UNIVERSITÉ DE MONTRÉAL

ENERGY-AWARE TRAFFIC ENGINEERING FOR WIRED IP NETWORKS

LUCA GIOVANNI GIANOLI

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE ÉLECTRIQUE)
JUILLET 2014

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

ENERGY-AWARE TRAFFIC ENGINEERING FOR WIRED IP NETWORKS

présentée par : GIANOLI Luca Giovanni
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. GIRARD André, Ph.D., président
Mme SANSÓ Brunilde, Ph.D., membre et directrice de recherche
M. CAPONE Antonio, Ph.D., membre et codirecteur de recherche
Mme CARELLO Giuliana, Ph.D., membre
M. KLEIN Thierry, Ph.D., membre

To my family, to Hélène,
and to all the friends
who supported me along these years...

ACKNOWLEDGEMENTS

There is a long list of people who deserve my most sincere thanks for their precious support along these four years. First of all, a special thanks goes to my supervisors, Antonio and Brunilde, who, in 2010, gave me the opportunity to start the Ph.D, come to Montreal and live this amazing adventure. I wanna thank them for their help, for their suggestions and for the attention that they always demonstrated in my regards. It has been a pleasure and an honour to work under their supervision.

My deepest gratitude goes also to Edoardo, Giuliana, Bernardetta, Stefano C., Carmelo and Erick, who taught me so many valuable things and whose contribution has been fundamental along these four years.

A special mention is also for Dr. A. Girard and Dr. T. Klein, who spent time an efforts to evaluate this thesis and gave me very precious suggestions to improve it. A particular thanks also to Dr. J.C. Grégoire, who was member of the commission for the preliminary exam and whose advice had been very useful to develop the research project.

I wanna thank all the guys of the labs in both Montreal and Milano, i.e., Silvia, who was the first to welcome me to Montreal and teach me everything about the city, Antimo, Alessandro, Arash, Stefano P., Federico, Ilario, Hadhami, Marnie, Michele and Carmelo (again!). My most sincere thanks for their precious companionship, for their support and for all the experiences that we shared along these years.

A special appreciation is for all the personnel of École Polytechnique de Montréal and Politecnico di Milano. In particular I would like to thank Nathalie Lévesque and Nadia Prada, who helped me out through all the bureaucratic requirements. Many thanks to the personnel of GERAD, too, and, in particular, to Marie Perrault, Francine Benoit, Carol Dufour, Marilyne Lavoie, and Pierre Girard.

Obviously, I cannot forget my parents, Andrea and Marina, who have never failed to support, and who I always felt very close to even when an entire ocean was among us (thanks Skype!): thanks for letting me go and find my own way in the world.

A very special thanks is for Hélène, who left everything she had in Europe to accompany me to Canada and live this adventure by my side: she is the only person who really knows how hard I worked to get this Ph.D., she has been always there for me.

I will never forget Roberto and all the guys from Tennis Canada, who have been always so kind and who gave me the possibility to play tennis here in Motreal too... during the most stressful periods of a Ph.D., a tennis racket in your hands and a yellow ball to hit are the best remedy!

My deepest thanks to my friends Nicolò, who taught me how to persist, Jacopo, who taught me the true values of life and where to buy the best home-made cakes in the world, Filippo, who forced me to go out beyond a radius of four-hundred meters from my place, Alejandro, my personal tequila supplier and great football player, and Nicola, a very inspiring master of life.

The final thanks are for all the other friends of both Milano and Montreal, with a special mention which goes to Johnny, who crossed the ocean to visit me in Montreal without my knowing it, and Alessandro, whose terrific backhand kept me well-trained along my stays in Milano. Please, forgive me if I've forgotten anybody!

RÉSUMÉ

Même si l'Internet est souvent considéré comme un moyen formidable pour réduire l'impact des activités humaines sur l'environnement, sa consommation d'énergie est en train de devenir un problème en raison de la croissance exponentielle du trafic et de l'expansion rapide des infrastructures de communication dans le monde entier. En 2007, il a été estimé que les équipements de réseau (sans tenir compte de serveurs dans les centres de données) étaient responsables d'une consommation d'énergie de 22 GW, alors qu'en 2010 la consommation annuelle des plus grands fournisseurs de services Internet (par exemple AT&T) a dépassé 10 TWh par an.

En raison de cette tendance alarmante, la réduction de la consommation d'énergie dans les réseaux de télécommunication, et en particulier dans les réseaux IP, est récemment devenue une priorité. Une des stratégies les plus prometteuses pour rendre plus vert l'Internet est le sleep-based energy-aware network management (SEANM), selon lequel la configuration de réseau est adaptée aux niveaux de trafic afin d'endormir tous les éléments redondantes du réseau.

Dans cette thèse nous développons plusieurs approches centralisées de SEANM, afin d'optimiser la configuration de réseaux IP qui utilisent différents protocoles (open shortest path first (OSPF) or multi protocol Label Switching (MPLS)) ou transportent différents types de trafic (élastique or inélastique). Le choix d'adresser le problème d'une manière centralisée, avec une plate-forme de gestion unique qui est responsable de la configuration et de la surveillance de l'ensemble du réseau, est motivée par la nécessité d'opérateurs de maintenir en tout temps le contrôle complet sur le réseau.

Visant à mettre en œuvre les approches proposées dans un environnement réaliste du réseau, nous présentons aussi un nouveau cadre de gestion de réseau entièrement configurable que nous avons appelé JNetMan. JNetMan a été exploité pour tester une version dynamique de la procédure SEANM développée pour les réseaux utilisant OSPF.

ABSTRACT

Even if the Internet is commonly considered a formidable means to reduce the impact of human activities on the environment, its energy consumption is rapidly becoming an issue due to the exponential traffic growth and the rapid expansion of communication infrastructures worldwide. Estimated consumption of the network equipment, excluding servers in data centers, in 2007 was 22 GW, while in 2010 the yearly consumption of the largest Internet Service Providers, e.g., AT&T, exceeded 10 TWh per year. The growing energy trend has motivated the development of new strategies to reduce the consumption of telecommunication networks, with particular focus on IP networks. In addition to the development of a new generation of green network equipment, a second possible strategy to optimize the IP network consumption is represented by sleep-based energy-aware network management (SEANM), which aims at adapting the whole network power consumption to the traffic levels by optimizing the network configuration and putting to sleep the redundant network elements. Device sleeping represents the main potential source of saving because the consumption of current network devices is not proportional to the utilization level: so that, the overall network consumption is constantly close to maximum. In current IP networks, quality of service (QoS) and network resilience to failures are typically guaranteed by substantially over-dimensioning the whole network infrastructure: therefore, also during peak hours, it could be possible to put to sleep a non-negligible subset of redundant network devices.

Due to the heterogeneity of current network technologies, in this thesis, we focus our efforts to develop centralized SEANM approaches for IP networks operated with different configurations and protocols. More precisely, we consider networks operated with different routing schemes, namely shortest path (OSPF), flow-based (MPLS) and take into account different types of traffic, i.e., elastic or inelastic. The centralized approach, with a single management platform responsible for configuring and monitoring the whole network, is motivated by the need of network operators to be constantly in control of the network dynamics. To fully guarantee network stability, we investigate the impact of SEANM on network reliability to failures and robustness to traffic variations. Ad hoc modeling techniques are integrated within the proposed SEANM frameworks to explicitly consider resilience and robustness as network constraints. Finally, to implement the proposed procedures in a realistic network environment, we propose a novel, fully configurable network management framework, called JNetMan. We use JNetMan to develop and test a dynamic version of the SEANM procedure for IP networks operated with shortest path routing protocols.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ANNEXES

LIST OF ACRONYMS

API	application program interface
AS	autonomous system
BGP	border gateway protocol
EANI	energy-aware network intelligence
EA-LG	energy-aware lexicographic GRASP
EA-STH	energy-aware single time-period heuristic
EA-STH-RP	energy-aware single time-period heuristic with restricted paths
EANM	energy-aware network management
EATE	energy-aware traffic engineering
ECMP	equal cost multi-path
E-TESP	energy-aware traffic engineering with shortest path routing
GA-ES	greedy algorithm for energy saving
GHG	global greenhouse gas
GRASP	greedy randomized adaptive search procedure
GRA-ES	greedy randomized algorithm for energy saving
ICT	information and communication technology
IETF	Internet engineering task force
IGP-WO	interior gateway protocol weight optimization
IP	Internet protocol
ISP	Internet service provider
JNetMan	java-based network management platform
LF	least-flow
LL	least-link
LP	linear programming
LPI	low power idle
LR	Lagrangian relaxation
MCND	multi-commodity network design
MIB	management information base
MILP-EWO	MILP-based algorithm for energy-aware weight optimization
MLR	mixed line rates
MMF	max-min-fairness
NGOA	next-generation optical access
NMP	network management platform

OID	object identifier
OSPF	open shortest path first
OSI	open systems interconnection
QoS	quality of service
PAFRP	power-aware fixed routing problem
PAVRP	power-aware variable routing problem
PF	proportional fairness
PoP	point of presence
RCL	restricted candidate list
RED	random early detection
RO	robust optimization
RTT	round-trip-time
SEANM	sleep-based energy-aware network management
SEANM-FB	SEANM with flow-based routing
SEANM-SP	SEANM with shortest path routing
SEANM-ET	SEANM with elastic traffic
SMI	structure of management information
SNMP	simple network management protocol
SW	sum-of-weights
TA-ES	two-stage algorithm for energy saving
TAM	topology abstraction manager
TCP	transport control protocol
TE	traffic engineering
UDP	user datagram protocol
WDM	wavelength division multiplexing

LIST OF SYMBOLS

	Sets
G	Network topology
V	Set of network nodes
V^e	Set of edge nodes
V^c	Set of core nodes
A	Set of unidirectional network links
D	Set of traffic demands
D^e	Set of elastic traffic demands
P^d	Set of paths for demand $d \in D$
S	Circular set of time periods
H	Set of pieces of a piece-wise linear function
U_{ij}^{σ}	Set of demands which are uncertain during time period $\sigma \in S$
V^{o^d}	Set of nodes from which the source node o^d of demand $d \in D^e$ has been excluded
	Parameters
o^d	Origin of demand $d \in D$
t^d	Destination of demand $d \in D$
r^d	Bandwidth request of demand $d \in D$
$r^{d\sigma}$	Bandwidth request of demand $d \in D$ during time period $\sigma \in S$
$\bar{r}^{d\sigma}$	Average bandwidth request of demand $d \in D$ during time period $\sigma \in S$ (robust variant)
$\hat{r}^{d\sigma}$	Worst case bandwidth request deviation of demand $d \in D$ during time period $\sigma \in S$ (robust variant)
ρ^d	Peak traffic value or nominal value of demand $d \in D$
n_{ij}	Number of line cards installed on link $(i,j) \in A$
c_{ij}	Capacity of a line card installed on link $(i,j) \in A$
$\bar{c}_{ij}^{\sigma}$	Capacity already used on link $(i,j) \in A$ during time period $\sigma \in S$
C_i	Capacity of node $i \in V$
$\bar{C}_i^{\sigma}$	Capacity already used on node $i \in V$ during time period $\sigma \in S$
μ_{ij}	Maximum link utilization allowed on link $(i,j) \in A$
μ_{ij}^{bck}	Maximum link utilization allowed on link $(i,j) \in A$ during failure periods
π_{ij}	Power required to keep activated a single line card of link $(i,j) \in A$

π_i	Power required to keep activated node $i \in V$
Π_{ij}	Energy profile function of a single line card of link $(i,j) \in A$
Π_i	Energy profile function of node $i \in V$
δ	Power required to switch on a chassis from the sleeping state, as a fraction of the hourly chassis consumption π_i
σ_{ij}^{dp}	Binary parameter, equal to 1 if link $(i,j) \in A$ belongs to path $p \in P^d$
ω_{max}	Maximum link weight allowed
h^σ	Duration of time period $\sigma \in S$
η_{on}	Maximum number of card switching on allowed over the entire set of periods S
E_c^{MIN-EN}	Energy budget to be respected during the *Min-Congestion* Stage of the Energy-Aware Lexicographic GRASP heuristic
$Cong_{(i,j)}$	Congestion cost function for link $(i,j) \in A$
$Cong_{(i,j)}^\sigma$	Congestion cost function for link $(i,j) \in A$ during time period $\sigma \in S$
α_h	Slope of piece $h \in H$
β_h	Offset of piece $h \in H$
Γ_{ij^σ}	Robustness parameter: total deviation allowed on link $(i,j) \in A$ during time period $\sigma \in S$
m^d	Maximum flow for demand $d \in D$ computed by considering unitary link capacity
Ω	Number of iterations of the path generation algorithm
Ξ	Scaling parameter for the traffic demands
g_i	Number of links connected to node $i \in V$
Δ^d	Number of TCP connections aggregated into an elastic demand $d \in D^e$
B	Maximum energy consumption allowed for the network
	Variables
f_{ij}	Real, non-negative: total amount of traffic carried by link $(i,j) \in A$
f_i	Real, non-negative: total amount of traffic carried by node $i \in V$
f_{ij}^d	Real, non-negative: amount of traffic of demand $d \in D$ carried by link $(i,j) \in A$
f_{ij}^t	Real, non-negative: amount of traffic carried by link $(i,j) \in A$ an destined to node $t \in V$
ϕ^d	Real, non-negative: transmission rate of elastic demand $d \in D^e$
y_i	Binary: equal to 1 if node $i \in V$ is powered on
y_i^σ	Binary: equal to 1 if node $i \in V$ is powered on during period $\sigma \in S$

x_{ij}^{d}	Binary: equal to 1 if the primary path of demand $d \in D$ uses link $(i,j) \in A$
$x_{ij}^{d\sigma}$	Binary: equal to 1 if the primary path of demand $d \in D$ uses link $(i,j) \in A$ during period $\sigma \in S$
ξ_{ij}^{d}	Binary: equal to 1 if the backup path of demand $d \in D$ uses link $(i,j) \in A$
$\xi_{ij}^{d\sigma}$	Binary: equal to 1 if the backup path of demand $d \in D$ uses link $(i,j) \in A$ during period $\sigma \in S$
x^{dp}	Real, non-negative in $[0,1]$: fraction of r^d sent along path $p \in P^d$
$\chi_{p}^{d'}$	Binary: equal to 1 if the primary path of demand $d \in D$ uses path $p \in P^d$
$\chi_{p}^{d''}$	Binary: equal to 1 if the backup path of demand $d \in D$ uses path $p \in P^d$
g_{ijkl}^{d}	Binary: equal to 1 if the backup path of demand $d \in D$ uses link $(i,j) \in A$, while the primary one uses link $(k,l) \in A$
w_{ij}^{d}	Integer in $[0, n_{ij}]$: number of active line cards on link $(i,j) \in A$
$w_{ij}^{d\sigma}$	Integer in $[0, n_{ij}]$: number of active line cards on link $(i,j) \in A$ during period $\sigma \in S$
u_{ij}^{t}	Binary: equal to 1 when link $(i,j) \in A$ belongs to a shortest path between i and t
z_{i}^{t}	Real, non-negative: traffic on each outgoing link of i which belongs to a shortest path between i and t
z_{i}^{σ}	Real, non-negative: energy to activate node i passing from period $\sigma - 1 \in S$ to period $\sigma \in S$
ω_{ij}	Real, non-negative: link weight of link $(i,j) \in A$
u_{ijk}^{σ}	Binary: equal to 1 if the k-th line card of link $(i,j) \in A$ is activated at the beginning of period $\sigma \in S$
l_{j}^{t}	Real, non-negative: minimum distance (according to the link weights) between j and t
$U(\phi)$	Real, non-negative: network utility related to demand rates
ι_{ij}	Real, non-negative: congestion cost on link $(i,j) \in A$
ι_{ij}^{σ}	Real, non-negative: congestion cost on link $(i,j) \in A$ during time period $\sigma \in S$
Θ_{ij}^{σ}	Worst-case deviation on link $(i,j) \in A$ during time period $\sigma \in S$
$v_{ij}^{d\sigma}$	Real, non-negative in $[0,1]$: fraction of traffic deviation of demand $d \in D$ on link $(i,j) \in A$ during time period $\sigma \in S$

$\epsilon_{ij}^{\sigma'}$	Real, non-negative: dual variables corresponding to Constraints (4.24)
$\epsilon_{ij}^{d\sigma''}$	Real, non-negative: dual variables corresponding to Constraints (4.25)
ϖ_{ij}^{d}	Binary: equal to 1 if link $(i,j) \in A$ is the bottleneck arc of demand $d \in D^e$
ϑ_{ij}	Real, non-negative: largest amount of flow routed on link $(i,j) \in A$
γ_{ijh}^{d}	Binary: equal to 1 when link $(i,j) \in A$ lies on the portion of the path used by demand $d \in D^e$ which connects o^d to $h \in V$
ψ_{h}^{d}	Binary: equal to 1 if demand $d \in D^e$ flows through node $h \in V^{o^d}$
ϱ^{d}	Real: piece-wise log-utility of each TCP connection carried by demand $d \in D^e$
λ_{h}^{d}	Real, non-negative: dual variables corresponding to Constraints (7.27)
κ_{i}^{d}	Real: dual variables corresponding to Constraints (7.9)
μ_{ij}	Real, non-negative: dual variables corresponding to Constraints (7.10)
θ_{ij}^{d}	Real, non-negative: dual variables corresponding to Constraints (7.11)
υ_{ij}^{d}	Real, non-negative: linearization variables

CHAPTER 1

Introduction

1.1 Definitions and Basic Concepts

Even if Internet and its related information and communication technology (ICT) are commonly considered as a powerful means to reduce the negative impact of human activities on the environment in terms of global warming and overall pollution through smart energy production and distribution, energy demand management, reduced travels, telework, etc, it cannot be ignored that, due the rapid expansion of the communication infrastructures worldwide and to the exponential traffic growth observed in the last decade, their energy footprint has reached by now a significant level. (Group, 2008).

The ICT sector is every year responsible for at least 2% (0.8 Gt CO_2) of global global greenhouse gas (GHG) emissions (Toure, 2008; Group, 2008; Forster *et al.*, 2009), a value which exceeds even the GHG emissions of the aviation sector (Ajmone Marsan *et al.*, 2009). According to (Vereecken *et al.*, 2008), at the current expansion rate, the emission amount of ICT is expected to grow by the year 2020 up to 1.4 Gt of CO_2, approximately 2.8% of global emissions.

In terms of energy, in 2007, ICT alone was estimated to be responsible from 2% to 10% of the world power consumption (Lubritto *et al.*, 2008; Koomey, 2007). According to recent studies, always in 2007, the energy demand of network equipment, excluding servers in data centers, was around 22 GW and should reach 95 GW in 2020 (Vereecken *et al.*, 2008), while the overall yearly consumption of broadband modems and routers was 35 TWh/y (Malmodin *et al.*, 2010). Other work states that the worldwide electricity consumption of Telecom operators grew from 150 Twh/y in 2007 to 260 TWh/y in 2012, around 3% of the total worldwide (Lambert *et al.*, 2012). Finally, data from single Internet service provider (ISP)s show that the largest ISPs, e.g., AT&T or China Mobile, consumed up to 11 TWh per year in 2010, while medium sized ones like Telecom Italia and GRNET are expected to reach 400 GWh in 2015 (Bolla *et al.*, 2012).

By looking more closely at the consumption distribution inside the network of a single ISP, recent studies show that about 20% of the power demand comes from the core/backbone part of the network, while the remaining 75% is consumed by the access/edge one (Bolla *et al.*, 2011b,c). However, due to technology evolution and to the increasing traffic volume, backbone networks consumption is expected to reach 40% of the total in 2017 (Lange, 2009), and exceed

50% in 2020 (Lange *et al.*, 2011; Hinton *et al.*, 2011). Furthermore, by considering the different hierarchical layers which Internet is built on, it is important to point out that IP/MPLS is currently responsible for at least 60% of the overall consumption (Van Heddeghem *et al.*, 2012b; Lange, 2009). By contrast, the optical layer is largely the more energy efficient one (Vereecken *et al.*, 2011).

For these reasons, the issues of energy saving in IP networks and of power awareness in network design (*Green Networking*), have recently become of great interest in the scientific community and have attracted the attention of device manufacturers and ISP (GreenTouch consortium, s.d.; Bolla *et al.*, 2011b; Bianzino *et al.*, 2012b; Zeadally *et al.*, 2011).

Starting from the seminal work by Gupta and Singh (Gupta et Singh, 2003), which evaluates the power demand of Internet and first proposes possible approaches to save energy, the research community has tried to reduce network energy consumption by pursuing three different, and at the same time, complementary strategies: (i) development of a new generation of green network devices with a higher energy efficiency, (ii) definition of methodologies for power aware network design, as well as (iii) energy management strategies to adapt the power consumption of networks to the observed traffic levels (Mellah et Sansò, 2009).

This class of techniques, which is at the core of this thesis, aims at dynamically performing traffic engineering (TE), i.e., adjusting the routing paths used to carry the Internet traffic, so as to jointly minimize the network power consumption and guarantee the desired network performance (see, e.g., Vasić et Kostić, 2010; Amaldi *et al.*, 2013c). These approaches are typically known under the name of energy-aware traffic engineering (EATE) or energy-aware network management (EANM).

