at every TM change. After an initial transient, the power saving of GRiDA is constant: this is due to the fact that $\delta = 1$ and the network is largely over-provisioned; thus the algorithm converges to a solution that does not involve any increment in the penalty function. Interestingly, GRiDA outperforms both the LF and MP heuristics, saving 52% of power

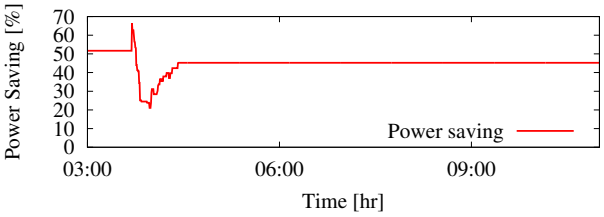

Figure 5.3: TIGER2 network: Power saving with a fault occurring after convergence is reached.

after convergence.

To give more insight, Fig.5.2 (right) reports the cumulative number of link reconfigurations due to network violations, for different values of β. For $\beta > 1$ reconfigurations occur only during the initial transient. To this extent, low values of β result in a large number of reconfigurations, since the learning rate of the algorithm is lower. The intuition suggests that in this case the predominant term in the utility function is the power consumption, thus each node always selects the most aggressive configuration in term of power savings, resulting in a large number of violations. On the contrary, when $\beta > 1$ the number of reconfigurations steadily decreases. Thus, a trade off emerges among responsiveness of the algorithm and number of reconfigurations.

We now evaluate the performance of GRiDA under *anomalous network conditions*. In particular, a node failure is simulated after convergence of GRiDA. Fig.5.3 reports the power saving before and after the failure event. GRiDA is able to wisely adapt to a new configuration with only 9 reconfigurations due to network violations. In fact, as soon as the failure is detected, GRiDA starts turning on links given LSAs reporting network anomalies. Then, the algorithm starts again to switch off links until a stable configuration is reached. While GRiDA has not been designed to explicitly handle failures, it helps the failure management algorithm to recover from critical conditions.

We consider now the *Geant* topology with parameter set reported in Tab. 5.1. Fig. 5.4 (left) reports the power saving versus time. Also in this case GRiDA outperforms both the LF and MP centralized heuristics. Notice that here we set $\delta = 0.999$, thus GRiDA does not converge to a stable solution, since the penalty costs are decreasing with time.[3] This allows GRiDA to adapt the power saving to the actual traffic. Fig. 5.4 (right) reports instead the cumulative number of reconfigurations. Also in this case, the number of unaccepted changes decreases as β increases.

Finally, the *Italian ISP* topology is considered. In this case, we have taken as reference the optimal solution as formulated in [36] (i.e., taking advantage of the particular network structure), solved for each TM,[4] and the MP-MP and LF-LF centralized heuristics, which has been proven in [25,65] to be the most effective ones for this topology. In particular, both MP-MP and LF-LF try to switch off first all the links incident to a node (which are sorted according to a MP or LF criterion, respectively). Then, as a second

[3]Results presented in [59] show that a stable solution is reached also in this scenario setting $\delta = 1$.

[4]The solution has been obtained running CPLEX on a high performance cluster hosted in the Politecnico di Torino Campus [66].

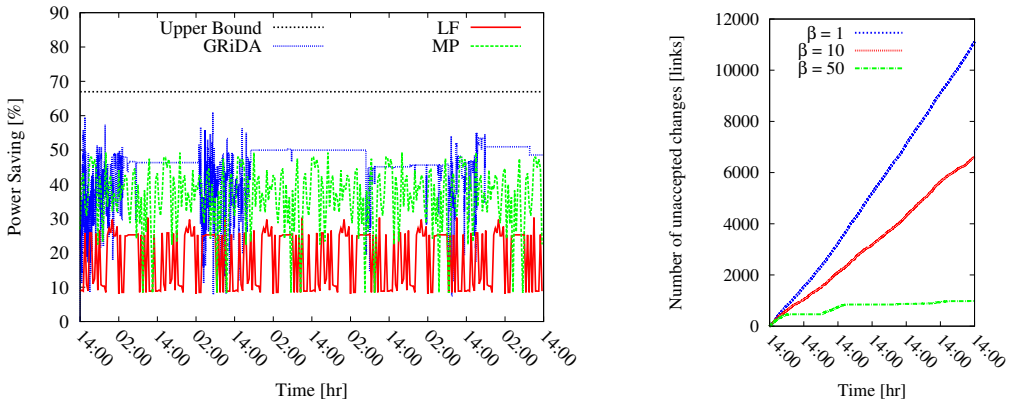

Figure 5.4: Geant network: (left) Variation of power saving versus time, (right) Cumulative number of unaccepted changes.

step, the remaining links are eventually powered off individually (according to a MP or LF ordering). The two algorithms are further detailed in [25, 65].

Fig. 5.5 (top) reports the algorithm comparison in terms of power saving. Interestingly, saving follows a strong day-night trend for all algorithms. In particular, more power saving is possible when the network is lightly loaded, i.e., during night. In this case, GRiDA is able to save an amount of power comparable to centralized heuristics, but without requiring the knowledge of the current TM. Moreover, the variability of the traffic impose GRiDA to quickly adapt the configurations. To give more insight, Fig. 5.5 (bottom-left) reports the average and maximum link load in the network running GRiDA. Average link loads are computed for each TM. Interestingly, during night-time the maximum link load is below 30%, i.e., far from the load threshold $\phi = 0.5$. This suggests that the connectivity constraint is stricter than the maximum load constraint. During high traffic periods, some link loads actually gets close to 0.5. Indeed, some violations are present, even if of short duration and of small intensity. We will quantify violations better in the remaining of the section. Moreover, the average link load is always lower than 10%, suggesting that most links are lightly loaded even when GRiDA is run over the network.

Finally, Fig. 5.5 (bottom-right) reports the average number of OFF-ON and ON-OFF link choices per node, per Δ_{TM} interval, in the network running GRiDA. Note that here we are accounting also the link reconfigurations triggered by a negative LSA. The figure reports also the average node degree $\frac{L}{N}$. Interestingly, GRiDA tries to turn off on average less than one link per node every Δ_{TM}. On the contrary, during the morning GRiDA quickly reacts to traffic increase, and about two links per node are powered on, per Δ_{TM} interval, to increase the network capacity.

Average Performance and Sensitivity Analysis

We now investigate how the parameter settings impact the performance of the algorithm. In this case, we consider only the Italian ISP scenario since it is the largest one in terms of nodes and links. We evaluate the performance considering the following metrics: *energy saving*, *number of unaccepted choices*, and *network overload*. The intuition is to have

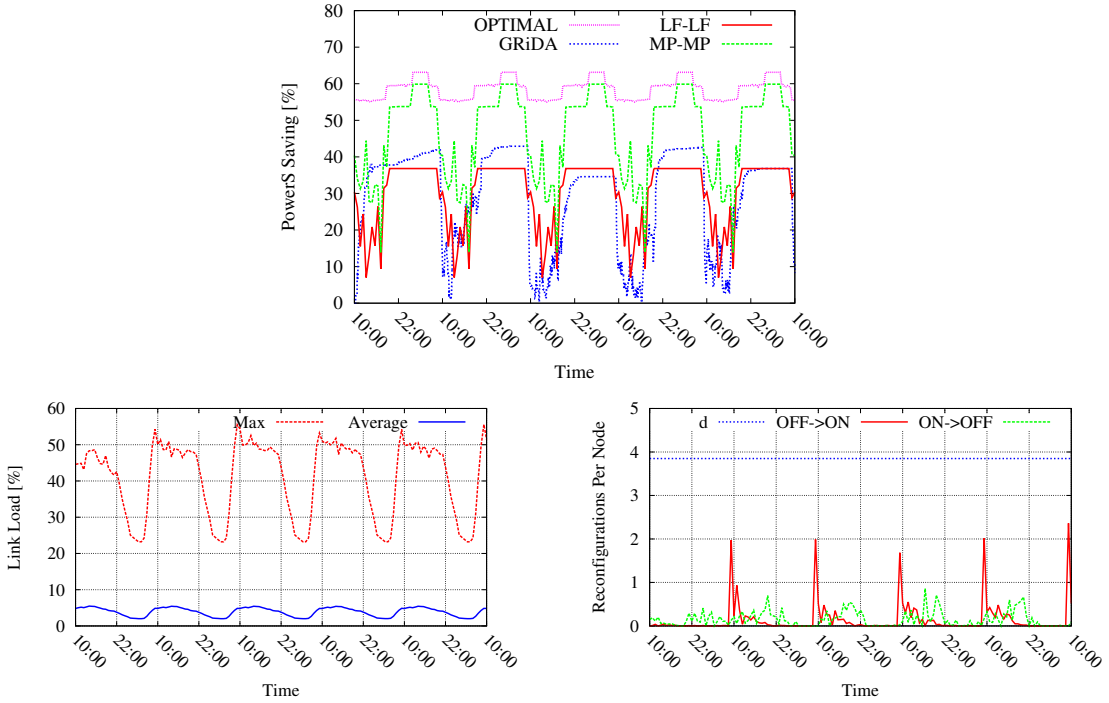

Figure 5.5: Italian ISP network: (top) power saving, (bottom-left) link load, (bottom-right) OFF⇒ON and ON⇒OFF events per Δ_{TM} per node.

a set of metrics to quantify the gains from saving while monitoring QoS for users. In particular, energy saving is evaluated as the integral of link power saving over a one week long time interval. Unaccepted choices account the percentage of switch off choices which are undone due to the immediate critical state indication by LSA. Percentage is computed with respect to the total number of switch off attempts. The network overload is defined as the fraction of traffic exceeding the load threshold ϕ with respect to the total carried traffic, i.e.:

$$\xi = \frac{\int_t \sum_{l \in E} \max(\rho^{(l)}(t) - \phi, 0) dt}{\int_t \sum_{s,d \in V} r^{sd}(t) dt} \tag{5.4}$$

where $r^{sd}(t)$ is the traffic request from node s to node d at time instant t. This is a relative indicator for the network congestion level, averaged over the simulation period, accounting for the *number of load violations*, their *entity*, and their *duration*. Note that we refer here to *violations* for link load overcoming the ϕ threshold, i.e., 50% or 70% of the link capacity, in the considered network scenarios. Link load never overcomes the full link capacity, i.e., 100%, in the considered scenarios. Let us consider, e.g., an average of 20 violation occurrences per hour, with $\phi = 50\%$, each one lasting 20 seconds on average, each one corresponding to a load of 5.5 Gbps over 10 Gbps links (i.e., $l = 55\%$). This will correspond on the considered network to an overload in the order of magnitude of e-3.

Learning Update We first evaluate the impact of the δ parameter. Intuitively, this multipicative parameter affects how much the past choices impact the current decision, i.e., if $\delta = 0$, the penalty function is reset to 0 for the current state, every time that a

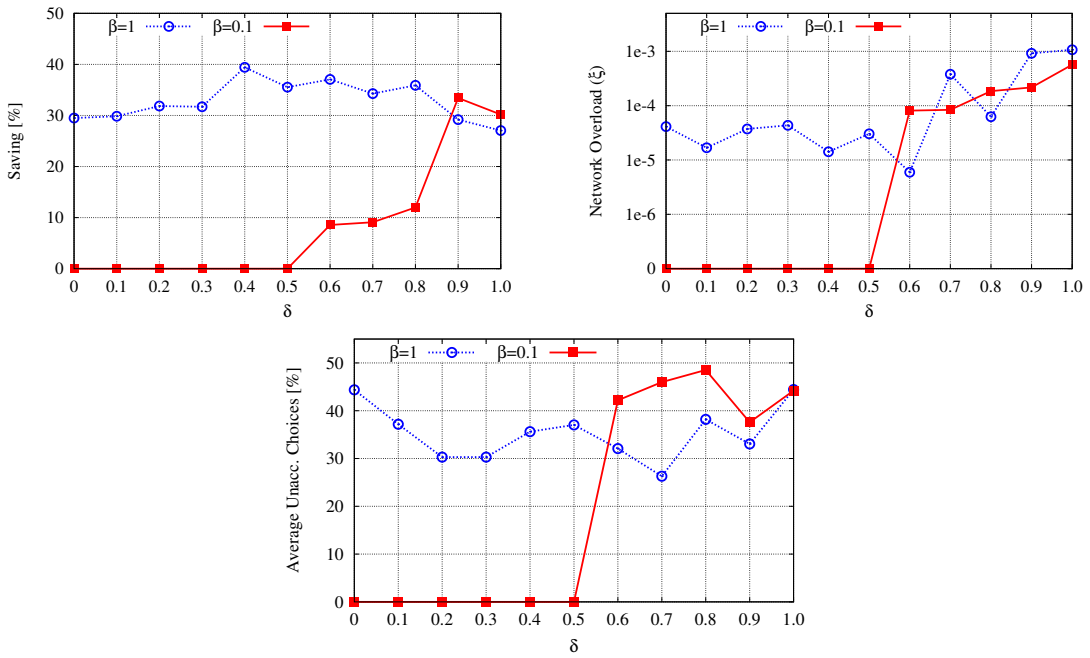

Figure 5.6: Italian ISP: Impact of δ: (top-left) Saving, (top-right) Network Overload, (bottom) Unaccepted choices.

positive LSA is received, while if $\delta = 1$, penalties obtained by learning are kept forever. Fig.5.6 (top-left) reports the average link power saving for $\delta \in [0, 1]$ and different values of β. Interestingly, with $\beta = 0.1$ the saving rapidly increases for increasing δ. In particular, saving is 0 if $\delta \leq 0.5$. This is due to the fact that the penalty β is not strong enough to choose a candidate configuration different from the all-off one, which is then undone since the connectivity check fails. Let us explain better this behavior with an example, supposing that a generic node n is running GRiDA. During the first choice of n, the "all-off" configuration is selected, since it is the most convenient in terms of energy, i.e. $c(K) = 0$. The penalty function is zero (we neglect the impact of θ_l). This causes the connectivity check to fail and consequently the penalty function is updated to β. During the following choice of n, the minimum utility function is recomputed. If the all-off configuration is still the most convenient one, the connectivity check fails again and its associated penalty becomes $\beta + \beta\delta$. After Z iterations with failed connectivity check the utility function for the all-off configuration becomes:

$$U(K, S) = p(K, S) = \sum_{z=1}^{Z} \beta\delta^z = \beta\frac{1 - \delta^{Z+1}}{1 - \delta} \tag{5.5}$$

This happens until the best current utility is lower than the utility function with at least one link on, i.e., $U(K, S) < U(K_{min}, S)$, with $U(K_{min}, S) = p(K_{min}, S) + c_{min} = c_{min} = \min_{J \in \mathcal{K}}\{c(J) | \sum_{i=1}^{d} j_i \geq 1\}$. Thus, only if $p(K, S) > c_{min}$ a different configuration is tested. For $Z \to \infty$, the algorithm selects another configuration different from the all-off one if and only if $\frac{\beta}{1-\delta} > c_{min}$. In our case, for $\beta = 0.1$, $\delta > 0.5$ is necessary.

Fig.5.6 (top-left) reports also the curve for $\beta = 1$. Saving is averaged over a one week interval. The maximum error for saving is 3% with 95% of confidence. In this case, the initial penalty β is strong enough to let succeed the connectivity check, and savings between 25% and 40% are achieved. However, savings depend on δ also in this case.

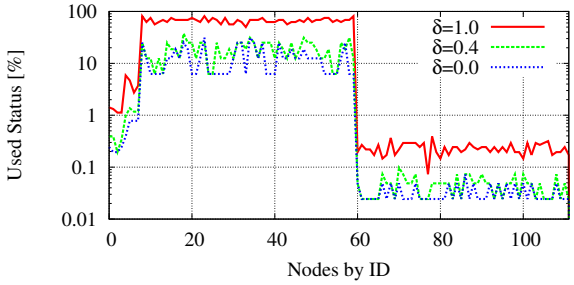

Figure 5.7: Italian ISP: Impact of δ on the status exploration.

Table 5.2: Italian ISP: algorithm comparison, with $\Delta_{LSA} = 5\,\text{s}$ and $\Delta_c = 50\,\text{s}$

Algorithm	Saving [%]	Unacc. Choices [%]	ξ
OPTIMAL [36]	58.56	N.A.	N.A.
MP-MP [65]	46.28	N.A.	N.A.
LF-MP [25]	30.24	N.A.	N.A.
GRiDA,$\beta = 0.1$	30.18	0.44	5.67e-04
GRiDA,$\beta = 1$	29.18	0.33	9.18e-04
GRiDA,$\beta = 10$	25.38	0.42	1.00-03
GRiDA,$\beta = 100$	26.88	0.42	1.10e-03
GRiDA,$\beta = 1000$	25.75	0.44	1.14e-03

Fig.5.6 (top-right) and Fig.5.6 (bottom) report the network overload and the percentage of unaccepted choices, respectively. Interestingly, both metrics are minimized for intermediate values of δ, i.e., when the algorithm trades between full knowledge of past learning ($\delta = 1.0$) and power consumption ($\delta = 0.0$). Interestingly, network overload is always extremely small, i.e., typically smaller than 10^{-4} with $\beta = 1$, suggesting that GRiDA is very effective in limiting the amount of traffic rerouted over congested links, with more than 60% of choices that are accepted over one week.

Fig. 5.7 reports the percentage of the explored states over all the possible ones for each node running GRiDA. As expected, for $\delta = 1.0$ the percentage of exploration tops 90%. This value is reached by backbone nodes that are connected by few links whose states change quite frequently. On the contrary, when $\delta = 0.0$, the percentage of exploration is below 30%, confirming that the network reaches a stable configuration which does not involve frequent changes of the node states.

Penalty Update We now evaluate the impact of β. In particular, we keep $\delta = 0.9$. Tab. 5.5 reports the average performance metrics. Interestingly, the best results are obtained with lower values of β, suggesting that larger β tend to penalize both power savings and overload, since frequent reconfigurations occur.