Energy inefficiency in current IP networks is mainly due to two different reasons: (i) in current network devices, power consumption is not proportional to the utilization level, i.e., an active but idle device consumes almost as a fully used one (Chabarek *et al.*, 2008; Mahadevan *et al.*, 2009; Adelin *et al.*, 2010; Bonetto *et al.*, 2012), and (ii) network resources are typically over-dimensioned by network providers to guarantee the quality of service (QoS), so that during peak and low-traffic hours the maximum link utilization remains steadily below 50% and 10%, respectively (Fraleigh *et al.*, 2003). In this scenario, the most promising way to reduce the network power demand consists in putting to sleep the redundant network devices such as router chassis and line cards, and re-routing the traffic on the active elements. Note that a sleeping device may consume up to 90% less than an active idle one (Chabarek *et al.*, 2008). We pass thus from simple EANM to sleep-based energy-aware network management (SEANM). The validity of the idea of reducing network consumption by putting network elements to sleep has been confirmed, directly, by analytical studies on its applicability and its potentialities in terms of energy savings (see, e.g., Bolla *et al.*, 2012; Vetter *et al.*, 2012;

Idzikowski *et al.*, 2013a,b; Chiaraviglio *et al.*, 2013a), and, indirectly, by the large body of work exploiting sleeping approaches that appeared in the last five years (see, e.g., Amaldi *et al.*, 2013c; Zhang *et al.*, 2010; Lee *et al.*, 2012; Cianfrani *et al.*, 2012b; Bianzino *et al.*, 2012b).

Furthermore, despite some practical problems that must be addressed to transparently run a network with sleeping capabilities, like, for instance, how quickly the system needs to restart the sleeping devices in response to unexpected events, or how the connectivity of the latter is guaranteed along inactivity periods, recent technology improvements have concurred to overcome several implementation issues. Among these, we mention sleep-capable routers (Bolla *et al.*, 2010), new hardware which allows to wake-up a line-card in a few microseconds (Hays, 2007) and virtualization-based approaches which exploit the active routers to run the basic operations of the sleeping ones (Bolla *et al.*, 2011a). Moreover, most of the manufacturers are currently working on network devices to improve the efficiency of low energy modes and make their management more agile (GreenTouch consortium, s.d.).

1.2 Elements of the Problem

The aim of SEANM strategies is to maximize the portion of the network that can be put to sleep while guaranteeing the desired QoS to each traffic demand. To achieve this goal, the routing path of each traffic demand has to be optimized according to the incoming level of traffic. The development of a specific SEANM approach requires the careful evaluation of five different aspects: (i) the routing protocol, which tells us how TE has to be performed, (ii) the QoS requirements to be guaranteed to each traffic flow, (iii) when and how frequently the re-configurations should be performed, i.e., in advance off-line or in real-time on-line, (iv) where to place the intelligence of the system (centralized or distributed intelligence) and, finally, (v) how to guarantee network survivability.

1.2.1 Routing Protocols

The routing protocol is the key element of SEANM. It is the entity which dictates the rules used by the network to compute the routing paths, and naturally influences the way TE, i.e., the routing optimization, is performed by the SEANM procedure. From the optimization point of view, the routing protocol defines the structure of the underlying TE problem solved by the SEANM framework, while from the practical one, it determines how to apply the computed solutions in the real network.

In IP networks, there exist two main families of widely used routing protocols, which are

commonly identified as (i) flow-based routing (FBR), e.g., multi protocol Label Switching (MPLS) and (ii) shortest path routing (SPR), e.g., open shortest path first (OSPF).

With FBR, each traffic demand k having s_k and t_k as source and destination node, is routed along a dedicated path freely chosen between all the possible paths connecting s_k and t_k (Wang *et al.*, 2008). Performing TE constrained by FBR guarantees high flexibility because the routing paths are explicitly selected. By contrast, since each traffic demand is managed separately, a growing number of demands may considerably increase both the problem complexity, and, from a practical point of view, the time and overhead required to switch a flow from the old path to a new one.

With SPR, each traffic demand k having s_k and t_k as source and destination node, is routed along the shortest path between s_k and t_k (Altin *et al.*, 2009). The shortest paths are defined by the set of administrative weights assigned by the network operator to all the network links. The choice of the routing path is not free but constrained by the link weights. In this case, performing TE is equivalent to adjusting the link weights to obtain the optimal set of shortest paths. It is worth pointing out that, in the SEANM context, putting to sleep a certain link or router is equivalent to exclude it from all the shortest paths. This can be naturally accomplished by setting the corresponding weight to a very large value. The advantages of TE with SPR consists in the problem complexity strictly depending on the number of links (one weight for each link), which remain fixed independently of the number of traffic demands, and in the relative simplicity of the routing adjustment process, which only requires the update of few administrative weights stored in each router. As drawback, TE is constrained by the shortest path criterion, which restricts the choice of the routing paths and makes the corresponding path selection problem much more complex.

It is worth pointing out that, even if not directly considered in this thesis, in addition to FBR and SPR, there exists other alternative routing schemes, among which the more popular are those known as *tree-based*, which are used, for instance, to route multicast traffic or to build layer two VLANs (see, e.g., Capone *et al.*, 2012).

1.2.2 QoS Constraints

The possibility of putting to sleep a network element is naturally constrained by the capability of the network to provide the required QoS to the users. A first trivial way to avoid congestion and guarantee QoS is to impose link maximum capacity constraints. However, due to the uncertain nature of Internet traffic, it is commonly known that fully utilized link leads to infinite delays. A straightforward way to avoid this situation consists in imposing maximum link utilization constraints, e.g., 50%, to reserve enough spare capacity to absorb unexpected traffic variations or possible device failures. Another option, as done for instance

in (Fortz et Thorup, 2002), is to directly consider the non-linear nature of queuing delay in telecommunication networks so as to explicitly maintain a certain measure of network congestion under the acceptable limit.

It is worth pointing out that these classic approaches to guarantee QoS work correctly when all the traffic demands are considered inelastic, i.e., with a given transmission rate. For elastic traffic demands whose rate is dynamically adjusted by distributed congestion mechanisms (TCP), guaranteeing the QoS is equivalent to optimizing a certain utility measure related to the bandwidth used by each demand.

1.2.3 Optimization Frequency

Internet traffic is variable but characterized by a daily/weekly periodic profile (Bolla *et al.*, 2012) which allows ISPs to predict future traffic levels with very good accuracy. However due to the intrinsic variability of Internet traffic, a certain degree of uncertainty obviously affects the most accurate predictions. For this reason, two important questions arise when designing new SEANM approaches: (i) should network optimization be performed off-line during a planning phase according to traffic predictions, or on-line in real-time by considering direct traffic measurements? (ii) How often should the network be reconfigured in order to react to varying traffic conditions?

Performance of off-line approaches are clearly linked to the accuracy of the traffic predictions. Assumed that predictions are reliable, off-line methods have two main advantages: (i) more computing time available (in the order of hours), which allows to increase the complexity of the optimization framework, and (ii) direct control by network administrators, who can verify and evaluate the SEANM solutions before their applications. However, off-line methods do not offer any flexibility in reacting to unexpected conditions. At the opposite extreme, on-line approaches adapt the network configuration according to real-time measurements. On-line optimization guarantees a rapid reaction to condition variation, but, due to its real-time nature, is constrained by scarce time resources which typically do not allow to compute optimal solutions.

The second point concerning the re-configuration frequency is strictly related to the off-line/on-line nature of the approach. For off-line methods, due to the slow and periodic dynamic of Internet traffic, a reasonable strategy is to compute a set of configurations which are each one optimized for a certain time period identified in the planning phase, and then applied at a pre-determined moment during the day. On-line approaches continuously re-configure the network when needed. However, we need to avoid too frequent configurations in order to preserve network stability and limit the overhead.

As we will show in the reminder of the thesis, it is also possible to effectively combine both

off-line and on-line optimization to jointly offer configuration optimality, network stability and responsiveness to failures.

1.2.4 Decision Points

The energy-aware configuration of the network can be performed globally by a centralized entity, e.g., by a centralized network management platform or controller, or locally by multiple network agents, that take distributed decisions which should converge towards a stable network configuration. Centralized approaches typically rely on large sets of global measurements, which, if accurate, potentially allow to compute quasi-optimal solutions. However, to correctly operate, centralized mechanisms require an higher complexity, a substantial amount of computing power, and a dedicated architectures to collect data and disseminate the configuration parameters. As for distributed schemes, they rely on a limited amount of local data and base their strength on speed and simplicity. As drawback, they do not provide any guarantee of optimality. It is worth pointing out that centralized SEANM approaches typically rely on off-line methodologies, while distributed ones are naturally implemented as on-line mechanisms.

1.2.5 Network Survivability and Robustness

Despite the highest energy savings that can be obtained by perfectly tailoring the active capacity to the traffic level, there are certain network functions that require the presence of spare capacity to be used in response to unexpected events. This includes device failures, which are managed by specific protection techniques to reserve backup capacity to serve the ongoing traffic, and unexpected traffic variations, which can be easily absorbed in case some spare capacity is made available on the links. SEANM approaches should not ignore these network requirements to further reduce the network consumption. There exists thus a clear trade off between energy efficiency and network resilience/robustness.

1.3 Research Goals

The aim of our research plan was to propose a group of SEANM approaches to optimize the network consumption of backbone IP networks operated with different routing protocols and carrying different types of traffic. To guarantee network stability, solution optimality and full control by the network operator, we mainly focused our efforts on centralized approaches for three different scenarios, i.e., SEANM with shortest path routing (SEANM-SP), SEANM with flow-based routing (SEANM-FB) and SEANM with elastic traffic (SEANM-ET). Note

that other recent work on distributed SEANM mechanisms (see, e.g., Bianzino *et al.*, 2012d,c) showed that traffic unawareness makes this type of approaches very unstable.

For SEANM-SP we developed methods to find the optimal set of link weights which minimizes energy consumption and limits network congestion (Amaldi *et al.*, 2011a,b, 2013c).

For SEANM-FB we focused on methods to determine, along an entire day split in different sub-periods, the optimal network routing and the optimal element state (on or off) which maximally reduce the daily network energy consumption (Addis *et al.*, 2012b, 2014a). Special constraints to limit the routing choices along the entire day and preserve the life-time of the network devices have been considered, as well as additional restrictions to guarantee network resilience to single link failures (Addis *et al.*, 2012c,a, 2014b) and network robustness to unexpected traffic variations (Addis *et al.*, 2013).

Finally, for SEANM-ET we proposed a new framework to determine the optimal network routing for a set of elastic traffic which optimally balances network utility and network consumption according to trade-off parameters chosen by the network operator. Resource allocation for the elastic demands has been computed according to different paradigms such as max-min-fairness (MMF) or proportional fairness (PF), which allow to model the bandwidth allocation determined in IP networks by widely used congestion control mechanisms, e.g., TCP or per-flow fair queuing, (see Amaldi *et al.*, 2013d,b,a, where the first two work discuss general traffic engineering with elastic demands, while the latter specifically addresses the SEANM-ET).

In the context of SEANM-SP, we also developed a new open source management framework that we used to realistically implement a combined off-line/on-line variant of the original off-line methods (Capone *et al.*, 2013, 2014).

1.3.1 Document Structure

In chapter 2, we present an accurate literature review of green networking and related optimization problems. In Chapter 3, we present our optimization approaches for SEANM-FB, while in chapter 4, we discuss how to explicitly consider resilience to failures and robustness to traffic variations in the context of SEANM-FB. In chapter 5, we present some centralized approaches for SEANM-SP, and in chapter 6, we propose and on-line implementation of the algorithms proposed in chapter 5. Finally, we discuss in chapter 7 our novel optimization framework for SEANM-ET, and present some concluding remarks in chapter 8.

CHAPTER 2

Green Networking in IP Networks: an Overview

In this chapter we exhaustively discuss the state of the art literature on green networking and, more specifically, on energy-aware network management (EANM). EANM proposals are categorized according to the different features of the corresponding network optimization problem. In the mean time, we take this opportunity to contextualize our work within the state of the art and to point out the innovative aspects of our contributions.

2.1 Power Models

Since the first seminal work of Gupta et al. (Gupta et Singh, 2003), the scientific community as well as manufacturers and Telecom operators have focused significant efforts to make Internet greener and improve its energy efficiency.

The first step toward a more ecological and sustainable Internet is represented by the analysis of the power consumption profile of a single network element, which is crucial to correctly understand how energy consumption might be reduced and by which order of magnitude. According to different studies appeared in the last five years, network elements such as router chassis and line cards are characterized by a consumption profile quite well approximated by an ON-OFF (step) curve (Chabarek *et al.*, 2008; Mahadevan *et al.*, 2009; Mellah et Sansò, 2009; Van Heddeghem *et al.*, 2012a). That means that the energy required to keep a device in an idle active state makes up around 90% of the total power consumed when the utilization is at 100%.

In addition, (Adelin *et al.*, 2010) shows that (i) consumption peaks are observed whenever the quality of service (QoS) router configuration, e.g., number and type of active queues, is changed, (ii) border gateway protocol (BGP) route updates do not produce consumption increases, (iii) the energy cost of different queuing policies is nearly the same, and (iv) the power consumed by a rebooting router is on average 20% lower than that of an active one. To simplify the life of green networking researchers, Van Heddeghem et al. have recently published the open *Powerlib* (Van Heddeghem et Idzikowski, 2012) database, which contains detailed consumption figures of several networking devices, both IP and optical, produced by different vendors. Another recent work (Di Gregorio, 2013) has showed that packet inspection can be exploited to decrease router power consumption by serving throughput-critical and delay-critical traffic with two different queues.

Other work has been focused on the power consumption profiles of aggregate systems. Data reported in (Bonetto *et al.*, 2012) show that the power consumption of the point of presence (PoP) of a real Italian operator is mainly influenced by the external temperature rather than by the incoming traffic: the cooling systems are the main consumption sources. The aggregate power requirement of different logical parts of Internet (e.g home, access, metro and core networks) has been discussed in (Lange, 2009; Bolla *et al.*, 2011c, 2012), while the energy efficiency of specific network technologies (e.g DSL, FTTH, WDM) has been studied (Hinton *et al.*, 2011; Baliga *et al.*, 2011). Note that this work provides estimates for the future energy trend up to 2020.

2.2 Green Networking

Once network consumption figures are well understood at both single device and more aggregated levels, it is quite natural to identify the main guidelines that should be followed to improve the energy efficiency of Internet. As suggested in (Mellah et Sansò, 2009; Minami et Morikawa, 2008), this target can be pursued by (i) developing a new generation of green hardware which, in addition to novel energy efficient architectures, offers support for energy-aware primitives such as sleeping (Bolla *et al.*, 2010) or *pipeline Internet protocol (IP) forwarding* (Baldi et Ofek, 2009), (ii) implementing local management approaches which allow each single device to dynamically and independently adjust its own state (e.g the maximum link rate or the on/off state) according to real-time measurements (Nedevschi *et al.*, 2009), and (iii) performing at the network level the overall and coordinated management of network routing and device states (EANM), so as to jointly optimize power consumption and network performance (see e.g., Amaldi *et al.*, 2013c; Zhang *et al.*, 2010; Chiaraviglio *et al.*, 2012).

Due to the consumption inelasticity of current network devices, putting network elements in sleep or low-power mode is commonly considered as the most promising and effective strategy to reduce current network consumption and adapt it to the traffic levels. Some analytical studies like (Bolla *et al.*, 2011c, 2012) estimate around 50% the energy savings that could be potentially achieved with current network infrastructure.

Sleeping approaches can be applied to reduce single device consumption by means of so-called low power idle (LPI) techniques, which aim at dynamically putting to sleep and reactivating a single line card according, for instance, to the presence of packets queued in the input buffer (Gunaratne et Christensen, 2005; Nedevschi *et al.*, 2008). The effectiveness of such approaches strictly depends on the traffic characteristics, which determine how frequently and for how long an interface can be put to sleep without affecting the QoS, and on the amount of energy required to reactivate the interface itself (Bolla *et al.*, 2013). While

these distributed sleeping-based schemes do not require any routing optimization at the network level and are executed in an independent manner by each capable device, another class of sleep-based strategies, to which we refer in this thesis as sleep-based energy-aware network management (SEANM) approaches, aim at jointly adjusting both network routing and device states in a coordinate manner according to predicted or real-time traffic conditions.

SEANM, which is a branch of the more general EANM, is one of the most studied approaches in the field of green IP networks (Bolla *et al.*, 2011b; Bianzino *et al.*, 2012b; Zeadally *et al.*, 2011). A substantial body of work is focused on understanding how SEANM performance are influenced by different factors. (Chiaraviglio *et al.*, 2011, 2013b; P. Charalampou *et al.*, 2013) show that sleeping-based strategies maintain their effectiveness also in the presence of more energy-proportional devices provided that the fixed and proportional parts are at least of the same order of magnitude. (Bianzino *et al.*, 2010) evaluates the relationship between SEANM performance, energy profiles, topology structures and QoS constraints, (Cardona Restrepo *et al.*, 2009) introduces the concept of *Energy Profile Aware Routing*, (Song et Liu, 2012) derives some guidelines for green routing according to which traffic demands should be routed on the shortest per-hop paths and active links should be maximally utilized, and (Garroppo *et al.*, 2013b) points out that in case of convex energy profiles it would be better to maximally balance the network traffic on all the network resources rather than concentrating it on a smaller subset.

A key issue for SEANM is the need to limit the number of reconfigurations to protect network stability and performance. Limited reconfigurations bring problem in terms of (i) energy gap between the current configuration and the optimal one, and (ii) temporary inability of the running configuration to meet the requested QoS requirements (the most crucial one). However, recent work has showed that, due to the slow and periodic dynamic of Internet traffic, quasi-optimal savings and acceptable network performance could be achieved through a limited number of network configurations (in terms of routing and device states) to be efficiently applied along an entire day (Chiaraviglio *et al.*, 2011, 2013b). An alternative study on the trade-off between network performance and energy saving levels considers a scenario wherein a subset of network routers is randomly put to sleep for a given amount of time, and shows how average delay and average packet loss are negatively influenced by router sleeping (Sakellari *et al.*, 2013).

It is worth pointing out that in the literature on SEANM, it is a common practice to assume that once a certain network element (a router or a link) has no traffic, there should exist some automatic mechanisms able to put to sleep the element itself, and successively awake it in case of need. These mechanisms can rely, for instance, on network virtualization (Bolla *et al.*, 2011a) or LPI schemes (Nedevschi *et al.*, 2009).

Along with SEANM and LPI, an alternative EANM strategy exhaustively investigated in the last decade is commonly known with the name of adaptive link rate (ALR) (Bilal *et al.*, 2012). According to ALR, the capacity of each Ethernet links is adjusted, e.g., from 100 Mbps to 1 Gbps, to satisfy the incoming traffic. Since the higher the link capacity the higher the link consumption, the system should select, from time to time, the less consuming operative rate able to satisfy the incoming flows (Gunaratne et Christensen, 2006; Gunaratne *et al.*, 2006, 2008). The standardization of ALR and sleeping capabilities for Ethernet links is handled by Energy Efficient Ethernet (EEE), which is the standardized by the IEEE 802.3az engineering task force (Christensen *et al.*, 2010). To date, the widespread of such practices in real networks has been being limited by hardware and implementation issues. The generalization of the ALR concept is known as mixed line rates (MLR), according to which several discrete capacity values, included the zero-capacity state corresponding to the sleep mode, are available on each link. The analytical study presented in (Idzikowski *et al.*, 2013b) states that MLR approaches are more energy efficient than pure sleep-based ones.

EANM approaches can be naturally modelled as specific network optimization problems aiming at minimizing the network consumption while being constrained by QoS constraints (see e.g., Garroppo *et al.*, 2013a). The main structure of each optimization problem is typically determined by the routing protocol considered (e.g open shortest path first (OSPF) or multi protocol Label Switching (MPLS)), the network topology structure, e.g., single or multi layer, bundled or single links, etc., and the power saving mechanisms, e.g., sleeping, load balancing or ALR. In the next sections, we define the main EANM problems, discuss their related literature and point out the novelties of our novel approaches.

2.3 Energy-Aware Network Optimization

Let us first define some basic concepts and notation common to all EANM problems.

2.3.1 Problem Sets and Parameters

Given a directed graph $G = (V, A)$ to represent the network topology, where V and A are the sets of nodes and links and given a set of traffic demands D where each traffic demand $d \in D$ is uniquely determined by an origin node o_d, a destination node t_d and a bandwidth request r^d, the aim is to adjust the routing path used by each traffic demand $d \in D$ so as to minimize the overall network consumption while satisfying each traffic request.