Finally, the table reports also the optimal power saving and the performance metrics for the MP-MP and LF-LF heuristics, showing that GRiDA saves a comparable amount of power without requiring the knowledge of the actual TM, nor a centralized coordination or synchronization.

Choice Interval We look at the sensitivity of GRiDA to the time intervals at which

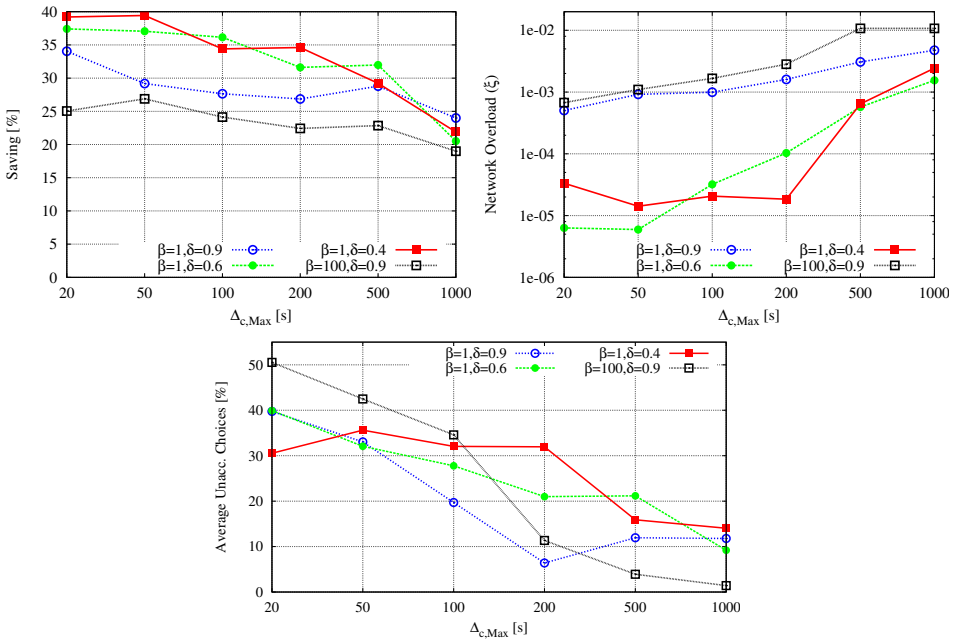

Figure 5.8: Italian ISP: Impact of $\Delta_{c,Max}$: (top-left) Saving, (top-right) Network Overload, and (bottom) Unaccepted choices.

choices about links are made, i.e., Δ_c. Fig. 5.10 reports the performance metrics for increasing values of $\Delta_{c,Max}$, $\beta = \{1, 100\}$ and $\delta = \{0.4, 0.6, 0.9\}$. Interestingly, large $\Delta_{c,Max}$ will slow down the algorithm convergence, while small values of $\Delta_{c,Max}$ may cause unnecessary changes to the network topology that have to be quickly undone. This intuition is confirmed by Fig. 5.10 (bottom), in which the percentage of unaccepted changes rapidly decreases as $\Delta_{c,Max}$ increases, for all the cases. However, this is not beneficial for the network since the network overload steadily increases while the saving decreases, since the system becomes slower in reacting to the changes of traffic. For example, with $\Delta_{c,Max} = 1000\,\text{s}$, $\beta = 1$ and $\delta = 0.4$ the average percentage of unaccepted choices is below 15%, but the network overload is two orders of magnitude higher than the $\Delta_{c,Max} = 50\,\text{s}$ case.

LSA Interval Finally, we vary the LSA interval Δ_{LSA}. Intuitively, a low LSA rate may deteriorate the algorithm performance since in this scenario node choices are based on outdate network states and traffic changes can cause overload situations to which the system does not promptly react. Tab. 5.3 reports the variations of the performance indicators with $\Delta_{LSA} \in [5s, 30s]$, as commonly adopted by OSPF. Results are obtained setting $\beta = 1$, $\delta = 0.9$, and $\Delta_{c,Max} = 50\,\text{s}$. Interestingly, all metrics present just minor oscillations with respect to Δ_{LSA}, suggesting that the algorithm is robust even for large values of the parameter.

5.2 DLF and DMP: Distributing Centralized Heuristics for Energy-Aware Routing

Another natural direction for the distributed solution of the energy-aware routing problem is represented by the distribution of the centralized heuristics described in Chapter 4. In particular, we device a novel algorithm, which is able to automatically adapt the

Table 5.3: Italian ISP: Δ_{LSA} variation.

Δ_{LSA}	Saving [%]	Unacc. Choices [%]	ξ
5	29.18	0.33	9.18e-04
10	29.40	0.33	9.00e-04
20	28.03	0.28	9.49e-04
30	28.43	0.25	1.07e-03

state of network links to the actual traffic in the network. The algorithm is able to considerably reduce the network energy consumption, while requiring only a small number of reconfigurations.

In more details, the algorithm is a *fully distributed solution* that leverages the knowledge of *current link load* instead of current traffic matrix (as instead was, in the case of the centralized versions). The solution takes advantage of a traditional Link-State routing protocol, e.g., OSPF, properly augmented to exchange information about the link power status (i.e., *on* or *off*) and current load, on the basis of which decisions are taken. With respect to the solution described in Section 5.1, this algorithm requires a set of input parameters whose values are intuitive and easy to set, and is able to achieve higher energy savings. The simplicity and better performance come at the price of an higher amount of exchanged information and higher level of coordination among nodes.

5.2.1 Algorithm Description

The proposed solution relies only on the knowledge of (i) the current topology configuration (i.e., $\{x_{ij}\}$, a binary variable, taking the value of 0 if link (i, j) is in off state, 1 otherwise), and of (ii) the traffic load on the links (i.e., $\{l_{ij}\}$). Periodic LSA are broadcasted in the network, describing the state of the links. LSAs are also used to broadcast eventual *critical states*, e.g., presence of unreachable destinations. This guarantees all nodes have the same knowledge of network status, and can take consistent decisions. Finally, link power consumption ($P_{ij} > 0$) can also be shared by means of LSAs.

Distributed choices regarding the power state of links are made. All nodes run the same algorithm to find the link to target. Two cases are possible, based on the critical state information carried by the last received LSA:

Last LSA critical state OK – If the network is in a normal working state, i.e., last LSA did not signal link load violations or disconnected source/destination pairs (line 1 of Alg. 3), one link is selected to be possibly switched off ($(i, j)^*$ in Alg. 3). An unambiguous policy must be defined to select which link is target of the switch off attempt, on the basis of the local knowledge available to all the nodes. Possible choices may be, but not limited to, (i) selecting the least loaded link (Distributed Least Flow - DLF - hereafter), a choice that would have the lowest possible impact on current traffic routing, or (ii) selecting the most power hungry link (Distributed Most Power - DMP - hereafter), a choice that would have the highest possible impact on the energy saving. We suppose that a tie-breaking rule is defined as well, e.g., using lexicographical order.

Each node maintains three FIFO queues to store the last links that (i) have been

switched off but no LSA confirmed yet that constraints are not violated – `to_be_verified` list; (ii) are in off state and caused no constraint violations – `offLinks` list; (iii) caused a violation and thus should not be switched off anymore – `tabu` list. `tabu` list has a maximum length of `maxLength` links.

Being $E^* = \{(i,j)|x_{(i,j)\in E} = 1 \wedge (i,j) \notin \texttt{tabu}\}$, the link selection policies may be formalized as:

$$\text{DLF}: \qquad (i,j)^* = \arg\min_{(i,j)\in E^*}\{l_{ij}\} \tag{5.6}$$

$$\text{DMP}: \qquad (i,j)^* = \arg\max_{(i,j)\in E^*}\{P_{ij}\} \tag{5.7}$$

Before switching link $(i,j)^*$ off, all nodes check if the network would still be connected after its removal (i.e., *connectivity check* - line 2 of Alg. 3). The check is performed through a simple graph exploration algorithm, like a Breadth-First Search. If the connectivity check fails, then all nodes append $(i,j)^*$ to the `tabu` list and no further action is taken.

If the *connectivity check* is positive, $(i,j)^*$ can be switched off. Nodes i and j take care of this by means of some signaling protocol if required, and insert $(i,j)^*$ in the `to_be_verified` list. Finally, they broadcast a new LSA to share the new state $x_{(i,j)^*}$ (lines 3-4 of Alg. 3).

Nodes i and j then wait for the first LSA after a switch off decision. If it reports constraint violations (unreachable node or link overload), they quickly undo the last move by popping all links $(i,j)^*$ from the `to_be_verified` list and inserting them into the `tabu` list (lines 3 to 7 of Alg. 4). Otherwise, if the LSA does not advertise any problem, elements from the `to_be_verified` list are moved to the `offLinks` list (lines 8-9 of Alg. 4).

Last LSA critical state KO – If the last LSA before a choice is reporting any constraint violation, nodes react by bringing back to operational state some link which was put into off state (lines 9 to 12 of Alg. 3). Also the choice of the link to be switched on must be unambiguous with respect to the distributed knowledge. Possible criteria may be, but not limited to, selecting (i) the last link being switched off (*LastOff* hereafter), or (ii) the closer off link to the congestion point (*Distance* hereafter). The *LastOff* criterion is based on the intuition that a recently made change is more likely responsible for the current congestion state. On the other hand, the *Distance* criterion is based on the intuition that the link which is closer to the congested point may more likely help in draining the extra traffic flow and relieve congestion. Distance between a couple of links is defined as the number of nodes on the shortest path between the nodes responsible for such links. Given a pair of nodes connected by a common link, we select as node responsible for the link the one with lowest ID between them, in order to compute link distances.

The node responsible for the selected link switches it on. This mechanism allows the algorithm to react to traffic surges, and to link failures that have to be recovered by turning on other resources.

```
Distributed Choice
Input: (i,j)*, lastLSACriticalState
1    if lastLSACriticalState == OK:
2      if connectivityCheck(x_(i,j)* = 0) == OK:
3        x_(i,j)* = 0
4        to_be_verified.append(x_(i,j)* = 0)
5      else
6        tabu.append((i,j)*)
7        if length(tabu) > maxLength:
8          removeOlder(tabu)
9    else:
10    ij = selectLink(offLinks)
11    x_ij = 1
12    offLinks.remove(ij)
```

Alg. 3: The pseudo-code of the *choice* event.

```
LSA receipt
Input: LSACriticalState
1  while length (to_be_verified) > 0:
2    ij = removeOlder(to_be_verified)
3    if LSACriticalState == KO:
4      tabu.append(ij)
5      if length(tabu) > maxLength:
6        removeOlder(tabu)
7      x_ij = 1
8    else:
9      offLinks.append(ij)
```

Alg. 4: The pseudo-code of the *LSA critical state reception* processing.

Simulator Details

Algorithms have been implemented in a custom event-based simulator starting from the software used in [59]. Events correspond to traffic changes, LSA broadcasting events, and choice events. The choice procedure is described by the pseudocode reported in Alg. 3, while the procedure nodes execute at every LSA critical state processing is described by the pseudocode in Alg. 4.

LSAs are broadcasted every Δ_{LSA}. The time interval between two consecutive choices is a random variable, t_c, which is uniformly distributed between Δ_{LSA} and Δ_c seconds. Indeed, LSA should be more frequent than choices (i.e., a choice, and its result, are notified before a new one takes place). The offered traffic is defined by a TM, which changes to a new TM every Δ_{TM}. Traffic is modeled as fluid, which is routed according to a minimum cost path algorithm. Link weights are given and known. Note that loose synchronization is achieved among nodes by means of LSA messages. Indeed, since the goal of the algorithm is to track the slow variation of traffic during the day, responsiveness to traffic changes is not critical.

Table 5.4: Simulation scenario characteristics.

Parameter	Symbol	Value
Maximum Link Load	ϕ	50%
TM change interval	Δ_{TM}	48 min
Number of nodes	N	373
Number of links	L	718
Average link length	$E[m_{ij}]$	41 km
Average link capacity	$E[c_{ij}]$	6 Gb/s

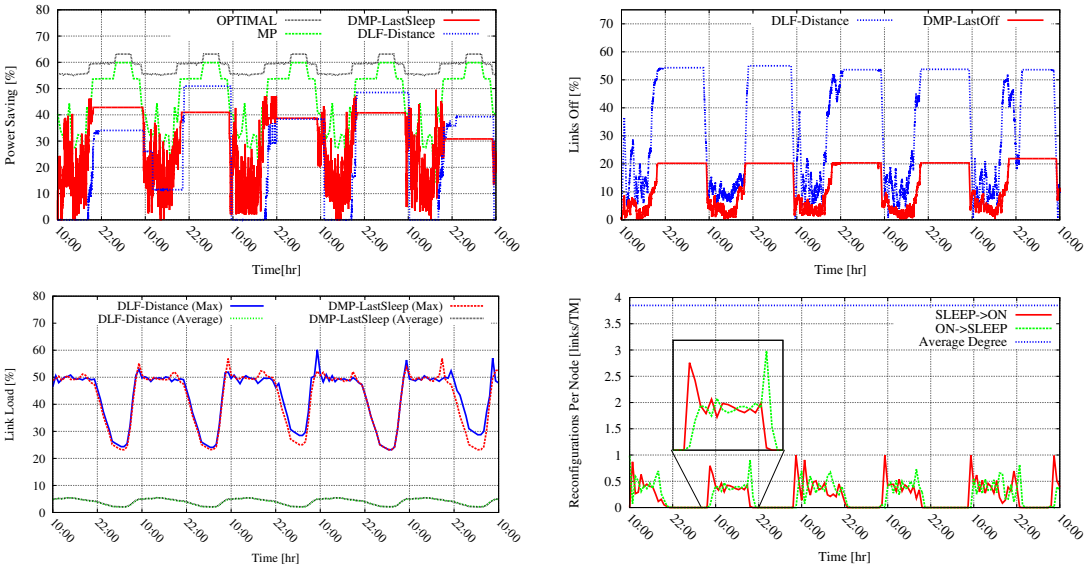

Figure 5.9: Italian ISP: (top-left) Power Saving Comparison, (top-right) Percentage of Links Off, (bottom-left) Maximum and Average link load with DLF-Distance and DMP-LastOff, (bottom-right) Average Number of Reconfigurations with DLF-Distance.

5.2.2 Results on Real Network Scenarios

To provide a relevant evaluation of the described algorithm, we consider a benchmarking scenario obtained from an actual nation-wide ISPs in Italy, namely **Italian ISP**. The scenario has already been used in the performance evaluation of previously described solutions. Its topology is reported in Fig. 5.1 (right). A summary of the main characteristics of the scenario is reported in Tab. 5.4. The adopted power model is the same already used in the performance evaluation of Section 5.1.2.

All simulations consider a one-week long period of time. This allows us to obtain average performance estimation, with negligible variations among different runs.

Time Evolution and Transient Analysis

Unless otherwise specified, we adopt the following set of parameters: $\Delta_c = 20$s, $\Delta_{LSA} = 10$s, `maxLength` $= 70$ links, which corresponds to nearly 10% of links, and maximum link utilization $\phi = 0.5$. Note that LSA timing matches current OSPF specifications [63].

Fig. 5.9 (top-left) reports the power saving computed with respect to the scenario in

which all links are on. DLF-Distance and DMP-LastOff algorithms are shown during the initial 5 days of system evolution using dotted blu and solid red lines, respectively. The solid black curve reports the optimal solution computed solving the formulation of [36] for each TM[5]. The dotted green line shows the power saving of the MP centralized heuristic, which has been proven in [36] to be the most effective one for this topology. Recall that the ILP solution exploits fluid routing, and guarantees higher power savings due to the fact that single traffic requests may be split over multiple paths. It has thus to be considered as an upper bound.

Several considerations hold. First, both DLF-Distance and DMP-LastOff are able to quickly follow traffic variation, and to achieve a good power saving. Both guarantee 30-50% of saving during night-time, which is comparable to the MP centralized heuristic, but obtained without the perfect knowledge of the TM. Second, constant saving during night-time suggests that algorithms have converged to a solution which remains stable. Third, DMP-LastOff algorithm provides better performance than DLF-Distance in terms of power saving. To give more insight about why this happens, Fig. 5.9 (top-right) reports the time evolution of the percentage of powered off links. The plot clearly shows that DLF-Distance is able to turn off a larger number of links than DMP-LastOff. This is due to the fact that the Italian ISP links that carry the least amount of traffic are the ones that are found at the edge of the topology. DLF targets thus these links, which unfortunately consume a negligible amount of power compared to the long haul links found in the core of the network. Power-hungry links are instead targeted by DMP which thus can achieve a better power saving even if switching off a smaller number of links.

Finally, note that during the day, i.e., when more capacity is needed to meet traffic demand, both algorithms keep looking for possible links to be switched off. DMP targets the most expensive links whose power cycling is reflected in the noisy power saving of Fig. 5.9 (top-left).

Fig. 5.9 (bottom-left) details the average and the maximum link load obtained by running DLF-Distance and DMP-LastOff. Considering the average load, we observe that its variation is limited during the day, suggesting that the algorithms efficiently match the current network capacity to the actual demands. Still, an over-provisioning of capacity is present since the average link load is smaller than 10%. Interestingly, during night-time the maximum link load is below 30%, i.e., far from the load threshold $\phi = 0.5$. This suggests that the connectivity constraint is stricter than the maximum load constraint. During high traffic periods, the maximum load constraint instead kicks in, and some link loads actually get close to 0.5. Indeed, some violations are present, even if of short duration and of small intensity. We will quantify violations better in the remaining of this Section.

Finally, Fig. 5.9 (bottom-right) reports the average number of OFF→ON (solid red line) and ON→OFF (dotted green line) changes per node per Δ_{TM}. DLF-Distance is considered, being results similar for other algorithms. Average node degree $\frac{2L}{N}$ is plotted as reference, using a dotted blu. During the night no choices occur confirming that the

[5]The solution has been obtained running CPLEX on a high performance cluster hosted in the Politecnico di Torino Campus [66].