Each network link $(i, j) \in A$ is equipped with n_{ij} line cards of capacity c_{ij} and network congestion is typically prevented by the network operator by imposing the total flow on each link to be smaller than $\mu_{ij} n_{ij} c_{ij}$, where $\mu_{ij} \in [0, 1]$ is the maximum utilization allowed on link

$(i,j) \in A$. Network nodes are split between *edge* nodes V^e, which can be both source and destination of traffic demands, and *core* nodes V^c, which are simple transit routers. As for the energy aspects, let π_i and π_{ij} be the power required to keep activated node $i \in V$ and a single line card of link $(i,j) \in A$ and let energy profile functions $\Pi_i(f)$ and $\Pi_{ij}(f)$ be the flow proportional power consumption component for node $i \in V$ and link $(i,j) \in A$.

2.3.2 Problem Variables

Suitable variables are defined to optimize both network routing and device power states. Let the flow scheme be multi-commodity. The total flow variables f_{ij} and f_i, which represent the total flow carried by each link $(i,j) \in A$ and by each node $i \in V$, are related to the non-negative continuous variables f_{ij}^d, representing the amount of traffic carried on link $(i,j) \in A$ and belonging to traffic demand $d \in D$. In case of single-path routing, we use the binary variables x_{ij}^d, which are equal to 1 if link $(i,j) \in A$ is used by the routing path of demand $d \in D$, and define the traffic generated by demand $d \in D$ flowing through link $(i,j) \in A$ as $r^d x_{ij}^d$. Note that this *per-arc* approach is not the only way to represent routing paths and link flows. The alternative is to adopt a *per-path* approach, according to which P^d is the set of paths connecting node o_d to node t_d, σ_{ij}^{dp} is the binary parameters equal to 1 if link $(i,j) \in A$ belongs to path $p \in P^d$, and x^{dp} are the binary (single path routing) or non-negative real variables in $[0,1]$ (multi-path routing) representing the fraction of r^d sent along path $p \in P^d$.

In case of sleeping-based approach, we define the binary variables y_i which are equal to 1 when node $i \in V^c$ (note that edge nodes $i \in V^e$ cannot be put to sleep because they are source or destination of traffic demands) is active, and the non-negative integer variables $w_{ij} \in [0, \ldots, n_{ij}]$ to represent the number of active line cards on link $(i,j) \in A$. It is worth pointing out that a MLR or ALR based approach can be modelled by correctly setting the n_{ij} parameters or by defining multiple parallel links, each one with its own capacity, of which only one can be activated at a certain time. As for methods aiming at optimizing the flow proportional power component, a proper energy profile functions $\Pi_i(f)$ and $\Pi_{ij}(f)$ have to be defined for each component.

2.4 Energy-Aware Network Management with Flow-Based Routing

One of the most popular routing protocol for IP networks is MPLS. In MPLS a dedicated routing path known under the name of Label Switched path (LSP) is explicitly assigned to each traffic demand $d \in D$. Most importantly, the path is freely chosen between all the possible paths connecting the source o^d to the destination t^d. Furthermore, an advanced function of MPLS allows to configure multiple paths for the same demand and adjust the

fraction of traffic carried by each path by properly setting the splitting ratios at each node crossed by the flows (Hannan *et al.*, 2000). The routing scheme of MPLS is a typical example of *flow-based* or *per-flow* routing. The problem of minimizing the network energy consumption in IP networks operated with such class of routing protocols belongs to the well known class of multi-commodity network design (MCND) optimization problems (Gendron *et al.*, 1998; Pióro et Medhi, 2004). A general formulation for EANM with flow-based multi-path routing and sleep capable devices (SEANM with flow-based routing (SEANM-FB) problem) can be expressed by the following mixed integer linear programming (MILP) formulation (a generalized version of that proposed in Garroppo *et al.*, 2013a; Addis *et al.*, 2014a):

$$\min \sum_{(i.j)\in A} \left(\pi_{ij} w_{ij} + \Pi_{ij}\left(f_{ij}\right)\right) + \sum_{i\in V} \left(\pi_i y_i + \Pi_i\left(f_i\right)\right) \tag{2.1}$$

$$w_{ij} \leq n_{ij} y_i, \qquad \forall (i,j) \in A \tag{2.2}$$

$$w_{ij} \leq n_{ij} y_j, \qquad \forall (i,j) \in A \tag{2.3}$$

$$\sum_{\substack{j\in V:\\ (i,j)\in A}} f_{ij}^d - \sum_{\substack{j\in V:\\ (j,i)\in A}} f_{ji}^d = \begin{cases} r^d & \text{if } i = o_d, \\ -r^d & \text{if } i = t_d, \\ 0 & \text{otherwise} \end{cases}, \qquad \forall i \in V,\, d \in D \tag{2.4}$$

$$\sum_{d\in D} f_{ij}^d \leq f_{ij}, \qquad \forall (i,j) \in A \tag{2.5}$$

$$\sum_{(j,i)\in A} f_{ji} + \sum_{\substack{d\in D:\\ o_d = i}} r^d \leq f_i, \qquad \forall i \in V \tag{2.6}$$

$$f_{ij} \leq \mu_{ij} c_{ij} w_{ij}, \qquad \forall (i,j) \in A \tag{2.7}$$

$$f_{ij},\, f_{ij}^d,\, f_h \geq 0, \quad \forall (i,j) \in A,\, h \in V,\, d \in D \tag{2.8}$$

$$y_i \in \{0,1\}, \qquad \forall i \in V \tag{2.9}$$

$$w_{ij} \in [0, \ldots, n_{ij}], \qquad \forall (i,j) \in A. \tag{2.10}$$

The Objective function (2.1) minimizes both fixed and proportional consumption components of routers and links. Constraints (2.2–2.3) force to put in stand by mode all the links connected to a sleeping router (in the reminder of the paper we use standby mode, sleeping mode or simply off to indicate that a device is in a low consumption state), while Equation (2.4) represents the multi-path flow conservation constraints. Finally, Constraints (2.5–2.6) determine the total amount of traffic flowing through a link or a node, respectively, and Equation (2.7) represents link maximum utilization constraints.

2.4.1 State of the Art for SEANM-FB

By taking as reference the basic SEANM-FB problem (2.1–2.10), we now specifically survey the recent literature on EANM dealing with flow-based routing protocols and point out the differences with respect to the reference problem (2.1–2.10). We will also analyze the difference between state of the art methodologies and our novel approaches for SEANM-FB discussed in Chapter 3.

The first work on SEANM-FB deals with a simplified version of (2.1–2.10) where only the fixed power consumption component is considered, i.e., $\Pi_{ij}(f_{ij}) = 0$ and $\Pi_i(f_i) = 0$, (Chiaraviglio *et al.*, 2009, 2012). Note that, due to the relative small incidence of the proportional component (smaller than 10% of the total consumption) (Chabarek *et al.*, 2008), this is often ignored in state of the art EANM procedures. In their work, Chiaraviglio et al. propose some greedy heuristics to maximize the number of sleeping nodes and links (with only one card per link, e.g., $n_{ij} = 1$, $\forall (i,j) \in A$) which first, sort nodes and links according to a specific given criterion, and then, if capacity constraints are respected, put to sleep one by one each element of the sorted list. The procedures take as input a traffic matrix and a network topology with link capacity and a non-negative weight assigned to each link. Traffic is routed on the shortest paths determined by the link weights. An element can be put to sleep only if the routing determined by the link weights of the active links allows to satisfy maximum utilization constraints (2.7). These heuristics should be performed in real-time by a centralized network controller able to dynamically estimate real time traffic matrices. The authors do not explicitly investigate the issues concerning network stability and re-configuration frequency. The same problem addressed in (Chiaraviglio *et al.*, 2012) is also tackled in (Lee et Reddy, 2012), where the authors propose a novel greedy heuristic that builds the sub-optimal network, in terms of energy consumption, by adding the required links and nodes to the minimal steiner,tree that all the edge nodes.

Proportional non-linear power consumption components are considered by other heuristic algorithms which cope with a variant of the problems addressed in (Chiaraviglio *et al.*, 2012; Lee et Reddy, 2012) (Garroppo *et al.*, 2011, 2012; Giroire *et al.*, 2010). The logical scheme of the heuristics is very close to that of (Chiaraviglio *et al.*, 2009, 2012). The only difference concerns the feasibility test performed to determine if an element can be put to sleep, which is based on the solution of a convex formulation (Garroppo *et al.*, 2011, 2012), or on a novel shortest-path based algorithm (Giroire *et al.*, 2010). The non-linear formulations heuristically solved in (Garroppo *et al.*, 2011, 2012) are successively analyzed and tested in (Garroppo *et al.*, 2013a). It is worth pointing out that one of the formulations discussed in (Garroppo *et al.*, 2013a) refers to the case of bundled links characterized by multiple independent line cards available on each link ($n_{ij} > 1$).

This scenario is closely related to MLR and ALR problems and has been addressed by a large body of work proposing new heuristics for on-line optimization (algorithms run periodically every 5 or 15 minutes). (Fisher *et al.*, 2010; Lin *et al.*, 2013) present some heuristic algorithms to quickly determine the sub-optimal set of line cards to be put to sleep. The procedures proposed in (Fisher *et al.*, 2010) are based on multiple resolutions of a specific linear programming (LP) formulation that maximizes the network spare capacity, while those presented in (Lin *et al.*, 2013) exploit some k-shortest-path based algorithms aiming at identifying the most efficient routing paths. While these approaches put to sleep only line cards, the methods proposed in (Mumey *et al.*, 2012), which exploit some shortest-path and tree based algorithms, allow switching-off network nodes. A multi-layer IP over wavelength division multiplexing (WDM) variant of the classic bundled links problem is addressed in (Wu *et al.*, 2013), where the authors relate the number of active interfaces to the number of active logical links at the optical layer. Both a MILP formulation and a Lagrangian relaxation (LR) heuristic are presented and tested. Another heuristic to optimize the number of sleeping line cards in each bundle is presented in (Charalambides *et al.*, 2013). The approach is based on the concept of multi-topology routing and aims at computing the optimal node splitting ratio between the different virtual topologies. Finally, to end the survey on bundled links based approaches we mention the work published in (Galán-Jiménez et Gazo-Cervero, 2013b,a), where the authors evaluate how the number and the size of link energy levels, i.e., number ad capacity of a group of line cards of the same bundled link, influence the consumption reduction. The analysis is performed by executing two novel heuristic procedures based, respectively, on a genetic algorithm and on particle swarm optimization.

Some novel approaches which, along with flow-based routing, consider special stand by states for both routers and line cards are presented in (Kist et Aldraho, 2011; Cianfrani *et al.*, 2012a; Coiro *et al.*, 2013b). The work of Kist et al. (Kist et Aldraho, 2011) is the first proposing to model special stand by states which allow to modularly deactivate different router capabilities (each one responsible for a certain amount of energy consumption). In particular, in addition to classic sleeping and active states, the authors introduce (i) *bridged-all*, (ii) *bridged-local*, (iii) *default-gateway* and (iv) *bridged-many* states. An exact MILP formulation able to correctly model all these states is presented and evaluated. A novel heuristic for the same problem exploiting the Floyd-Warshall algorithm is proposed in (Cianfrani *et al.*, 2012a). Note that the definition of new stand by states for network devices can be applied to line-cards (instead of routers) too, and a novel ad-hoc heuristic method has been presented in (Coiro *et al.*, 2013b).

A different perspective is assumed in (Vasić et Kostić, 2010), where the authors present some on-line distributed procedures for energy-aware traffic engineering (EATE) aiming at

dynamically adjusting the amount of traffic sent by each edge router through a pre-determined set of paths, in order to minimize the current network consumption. The reference optimization problem addressed in (Vasić et Kostić, 2010) is exactly (2.1–2.10). The algorithms, which are based on a local search scheme that shifts traffic between different paths, are tuned to work with both on-off (SEANM case) and stepwise proportional (ALR or MLR case) energy profiles. As with other work, instability and reconfiguration frequency are not completely addressed. A similar on-line approach which exploits off-line optimization to pre-compute routing paths is proposed in (Vasic *et al.*, 2011). The authors present a novel distributed architecture called REsPoNse which, at each node, adjusts the fraction of traffic sent on each stored path. The set of pre-computed paths is split among critical always-on paths and on-demand paths that can be exploited in case of congestion or device failures. Joint dynamic routing optimization and admission control are instead considered in (Avallone et Ventre, 2012), where a framework which progressively computes the less consuming path for each incoming demand is proposed.

DAISES, a new distributed architecture for IP/MPLS network, is illustrated in (Coiro *et al.*, 2013a). The framework aims at dynamically adjusting the multiple LSPs used by each traffic demand by means of shortest-paths determined by a set of link weights which are both energy consumption and congestion related. Energy/congestion related weights are considered in (Hou *et al.*, 2014) too, where a new hop-by-hop routing protocol to implement in a distributed way a modified version of the well-known flow-deviation method (Fratta *et al.*, 1973) is presented.

The only work which explicitly considers multi-period optimization in SEANM-FB is that of Francois et al. (Francois *et al.*, 2013a). Their centralized off-line approach is built on the observation that along an entire day it should be possible to implement a low consumption configuration during the low traffic periods, and a more consuming one during peak phases. The authors propose a heuristic algorithm which optimizes the low traffic configuration so as to jointly minimize the network power consumption and the length of its applicability window (a configuration which saves 40% of energy and can be applied for 8 hours is better than another one that reduces consumption up to 60% but is feasible for only one hour).

Finally, we mention some work dealing with the SEANM-FB problem by means of game-theory approaches (Bianzino *et al.*, 2012a; Zhao *et al.*, 2013), a MILP formulation and a heuristic algorithm to solve a problem where network routers can perform the so-called redundancy elimination functionality (Giroire *et al.*, 2012), a new management architectures for IP/MPLS networks which relies on a novel heuristic to optimize both routing paths and device states (Niewiadomska-Szynkiewicz *et al.*, 2013), a study on the complexity of SEANM-FB problems aiming at identifying possible practices, such as use of pre-computed

paths, to make the optimization models more scalable (Arabas *et al.*, 2012).

2.4.2 Our Contribution

In this thesis we address a completely novel multi-period planning problem for IP/MPLS networks. We imagine that, due to the slow and periodic dynamic of Internet traffic, network operator can split a single day among multiple fixed time periods characterized by a certain level of traffic. Assuming the availability of predicted traffic matrices for the considered time intervals during the following days (Rahman *et al.*, 2006), we address the problem of optimizing, in advance and in a centralized manner, the LSPs (the dedicated routing paths) used by the expected traffic demands, and the power state (active or sleeping) of router chassis and router line cards. We consider single path unsplittable routing, multiple line cards of the same capacity available on each link (bundled links), routing constraints to maintain the routing unchanged along all the time periods, and novel card-state constraints to avoid reactivating a line card too many times during the same day in order to preserve its lifetime. We consider also survivable and robust variants that will be discussed later. We have first formulated two exact MILP formulations, (i) power-aware fixed routing problem (PAFRP) and (ii) power-aware variable routing problem (PAVRP), that can be solved at optimality in an acceptable amount of time (a few hours) by commercial solver (like CPLEX) for networks up to 20 nodes and 40 links (Addis *et al.*, 2014a). We have then developed both a greedy heuristic called energy-aware lexicographic GRASP (EA-LG) to rapidly obtain near-optimal solutions (Addis *et al.*, 2012b) and a mathematical programming heuristic algorithm named energy-aware single time-period heuristic (EA-STH) that deals one by one with each time period by solving a single period version of the original formulation (Addis *et al.*, 2014a). Note that, thanks to its reduced complexity, this algorithm can be implemented online based on traffic matrices measured in real time in the network.

2.5 Energy-Aware Network Management with Shortest Path Routing

Along with MPLS and its per-flow routing scheme, a second class of intra-domain routing protocol largely used in IP networks adopts a pure shortest path approach which does not allow the network administrator to explicitly configure the routing paths. The most popular shortest path routing protocol is OSPF. In OSPF an administrative weight is assigned by the network operator to each link, and traffic demands are then routed on the shortest paths determined by the link weights themselves. In case of multiple shortest paths available for a given demand, it is possible to enable the equal cost multi-path (ECMP) functionality,

according to which at a given node the traffic directed toward the same destination is equally split among the multiple shortest-paths.

In IP networks operated with shortest path routing protocols, optimizing the routing paths (traffic engineering) is not as straightforward as in those running flow-based ones. With shortest path routing, paths cannot be explicitly and freely selected as in flow-based one, because constantly constrained by the shortest path rule. To adjust the routing path it is instead necessary to modify the link weights used to compute the shortest path trees (see Altin *et al.*, 2009, for a survey on traffic engineering (TE) with shortest path routing).

From the practical point of view, this feature considerably simplifies the routing implementation, which in this case involves the management of a limited number of administrative weights ($|A|$ weights) and does not depend on the number of traffic demands. However, the shortest path limitation reduces the feasible path configurations and adds a substantial degree of complexity to the corresponding TE problem. While it has been demonstrated that the first issue does not represent a real limitation in large networks (see Proposition 4.1 in Pióro et Medhi, 2004), the complexity increase makes formulation for TE with shortest path routing hardly tractable also for small-sized networks.

Using the notation previously defined at the beginning of the chapter and by introducing proper variables and constraints to model shortest path routing, it is possible to write a general formulation (restated version of that proposed in Amaldi *et al.*, 2013c) for energy minimization in networks operated with OSPF and ECMP enabled, i.e., SEANM with shortest path routing (SEANM-SP). Shortest path routing is defined by means of four new groups of variables: binary variables u_{ij}^t, which are equal to 1 when link $(i,j) \in A$ belongs to at least one shortest path connecting node i to node t, non-negative real variables z_i^t and ω_{ij}, which represent, respectively, the flow value assigned to all the outgoing links of i which belong to the shortest paths from i to t and the link weight of link $(i,j) \in A$, and, finally, the non-negative real variables l_j^t, which are equal to the minimum distance (according to the link weights) between node j and node t. We add to problem (2.1–2.10) the following constraints:

$$0 \leq z_i^t - \sum_{\substack{d \in D: \\ t_d = t}} f_{ij}^d \leq (1 - u_{ij}^t) \sum_{\substack{d \in D: \\ t_d = t}} r^d, \quad \forall t \in V^e, (i,j) \in A \tag{2.11}$$

$$\sum_{\substack{d \in D: \\ t_d = t}} f_{ij}^d \leq u_{ij}^t \sum_{\substack{d \in D: \\ t_d = t}} r^d, \quad \forall t \in V^e, (i,j) \in A \tag{2.12}$$

$$0 \leq l_j^t + \omega_{ij} - l_i^t \leq (1 - u_{ij}^t)M, \quad \forall t \in V^e, (i,j) \in A \tag{2.13}$$

$$1 - u_{ij}^t \leq l_j^t + \omega_{ij} - l_i^t, \quad \forall t \in V^e, (i,j) \in A \tag{2.14}$$

$$u_{ij}^t \leq w_{ij}, \quad \forall t \in V^e, (i,j) \in A \tag{2.15}$$

$$\omega_{ij} \geq (1 - w_{ij})\omega_{max}, \quad \forall (i,j) \in A \tag{2.16}$$

$$1 \leq \omega_{ij} \leq \omega_{max}, \quad \forall (i,j) \in A \tag{2.17}$$

$$u_{ij}^t \in \{0,1\}, \quad \forall t \in V^e, (i,j) \in A \tag{2.18}$$

$$l_h^t,\ z_h^t,\ \omega_{ij} \geq 0, \quad \forall h,t \in V^e, (i,j) \in A. \tag{2.19}$$

Objective function (2.1) and Constraints (2.2–2.10) are taken untouched from the general formulation for SEANM-FB. However, the choice of the routing paths, which is determined by the non-negative values of the flow variables f_{ij}^d, is now constrained by the new group of Constraints (2.11–2.19) to define a shortest path routing compatible with the link weights.

Constraints (2.11) make sure that the value of traffic destined to node $t \in V^e$ and carried by link $(i,j) \in A$ is equal to z_i^t if link $(i,j) \in A$ belongs to a shortest path from i to t. Constraints (2.12) set to 0 the traffic destined to node $t \in V^e$ and carried by link $(i,j) \in A$ if it is not on the shortest path to node t. Constraints (2.13–2.14) determine a shortest path routing vector u consistent with the link weight vector ω and state the optimality conditions for the shortest path problem, while Equation (2.15) excludes sleeping links from any shortest paths. Finally, Constraints (2.16) force a sleeping link to assume the highest feasible weight ω_{max} and Constraints (2.17–2.19) define the variable domains.

2.5.1 State of the Art for SEANM-SP

Due to the extreme complexity of (2.1–2.19) and its related variants (see Amaldi *et al.*, 2013c, for a performance analysis), SEANM-SP approaches has received less attention than SEANM-FB problems. However in the last two years, driven by the promising performance of the first work appeared in literature on the energy-aware optimization of the link weights (see e.g., Amaldi *et al.*, 2011a,b; Phillips *et al.*, 2011; Lee *et al.*, 2012), this trend has been reversed.

The problem of centrally optimizing the link weights to jointly minimizing network consumption and network congestion has been addressed in (Phillips *et al.*, 2011; Lee *et al.*, 2012; Shen *et al.*, 2012; Francois *et al.*, 2013b).