Table 5.5: Algorithm Comparison, $\Delta_{LSA} = 10$, $\Delta_c = 20$.

Algorithm	Saving [%]	Unacc. Choices [%]	ξ
OPTIMAL	58.56	–	–
MP [36]	46.28	–	–
GRiDA [59]	19.73	52	5.93e-4
DMP-Distance	32.35	20	3.23e-3
DMP-LastSleep	30.30	23	4.37e-3
DLF-Distance	25.45	17	1.63e-3
DLF-LastSleep	19.66	18	4.81e-4

algorithm has converged to a stable solution limited by connectivity check. During early morning, the OFF→ON events are predominant and nodes switch back on some links to quickly react to traffic surge (see the inset). During the day both events occur since the system continuously tries to adapt the network capacity to the actual traffic demand, but most changes are then undone due to temporary overload. Finally, during the evening the OFF→ON events become predominant due to traffic drop. Note that the number of reconfigurations per node is always limited to 1 per Δ_{TM} (i.e., 48 min), and much lower on average.

Average Performance

We now compare the performance of different algorithms to assess which policy performs better. We consider *energy saving*, *number of unaccepted choices*, and *network overload* as performance metrics. Energy saving is evaluated as the integral of link power saving over a one week long time interval. Unaccepted choices account the percentage of sleep choices which are undone due to the immediate critical state indication by LSA. Percentage is computed with respect to the total number of sleep attempts. The network overload has been defined in Section 5.1, and is a relative indicator for the network congestion level, averaged over the simulation period, accounting for the number of load violations, their entity, and their duration.

Tab. 5.5 reports results considering the distributed algorithms. For comparison, we include the optimal solution, the MP centralized policy [36], and GRiDA (Section 5.1). As the intuition already suggested from Fig. 5.9, higher energy savings are guaranteed by the DMP policies given that DLF policies are able to put in sleep mode links that do not consume much power. Note that up to 32.35% of energy saving can be reached for the considered network scenario, which is smaller than the centralized solutions but higher than GRiDA. On the other hand, GRiDA generally guarantees lower network overloads.

The DMP algorithms are also more aggressive than the DLF policies in terms of sleep attempts, thus the percentage of unaccepted changes and network overload are higher for the formers. This is due to the fact that DMP targets energy hungry links, which are also the ones that carry lot of traffic being backbone links. Putting in sleep state one of these links results in a large amount of traffic to be re-routed over alternative paths. This causes a larger number of violations. Note that only less than 23% of choices results in a traffic violation.

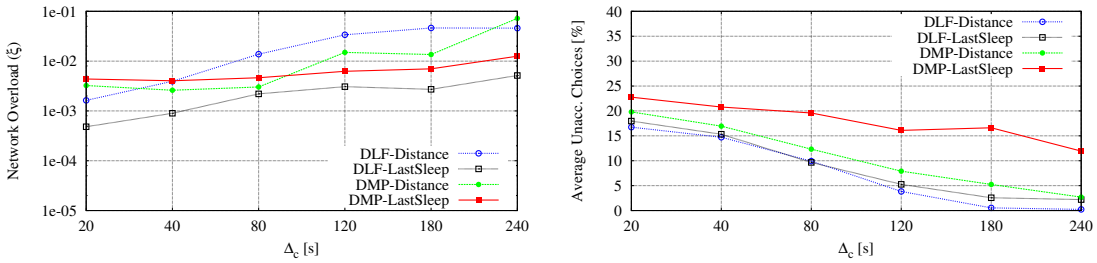

Figure 5.10: Impact of Δ_c. (left) Network Overload, (right) Unaccepted choices.

To gauge how critical are those violations we focus our attention to the network overload ξ. Results confirm that violations are overall very small (see Fig. 5.9 (bottom-left)) . This because the algorithms react to a critical state by immediately undoing the last power off attempt. Since LSA are frequently exchanged, the overload condition lasts no more than Δ_{LSA} in the worst case, i.e., few seconds.

Furthermore, comparing the LastSleep and Distance policies, we observe that the latter generally saves more power while showing quicker reaction to critical situation. This is due to the different reactions to traffic surges. Indeed, when a link is overloaded due to an increase of some r^{sd}, it is better to turn on some adjacent link, rather than the last link that has been switched off (being such link in any uncorrelated place in the network).

Finally, observe that all proposed algorithms outperform GRiDA. In particular, the number of unaccepted choices is much higher for GRiDA which is designed to turn on/off more than one link per choice, thus causing a large number of "wrong" decisions. On the other hand, nodes in GRiDA only exchange information about eventual anomalous network states, but perform independent asynchronous choices, only based on local knowledge, while, when running DMP or DLF, nodes exchange information about link loads and link power consumptions, and have to achieve a loose synchronization, by means of the LSAs.

Sensitivity to Parameter Setting

We evaluate the impact of parameter choice on the performance. In particular, we consider the sensitivity to choice interval Δ_c, size of system memory `maxLength`, and LSA interval Δ_{LSA}. Parameters are varied one at a time, keeping the others set as reported in the previous section. Average results are computed over 7 days of simulations.

We start looking at the sensitivity of the algorithms to the time interval at which choices about links are made (Δ_c). Fig. 5.10 reports the variation in terms of network overload (ξ), and percentage of unaccepted choices, for increasing values of Δ_c. Interestingly, the percentage of unaccepted changes rapidly decreases as Δ_c increases, for all the algorithms. However, this is not beneficial for the network since the network overload steadily increases, suggesting that the system becomes slower in reacting to the changes of traffic. For example, with a choice made every two minutes on average, i.e., $\Delta_c = 240\,\text{s}$, the average percentage of unaccepted choices is steadily below 5% for DLF-Distance, but the network overload is nearly two orders of magnitude higher than in the $\Delta_c = 20\,\text{s}$ case. Energy saving is 25% and 20% for $\Delta_c = 20\,\text{s}$ and $\Delta_c = 240\,\text{s}$,

Table 5.6: Impact of `maxLength` on DLF-LastSleep (LS), and DLF-Distance (Di) algorithms.

maxLength [links]	Saving [%]		Unacc. Choices [%]		ξ	
	Di	LS	Di	LS	Di	LS
1	22.10	22.43	19	20	6.91e-4	1.29e-3
2	23.39	22.30	17	20	9.54e-4	1.13e-3
4	25.58	23.83	17	20	9.34e-4	1.23e-3
20	26.21	22.51	16	19	1.68e-3	8.48e-4
70	25.45	19.66	17	18	1.63e-3	4.81e-4

Table 5.7: Variation of Δ_{LSA}.

$\Delta_{LSA}[s]$	Saving [%]	Unacc. Choices [%]	ξ
2	24.22	18	8.84e-04
10	25.45	17	1.63e-03
20	22.53	17	1.13e-03
30	23.33	17	8.16e-04

respectively, confirming that the algorithm performance degrades for large values of Δ_c.

We then investigate the impact of the `tabu` size on the algorithm performance. Tab. 5.6 reports the metrics for different values of `maxLength`, obtained by running DLF-Distance and DLF-LastSleep on the considered scenario. Interestingly, the size of the `tabu` list on this scenario has a rather limited impact on performance. In particular, the percentage of unaccepted choices is decreasing as `maxLength` is increasing, suggesting that, as the system exploits more memory, the number of choices leading to negative LSA is decreased. However, it is also crucial not to have a too big memory to follow the traffic fluctuations. If the `tabu` list is too long, indeed, links are blacklisted for long periods of time during which they are kept on. Thus, the best savings are obtained as the length of the buffer is set to intermediate values.

Finally, we vary the LSA frequency, considering also the case when Δ_{LSA} is greater than Δ_c. Intuitively, a low LSA rate may deteriorate the algorithm performance since in this scenario node choices are based on a network status not constantly updated so that traffic changes can cause overload situations to which the system does not promptly react. Tab. 5.7 reports the variations of the performance indicators for the considered network scenario with $\Delta_{LSA} \in [2s, 30s]$. Results consider the DLF-Distance. Interestingly, all metrics present just minor oscillations with respect to Δ_{LSA}, suggesting that the algorithm is robust even for high values of the parameter.

5.3 Implementation Issues

In both the presented distributed solutions, we suppose network devices to support a power saving state for links, which can be selected by any of its two adjacent nodes, by means of a simple signaling protocol. We suppose undirected links, i.e., the power state of the link has to be the same for both directions. The extension to support unidirectional power states is straight forward.

Considering the link switching off procedure, it can occur without any traffic losses. The link can indeed be switched off after all nodes routing tables have been properly changed, to allow a smooth traffic migration.

Our solutions requires nodes to run a link-state routing algorithm, through which eventual link overload occurrences are signaled to all the nodes in the network. This allows nodes to timely signal and quickly react to eventual network congestions. Opaque LSAs [67] may allow to easily carry the additional information in practical implementations, without any change to the link-state protocol. LSA timings can be set in accordance to OSPF specifications [63], so that on average each node has to process N LSA messages every Δ_{LSA}. As standard practice, overhead can be controlled by dividing the routing domain into different OSPF areas and running separate instances of the distributed algorithm on each area. LSA messages allow also nodes to achieve a loose synchronization, thus a strict synchronization is not required. Indeed, since the goal of the algorithm is to track the slow variation of traffic during the day, responsiveness to traffic changes is not critical.

It should be noted that node disconnections from the rest of the network are prevented through the connectivity check mechanism, which is run before trying to power off a link, for both the presented solutions. In the rare case some node is going to be disconnected due to incongruent information, a recovery phase can be implemented using some signaling protocol on links.

Chapter 6

Conclusions and Future Work Directions

In this thesis work, we studied solutions to push energy-awareness into wired networks, following the resource consolidation principle. In last years, the issue of energy efficiency has become of paramount importance for both the industries and the research community, because of its potential economical benefits and of its expected environmental impact. In this context, although the green networking field is still in its infancy, a number of interesting works have already been carried out, exploring different research directions. In particular, we mainly leveraged on the *energy-aware routing* paradigm, proposing and analysing both *centralized* and *distributed* technical solutions.

6.1 Summary

A Picute of the Green Networking Research

Our first work toward the greening of networks aimed at individuating the different paradigms currently explored to reduce the network energy expenditure, and the ones that still may be explored. For each paradigm, we draw the state of the art, and highlighted the points that need further investigation. The main branches of the green networking research explored so far are reported in Fig. 6.1, classified by timescale and architectural level. We proposed a taxonomy of the relevant works in green networking. Our main contribution to the green networking field regards the energy-aware routing, which represented a marginally explored paradigm, while being promising in terms of achievable energy savings. Furthermore, it represents a real challenge, as energy savings are achieved at the price of a reduction of the redundancy level, and of the offered QoS. A good trade off should hence be defined.

The Benchmark Issue

Overviewing the current green networking research, we highlighted a lack of common evaluation scenarios and metrics for the analysis of energy saving solutions, as well as of reliable energy consumption figures and common measuring methodologies. We believe that a community-wide effort is necessary toward the definition of a comprehensive methodology for measuring and reporting the energy consumption of networks, and characterize any possible tradeoff between energy consumption and system performance. We contributed to this effort by (i) profiling the end-user power consumption related to the Web browsing and Flash plug-in loading, and by (ii) comparing and contrasting various

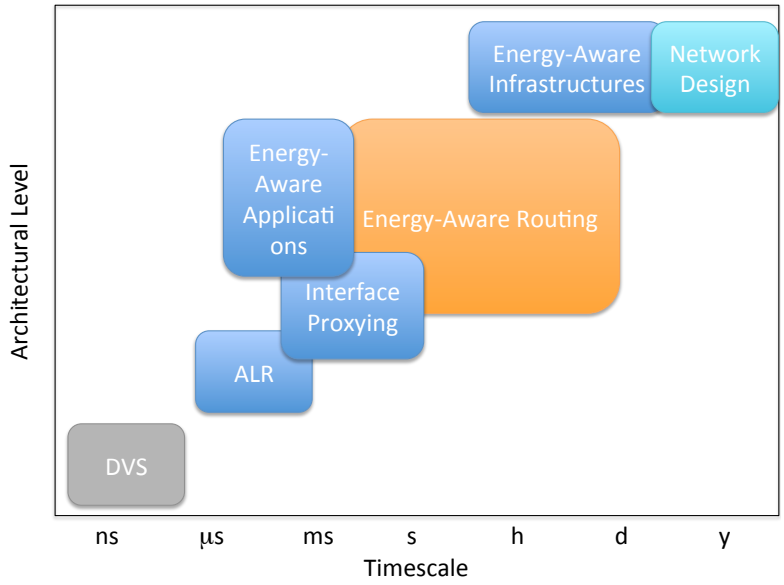

Figure 6.1: The main directions of the current green networking research.

energy-related metrics used in the recent literature, and defining a taxonomy to classify them.

An Optimization Problem Formulation for Energy-Aware Routing

Following the resource consolidation paradigm, we evaluated the formulation of the energy-aware routing as an optimization problem, and present its solution results for real network scenarios, considering different power models, representative of different technological scenarios, and analysing the resulting tradeoff between achievable energy saving and offered QoS.

The Evaluation of Network Device Criticality

While solving the energy-aware routing as an optimization problem, the set of devices to be switched off to save energy is chosen on the basis of the sole energy costs, and does not take into account the "criticality" of such devices in the specific network scenario. We model the network scenario as a cooperative transferable utility game, in order to exploit the game theory powerful tools and define a criticality ranking among network devices. The defined game accounts for both the network topology, and its traffic conditions, moreover, device criticality is accounted for the number of primary and backup paths devices lay on, their importance in building the paths, and the amount of traffic actually routed on that paths, on average over the different possible network configurations. The criticality ranking has been defined in a similar way, both for nodes, and for links, and can be profitably used to drive the resource consolidation process.

Centralised Heuristics for the Energy-Aware Routing Problem

Heuristics have been proposed and evaluated for the energy-aware routing problem, because of the criticality unawareness of its modeling as an optimization problem, and the huge solution times it requires in case of big size network scenario. In particular, we

proposed a resource consolidation algorithm, based on the criticality ranking previously defined by means of the game theory formulation. The proposed solution, contrasted to the other solutions present in the literature, considering real network scenarios, resulted being able to achieve the better tradeoff between achievable energy saving, and offered QoS, also under different technological assumptions.

Distributed Solutions for the Energy-Aware Routing Problem

All the previously proposed solutions for the energy-aware routing problem suppose a technological scenario in which a central control entity is present, having a global view of the instantaneous network status and traffic requests, and being able to compute the best network configuration and to coordinate devices to achieve such configuration. These assumptions limit the applicability and deployment of *centralised* solutions to specific cases, considering current and foreseen network technologies. Distributed solutions have hence been developed and evaluated. In particular, we proposed different solutions to account for different levels of coordination among nodes, and different kinds of exchanged information. The proposed solutions resulted in stable network behaviors, when considering real network scenarios. Moreover, they resulted able to achieve energy savings comparable to the ones of centralised solutions, while keeping network performance under control.

6.2 Future Work Directions

Despite we did our best in making this work as complete as possible, inevitably, there are points that we did not deal with yet, for lack of time. In the following, we report issues that we think represent still open points, and that we would like to pursue in the future.

Energy-Aware Routing

The problem of energy aware routing may also be formulated as a robust network-design problem, in which, potential failure devices (i.e., links and/or nodes) correspond to devices that can be switched off. It may hance be possible to define different working statuses, each one of which corresponds to a specific set of devices being powered off. In this case, it may be possible to optimize the design process as a global optimization over the different possible working statuses. An analysis of such formulation may be interesting, especially to evaluate the trade off between energy saving and network robustness. Actually, an explicit evaluation of the solution robustness is missing also for what concerns the current and classical solutions for the energy-aware routing problem.

For what concerns the evaluation of the criticality of network devices, as defined in Chapter 4, it would be interesting to analyse the time evolution of the criticality ranking, especially when considering network scenarios including nodes subject to different night/day behaviors (e.g., network spreading across different time-zones). In a similar variable scenario, it would also be interesting to evaluate the correlation between different, but close in time, resource consolidation solutions. An high number of network reconfigurations may, indeed, degrade the network performance (i.e., each reconfiguration requires a new convergence period for the routing protocol). Different tradeoffs between energy savings and number of network reconfigurations may hence be precisely

defined and evaluated. Finally, the impact of such network reconfigurations on traffic should be precisely evaluated (e.g., routing protocol convergence time, or eventual traffic losses, depending on the technological scenario).

Distributed Solutions for Energy-Aware Routing

In our work, we proposed distributed solutions for the energy-aware routing problem, which allow avoiding the need of a central control entity, coordinating the network configuration, and having a global knowledge of the network traffic flows. Furthermore, distributed solutions drastically reduce the problem complexity with respect to the centralised ones. Modeling such distributed solutions as evolutionary games (or as perfect information games) may allow a better understanding of the algorithm convergence, and of the eventual steady states.

Further distributed solutions may be defined considering different levels of available information at nodes. For instance, higher level of information may be considered, as supposing nodes collecting information about the set of active destinations for the traffic they are routing. Similar information availability may allow achieving higher energy saving, by loosing (or dynamically adapting) the connectivity check constraints.

Finally, a detailed evaluation of the overhead coming with the distributed solutions for energy-aware routing, and of its practical issues, represents a necessary step toward any real implementation. For this purpose, we are currently working on a testbed implementation of the proposed solutions.

Energy-Aware Network Design

The current network design techniques are based on the peak foreseen traffic requests. It would be interesting to evaluate how the network design process changes, when taking into account the possibility of successive dimensioning according to the variable network load (i.e., *energy-aware routing*). This process will include a design based on a set of traffic matrixes, covering the all foreseen traffic variations, instead of the sole peak traffic matrix.