The approach for off-line optimization presented in (Phillips *et al.*, 2011) aims at computing in centralized manner a finite set of energy-aware configuration for OSPF-operated IP networks to be periodically applied during the day (according to a computed schedule). A first genetic algorithm is exploited to find the sub-optimal configurations that satisfy an esti-

mated traffic matrix representing specific levels of traffic. A single configuration determines the states of link and routers. Note that in this work, the link weights of active elements are kept unchanged. A second genetic algorithm is then exploited to select which of the computed configurations should be used and for how long, in order to guarantee a quasi-optimal energy consumption without the need of continuously adjusting the configurations.

A comprehensive optimization approaches where both device states and link weights are jointly optimized is proposed in (Lee *et al.*, 2012). Starting from a per-path formulation to maximize the energy savings by putting to sleep the network links, the authors first derive a Lagrangian relaxation of the basic problem. Then, they develop three heuristic algorithms: (i) the LR algorithm based on the progressive update of the dual multipliers of the capacity constraints, (ii) the Harmonic Series (HS) algorithm which exploits a local search scheme to update and improve an input set of link weights, (iii) a combined LRHS heuristic where the weights computed by the LR algorithm are given as input to the HS. The authors believe that the proposed algorithms, which have been originally designed to be executed off-line by a centralized controller, are fast enough to be integrated in an on-line optimization framework too.

The link weight optimization method presented in (Shen *et al.*, 2012) addresses a different problem where it is assumed that the network operator can freely configure the splitting ratios between multiple shortest-paths. The proposed heuristic approach finds the energy-aware link weights which maximize the number of sleeping links and then adjusts the splitting ratios so as to not violate the maximum link utilization. It is worth pointing out that, though technically possible, the splitting ratios configuration is typically avoided by network operators and not available in all the network devices.

The joint optimization of network consumption and network congestion is tackled in (Francois *et al.*, 2013b), where the authors present an optimization scheme based on a genetic algorithm tested with multiple objective functions.

Another work considering shortest path routing but without the possibility of optimizing the link weights has been very recently proposed in (Coiro *et al.*, 2014). The authors consider the table-look operation first discussed in (Coiro *et al.*, 2013b) with flow-based routing, and present a new ad-hoc genetic algorithm to solve the energy minimization problem with shortest path routing.

Along with centralized off-line approaches, energy-aware link weight optimization in IP networks operated with OSPF is addressed by the fully distributed on-line frameworks proposed in (Bianzino *et al.*, 2012c,d). These methods exploit the OSPF link state packets to disseminate link load information among all the routers. Then, according to a given policy, the whole set of routers (Bianzino *et al.*, 2012c), or each single router independently (Bianzino

et al., 2012d), is periodically responsible for putting to sleep the most suitable link (it can be done by using a very high weight for the considered link). When the flow redirection caused by the link switching-off produces a maximum utilization violation, the last switched-off link is immediately reactivated.

Finally we mention the Energy Aware Routing (EAR) algorithm first presented in (Cianfrani *et al.*, 2010) and successively enhanced in (Cianfrani *et al.*, 2011, 2012b), which assumes a completely different approach based on a restated version of the OSPF protocol. This new algorithm first select a subset of routers identified as Importer Routers (IR), that are exempted from computing their own shortest path trees (SPTs). Then IRs route the incoming traffic by considering the SPT of a neighbouring router, identified as Exporter Router (ER). The basic idea behind this protocol modification is that a smallest number of active SPTs should naturally increase the number of links to be put to sleep because not belonging to any shortest path. These links are clearly the ones that can be switched off. It is very important to remark that EAR has no knowledge of the traffic matrices, and as highlighted by the authors, should be implemented only during low traffic periods characterized by a very low link utilization (under 10%).

2.5.2 Our Contribution

We consider IP networks operated with OSPF and ECMP enabled. We address the problem of optimizing the administrative link weights of OSPF in a centralized manner, by means of a network management platform which is responsible for both monitoring network conditions and modifying network configurations. Our aim is to find the set of link weights that lexicographically minimizes, in the following order, network energy consumption and network congestion. Practically speaking, *lexicographic* means that power consumption is the first objective to be minimized, while network congestion is optimized in a second phase by considering the reduced energy-efficient topology. Energy is minimized by putting to sleep both network nodes and network links (not only links as in Lee *et al.*, 2012; Shen *et al.*, 2012) by means of very high link weights which shift all the traffic away from the powered-off elements. We assume, as typically done in literature, that there exist some sort of mechanisms which autonomously puts the idle devices in a low consumption mode.

Due to the slow and periodic dynamic of Internet traffic, as later suggested in (Chiaraviglio *et al.*, 2013a; Francois *et al.*, 2013b) we assume that, within a single day, network operators can identify a limited subset of periods characterized by a specific and quite constant level of traffic, e.g., morning, lunch time, afternoon, evening, night. A dedicated set of energy-aware OSPF link weights is then computed for each time period by considering, for each of them, a predicted traffic matrix. Note that, to guarantee network stability and

solution optimality, and provide 100% control on the optimization process to network operators, which otherwise would be reluctant to dynamically adjust network configuration, we place the weight optimization within an off-line planning phase. We believe that frequent reconfigurations should be avoided to preserve network stability and performance, and each weight configuration should be ideally maintained for the entire duration of the corresponding time period. In this regard, suitable values for the link maximum utilization parameters $\mu_{ij} \in [0, 1]$ preserve the capability of the network to support unpredictable traffic variations, also when a subset of network devices is powered off. It is worth pointing out that since we leave the OSPF protocol unchanged (differently from Cianfrani *et al.*, 2012b), the simple optimization of the link weights does not interfere with other important network routines like failure management.

To solve the single period link weight optimization problem, where, given a network topology, a traffic matrix and maximum link utilization values, we look for the OSPF weight set which lexicographically minimizes energy consumption and network congestion, we developed: (i) a greedy-based algorithm called greedy algorithm for energy saving (GA-ES) (Amaldi *et al.*, 2011a) and (ii) its greedy randomized adaptive search procedure (GRASP) variant called greedy randomized algorithm for energy saving (GRA-ES) (Amaldi *et al.*, 2011b), (iii) a mathematical programming heuristic named two-stage algorithm for energy saving (TA-ES) (Amaldi *et al.*, 2011a) and (iv) a final heuristic known as MILP-based algorithm for energy-aware weight optimization (MILP-EWO) derived from the efficient combination of GA-ES and TA-ES (Amaldi *et al.*, 2011b, 2013c). Besides being, to the best of our knowledge, the first algorithms to consider link weight optimization as mean to reduce energy consumption, our procedures have been proved to outperform other state of the art algorithms like those presented (Lee *et al.*, 2012) (see experimental results in Amaldi *et al.*, 2013c).

A complementary problem to be addressed to make our approach complete is that concerning the scheduling of the computed OSPF configurations along an entire day. Although this operation may be planned off-line according to the time intervals previously identified, we opted to take an on-line approach where configuration switching decisions are driven by real-time measurements on the network conditions. We therefore developed an open source network management framework called JNetMan (Capone *et al.*, 2014), which allowed us to implement a real network controller able to dynamically monitor the link loads and dynamically apply the link weight sets computed off-line by one of our heuristic algorithms (Capone *et al.*, 2013). Preliminary results on emulated test-beds show that a dynamic switching approach based on average and maximum utilization criteria guarantees network performance and prevent the network from oscillating between different configurations.

2.6 Energy Minimization and Network Survivability

The aim of EANM is to drastically reduce the energy footprint of redundant network resources while guaranteeing that enough capacity is constantly available to satisfy the incoming traffic. However, it is crucial to keep in mind that redundant resources, although energy consuming and often strongly underutilized, play a key role to guarantee the network capability to react to unexpected events like device failures or significant traffic variations. For this reason, the energy cost evaluation of protection techniques (to guarantee network resilience) and their explicit implementation within traditional EANM approaches have received an increasing attention in the last three years.

Although several different protection and restoration techniques have been proposed in the last decades (Vasseur *et al.*, 2004), a typical way to implement network protection against link (node) failures consists in defining a backup path for each traffic demand. The backup path has to be link (node) disjoint from the main one and it is used to carry traffic only in case of failure of one of the links (nodes) used by the primary path. Due to the possibility of explicitly expressing the routing paths, MPLS, and more in general flow-based protocols, are naturally used to implement such types of protection approaches.

Two very popular protection schemes that can be implemented in IP/MPLS networks are known as *dedicated protection* and *shared protection*. According to them, a single unsplittable backup path is defined for each traffic demand, and a certain amount of spare capacity is reserved along links and nodes used by the backup path itself.

Since single node or multiple link failures are very unlikely in common IP network scenarios, dedicated and shared protection are typically designed to protect the networks against single-link failures. With respect to the basic formulation (2.2–2.10) for SEANM-FB, the implementation of some sort of protection requires (i) the transition from a splittable multipath routing to an unsplittable single path one, (ii) the introduction of new routing variables to define the backup paths, and (iii) the addition of new constraints to compute the resource allocation on the backup paths themselves.

Let us introduce the new binary variables x_{ij}^d (ξ_{ij}^d) which are equal to 1 when link $(i,j) \in A$ belongs to the primary (backup) path of demand $d \in D$. To determine a single unsplittable primary path for each demand, original flow conservation Constraints (2.4) are modified as follows

$$\sum_{\substack{j\in V:\\(i,j)\in A}} x_{ij}^d - \sum_{\substack{j\in V:\\(j,i)\in A}} x_{ji}^d = \begin{cases} 1 & \text{if } i = o_d, \\ -1 & \text{if } i = t_d, \\ 0 & \text{otherwise} \end{cases}, \quad \forall i \in V,\, d \in D. \tag{2.20}$$

Then, we add three groups of constraints which define single unsplittable backup paths and force backup paths to not share any links with the corresponding primary paths:

$$\sum_{\substack{j\in V:\\(i,j)\in A}} \xi_{ij}^d - \sum_{\substack{j\in V:\\(j,i)\in A}} \xi_{ji}^d = \begin{cases} 1 & \text{if } i = o_d, \\ -1 & \text{if } i = t_d, \\ 0 & \text{otherwise} \end{cases}, \quad \forall i \in V,\, d \in D \tag{2.21}$$

$$x_{ij}^d + \xi_{ij}^d \leq 1, \quad \forall (i,j) \in A,\, d \in D \tag{2.22}$$

$$x_{ij}^d + \xi_{ji}^d \leq 1, \quad \forall (i,j) \in A,\, d \in D. \tag{2.23}$$

Finally, link capacity Constraints (2.7) have to correctly consider both primary and backup resources. The way spare capacity is assigned to backup paths differs from dedicated to shared protection. The first requires that the same amount of bandwidth is reserved on both primary and backup path:

$$\sum_{d\in D} r^d \left(x_{ij}^d + \xi_{ij}^d\right) \leq \mu_{ij} c_{ij} w_{ij}, \quad \forall (i,j) \in A. \tag{2.24}$$

The scheme adopted with shared protection is substantially different and allows a more efficient distribution of the backup resources. Assuming that simultaneous link failures never occur, shared protection allows backup paths whose corresponding primary paths are link-disjoint to share the same backup capacity. In fact, these backup paths will never be used simultaneously. To correctly compute the amount of backup capacity reserved on each link, we consider the link failure scenario which leads to the largest shift of traffic on the link. Note that the impact of a single link failure is determined by the sum of the demands whose backup paths are routed on the link and whose primary paths are affected by the failure. To correctly compute the backup bandwidth reserved on each link, we introduce a new group of binary variables g_{ijkl}^d, which are equal to 1 if traffic demand $d \in D$ is rerouted on link $(i,j) \in A$ when a failure occurs on link $(k,l) \in A$, i.e., if traffic demand $d \in D$ uses a primary and a backup paths routed, respectively, on link $(i,j) \in A$ and link $(k,l) \in A$. The following

constraints are then added instead of 2.7 to respect the link maximum utilization:

$$g^d_{ijkl} \geq x^d_{ij} + \xi^d_{kl} - 1, \quad \forall (i,j),(k,l) \in A,\, d \in D, \tag{2.25}$$

$$\sum_{d \in D} r^d \left(x^d_{ij} + g^d_{klij} \right) \leq \mu_{ij} c_{ij} w_{ij}, \qquad \forall (i,j),(k,l) \in A. \tag{2.26}$$

2.6.1 State of the Art on SEANM with Survivability Requirements

Resilience to failures

The negative impact of pure energy-efficient network design in terms of network reliability was investigated for the first time in (Sanso et Mellah, 2009). Driven by these considerations, the research community has recently started to increase its attention on network survivability requirements when performing EANM.

It is worth pointing out that, though EANM is very often implemented at the IP level, a large body of work (the most significant) on survivable EANM directly deals with the optical/WDM domain (see e.g., Cavdar *et al.*, 2010; Jirattigalachote *et al.*, 2011; Bao *et al.*, 2012). In this particular case, the target of the optimization is represented by the lighpaths, which are split among primary and backup light-paths. The resources on the backup light-paths are reserved according to the implemented protection scheme, and optical network devices which are idle or exploited to carry backup lightpaths only are put to sleep.

(Muhammad *et al.*, 2010; Monti *et al.*, 2011; Jirattigalachote *et al.*, 2011) implement dedicated protection. They propose an ILP formulation (Muhammad *et al.*, 2010) and some ad-hoc heuristics (Monti *et al.*, 2011; Jirattigalachote *et al.*, 2011) to choose the more energy-efficient lightpaths from a precomputed set.

Shared protection is considered in (Cavdar *et al.*, 2010; Bao *et al.*, 2012; He et Lin, 2013), which propose (i) an on-line procedure for dynamic energy-aware admission control where incoming connections are accepted only if spare resources can support the required additional traffic (He et Lin, 2013), and (ii) different heuristics to select demand lightpaths and to put to sleep redundant network devices (Cavdar *et al.*, 2010; Bao *et al.*, 2012).

A different perspective is adopted in (Wu et Mohan, 2012), where network survivability is not provided by backup paths, but guaranteed by considering he minimum level of reliability required by each incoming demand. Since each link is characterized by a certain probability of failure, demand reliability is computed according to the overall failure probability of the corresponding lightpath. The authors propose a heuristic algorithm to minimize network consumption by adjusting network routing and blocking incoming connections which cannot be served with the requested reliability level. A similar concept of reliability known as *Dif-*

ferentiated Reliability is exploited in (Muhammad *et al.*, 2013) along with shared protection to develop a heuristic on-line algorithm for dynamic routing and admission control over optical networks. Differentiated reliability means that each connection $d \in D$ has a reliability requirement Υ^d which can be satisfied by defining a proper backup path to protect the corresponding primary path against a subset of possible failures whose overall probability is larger than Υ^d.

Some recent work has focused on the special case of double-link failures (Zhang *et al.*, 2014; Xu *et al.*, 2013; Soh et Lazarescu, 2013). The work presented in (Zhang *et al.*, 2014) proposes a two-stage heuristic greedy algorithm for a shared protection scenario. The two stages, which are repeated until convergence is reached, are organized as follows. The first one sequentially allocates the given set of connections by establishing the necessary lightpaths, while the second one aims at rerouting some lightpaths so as to put to sleep the underutilized devices. (Xu *et al.*, 2013) considers only the IP level and presents some heuristics to minimize the cost of protecting the incoming demands, computed as sum of memory, computational and control overhead costs. Some constraints on the minimum network availability for both single and double link failures are considered. A shortest-path based algorithms to find, at the IP level, the disjoint paths which protect each traffic demand from double link failures is proposed in (Soh et Lazarescu, 2013).

To conclude, pure IP optimization is addressed in (Luo *et al.*, 2013; Lee *et al.*, 2013; Aldraho et Kist, 2012; Francois *et al.*, 2013a). In (Luo *et al.*, 2013) the authors address an ALR scenario with six different rate/energy states for each device, and propose three MILP formulations for dedicated (two formulations, the second one with the possibility to put to sleep network elements carrying only backup paths) and shared (one formulation) protection. The LR algorithm proposed in (Lee *et al.*, 2013), which is an extension of that presented in (Lee *et al.*, 2012), considers protection implemented through NotVia IP fast re-route to find the optimal routing configuration (in both normal and failure scenarios) which maximizes energy saving and offer single-node failure resilience. Note that, according to NotVia IP fast reroute, once a failure has been detected, each node evaluates an alternative failure-detected IP address to forward packets. Finally, in (Aldraho et Kist, 2012) the authors present some MILP formulation to save energy while providing protection to each single network link or demand, while in (Francois *et al.*, 2013a), a heuristic framework to compute different topology configurations by considering a single link failure protection is proposed.

Robustness to traffic variations

Robustness to traffic variations is obviously crucial for any type of TE approaches (not only for those which are energy-aware). However, to the best of our knowledge, traffic uncer-

tainty has been never explicitly integrated into EANM procedures. The only work to consider a certain form of traffic uncertainty is that presented in (Coudert *et al.*, 2013). The authors propose a novel approach to optimize network consumption by exploiting the so-called *redundancy elimination* mechanism to prevent the network from transmitting the same contents multiple times. In this particular problem, the uncertainty involves the redundancy degree of each demand, i.e., the amount of redundant data which may not be transmitted in presence of routers which are capable of storing the contents of the packets already transmitted.

It is worth pointing out that in the literature, the uncertainty of traffic variations is typically handled in an indirect way by applying on-line strategies that adjust the network configuration according to the observed traffic changes. We refer the reader to (Bertsimas *et al.*, 2011) and (Ben-Tal *et al.*, 2009) for general surveys on robust optimization applied in both general and network contexts.

2.6.2 Our Contribution

Our work on EANM with survivability and robustness requirements is an extension of the multi-period problem for IP/MPLS networks addressed by us in (Addis *et al.*, 2012b, 2014a). We have integrated both the exact MILP formulations and the heuristic algorithms with the proper variations required to consider (i) dedicated protection (Addis *et al.*, 2012c), (ii) shared protection (Addis *et al.*, 2012a), and (iii) robustness to uncertain traffic demands varying in a close symmetric interval (Addis *et al.*, 2013). Finally, we have developed an integrated solution to jointly include both resilience and robustness constraints, and conducted an overall trade-off evaluation between energy-efficiency and network survivability (Addis *et al.*, 2014b).

As for the implementation of the protection schemes, we have compared a *classic* approach where all network elements which carry at least a primary or a backup path must be completely powered-on, and a *smart* version in which line cards used only by backup paths can be put to sleep. In this case, we assume that a line card carrying only backup routes, due to the negligible amount of time required to wake up a sleeping card (in the order of milliseconds) from the low-power state (Hays, 2007), can be promptly powered on only when required by a failure. Note that the same assumptions are not valid for router chassis, which require considerably more time to change state.

Differently from previous work, we introduce a novel element to further reduce the energy consumption. Since link failure is typically very unlikely, we distinguished between the maximum utilization threshold that must be respected in normal conditions, and the so-called failure maximum utilization threshold to be respected when backup resources are used. Practically speaking, we believe that it is reasonable to allow the network to operate under higher

congestion conditions during the very limited failure intervals, in exchange of substantially higher energy savings during normal periods.

In our modeling framework, we have integrated the approach proposed in (Bertsimas *et al.*, 2011) for the management of uncertain traffic demands. We assume uncertain traffic demands to vary inside a close symmetric interval (without having any knowledge of the underlying distribution) and use a group of input parameters, one for each maximum utilization constraint, to easily tune the degree of conservatism of the solution (Addis *et al.*, 2013): a highly conservative solution tends to reserve more spare capacity to cope with the most unlikely traffic variations, i.e., the largest ones.

2.7 Energy-Aware Network Management with Elastic Traffic

A traffic demand is considered elastic if its rate is not known a priori, but it is determined by a certain congestion control mechanism e.g., the *additive increase/multiplicative decrease* of transport control protocol (TCP) or different queue management policies, which tries to maximize the network resources available on the selected routing path. In a complex network context, concurrent elastic demands naturally compete with each other to get as much bandwidth as possible. In this regard, the notion of fairness between traffic flows has been largely investigated to guarantee equality of treatment to all the incoming elastic demands.

The peculiarity of TE problems with elastic traffic is that resource allocation has to be optimized along with network routing. Practically speaking, it means that an elastic traffic demand $d \in D^e$ (D^e is the set of elastic demands) is characterized by source node o^d and destination node t^d, while its transmission rate is represented by non-negative real variables ϕ^d (instead of the fixed parameter r^d).

A typical joint TE-allocation problem with no energy management can be formulated as follows (general version of that proposed in Amaldi *et al.*, 2013b):

$$\max U(\underline{\phi}) \tag{2.27}$$

s.t.

$$\sum_{(i,j)\in A} f_{ij}^d - \sum_{(j,i)\in A} f_{ji}^d = \begin{cases} \phi^d & \text{if } i = o^d \\ -\phi_d & \text{if } i = t^d \\ 0 & \text{else} \end{cases}, \quad \forall i \in V,\, d \in D^e \tag{2.28}$$

$$\sum_{(i,j)\in A} x_{ij}^d - \sum_{(j,i)\in A} x_{ji}^d = \begin{cases} 1 & \text{if } i = o^d \\ -1 & \text{if } i = t^d \\ 0 & \text{else} \end{cases}, \quad \forall i \in V,\, d \in D^e \tag{2.29}$$

$$\sum_{(i,j)\in A} x_{ij}^d \leq 1, \quad \forall i \in V,\, d \in D^e \tag{2.30}$$

$$f_{ij} \leq \mu_{ij} c_{ij} n_{ij}, \qquad \forall (i,j) \in A \tag{2.31}$$

$$\phi^d \geq 0, \qquad \forall d \in D^e \tag{2.32}$$

$$(2.5\text{–}2.6),\ (2.8). \tag{2.33}$$

Objective function (2.27) maximizes a general utility function U related to the bandwidth allocated to each elastic demand $d \in D^e$. Constraints (2.28–2.30) are required to define a single unsplittable path for each elastic demand and compute the bandwidth allocated to each demand. Finally there are capacity Constraints (2.31) (note that in this case all the line cards are considered activated), link and node total flow Constraints (2.5–2.6), and variable domain Constraints (2.32), (2.8).

2.7.1 State of the Art on the Management of Elastic Traffic

In the literature the utility function $U(\underline{\phi})$ has been typically interpreted as fairness measure related to the amount of bandwidth assigned to each incoming elastic connection. The two most popular definitions of fairness are max-min-fairness (MMF) and proportional fairness (PF). In this context, MMF is strictly related to the lexicographic maximization of the flow vector $\underline{\phi}$. If $\sigma(\underline{\phi})$ is the allocation vector sorted in nondecreasing order and $\sigma(\underline{\phi})_i$ is the i-th element of $\sigma(\underline{\phi})$, $\underline{\phi}$ is lexicographically maximized (max min lex $\{\ldots\}$) if and only if the value of $\sigma(\underline{\phi})_i$ for any $i \in \{1, \ldots, |\underline{\phi}|\}$ cannot be increased without decreasing that bandwidth assigned to any other $\sigma(\underline{\phi})_j$ with $j < i$. The utility functions $U(\underline{\phi})$ achieving MMF and PF can be expressed, respectively, as (Pióro et Medhi, 2004):

$$\textbf{MMF} \qquad U\left(\underline{\phi}\right) = \max \min \text{lex}\left\{\underline{\phi}\right\} \tag{2.34}$$

$$\textbf{PF} \qquad U\left(\underline{\phi}\right) = \max \left\{ \sum_{d\in D^e} \log\left(\phi^d\right) \right\}. \tag{2.35}$$

The MMF utility function (2.34) states that an MMF allocation vector $\underline{\phi}$ has to be

lexicographically maximized (see, e.g., Nace et Pioro, 2008). As for PF, according to (2.35) an allocation vector is PF only if it maximizes the summation over the logarithms of all the rate values (Pióro et Medhi, 2004).

Some authors have suggested that the notion of fairness should be applied w.r.t to the corresponding utility values. This is based on the observation that flows belonging to different applications, e.g., IP-TV, HTTP, FTP, are characterized by specific rate utility functions U^d which could be related to bandwidth values, economical aspects, e.g., the price paid by the corresponding user, and application service requirements (Shenker, 1995).

Due to the extreme complexity of joint TE-allocation problems, a large body of work has addressed simple allocation problems where routing is given (both single and multi-path). The MMF allocation vector for the single-fixed-path routing is computed by the polynomial-time *Water filling* algorithm (Bertsekas *et al.*, 1987). The same problem can be solved by an NP-Hard MILP formulation (Pióro et Medhi, 2004; Tomaszewski, 2005) based on the notion of *bottleneck* link (Massoulie et Roberts, 2002). Other general utility optimization problems with given single path routing include non-linear programming to maximize the utility of triple play services (Shi *et al.*, 2008), centralized (Cao et Zegura, 1999) and distributed (Cho et Chong, 2007) algorithms to compute an allocation vector which is utility MMF, methods to achieve utility PF in a scenario with layered multimedia applications (Harks et Poschwatta, 2005), and an algorithm to guarantee utility maximal fairness in multicast networks (Sarkar et Tassiulas, 2002).

When routing is not given as input, a well studied problem is joint routing-allocation to compute an allocation vector which is overall MMF. Both algorithms to compute MMF allocation vectors with splittable (Pióro et Medhi, 2004; Nace et Pioro, 2008) and unsplittable (Nilsson, 2006; Ogryczak *et al.*, 2005) routing have been proposed. A general method to compute the MMF solution for convex problems not tractable with the *Water filling* algorithm has been proposed in (Radunovic et Le Boudec, 2007), while the non-convex cases have been addressed in (Pioro, 2007). From a practical point view, the joint problem of maximizing a general utility function by means of traffic engineering and resource allocation has been addressed in (J. Wang et Doyle, 2005; He *et al.*, 2006; Lin et Shroff, 2006; Han *et al.*, 2006; He *et al.*, 2007), where different flow control algorithms to maximize network utility functions related to congestion and throughput are presented. These approaches exploit multipath splittable routing based on pre-computed paths.

Other relevant work on fairness and resource allocation includes a different definition of fairness which allows for simpler algorithms (Danna *et al.*, 2012a), a linear programming formulation to jointly set rates and routing paths so as to balance throughput and fairness in case of splittable flows (Danna *et al.*, 2012b), a study which shows how to obtain MMF

solutions by optimizing suitably chosen (nonlinear) objective functions (Coluccia *et al.*, 2011).

To conclude this part, we mention the work of Gan et al. (Gan *et al.*, 2012), which proposes a novel distributed algorithm called *Dynamic Bandwidth Adjustment* (DBA), whose aim is to minimize the network energy consumption by dynamically adapting the active capacity of the links. The basic idea is to exploit instantaneous measurements on link buffer size to determine whether the amount of active bandwidth of a link can be reduced: in this way those links which are not the bottleneck of any traffic flow are artificially transformed into bottlenecks by reducing their available bandwidth.

2.7.2 Our Contribution

Some important studies have recently shown that each congestion control mechanism employed in IP networks tends to indirectly achieve a specific form of fairness. Thanks to a dual modeling approach, Kelly et al. (Kelly *et al.*, 1998) showed that each congestion control protocol implicitly solves a specific utility maximization problem: in particular, the allocation vector determined by TCP is very close to realize PF. The same model has been later exploited to derive the equivalent allocation utility function corresponding to different versions of TCP (Low, 2003), or to HTTP traffic over TCP (Chang *et al.*, 2004). Similarly, another key work of Massoulié and Roberts (Massoulie et Roberts, 2002) has pointed out the fairness realized by different queuing policies: in particular, first-in-first-out (FIFO), longest-queue-first (LQF) and per-flow fair queuing realize PF, throughput maximization and MMF, respectively.

Motivated by these observations, we have proposed a completely novel approach by formulating a new bi-level optimization problem for joint routing-allocation with elastic traffic. Our work is based on the fact that, once routing paths have been chosen, the rate allocation is uniquely defined by the multiple congestion control mechanisms implemented in the networks. The network operator has thus no direct control on the rate values, which, as shown in (Kelly *et al.*, 1998; Low, 2003; Massoulie et Roberts, 2002), are determined by distributed mechanism which implicitly solves the corresponding utility maximization problem on the given paths, e.g., Max-Min-Fairness, Proportional Fairness, etc.

To the best of our knowledge, we are the first to consider the rate allocation scheme as a hard network constraint determined by the network technology. We present a bi-level network utility maximization problem for networks with elastic traffic where, at the upper level, the network operator (leader) selects a single routing path for each elastic demand and, at the lower one, the congestion control mechanism (follower) determines the transmission rate of each elastic flow by maximizing the corresponding utility function.

Due to the novelty of our approach, we first developed an exact MILP formulation, a

restricted-path MILP and a column generation method for a utility maximization TE problem (Amaldi *et al.*, 2013b,d), which is energy-unaware, and thus more general. Successively, in (Amaldi *et al.*, 2013a) we have addressed a SEANM with elastic traffic (SEANM-ET) problem to find the trade-off between energy savings and network utility. We have adapted the original MILP of (Amaldi *et al.*, 2013b) to an energy-aware scenario and showed how discriminating between elastic and inelastic traffic demand allows us to detect when high levels of link utilization really correspond to network saturation. In the negative case, we show that some network elements can be safely put to sleep also when 100% utilization is observed on the links.

To the best of our knowledge, the explicit management of elastic traffic demands in EANM has been never addressed before our work presented in (Amaldi *et al.*, 2013a).

2.8 Other Approaches

In the next subsections we summarize some worth mentioning work which cannot be classified according to the proposed categorization and which is hardly comparable with our contributions.

2.8.1 Heuristic Algorithms

A specific scenario considering a multi-stage software router is handled in (Bianco *et al.*, 2012), where the authors define a MILP formulation for the energy minimization problem involving the efficient management of the distributed router components. They also propose a two-stages heuristic algorithm to rapidly solve the problem and overcome the MILP complexity. A multi-layer problem concerning the virtualization of logical routers over multiple physical locations (virtual network embedding) is addressed in (Botero et Hesselbach, 2013) by presenting a MILP formulation and a greedy heuristic.

Several traffic-unaware heuristic algorithms to put to sleep network elements are presented in (Cuomo *et al.*, 2011a,b, 2012). Surprisingly, these approaches do not consider any traffic data (neither traffic matrices nor link loads) and exploit topological features, like the number of shortest paths that use a certain link, to decide whether or not put to sleep a certain network element. The authors state that these procedures should be exploited only in case of low traffic conditions.

(Giroire *et al.*, 2012; Koster *et al.*, 2013) consider the problem related to green redundancy elimination without, in this case, any data uncertainty (see Section 2.6.1). A novel MILP formulation and a dedicated heuristic algorithm are presented in (Giroire *et al.*, 2012), while in (Koster *et al.*, 2013) some cutting planes for the same problem are derived.

Telecommunication networks operated with Carrier Grade Ethernet are considered in (Capone *et al.*, 2013): two different optimization frameworks to jointly minimize consumption and congestion by efficiently adjusting layer-2 Vlan configuration are proposed.

Finally, an hybrid MPLS/OSPF routing protocol is considered in (Zhang *et al.*, 2010), where a MILP formulation to switch off the network links while respecting maximum utilization and maximum path length constraints are proposed. The formulation takes in input a set of previously computed k-shortest paths, and the final solution is implemented in the real network by using the shortest paths of OSPF for the traffic demands whose computed routing matches with the first shortest path, and by exploiting properly configured LSPs to route the remaining demands.

2.8.2 Green Protocols

In (Gelenbe et Mahmoodi, 2011) the authors propose to use the Cognitive Packet Network (CPN) (Gelenbe *et al.*, 2004) so as to implement energy-aware routing. In this approach, the optimization is fully distributed and each node takes routing decisions determined by a random neural network (Gelenbe, 1993).

(Ho et Cheung, 2010) proposes a new overlay protocol called *General Routing Protocol for Power Saving*, where each capable router periodically asks to a designated router (a sort of controller) the permission to go into a sleep mode. The designated router is then responsible for monitoring the link load values through the information disseminated by the underlying protocols, e.g., it uses the link state packet of OSPF, and sends wake-up command to the routers in case of congestion detection.

Finally, in (Raman *et al.*, 2012) the authors present a modified version of BGP which, by introducing a new energy-related attribute and modifying the path selection algorithm, aims at adjusting the routing paths according to low-power path criteria.

2.8.3 Optical Networks

Part of the literature on EANM, rather than optimizing network configuration at the IP level, focuses on multilayer IP over WDM or even pure optical green optimization problems. This choice leads to the introduction of new physical devices such as fibers, optical amplifiers, optical cross-connects, optical transponders, and new elements like wavelengths, lightpaths, and virtual links. Although the optical layer contributes for less than 10% of total network consumption, the efficient management of the optical infrastructure as well as the implementation of optical bypass strategies may reduce the number of required IP ports (Mukherjee, 2000, 2002), and consequently further decrease the IP power requirement. However, it is

worth pointing out that the modeling of the optical layer further increase the complexity of pure IP EANM problems.

We mention below some of the most relevant work. One of the first article on multi-layer EANM is that of Shen et al. (Shen et Tucker, 2009), where both a MILP formulation and two greedy heuristic methods for the energy-aware optimization of both optical infrastructure and demand lightpaths are presented. In this case, the optical layer modeling is very rich, since it considers both optical amplifiers, optical transponders, optical fibers and wavelengths. A simpler multi-layer problem is addressed in (Idzikowski *et al.*, 2010, 2011), which consider an optical layer composed of optical fibers and lightpaths. Three different optimization approaches called *Fixed Upper Fixed Lower* (FUFL), *Dynamic Upper Fixed Lower* (DUFL) and *Dynamic Upper Dynamic Lower* (DUDL) are presented and evaluated by solving the related MILP formulation.

The problem of efficiently configuring the optical bypass so as to minimize the number of activated IP line cards and chassis is addressed in (Zhang *et al.*, 2011). The authors consider three different bypass approaches, i.e., *non-bypass*, *direct bypass* and *multi-hop bypass*, and develop some mathematical programming algorithms based on a novel MILP formulation. They aim at efficiently computing a set of configurations which can be optimally implemented along the different parts of a single day. In (Rizzelli *et al.*, 2012), a very detailed MILP formulation to reduce energy consumption and evaluate the energy efficiency of different topology architectures is proposed.

The *Energy Watermark Algorithm* (EWA) to minimize the consumption across two consecutive time periods while minimizing the required re-configurations and respecting capacity constraints is presented in (Idzikowski *et al.*, 2012; Bonetto *et al.*, 2013), which consider an optical layer composed of lightptahs, fibers and fabric card shelves.

To conclude, we mention an on-line local search procedure to maximize energy savings while implementing only a limited number of new lightpaths across two consecutive time periods (Yayimli et Cavdar, 2012), a MILP formulation and a heuristic for energy-aware multi-layer network design (Ahmad *et al.*, 2013) and a greedy algorithm for power-aware lightpaths provisioning (Xia *et al.*, 2011).

CHAPTER 3

SEANM with Flow-Based Routing

The majority of backbone IP networks is typically operated with multi protocol Label Switching (MPLS). MPLS offers network operators the possibility to explicitly define a dedicated path for each traffic demand (flow-based or per-flow routing). This greatly differs from other routing schemes like shortest path, where routing decisions are taken according to additional factors, e.g., the link weights. Path setup needs a lot of additional overhead and signaling, especially when the number of traffic demands substantially grows. MPLS, which is often used with other more scalable routing protocols, e.g., open shortest path first (OSPF), is typically destined to route only premium/high priority traffic.

Note that, although recent versions of MPLS offer support for multipath (splittable) routing, this practice requires additional management steps, i.e., configuration of the splitting ratios, introduces additional complexity in terms of low level operations to forward each packet and brings potential issues in terms of packet reordering and transmission delay jitter. For this reason unsplittable routing is typically preferred.

In MPLS/IP networks, optimization strategies to perform energy-aware network management (EANM), or its sleeping-based variant known as sleep-based energy-aware network management (SEANM), are substantially simplified by the possibility to explicitly define the single routing path of each demand by directly indicating each single hop. Furthermore, besides being conceptually quite elementary, flow-based routing can be modeled by means of quite standard and relatively efficient modeling techniques. We refer to SEANM problems involving MPLS or other generic flow-based (per-flow) routing protocols as SEANM with flow-based routing (SEANM-FB) problems.

These very important features make routing optimization (traffic engineering) very flexible and allow to introduce additional elements into the considered SEANM-FB problem, including for instance, multi-period modeling, inter-periods constraints, additional routing limitations, path or link protection and so on.

The reminder of the chapter is organized as follows. We formally present our general approach in Section 3.1. Then, we first discuss two different SEANM-FB problems by presenting the corresponding exact mixed integer linear programming (MILP) formulations in Section 3.2, and propose some heuristic methods in Section 3.3. Finally we present and comment the most relevant computational results in Section 3.4.

3.1 Our Approach for Energy-Aware Multi-Period Network Management

We consider IP networks operated with MPLS or any other flow-based routing protocols. Our aim is to minimize the daily network power consumption by jointly optimizing both network routing (traffic engineering (TE)) and network device power states.

We propose a centralized off-line approach for efficient operational planning of network configuration (for the following days/weeks). We assume that our management strategy can be naturally integrated in a network management platform, a tool typically used by network operators to monitor and adjust each aspect of the network configuration, e.g., route setting, device power state, IP configuration (Subramanian, 2000).

Applicability and performance of our management approach strictly depends on two crucial elements, i.e., the overall length of the time horizon to be optimized (typically a single day), and how traffic varies (traffic profile features) along the time horizon itself. These two elements are jointly evaluated to identify a subset of different time periods where we assume that traffic levels remain constant. Thus, the number of periods among which to split the time-horizon is strictly related to the traffic variability.

Due to the peculiar characteristics of Internet traffic, whose dynamic is relatively slow and strongly periodic (see e.g Mackarel *et al.*, 2011), we propose to consider a daily time-horizon to be split among a limited number of time periods (six in our case). In this regard we mention an analytical study on IP networks which confirms that optimizing over a limited set of time periods does not significantly affect solution optimality (Chiaraviglio *et al.*, 2013a).

For each single time interval, we assume that a predicted traffic matrix representing the corresponding traffic pattern is available. Thanks to Internet traffic periodicity, these matrices can be reasonably estimated by means of both direct (Cisco Systems, 2012) and indirect measurements (Casas *et al.*, 2009) collected during the previous days.

Our approach explicitly advocates a novel multi-period SEANM-FB problem where the overall daily network consumption is minimized by jointly optimizing the network configuration within each different time period. The basic idea is, as in classic SEANM-FB, to make network power consumption proportional to the predicted levels of traffic by putting to sleep the redundant network devices. The multi-period approach brings two substantial benefits: (i) it allows to directly optimize over the entire traffic variation range and (ii) makes straightforward to control network stability by means of inter-periods constraints. In our case, these are exploited to limit both signaling overhead and performance degradation produced by routing reconfigurations, minimize the consumption required to reactivate sleeping devices, and preserve device lifetime by forbidding too frequent state-switching.

According to the basic configuration of MPLS, routing is considered per-flow and unsplit-

table, i.e., a single dedicated path is assigned to each traffic demand. To evaluate a trade-off between routing stability and power consumption, we address two variants of the problem where, respectively, routing is maintained fixed along the entire day, i.e., power-aware fixed routing problem (PAFRP), or can be freely adjusted within each different time period, i.e., power-aware variable routing problem (PAVRP).

To accurately evaluate the impact of our approach on daily network consumption, we consider network routers to be composed of a chassis to which a set of line cards is connected. The first offers computation and switching capabilities, while the second provide communication interfaces and may be responsible for additional network processing operations. Both chassis and line cards can be put to sleep directly by the management platform or through some distributed mechanisms capable to detect the absence of traffic on the monitored interfaces.

For what concerns the consumption figures, we consider on/off consumption profiles characterized by negligible consumption levels during sleeping states. Furthermore, we keep track of the power spikes typically produced when a device is reactivated. Note that this may be seen also as a penalty factor to prevent the network from changing state too frequently (that would produce additional signaling overhead and may negatively affect the quality of service (QoS) of incoming flows). For the numerical analysis we consider public information on Juniper routers made available by the manufacturer.

3.2 Two Exact MILP Formulations

Let us consider an IP network represented by the directed graph $G(V, A)$, where V is the set of nodes and A the set of full-duplex links. Nodes are further split between *edge* nodes V^e which generate and receive traffic, and *core* nodes V^c which simply carry the network traffic. Each node $i \in V$ corresponds to a network router equipped with a chassis of capacity $C_i \; \forall i \in V$ and a set of line cards. The connectivity of each link $(i, j) \in A$ is provided by means of n_{ij} line cards installed on each side of the link itself (we use $n_{ij} = 2$ in most of our tests). While c_{ij} is the nominal transmission rate of a single line card installed on link $(i, j) \in A$, the total capacity available on the same link is $c_{ij}n_{ij}$. Note that all the line cards used on a specific link must be of the same type (same capacity). Since spare capacity is typically left on each link to absorb unexpected traffic variations, let us denote with μ_{ij} the maximum link utilization allowed by the network operator. Moreover, note that, to preserve the device lifetime, each single line card cannot be reactivated from its sleeping state more than η_{on} times along the entire set of periods S.

Let a single day be split among the circular set S of multiple time periods. Note that *circular* means that in our multi-period problem we consider also the transition from the last

to the first period contained in S. Each period has a duration h^σ $\forall \sigma \in S$ and corresponds to some specific hours of the day. Let us denote with D the set of traffic demands (traffic matrix). Each traffic demand $d \in D$ is characterized by a source node o^d, a destination node t^d and a traffic request of $r^{d\sigma}$ between o^d and t^d during time period $\sigma \in S$.

As in Chapter 2, let π_i be the power consumption of an active chassis installed at node i, and let π_{ij} be the energy consumed by a single line card of link $(i,j) \in A$. In addition we also denote with $\delta \in [0,1]$ the power required to switch on a chassis from a sleeping state, as a fraction of the hourly chassis consumption π_i.