We believe that the criticality index for network devices, as introduced in Chapter 4, may be profitably used also in the network design process. It may allow, for instance, to distribute the criticality as much as possible among network devices (i.e., introduce back-up devices aside the most critical ones, and remove the less critical devices, taking into account the corresponding investment).

Bibliography

[1] M. Webb, "SMART 2020: Enabling the Low Carbon Economy in the Information Age." The Climate Group. London, June 2008.

[2] Global Action Plan, "An Inefficient Truth." Global Action Plan Report, http://globalactionplan.org.uk, December 2007.

[3] International Energy Agency Web Page, "http://www.iea.org/journalists/fastfacts.asp," 2010.

[4] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures," *IEEE Communication Surveys and Tutorials, to appear*, 2011.

[5] A. Adelin, P. Owezarski, and T. Gayraud, "On the Impact of Monitoring Router Energy Consumption for Greening the Internet," in *IEEE/ACM International Conference on Grid Computing (Grid 2010)*, (Bruxelles, Belgique), October 2010.

[6] What Europeans do at Night, "http://asert.arbornetworks.com/2009/08/ what-europeans-do-at-night/," 2009.

[7] Y. Zhang, P. Chowdhury, M. Tornatore, and B. Mukherjee, "Energy Efficiency in Telecom Optical Networks," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 4, 2010.

[8] K. W. Roth, F. Goldstein, and J. Kleinman, "Energy Consumption by Office and Telecommunications Equipment in Commercial Buildings Volume I: Energy Consumption Baseline," Tech. Rep. Vol I, National Technical Information Service (NTIS), US Department of Commerce, Jan. 2002.

[9] B. Nordman and K. Christensen, "Reducing the Energy Consumption of Network Devices." IEEE 802.3 Tutorial, July 2005.

[10] C. Lange, "Energy-related Aspects in Backbone Networks," in *Proceedings of 35th European Conference on Optical Communication (ECOC 2009)*, (Wien, AU), Sept. 2009.

[11] R. Bolla, R. Bruschi, K. Christensen, F. Cucchietti, F. Davoli, , and S. Singh, "The Potential Impact of Green Technologies in Next Generation Wireline Networks – Is There Room for Energy Savings Optimization?," *IEEE Communication Magazine*, 2010.

[12] K. Christensen, C. Gunaratne, B. Nordman, and A. D. George, "The Next Frontier for Communications Networks: Power Management," *Computer Communications*, vol. 27, pp. 1758–1770, Dec. 2004.

[13] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A Survey of Green Networking Research," *IEEE Communication Surveys and Tutorials*, no. 2, 2012.

[14] IEEE P802.3az Energy Efficient Ethernet Task Force, "http://www.ieee802.org/3/az/index.html."

[15] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A Power Benchmarking Framework for Network Devices," in *Proceedings of IFIP Networking 2009*, (Aachen, Germany), May 2009.

[16] C. Gunaratne, K. Christensen, and B. Nordman, "Managing Energy Consumption Costs in Desktop PCs and LAN Switches with Proxying, Split TCP Connections, and Scaling of Link Speed," *International Journal of Network Management*, vol. 15, pp. 297–310, Sept. 2005.

[17] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR)," *IEEE Transactions on Computers*, vol. 57, pp. 448–461, Apr. 2008.

[18] G. Ananthanarayanan and R. H. Katz, "Greening the Switch," in *Proceedings of the USENIX Workshop on Power Aware Computing and Systems (HotPower), held at the Symposium on Operating Systems Design and Implementation (OSDI 2008)*, (San Diego, California, USA), Dec. 2008.

[19] S. Albers, "Energy-efficient algorithms," *Communications of the ACM*, vol. 53, pp. 86–96, May 2010.

[20] K. Sabhanatarajan and A. Gordon-Ross, "A Resource Efficient Content Inspection System for Next Generation Smart NICs," in *Proceedings of the IEEE International Conference on Computer Design 2008. (ICCD 2008)*, (Lake Tahoe, California, USA), pp. 156–163, Oct. 2008.

[21] Mac OS X v10.6: About Wake on Demand, "http://support.apple.com/kb/HT3774."

[22] J. Blackburn and K. Christensen, "A Simulation Study of a New Green BitTorrent," in *Proceedings of the 1st International Workshop on Green Communications (GreenComm) in conjunction with the IEEE International Conference on Communications*, (Dresden, Germany), June 2009.

[23] L. Irish and K. J. Christensen, "A "Green TCP/IP" to Reduce Electricity Consumed by Computers," in *Proceedings IEEE Southeastcon 1998 'Engineering for a New Era'*, (Orlando, Florida, USA), pp. 302–305, Apr. 1998.

[24] M. Gupta and S. Singh, "Greening of the Internet," in *ACM SIGCOMM 2003*, (Karlsrhue, Germany), August 2003.

[25] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *IEEE ICC'09*, (Dresden, Germany), June 2009.

[26] J. P. Jue and V. M. Vokkarane, *Optical Burst Switched Networks*. Springer, 2005.

[27] M. Baldi and Y. Ofek, "Time for a "Greener" Internet," in *Proceedings of the 1st International Workshop on Green Communications (GreenComm) in conjunction with the IEEE International Conference on Communications*, (Dresden, Germany), June 2009.

[28] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power Awareness in Network Design and Routing," in *Proceedings of the 27th IEEE Annual Conference on Computer Communications. (INFOCOM 2008)*, (Phoenix, AZ, USA), pp. 457–465, Apr. 2008.

[29] B. Sansò and H. Mellah, "On Reliability, Performance and Internet Power Consumption," in *Proceedings of 7th International Workshop on Design of Reliable Communication Networks (DRCN 2009)*, (Washington, D.C., USA), Oct. 2009.

[30] A. P. Bianzino, A. K. Raju, and D. Rossi, *Sustainable Green Computing: Practices, Methodologies and Technologies*, ch. Energy Consumption in the Internet Core: a Sensitivity Analysis. IGI Global, 2011.

[31] A. P. Bianzino, A. K. Raju, and D. Rossi, "Greening the Internet: Measuring Web Power Consumption," *IT Professional, Special Issue on Green IT*, vol. 13, pp. 48–53, January/February 2011.

[32] A. P. Bianzino, A. K. Raju, and D. Rossi, "Apples-to-apples: a framework analysis for energy-efficiency in networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, pp. 81–85, Dec. 2010.

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

[34] L. A. Barroso and U. Hölzle, "The Case for Energy-Proportional Computing," *IEEE Computer*, vol. 40, pp. 33 – 37, Dec. 2007.

[35] The GEANT network, "http://www.geant.net/."

[36] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing ISP Network Energy Cost: Formulation and Solutions," *IEEE/ACM Transactions on Networking, to appear*, 2011. http://www.telematica.polito.it/chiaraviglio/papers/GreenTon.pdf.

[37] C. Gunaratne, K. Christensen, and S. W. Suen, "Ethernet Adaptive Link Rate (ALR): Analysis of a buffer threshold policy," in *IEEE GLOBECOM*, (San Francisco, California, USA), Nov. 2006.

[38] R. Tucker, J. Baliga, R. Ayre, K. Hinton, and W. Sorin, "Energy consumption in IP networks," in *ECOC 2008*, (Brussells, Belgium), Sept. 2008.

[39] R. Hays, A. Wertheimer, and E. Mann, "Active/Idle Toggling with Low-Power Idle." Presentation for IEEE 802.3az Task Force Group Meeting., Jan. 2008.

[40] "Ampl, a modeling language for mathematical programming." `http://www.ampl.com/`.

[41] "Ibm ilog cplex optimizer homepage." `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`.

[42] The IGP-WO algorithm, "http://totem.run. montefiore.ulg.ac.be/algos/igpwo.html."

[43] A. Bianzino, C. Chaudet, F. Larroca, D. Rossi, and J. Rougier, "Energy-Aware Routing: a Reality Check," in *3rd International Workshop on Green Communications (GreenComm3)*, (Miami, USA), December 2010.

[44] C. Isci, A. Buyuktosunoglu, C.-Y. Cher, P. Bose, and M. Martonosi, "An Analysis of Efficient Multi-Core Global Power Management Policies: Maximizing Performance for a Given Power Budget," in *Proceedings of IEEE/ACM International Symposium on Microarchitecture (MICRO 39)*, (Orlando, Florida, USA), pp. 347–358, IEEE Computer Society, Dec. 2006.

[45] W. Fisher, M. Suchara, and J. Rexford, "Greening Backbone Networks: Reducing Energy Consumption by Shutting Off Cables in Bundled Links," in *1st ACM SIGCOMM Workshop on Green Networking*, (New Delhi, India), August 2010.

[46] R. Garroppo, S. Giordano, G. Nencioni, M. Pagano, and M. Scutella, "Models and Heuristic Approaches for Network Power Management," in *9th Italian Networking Workshop*, (Courmayeur, Italy), Jan. 2012.

[47] S. Moretti and F. Patrone, "Transversality of the Shapley value," *Top*, vol. 16, no. 1, pp. 1–41, 2008.

[48] L. Shapley, "A Value for n-Person Games," *Contributions to the Theory of Games II*, pp. 307–317, 1953.