The single unsplittable path used by each demand $d \in D$ is defined by means of the binary variables x_{ij}^d, which are equal to 1 if link $(i,j) \in A$ is used by demand $d \in D$. The chassis power state is represented by the binary variables y_i, which are equal to 1 if the chassis of node i is active. The number of active line cards on link $(i,j) \in A$ during time period $\sigma \in S$ is instead denoted by the integer non-negative variables $w_{ij}^\sigma \in [0, \ldots, n_{ij}]$. Finally, let z_i^σ be the non-negative real variables representing the energy consumed to activate the chassis of node i passing from period $\sigma - 1 \in S$ to period $\sigma \in S$, and let u_{ijk}^σ be the binary variables equal to 1 if the k-th line card of link $(i,j) \in A$ is activated from the sleeping state at the beginning of period $\sigma \in S$.

Let us exploit the above definitions to define two MILP formulations for both PAFRP and PAVRP problems.

3.2.1 Power Aware Fixed Routing Problem (PAFRP)

The first problem we address is the PAFRP, according to which each traffic demand $d \in D$ must use the same path over all time periods $\sigma \in S$. The goal is to minimize the the daily energy consumption (in Watt-hour) subject to single path routing and other operational constraints. Let us define the following MILP formulation for PAFRP:

$$\min \sum_{\sigma \in S} h_\sigma \sum_{j \in V} \pi_j y_j^\sigma + \sum_{\sigma \in S} h_\sigma \sum_{(i,j) \in A} \pi_{ij} w_{ij}^\sigma + \sum_{\sigma \in S} \sum_{j \in N} z_j^\sigma \tag{3.1}$$

s.t.

$$\sum_{\substack{j\in V:\\(i,j)\in A}} x_{ij}^d - \sum_{\substack{j\in V:\\(j,i)\in A}} x_{ji}^d = \begin{cases} 1 & \text{if } i = o_d, \\ -1 & \text{if } i = t_d, \\ 0 & \text{otherwise} \end{cases}, \qquad \forall i \in V,\, d \in D \tag{3.2}$$

$$\sum_{\substack{i\in V:\\(i,j)\in A}} \sum_{d\in D} r^{d\sigma} x_{ij}^d + \sum_{\substack{i\in V:\\(j,i)\in A}} \sum_{d\in D} r^{d\sigma} x_{ji}^d \le C_j y_j^\sigma, \qquad \forall j \in V,\, \sigma \in S \tag{3.3}$$

$$\sum_{d\in D} r^{d\sigma} x_{ij}^d \le \mu_{ij} c_{ij} w_{ij}^\sigma, \qquad \forall (i,j) \in A,\, \sigma \in S \tag{3.4}$$

$$z_j^\sigma \ge \delta \pi_j \left(y_j^\sigma - y_j^{\sigma-1} \right), \qquad \forall j \in V,\, \sigma \in S \tag{3.5}$$

$$w_{ij}^\sigma = w_{ji}^\sigma, \qquad \forall (i,j) \in A,\, \sigma \in S \tag{3.6}$$

$$\sum_{k=1}^{n_{ij}} u_{ijk}^\sigma \ge w_{ij}^\sigma - w_{ij}^{\sigma-1}, \qquad \forall (i,j) \in A,\, \sigma \in S \tag{3.7}$$

$$\sum_{\sigma\in S} u_{ijk}^\sigma \le \eta_{on}, \qquad \forall (i,j) \in A,\, k \in \{1,\dots,n_{ij}\} \tag{3.8}$$

$$y_h^\sigma,\, x_{ij}^d \in \{0,1\}, \qquad \forall h \in V,\, (i,j) \in A,\, d \in D\,, \sigma \in S \tag{3.9}$$

$$u_{ijk}^\sigma \in \{0,1\}, \qquad \forall (i,j) \in A,\, \sigma \in S,\, k \in \{1,\dots,n_{ij}\} \tag{3.10}$$

$$w_{ij}^\sigma \in \{0,\dots,n_{ij}\}, \qquad \forall (i,j) \in A,\, \sigma \in S \tag{3.11}$$

$$z_j^\sigma \ge 0, \qquad \forall j \in V,\, \sigma \in S. \tag{3.12}$$

The Objective function (3.1) minimizes the overall daily network consumption computed as the summation of three different power consumption components: in the following order we find the power consumed by (i) active chassis, (ii) active line cards and (iii) chassis reactivation, which is computed by means of Constraints (3.5), which force variables z_i^σ to be equal to $\delta\pi_i$ when chassis i is activated at the beginning of period $\sigma \in S$.

The flow conservation Constraints (3.2) define a single routing path for each demand, while Equations (3.3) and (3.4) represent, respectively, node maximum capacity constraints and link maximum utilization constraints. The link maximum utilization parameters μ_{ij} can be adjusted by the operator so as to reserve enough spare capacity to absorb unexpected traffic variations or possible device failures. Note that the available capacity on each link $(i,j) \in A$ during period σ depends on the number of activated line cards w_{ij}^σ.

The card state of each link $(i,j) \in A$ is regulated by Constraints (3.6–3.8). Equation (3.6) states that, due to hardware constraints, the same number of line cards must be activated on both direction of each link. Constraints (3.7) determine how many sleeping line cards must be awakened at the beginning of each time period, and Constraints (3.8) impose the limit η_{on} on the maximum number of card switching-on allowed over the entire day. Note that, according to Constraints (3.7), if m line cards are switched on on link $(i,j) \in A$ at

the beginning of time period $\sigma \in S$, then the value $w_{ij}^{\sigma} - w_{ij}^{\sigma-1}$ is equal to m, and m of the u auxiliary variables associated to link (i,j) must be equal to 1. Constraints (3.7–3.8), to which we commonly refer as card reliability constraints, preserve line card lifetime and correct functioning by avoiding too frequent state switching. Where not explicitly specified, we experiment with η_{on} equal to 1.

3.2.2 Power Aware Variable Routing Problem (PAVRP)

The variable routing variant PAVRP allows to adjust the routing of each demand within each single time period. PAFRP routing variables x_{ij}^{d} are replaced in PAVRP by the new $x_{ij}^{d\sigma}$ variables, which state if link $(i,j) \in A$ is used or not by demand $d \in D$ during period $\sigma \in S$. The MILP formulation for PAVRP can be expressed as follows:

$$(3.1) \tag{3.13}$$

s.t.

$$\sum_{\substack{j \in V: \\ (i,j) \in A}} x_{ij}^{d\sigma} - \sum_{\substack{j \in V: \\ (j,i) \in A}} x_{ji}^{d\sigma} = \begin{cases} 1 & \text{if } i = o_d, \\ -1 & \text{if } i = t_d, \\ 0 & \text{otherwise} \end{cases}, \qquad \forall i \in V,\, d \in D, \sigma \in S \tag{3.14}$$

$$\sum_{\substack{i \in V: \\ (i,j) \in A}} \sum_{d \in D} r^{d\sigma} x_{ij}^{d\sigma} + \sum_{\substack{i \in V: \\ (j,i) \in A}} \sum_{d \in D} r^{d\sigma} x_{ji}^{d\sigma} \leq C_j y_j^{\sigma}, \qquad \forall j \in V,\, \sigma \in S \tag{3.15}$$

$$\sum_{d \in D} r^{d\sigma} x_{ij}^{d\sigma} \leq \mu_{ij} c_{ij} w_{ij}^{\sigma}, \qquad \forall (i,j) \in A,\, \sigma \in S \tag{3.16}$$

$$x_{ij}^{d\sigma} \in \{0,1\}, \quad \forall (i,j) \in A,\, d \in D,\, \sigma \in S \tag{3.17}$$

$$(3.5),\ (3.6 - -3.8),\ (3.10 - -3.12).$$

With the PAVRP formulation, flow conservation Constraints (3.14) are defined for each time period $\sigma \in S$, and capacity Constraints (3.15–3.16) are modified to keep track of the routing variations over different periods.

3.3 Heuristic Methods

State-of-the-art commercial solvers, e.g., CPLEX, can solve at optimality the formulations for networks of 20 nodes and 50 bidirectional links. With larger networks, computing time significantly increases and too much memory (more than 8 GB) is required to conclude a

classic branch and bound (or branch and cut) procedure. To handle larger networks, we propose two very different heuristic approaches called, respectively, energy-aware lexicographic GRASP (EA-LG) and energy-aware single time-period heuristic (EA-STH).

3.3.1 A Quick Heuristic Algorithm

EA-LG is a mathematical programming GRASP heuristic to compute near-optimal solutions by sequentially routing all traffic demands into the network, and activating, at each iteration, the necessary network devices. With EA-LG we are able to efficiently solve instances with about 50 nodes and 600 traffic demands. Its pseudo-code is reported in Algorithm 1. The logical structure of EA-LG consists into a greedy procedure integrated into a greedy randomized adaptive search procedure (GRASP) (see, e.g., Resende et Ribeiro, 2003). Let us first analyze the greedy routine alone. We identify three stages which are called, respectively, (i) *Sorting*, (ii) *Min-Energy* and (iii) *Min-Congestion*.

Sorting Stage

The *Sorting* stage is the first to be performed and consists into sorting all the traffic demands $d \in D$ in a decreasing order w.r.t. their peak traffic request $\max_{\sigma \in S} \left(r^{d\sigma}\right)$. Traffic demands are then processed one by one according to the order of the sorted list.

Input: G Topology,
R Routing of demands already considered,
D Traffic Matrix
i_{max} Number of multi-start iterations,
l Percentage of demands belonging to the RCL
g Sorting criterion

for $it = 1 \dots i_{max}$ **do**
  RandomizedSortingOfDemandsDecreasingOrder(D, k);
  foreach *Element* $d \in D$, *according to the sorting* **do**
    SolveModelWithSelectedDemand(d, G, R);
    return *NetworkConsumption* N_c;
    if *infeasible* **then**
      break;
    end
    SolveModelToMinimizeCongestion(d, G, R, N_c) UpdateRouting(R);
  end
end

Algorithm 1: Energy-aware lexicographic GRASP.

Min-Energy Stage

The selected demand $\bar{d}$ is first provided as input to the *Min-Energy* stage, which determines its routing paths (one for each period) by solving a variant of the PAFRP/PAVRP formulation characterized by the following changes: $\bar{d}$ is the only demand contained in D and capacity is reduced by the amount of traffic already carried by each link $(i,j) \in A$, i.e., $\bar{c}_{ij}^{\sigma}$, and by each node $i \in V$, i.e., $\bar{C}_i^{\sigma}$:

$$\sum_{\substack{i \in V: \\ (i,j) \in A}} r^{\bar{d}\sigma} x_{ij}^{\bar{d}\sigma} + \sum_{\substack{i \in V: \\ (j,i) \in A}} r^{\bar{d}\sigma} x_{ji}^{\bar{d}\sigma} \leq \left(C_j - \bar{C}_j^{\sigma}\right) y_j^{\sigma}, \qquad \forall j \in V,\, \sigma \in S \tag{3.18}$$

$$r^{\bar{d}\sigma} x_{ij}^{\bar{d}\sigma} \leq \mu_{ij} \left(c_{ij} - \bar{c}_{ij}^{\sigma}\right) w_{ij}^{\sigma}, \quad \forall (i,j) \in A,\, \sigma \in S. \tag{3.19}$$

The modified formulation returns as output E_c^{MIN-EN}, i.e., the minimized network energy consumption obtained by optimizing the routing of demand $\bar{d}$ while keeping fixed all the routing paths used by traffic demands previously processed.

Min-Congestion Stage

Once E_c^{MIN-EN} is computed, we pass it to the *Min-Congestion* stage to minimize network congestion while keeping the energy consumption smaller or equal than E_c^{MIN-EN}. The basic idea is to prevent the algorithm from saturating active resources, which, due to the sequential nature of the greedy procedure, would cause a premature stop of the procedure (algorithm failure). If load balancing is not performed, no further resource might be available to route a certain demand $\bar{d}$, which would result in a failure of the whole algorithm.

The *Min-Congestion* stage consists in solving a second restated version of the original PAFRP/PAVRP formulation. A new objective function aims at optimizing network congestion by minimizing the summation over all the links $(i,j) \in A$ and over all the time periods $\sigma \in S$ of a convex piece-wise cost function $Cong_{ij}^{\sigma}\left(w_{ij}c_{ij}, r^{\bar{d}\sigma} + \bar{c}_{ij}\right)$ related to link saturation. This function, which was first proposed in (Fortz et Thorup, 2002), can be described by the following expression, which defines both slope and domain of each piece:

$$Cong^{\sigma'}_{(i,j)}\left(w_{ij}c_{ij}, r^{\bar{d}\sigma}+\bar{c}_{ij}\right) = \begin{cases} 1\,for & 0 \le \left(r^{\bar{d}\sigma}+\bar{c}_{ij}\right)/\left(w_{ij}c_{ij}\right) < 1/3 \\ 3\,for & 1/3 \le \left(r^{\bar{d}\sigma}+\bar{c}_{ij}\right)/\left(w_{ij}c_{ij}\right) < 2/3 \\ 10\,for & 2/3 \le \left(r^{\bar{d}\sigma}+\bar{c}_{ij}\right)/\left(w_{ij}c_{ij}\right) < 9/10 \\ 70\,for & 9/10 \le \left(r^{\bar{d}\sigma}+\bar{c}_{ij}\right)/\left(w_{ij}c_{ij}\right) < 1 \\ 500\,for & 1 \le \left(r^{\bar{d}\sigma}+\bar{c}_{ij}\right)/\left(w_{ij}c_{ij}\right) < 11/10 \\ 5000\,for & 11/10 \le \left(r^{\bar{d}\sigma}+\bar{c}_{ij}\right)/\left(w_{ij}c_{ij}\right) < \infty \end{cases}. \quad (3.20)$$

Note that $Cong^{\sigma'}_{(i,j)}\left(w_{ij}c_{ij}, r^{\bar{d}\sigma}+\bar{c}_{ij}\right)$ is the first derivative of the congestion function. According to its definition, $Cong^{\sigma}_{(i,j)}\left(w_{ij}c_{ij}, r^{\bar{d}\sigma}+\bar{c}_{ij}\right)$ is expressed through the following MILP formulation:

$$\min\left\{\sum_{(i,j)\in A,\sigma\in S}\iota^{\sigma}_{ij}\right\} \quad (3.21)$$

s.t.

$$\iota^{\sigma}_{ij} \ge \alpha_h\left(r^{\bar{d}}x^{\bar{d}\sigma}_{ij}+\bar{c}_{ij}\right) - \beta_h c_{ij}, \quad \forall(i,j) \in A,\, h \in H, \quad (3.22)$$

where Equation (3.22) is responsible for correctly modeling the set H of linear pieces used to represent the corresponding cost function. Each piece $h \in H$ is described by a slope α_h and an offset β_h. Both α_h and β_h are set to respect (3.20). The non-negative real variables ι^{σ}_{ij} assume the right congestion cost for each link $(i,j) \in A$ during period σ. A second modification concerns the network energy consumption, which is fixed to E_c^{MIN-EN}:

$$\sum_{\sigma\in S}h_\sigma\sum_{j\in V}\pi_j y^{\sigma}_j + \sum_{\sigma\in S}h_\sigma\sum_{(i,j)\in A}\pi_{ij}w^{\sigma}_{ij} + \sum_{\sigma\in S}\sum_{j\in N}z^{\sigma}_j = E_c^{MIN-EN} \quad (3.23)$$

At the end of the *Min-Congestion* stage, we update the $\bar{C}^{\sigma}_i$ and $\bar{c}^{\sigma}_{ij}$ parameters on both nodes and links used to route demand $\bar{d}$. The sequence composed of *Min-Energy* and *Min-Congestion* is progressively repeated for each demand $d \in D$ (according to the sorted list).

When integrated into the GRASP scheme, the overall greedy procedure is repeated several times (*Second-Level Multi-Start*), and only the best solution is returned at the end of the algorithm. However, if we use the same sorting criterion for each greedy run, each iteration will return the same solution. To prevent this phenomenon and better explore the solution space, the GRASP scheme introduces the concept of restricted candidate list (RCL): demand sorting is perturbed at each iteration by instructing the procedure to randomly choose one demand from the firsts $k\%$ of the sorted list (not always the first one). Sorting perturbation leads, from time to time, to different final solutions.

3.3.2 A Single-Period Heuristic

A natural approach to reduce problem complexity is to compute the overall solution by handling each single time period separately. The critical element is represented by the management of inter-period constraints. Two alternative strategies can be pursued. The first one is to compute the solution of each single time period without considering any form of inter-period constraints. The overall solution is then built by putting together all the single-period configurations. However, this step is not at all straightforward, since it requires that each single solution is somehow adjusted to guarantee compliance with inter period constraints. The second one is to consider all time periods one by one according to their order in the circular set S. Along the whole elaboration, the solution computed for a certain time period $\sigma \in S$ is constrained by inter-period constraints related to the time intervals already processed. If properly designed, these constraints guarantee that the solution progressively built by the procedure remains feasible along the whole elaboration.

We have opted for the second strategy and developed the EA-STH algorithm for the PAVRP problem. Although this scheme is applicable also in PAFRP, we believe that fixed-routing inter-period constraints are too strong and make the approach trivial. Once a solution is computed for the peak traffic time period, configuration in the remaining periods is computed by simply adapting the activated resources according to the pre-determined routing. Note that fixed-routing also adds some issues in terms of solution feasibility, since it is not possible to guarantee a priori that the partial solution will remain feasible along the entire elaboration.

Starting from a random time period, we optimize one by one the energy consumption of each time period $\sigma \in S$ by solving a restated version of the PAVRP model which considers S as being composed of only the considered time period $\bar{\sigma}$. When a new time interval is considered ($\bar{\sigma} = \bar{\sigma} + 1$), the appropriate parameters are defined to correctly compute activation energy cost and keep track of the number of times that each line card has been switched on. Once k line cards of link $(i, j) \in A$ have been already switched on η_{on} times, proper modifications

applied to Constraints (3.7–3.8) force the model to keep $w_{ij}^{\sigma} \geq k$ along all the remaining unprocessed $\sigma \in S$.

As we consider demand patterns to be cyclically repeated, to compare online results with off-line ones we assume that both routing and switching pattern are repeated every 24 hours. Thus, to guarantee that constraints on card reliability are not violated, if a card has been already switched on η_{on} times in the previously optimized time intervals it is forced to keep its current status (powered on) for the next time intervals. The status can be modified at the end of the cycle, e.g., after 24 hours all the u_{ijk}^{σ} variables corresponding to the previous optimized time periods are set to 0.

It is worth pointing out that the scheme adopted by EA-STH can be naturally adapted to on-line optimization approaches based on short term traffic estimations made in real-time by observing incoming levels of traffic. A strategy of this kind would rely on conservative link maximum utilization parameters μ_{ij} to absorb traffic variations, and would produce a network reconfiguration whenever utilization levels are too much below, or too much above the utilization thresholds themselves.

3.4 Computational Results

All tests have been carried out on Intel i7 processors with 4 core and multi-thread 8x, equipped with 8Gb of RAM. Mathematical formulations have been defined with AMPL and solved with CPLEX-12.4. We report in Table 3.1 the most important acronyms and parameters.

Table 3.1 Main parameters and acronyms

Symbol or acronym	Description
PAFRP	Fixed routing problem
PAVRP	Variable routing problem
MILP	Mixed-Integer Linear Programming
EA-LG	Energy-Aware Lexicographic GRASP
EA-STH	Energy-Aware Single Time-period Heuristic
n_{ij}	Number of line cards on each link
μ_{ij}	Maximum link utilization
η_{on}	State switching limit
δ	Reactivation penalty

3.4.1 Test Instances

Network Topologies

We have conducted our experiments on four network topologies, i.e., the 9-node network shown in Figure 3.1, the 25-node `france` , the 28-node `nobel-eu` and the 50-node `germany50` networks reported in Figure 3.2, 3.3 and 3.4. The last three, which are also the largest ones, belong to the widely known SNDLibrary (Orlowski *et al.*, 2010).

We consider the 9-node network to extensively analyze the optimal solutions of both PAFRP and PAVRP formulations, while we experiment with the largest topologies to confirm the findings with more realistic networks. The nodes of each network are randomly split among edge nodes V^e and core nodes V^c.

For each network we consider three different equipment configurations, namely A, B and C. In each scenario, we assume that each node is equipped with the same type of router, composed of a chassis and a specific type of line cards. Capacity and consumption values for each chassis and line card technology are reported in Table 3.2. Note that if the maximum number of ports supported by a single chassis is lower than the number of line cards installed on the incident links, for sake of simplicity we assume that a router is equipped with a sufficient number of interconnected chassis (capacity and consumption are scaled accordingly).

Network demand

To generate the amount of traffic $r^{d\sigma}$ requested by each demand $d \in D$ during period $\sigma \in S$, we first compute a nominal value ρ^d, representing the peak traffic value of each demand $d \in D$. For `france`, `nobel-eu` and `germany50` networks, ρ values are obtained by scaling the traffic matrices provided by the SNDLib with the largest fixed value which allows to route each traffic demand on a single unsplittable path without crossing a 50% threshold

Table 3.2 Router chassis and cards

Chassis features			
case	device	capacity	hourly power cons.
all	Chassis Juniper M10i	16Gbps	86.4 W
Cards features			
case	device	capacity	hourly power cons.
A	FE 4 ports	400 Mbps	6.8 W
B	OC-3c 1 port	155 Mbps	18.6 W
C	GE 1 port	1 Gbps	7.3 W

on maximum link utilization. Note that we set to 0 each entry of the SNDLib traffic matrices which corresponds to a traffic demand generated (destined) from (to) a core node.

The nominal values of the 9-node network are instead generated from scratch. We define a traffic demand between each pair of edge nodes. The set D is then split among two different subsets D^1 and D^2. Within each subset all traffic demands have the same traffic request,

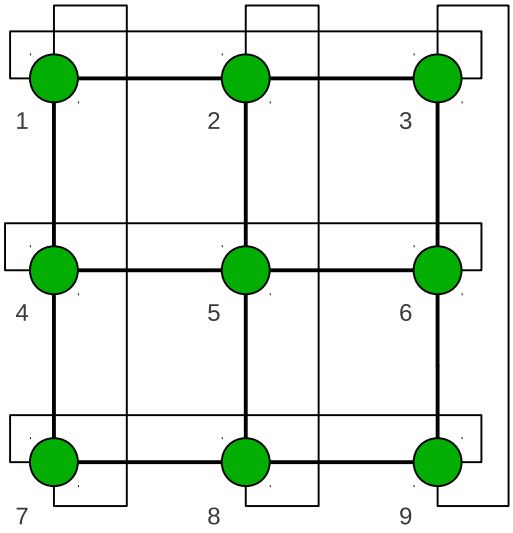


Figure 3.1 Network with 9 nodes

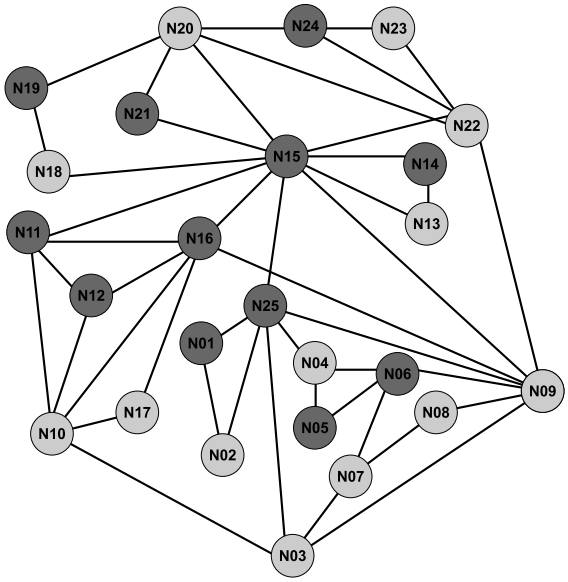


Figure 3.2 `France` network with 25 nodes and 90 links.

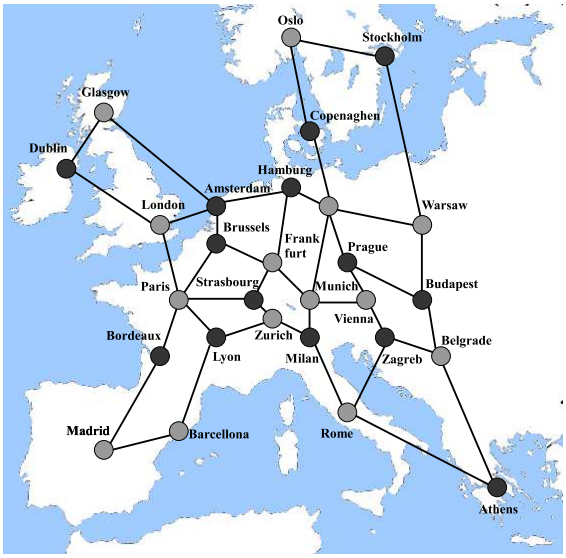


Figure 3.3 `Nobel-eu` network with 28 nodes and 82 links.

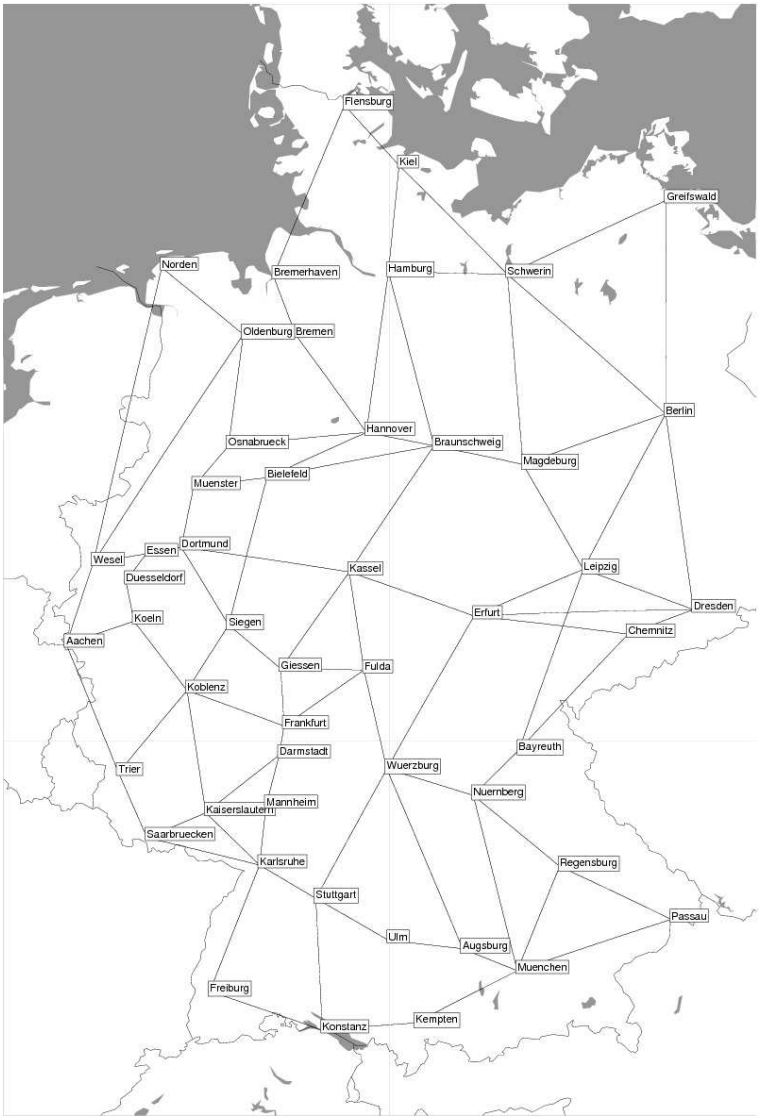


Figure 3.4 `germany50` network with 50 nodes and 176 links.

which is $\frac{1}{4}$ in D^1 and $\frac{1}{2}$ in D^2. All traffic demands are routed by solving a feasibility single scenario model expressed by Constraints (3.2), (3.3), (3.4) and (3.6). Note that we consider $c_{ij} = 1$ and $\mu_{ij} = 0.5$ for all $(i, j) \in A$. If a feasible solution is found, the value of each demand is increased by its starting value and the feasibility test is repeated again. When it fails, we finally multiply the last feasible traffic values by the line card capacity, obtaining in this way the final nominal values $\rho^d \ \forall d \in D$. The last feasible traffic values are reported in the second column of Table 3.4.2.

Nominal values are used as reference to determine the average amount of traffic generated by each traffic demand $d \in D$ in each period $\sigma \in S$. The time horizon consid-

ered in this study is an entire day. We split it among 6 periods corresponding to the following time intervals: 1) 8a.m.-11a.m., 2) 11a.m.-1p.m., 3) 1p.m.-2.30p.m., 4) 2.30p.m.-6.30p.m., 5) 6.30p.m.-10.30p.m., 6) 10.30p.m.-8a.m. For each time interval and topology, traffic demand intensity values $r^{d\sigma}$ are randomly generated by the uniform distribution $U\left(\rho^d(Av-0.2), \rho^d(Av+0.2)\right)$, where the average value Av of each scenario follows the profile shown in Figure 3.5. We experimented with three stochastic traffic realizations, named a, b and c, obtained by considering three different random seeds.

3.4.2 PAFRP and PAVRP Results

Small Networks

To exhaustively investigate the impact of our optimization approach on the underlying network, let us first analyze the optimal solution obtained from CPLEX for the (C,c, 9-nodes) instance (device C, random draw c, 9-node network topology). We consider η_{on} equal to 1 (maximum number of allowed switching on), δ equal to 0.25 (reactivation penalty parameter) and μ_{ij} equal to 0.5 for each $(i,j) \in A$. CPLEX can solve all these instances to optimality within a few seconds.

Switching Patterns

We report in Tables 3.3 and 3.4 the power status of each link and each chassis, respectively,

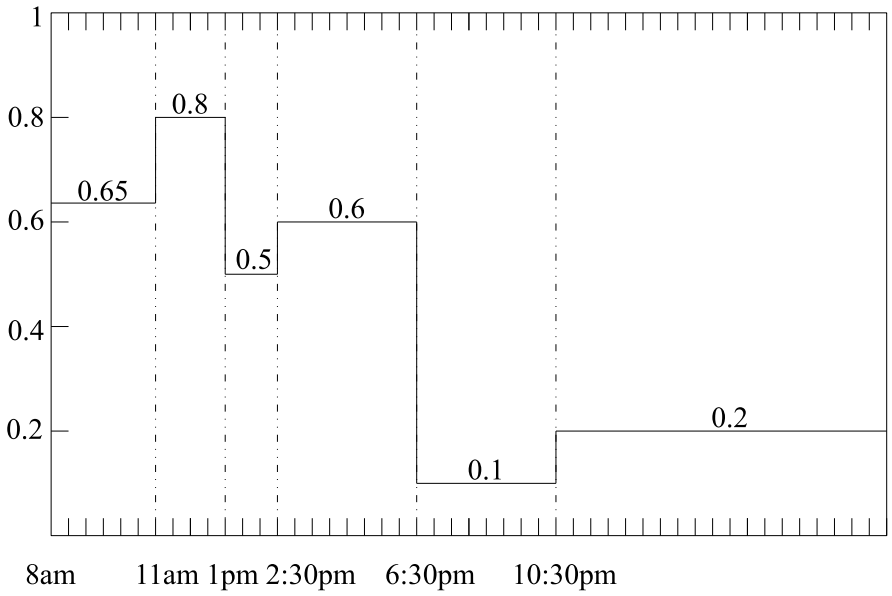


Figure 3.5 Traffic scenarios: average fraction (y-axis) of the nominal value used in a given scenario

Table 3.3 Card status for the 9-node network.

	Fixed Routing						Variable Routing					
Card	8	11	1	2.30	6.30	10.30	8	11	1	2.30	6.30	10.30
	11	1	2.30	6.30	10.30	8	11	1	2.30	6.30	10.30	8
2-5	0	0	0	0	0	0	0	0	0	0	0	0
4-5	0	0	0	0	0	0	0	0	0	0	0	0
5-6	0	0	0	0	0	0	0	0	0	0	0	0
5-8	0	0	0	0	0	0	0	0	0	0	0	0
1-2	0	0	0	0	0	0	0	2	0	0	0	0
2-3	0	0	0	0	0	0	0	2	0	0	0	0
2-8	0	0	0	0	0	0	0	2	0	0	0	0
7-8	0	0	0	0	0	0	0	2	0	0	0	0
8-9	0	0	0	0	0	0	0	2	0	0	0	0
1-4	2	2	1	2	1	1	0	0	0	1	0	0
6-9	2	2	1	2	1	1	0	0	0	0	0	0
4-7	2	2	2	2	1	1	0	0	0	1	0	0
1-7	2	2	2	2	1	1	2	2	2	2	0	1
1-3	2	2	2	2	1	1	2	2	2	2	1	1
3-6	2	2	2	2	1	1	0	0	0	0	0	0
3-9	2	2	2	2	1	1	2	2	2	2	1	1
4-6	2	2	2	2	1	1	0	0	0	0	0	0
7-9	2	2	2	2	1	1	2	2	2	2	1	1
sum	18	18	16	18	9	9	8	18	8	10	3	4

Table 3.4 Chassis status for the 9-node network.

Chassis	8a.m.	11a.m.	1p.m.	2.30p.m.	6.30p.m.	10.30p.m.
	11a.m.	1p.m.	2.30p.m.	6.30p.m.	10.30p.m.	8a.m.
5	0	0	0	0	0	0
2	0	0/**1**	0	0	0	0
8	0	0/**1**	0	0	0	0
1	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1/**0**	1/**0**	1/**0**	1	1/**0**	1/**0**
6	1/**0**	1/**0**	1/**0**	1/**0**	1/**0**	1/**0**
7	1	1	1	1	1	1
9	1	1	1	1	1	1

for both PAFRP and PAVRP approaches. Note that in Table 3.4, we report only one value when PAFRP and PAVRP obtain the same result. Due to the limited flexibility allowed by fixed routing, which does not allow to perfectly tailor the network configuration to the traffic levels, with PAFRP we observe that only three different switching patterns are used to optimize the six time periods. The first with 18 pairs of active line cards is used in three time intervals, i.e., (8–11a.m), (11a.m.–1p.m), and (2.30–6.30p.m.), the second with 16 pairs of active cards is applied within only the single time period (1–2.30p.m), while the third, which is the less consuming one with only 9 pairs of active cards, is employed during time periods (6.30–10.30p.m.) and (10.30–8a.m.). It is worth pointing out that in normal conditions (no failures or no special events), with fixed routing a chassis can be put to sleep only if it can be maintained deactivated for the whole time horizon.

Interestingly, while both PAFRP and PAVRP compute an equivalent configuration for the peak traffic time interval (11a.m.–1p.m), freedom in routing configuration allows PAVRP to double the number of line cards and chassis which are put to sleep during the other five time periods. The capability of PAVRP to accurately adapt network consumption to traffic variations seems not at all undermined by the switching-on limitations. The six switching patterns show remarkable differences, ranging from the 18 pairs of line cards powered on in time period (11a.m.–1p.m.) to the only 3 activated in (6.30p.m. to 10.30p.m.).

Routing Patterns

The higher flexibility of PAVRP is further proved the wider range of routing options exploited to satisfy the traffic demands (see Table 3.4.2). As expected, if routing can be adjusted along each different time interval, six different power state configurations, i.e., one for each time period (Table 3.3), are returned.

Energy Consumption

As natural consequence, results on energy consumption reveal the same trend previously emerged with the switching patterns. In Figure 3.6 we visually compare the power consumption profile of the fully active network (*reference* case, constant along the entire day) with that optimized with both fixed and variable routing. It is worth pointing out that variable routing yields an energy profile which is proportional to the traffic figure shown in Figure 3.5. Furthermore, except for the peak time interval (11a.m.–1p.m), where PAVRP performs sightly worse than PAFRP because of the energy consumed to reactivate two chassis which must be constantly kept on with PAFRP, we clearly notice that with variable routing the power consumption is practically halved w.r.t. fixed routing.

Table 3.5 Routing for the 9-node network.

Demand		Fixed Routing	Variable Routing					
$o^d - t^d$	Nom. value	Always	8-11a.m.	11a.m.-1p.m.	1-2.30p.m.	2.30-6.30p.m.	6.30-10:30p.m.	10.30p.m.-8a.m.
(1,3)	0.5	1-3	1-3	1-2-3	1-3	1-3	1-3	1-3
(1,9)	1.0	1-4-6-9	1-3-9	1-3-9	1-3-9	1-3-9	1-3-9	1-3-9
(1,7)	0.5	1-7	1-7	1-2-8-7	1-7	1-4-7	1-3-9-7	1-7
(3,9)	0.5	3-9	3-9	3-2-8-9	3-9	3-9	3-9	3-1-7-9
(3,7)	1.0	3-6-4-7	3-1-7	3-1-7	3-1-7	3-1-7	3-9-7	3-1-7
(9,7)	0.5	9-3-1-7	9-7	9-8-7	9-7	9-7	9-7	9-7
(3,1)	0.5	3-1	3-1	3-2-1	3-1	3-1	3-1	3-1
(9,1)	1.0	9-7-1	9-7-1	9-7-1	9-7-1	9-7-1	9-3-1	9-3-1
(7,1)	0.5	7-4-1	7-1	7-8-2-1	7-1	7-1	7-9-3-1	7-1
(9,3)	0.5	9-3	9-3	9-8-2-3	9-3	9-3	9-3	9-3
(7,3)	1.0	7-9-6-3	7-9-3	7-9-3	7-9-3	7-9-3	7-9-3	7-1-3
(7,9)	0.5	7-4-1-3-9	7-9	7-8-9	7-9	7-9	7-9	7-9

Table 3.6 Normalized consumption per hour (w.r.t the reference case) and congestion: fixed/-variable routing

Instance		Normalized Consumption							Congestion (ms/Mb)
dev.	μ	Daily cons.	8 11	11 1	1 2.30	2.30 6.30	6.30 10.30	10.30 8	see Appendix A
A	0.5	0.54/**0.38**	0.60/**0.50**	**0.60**/**0.60**	0.57/**0.36**	0.60/**0.46**	0.51/**0.30**	0.51/**0.31**	**5.33**/4.70
A	0.6	0.51/**0.35**	0.54/**0.36**	**0.56**/0.59	0.52/**0.35**	0.55/**0.36**	0.49/**0.30**	0.49/**0.30**	**5.63**/5.25
A	0.7	0.36/**0.32**	0.39/**0.35**	**0.39**/0.40	0.39/**0.34**	0.39/**0.35**	0.35/**0.30**	0.35/**0.30**	4.41/**6.46**
B	0.5	0.47/**0.31**	0.55/**0.42**	**0.55**/**0.55**	0.50/**0.30**	0.55/**0.38**	0.40/**0.22**	0.40/**0.23**	**13.73**/12.14
B	0.6	0.42/**0.27**	0.46/**0.30**	**0.49**/0.51	0.43/**0.29**	0.48/**0.30**	0.38/**0.22**	0.38/**0.22**	**14.52**/13.34
B	0.7	0.28/**0.25**	0.32/**0.29**	**0.33**/**0.33**	0.32/**0.27**	0.32/**0.28**	0.26/**0.22**	0.26/**0.22**	11.38/**16.50**
C	0.5	0.54/**0.38**	0.60/**0.49**	**0.60**/**0.60**	0.56/**0.35**	0.60/**0.45**	0.50/**0.30**	0.50/**0.31**	**2.13**/1.87
C	0.6	0.51/**0.34**	0.53/**0.35**	**0.55**/0.59	0.51/**0.34**	0.54/**0.35**	0.49/**0.30**	0.49/**0.30**	**2.25**/2.11
C	0.7	0.36/**0.32**	0.38/**0.34**	**0.38**/0.39	0.38/**0.33**	0.38/**0.34**	0.34/**0.30**	0.34/**0.30**	1.76/**2.55**

The overall energy savings achieved with both PAFRP and PAVRP in each single time period are shown in Figure 3.7. The trends already observed in Figure 3.6 are confirmed. In addition, we observe that the highest amount of savings is concentrated during the night time interval (10.30p.m.-8a.m.s), which, although slightly more loaded than (6.30p.m.-10.30p.m), has more influence on the overall savings because it is longer (9 hours and half).

Extensive results in terms of energy savings (w.r.t. to the always-on reference case) are summarized in Table 3.6, to compare the performance of PAFRP and PAVRP. All values are averaged over the three different traffic realizations a, b and c, while η_{on} and δ are set equal to 1 and 0.5, respectively. To facilitate the comparison, we highlight in bold the lowest consumption value for both the whole day and each single time period. As expected, also in this case we find out that the average consumption values achieved by variable routing are much lower than in the fixed routing scenario. It is worth pointing out that only within the peak traffic period, i.e., (11a.m.–1p.m), both models reach almost the same level of consumptions, with PAFRP which even outperforms PAVRP five times out of 9. This result

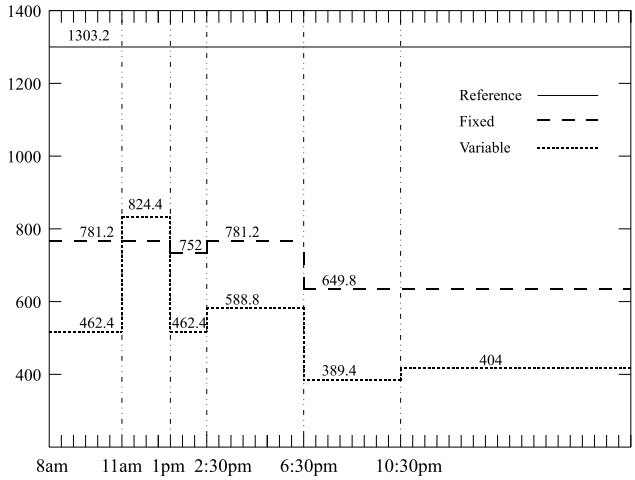


Figure 3.6 Hourly power consumption (in W): reference (all devices switched on), fixed routing, and variable routing.