[49] D. Gómez, E. González-Arangüena, C. Manuel, G. Owen, M. del Pozo, and J. Tejada, "Splitting graphs when calculating Myerson value for pure overhead games," *Mathematical Methods of Operations Research*, vol. 59, no. 3, pp. 479–489, 2004.

[50] R. Myerson, "Graphs and cooperation in games," *Mathematics of Operations Research*, vol. 2, no. 3, pp. 225–229, 1977.

[51] A. Bianzino, C. Chaudet, S. Moretti, D. Rossi, and J. Rougier, "The Green-Game: Striking a Balance between QoS and Energy Saving," in *23rd International Teletraffic Congress (ITC 2011)*, (San Francisco, USA), September 2011.

[52] F. Glover and M. Laguna, *Tabu Search*. Kluwer Academic Publishers, 1997.

[53] M. Shaw, "Group structure and the behaviour of individuals in small groups," *Journal of psychology*, no. 38, pp. 139–149, 1954.

[54] A. Bavelas, "A mathematical model for small group structures," *Human Organization*, no. 7, pp. 16–30, 1948.

[55] M. Beauchamp, "An improved index of centrality," *Behavioral Science*, vol. 10, no. 2, pp. 161–163, 1965.

[56] P. Bonacich, "Factoring and Weighting Approaches to Status Scores and Clique Identification," *Journal of Mathematical Sociology*, no. 2, pp. 113–120, 1972.

[57] S. Balon, J. Lepropre, O. Delcourt, F. Skivée, and G. Leduc, "Traffic Engineering an Operational Network with the TOTEM Toolbox," *IEEE Transactions on Network and Service Management*, vol. 4, pp. 51–61, June 2007.

[58] N. Feamster, H. Balakrishnan, and J. Rexford, "Some foundational problems in interdomain routing," in *Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III)*, (San Diego, CA, USA), Citeseer, Nov. 2004.

[59] A. Bianzino, L. Chiaraviglio, and M. Mellia, "GRiDA: a Green Distributed Algorithm for Backbone Networks," in *2011 IEEE Online Green Communications Conference (GREENCOM 2011)*, September 2011. http://www.telematica.polito.it/chiaraviglio/papers/GRiDA.pdf.

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

[61] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini, "An Energy Saving Routing Algorithm for a Green OSPF Protocol," in *IEEE INFOCOM, 2010*, (San Diego, USA), March 2010.

[62] D. Arai and K. Yoshihara, "Eco-friendly distributed routing protocol for reducing network energy consumption," in *Network and Service Management (CNSM), 2010 International Conference on*, (Niagara Falls, Canada), pp. 104–111, IEEE, October 2010.

[63] J. Moy, "OSPF Version 2." RFC 2328, April 1998.

[64] C. Watkins and P. Dayan, "Q-learning," *MACHINE LEARNING*, vol. 8, no. 3, pp. 279–292, 1992.

[65] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware backbone networks: a case study," in *First Int. Workshop on Green Communications (GreenCommâĂŹ09)*, (Dresden, Germany), June 2009.

[66] POLITO HPC Initiative, "http://dauin-hpc.polito.it/."

[67] R. Coltun, "The OSPF Opaque LSA Option." RFC 2370, July 1998.

Appendix A

List of Publications

A.1 International Journals with Peer Review

- Aruna Prem BIANZINO, Claude CHAUDET, Dario ROSSI, Jean-Louis ROUGIER, "A Survey of Green Networking Research", to appear in *IEEE Communications Surveys & Tutorials*, no. 2, 2012.

- Aruna Prem BIANZINO, Anand Kishore RAJU, Dario ROSSI, "Greening the Internet surf: Experimental measurements of Web power-consumption", in *IT Professional, Special Issue on Green IT*, vol. 13, no. 1, IEEE Computer Society, Los Alamitos, CA, USA, January/February 2011.

- Aruna Prem BIANZINO, Anand Kishore RAJU, Dario ROSSI, "Apples-to-apples: a framework analysis for energy-efficiency in networks", in *ACM SIGMETRICS Performance Evaluation Review*, Volume 38 Issue 3, Pages 81-85, New York, USA, December 2010.

A.2 International Conferences and Workshops with Peer Review

- Aruna Prem BIANZINO, Claude CHAUDET, Stefano MORETTI, Jean-Louis ROUGIER, Luca CHIARAVIGLIO, Esther LE ROUZIC, "Enabling Sleep Mode in Backbone IP-Networks: a Criticality-Driven Tradeoff", in *IEEE ICC'12 Workshop on Green Communications and Networking*, Ottawa, Canada, June 15 2012.

- Aruna Prem BIANZINO, Luca CHIARAVIGLIO, Marco MELLIA, "Distributed Algorithms for Green IP Networks", in *The 1st IEEE INFOCOM Workshop on Communications and Control for Sustainable Energy Systems: Green Networking and Smart Grids*, Orlando, Florida, USA, March 30 2012.

- Aruna Prem BIANZINO, Luca CHIARAVIGLIO, Marco MELLIA, "GRiDA: a Green Distributed Algorithm for Backbone Networks", in *Proceedings of 2011 IEEE Online Green Communications Conference*, September 26-29 2011.

- Aruna Prem BIANZINO, Claude CHAUDET, Stefano MORETTI, Dario ROSSI, and Jean-Louis ROUGIER, "The Green-Game: Striking a Balance between QoS

and Energy Saving", in textitProceedings of the 23rd International Teletraffic Congress (ITC 2011), San Francisco, CA, USA, September 6-8 2011.

- Aruna Prem BIANZINO, Claude CHAUDET, Federico LARROCA, Dario ROSSI, Jean-Louis ROUGIER, "Energy-Aware Routing: a Reality Check", in *3rd International Workshop on Green Communications (GreenComm3)*, in conjunction with IEEE GLOBECOM 2010, Miami, Florida, USA, December 10 2010.

- Aruna Prem BIANZINO, Anand Kishore RAJU, Dario ROSSI, "Apple-to-Apple: A Framework Analysis for Energy-Efficiency in Networks", in *Second GreenMetrics Workshop*, in conjunction with ACM SIGMETRICS 2010, New York, USA, June 14-18, 2010.

- Aruna Prem BIANZINO, Jean-Louis ROUGIER, Stefano SECCI, Ramon CASELLAS, Ricardo MARTINEZ, Raul MUNOZ, Nabil Bachir DJARALLAH, Richard DOUVILLE, HÃľlia POUYLLAU, "Testbed Implementation of Control Plane Extensions for Inter-Carrier GMPLS LSP Provisioning", in *Proceedings of 2009 5th Int. Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2009)*, Washington, USA, April 6-8 2009.

A.3 Book Chapters

- Aruna Prem BIANZINO, Anand Kishore RAJU, Dario ROSSI, "Energy Consumption in the Internet Core: a Sensitivity Analysis", in *Sustainable Green Computing: Practices, Methodologies and Technologies*, IGI Global, K. Naima and W. Chen, 2011.

A.4 Other Publications and Research Reports

- Aruna Prem BIANZINO, Claude CHAUDET, Federico LARROCA, Dario ROSSI, Jean-Louis ROUGIER, "Taking into account energy awareness into network optimization", in *The First TELECOM ParisTech − Keio University Joint Workshop on Future Networking*, Invited Communication, Paris, France, September 29 2010.

- Aruna Prem BIANZINO, Claude CHAUDET, Stefano MORETTI, Dario ROSSI, and Jean-Louis ROUGIER, "The Green-Game: Further Results and Discussions", *Technical Report, TELECOM ParisTech*, August 2010.

- Aruna Prem BIANZINO, Dario ROSSI, Jean-Louis ROUGIER, Stefano MORETTI, "The G-Game: A Cooperative Game Approach for Resource Consolidation in Network Dimensioning", in *the 24th European Conference on Operational Research (EUROXXIV)*, Invited Communication, Lisbon, Portugal, July 11-14 2010.

- Aruna Prem BIANZINO, Claude CHAUDET, Dario ROSSI, Jean-Louis ROUGIER, "Energy-Awareness in Network Dimensioning: a Fixed Charge Network Flow Formulation", in *e-Energy'10*, Extended Abstract, Passau, Germany, April 13-15 2010.

A.5 Submitted

- Aruna Prem BIANZINO, Luca CHIARAVIGLIO, Marco MELLIA, Jean-Louis ROUGIER, "GRiDA: Green Distributed Algorithm for Energy-Efficient IP Backbone Networks", submitted to *Computer Networks, The International Journal of Computer and Telecommunications Networking*, ELSEVIER.

- Aruna Prem BIANZINO, Claude CHAUDET, Dario ROSSI, Jean-Louis ROUGIER, "The Green-Game: Accounting for Device Criticality in Resource Consolidation for Backbone IP Networks", submitted to *Strategic Behavior and the Environment, Special Issue on: "ICT-based strategies for environmental conflicts"*, Mike Casey.

Institute Télécom ParisTech
Département Informatique et Réseaux
23 avenue d'Italie - 75013 Paris