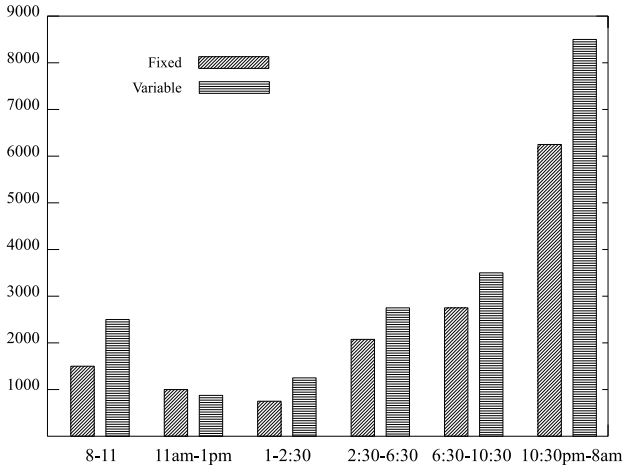


Figure 3.7 Overall energy savings (in Wh) with respect to the reference case (all devices switched on): fixed and variable routing.

has a quite straightforward explanation. The routing scheme configured by PAFRP must be valid for the whole day: since the relative intensity (w.r.t. the total incoming traffic) of each traffic demand remains almost constant along the day (it depends on the nominal value), a feasible routing is naturally obtained by optimizing w.r.t. the peak traffic time-period. With fixed routing, this is the only time period with a configuration optimally adapted to the traffic level. In addition, here PAFRP performs better than PAVRP because no chassis switching-on is required when routing is fixed (sleeping chassis maintains their state along the whole day).

Table 3.7 Total and average number of cards having been switched on during the study period for different values of δ, comparison fixed and variable routing

# switch-on	$\delta = 0.5$ nomax	$\delta = 0.25$ nomax	$\delta = 0.0$ nomax	$\delta = 0.5$ max1	$\delta = 0.25$ max1	$\delta = 0.0$ max1
Total number of card activations						
0	1602/1318	1602/1326	1602/1304	1500/1262	1500/1260	1500/1250
1	240/536	240/516	240/548	444/682	444/684	444/694
2	102/90	102/102	102/88	-	-	-
3	0/0	0/0	0/4	-	-	-
Average number of card activations						
0	59.33/48.81	59.33/49.11	59.33/48.30	55.56/46.74	55.56/46.67	55.56/46.3
1	8.89/19.85	8.89/19.11	8.89/20.30	16.44/25.26	16.44/25.33	16.44/25.7
2	3.78/3.33	3.78/3.78	3.78/3.26	-	-	-
3	0.00/0.00	0.00/0.00	0.00/0.15	-	-	-

Network Congestion

Besides power consumption values, in the last column of Table 3.6 and in Figure 3.8, we report a measure of average congestion computed in ms/Mb (see Appendix A) observed on network links. It is worth pointing out that variable routing yields the highest congestion only in two instances out of nine. That means that, although the higher number of switched-off elements, routing adaptation significantly improves the way network resources are exploited.

Impact of Card Reliability Constraints

To conclude this group of experiments, we show in Table 3.7 the results obtained by considering different values for both δ, i.e., 0, 0.25, 0.5 and η_{on}, i.e., 1 and 3. Note that using η_{on} larger or equal than $|S|/2$, which in our case is exactly 3, is equivalent to impose no limitation on the number of allowed switching-on. Consumption values are not reported because no significant difference has arisen by varying the parameters. However, this result suggests that switching limitations do not negatively affect energy savings. Furthermore, note that by imposing $\eta_{on} = 1$ we earn to significantly reduce the number of switching. Table 3.7 and Figure 3.9 clearly show that, on average, with no limitation almost four line cards are activated two times during a single day (one card even three times in one specific instance). The variation of δ seems to not produce any significant effects on the optimal solutions.

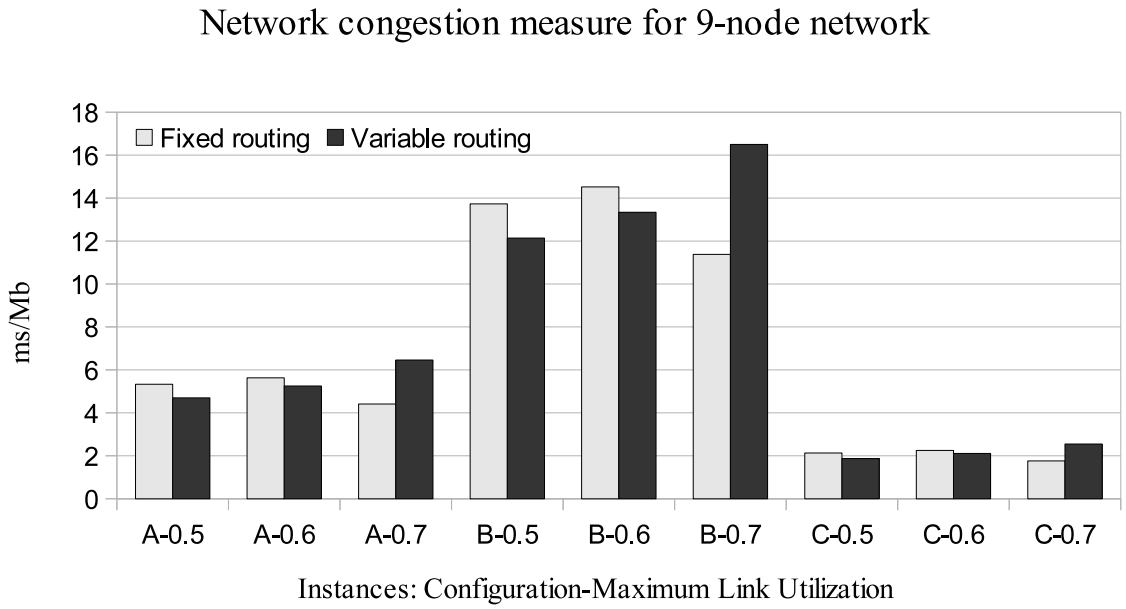


Figure 3.8 Overall congestion values for 9-node network. Comparison between fixed and variable routing solutions.

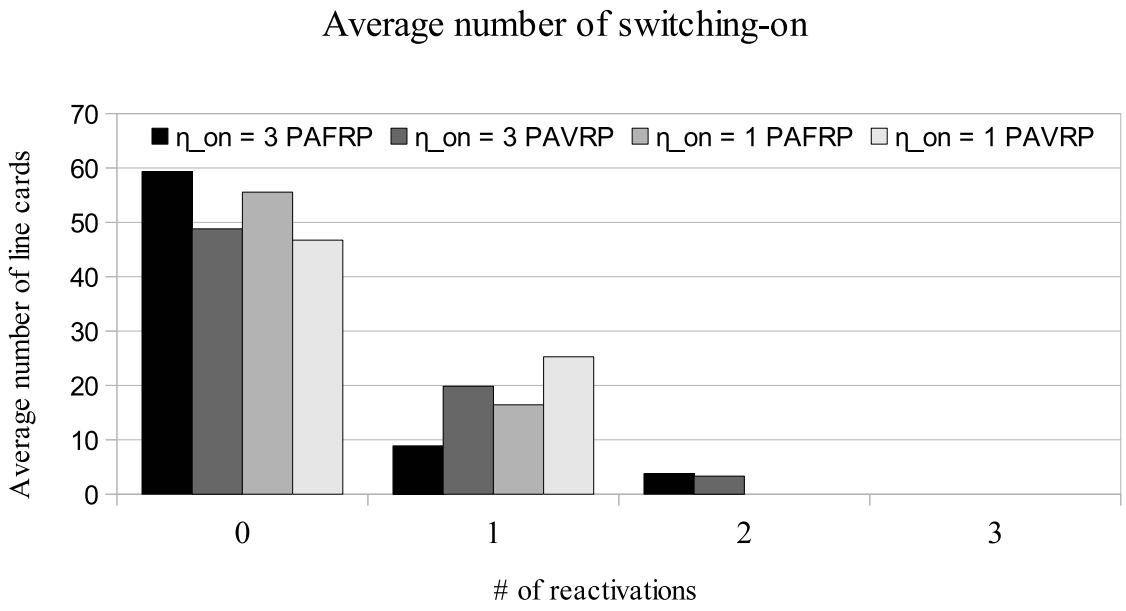


Figure 3.9 Average number of cards having been switched on during the study period for $\delta = 0.5$, comparison between fixed and variable routing.

Table 3.8 MILP formulation: normalized consumption (with respect to the reference case) and number of daily card switching on.

Instance			VAR, $\delta = 0.25$, $n_{ij} = 2$				VAR, $\delta = 0.25$, $n_{ij} = 1$				FIX, $\delta = 0.25$, $n_{ij} = 2$			
			Energy		Num switch-on		Energy		Num switch-on		Energy		Num switch-on	
ID	dev	rp	max1	nomax	max1	nomax	max1	nomax	max1	nomax	max1	nomax	max1	nomax
france network														
1	A	a	0.57	0.57	76	78	0.60	0.59	36	48	0.63	0.63	44	46
2	A	b	0.58	0.57	80	82	0.60	0.60	40	46	0.66	0.66	48	46
3	A	c	0.57	0.57	72	76	0.59	0.59	34	50	0.63	0.63	44	44
4	B	a	0.47	0.47	74	78	0.54	0.54	38	50	0.54	0.54	44	44
5	B	b	0.47	0.47	80	82	0.54	0.55	36	44	0.56	0.56	46	48
6	B	c	0.46	0.46	72	80	0.53	0.53	32	48	0.54	0.54	44	44
7	C	a	0.56	0.56	72	80	0.59	0.59	38	52	0.63	0.63	44	44
8	C	b	0.57	0.57	80	76	0.60	0.60	42	50	0.70	0.70	44	48
9	C	c	0.56	0.56	72	80	0.59	0.59	40	52	0.63	0.63	44	44
nobel-eu network														
10	A	a	0.59	0.59	56	50	0.66	0.66	30	32	0.67	0.67	40	32
11	A	b	0.59	0.59	54	58	0.66	0.66	26	30	0.67	0.67	32	38
12	A	c	0.59	0.59	60	64	0.66	0.66	32	24	0.67	0.67	36	34
13	B	a	0.49	0.49	58	54	0.62	0.61	22	32	0.57	0.56	26	34
14	B	b	0.49	0.49	64	62	0.61	0.61	40	32	0.56	0.56	38	34
15	B	c	0.49	0.49	64	56	0.61	0.61	44	32	0.56	0.56	34	36
16	C	a	0.59	0.59	50	54	0.66	0.65	20	30	0.66	0.66	32	40
17	C	b	0.58	0.58	54	60	0.65	0.65	32	40	0.66	0.66	36	40
18	C	c	0.58	0.58	62	62	0.66	0.65	40	30	0.66	0.66	30	38

Results for Larger Network Topologies

Let us now verify if the experiments with mid-sized realistic networks, i.e., `france` and `nobel-eu`, confirm the trends emerged for the 9-node network. As demonstrated by the overall results summarized in Table 3.8, the answer is positive. We consider nine instances for each topology, by combining the three equipment configurations A, B and C with the three randomly generated demand patterns a, b and c. Maximum utilization μ is set to 0.5 on all links. We have randomly selected 13 edge nodes in `france` and 14 edge nodes in `nobel-eu`. The time limit of CPLEX has been set to six hours.

The first group of columns describes describes the parameters, i.e., instance ID, equipment configuration, and random traffic profile *rp*. In the second block we report normalized consumption and switching-on number for three main groups of parameter setting: (i) variable routing, $\delta = 0.25$ and $n_{ij} = 2$, (ii) variable routing, $\delta = 0.25$ and $n_{ij} = 1$, (iii) fixed routing, $\delta = 0.25$ and $n_{ij} = 2$. For each class we distinguish between two cases, i.e., η_{on} equal to 1 ($max1$), and η_{on} equal to 3 ($nomax$).

Impact of the Number of Line Cards per Link

Results reported in Table 3.8, Figure 3.10 (`france` network) and Figure 3.11 (`nobel-eu` network) show that energy savings between 50% and 40% are achieved with variable routing.

As expected, an higher number of cards n_{ij} improves the overall savings: using more cards on a link is equivalent to increasing the number of link energy levels, which naturally makes energy management more agile and precise. In contrast, the total number of switching on is obviously reduced when $n_{ij} = 1$. Fixed routing decreases by about 8% the overall savings, but significantly reduces the total number of switching-on: one card on each used link has to be maintained activated along the entire day, while all the sleeping cards connected to a sleeping chassis remain constantly deactivated. Note that also in this case switching-on constraints have no negative effect on the optimal energy savings.

Comparison with the Fully Proportional Scenario

To conclude this part, we show in Figure 3.12 a comparison between the power consumption of PAVRP solutions and that achieved by an alternative model which considers the energy profile of each network device to be perfectly load proportional. That means that during period $\sigma \in S$, the consumption of a link $(i, j) \in A$ equipped with two line cards is equal to:

$$\pi_{ij} \frac{\sum_{d \in D} r^{d\sigma} x_{ij}^{d\sigma}}{n_{ij} c_{ij}}$$

In the fully proportional scenario the model is greatly simplified since all network devices are always active and, consequently, no inter-period constraint is considered. The results are quite impressive: load proportionality would allow to save up to 96% with respect to the classic full-active network with inelastic consumption. With a maximum utilization kept below 50% also during peak traffic periods, the consumption of the most loaded links is always below 50% , while that of the network chassis, which are the most consuming elements, hardly overcomes 10%. Note that with the equipment configuration B, due to the limited capacity of the line cards (155 Mbps in each direction) with respect to that of the chassis (16 Gbps), the utilization of the latter cannot overcome 15% in the most saturated case (all line cards with a utilization of 100%). The availability of fully proportional equipment would represent a very significant improvement in the field of green networking.

3.4.3 EA-LG

To evaluate the performance of EA-LG in terms of both energy savings and computing times, we compare in Table 3.9 the solutions obtained with EA-LG and those computed by

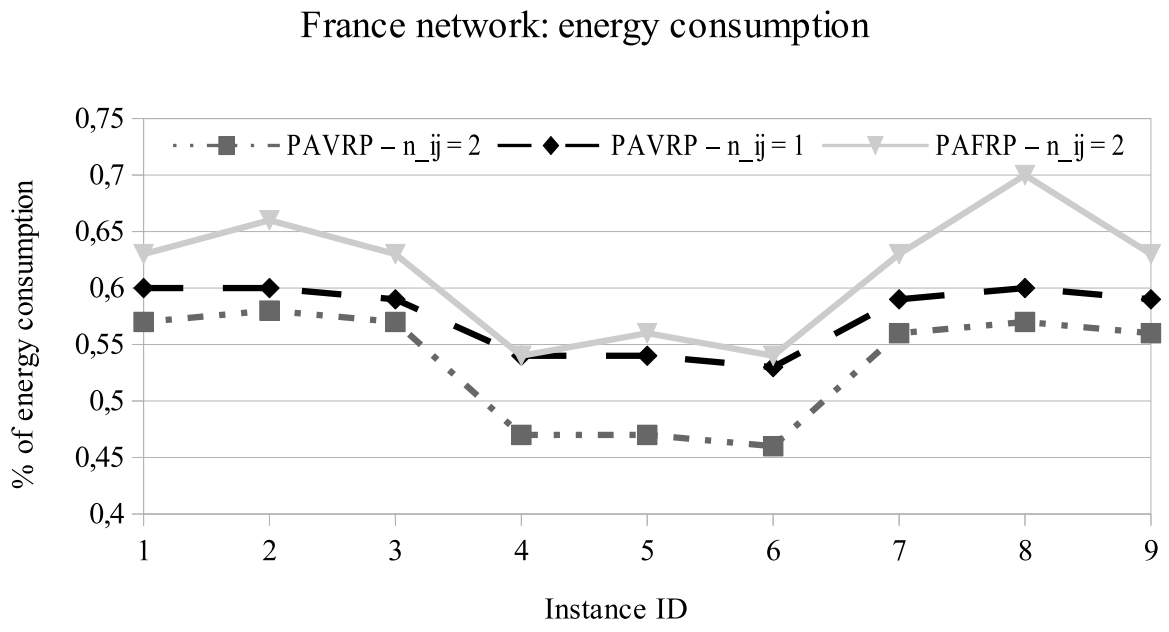


Figure 3.10 Energy consumption obtained with `france` network by considering different numbers of line cards per link and different routing constraints.

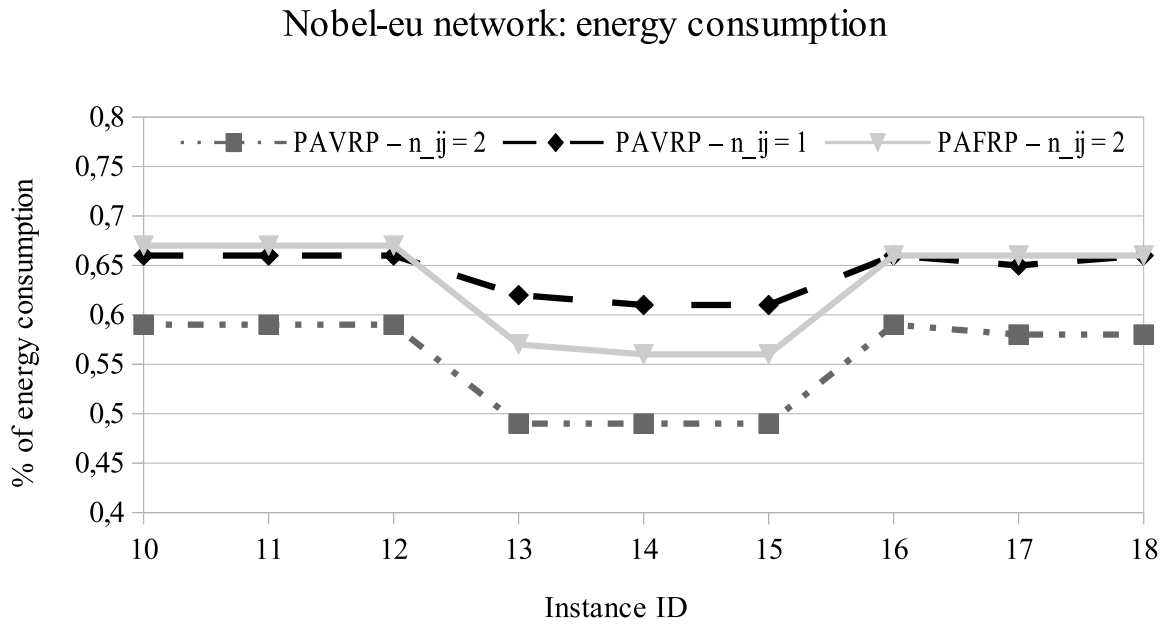


Figure 3.11 Energy consumption obtained with `nobel-eu` network by considering different numbers of line cards per link and different routing constraints.

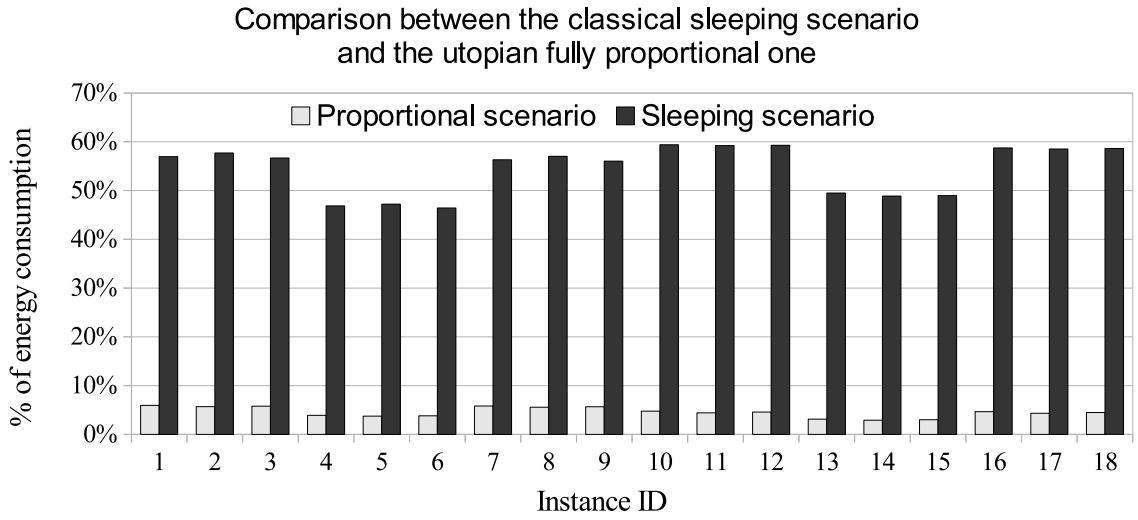


Figure 3.12 Power consumption comparison between the classic sleeping scenario solved with the PAVRP MILP formulation and the utopian fully proportional one.

solving the exact MILP formulation within a time limit of 6 hours. We consider variable routing (PAVRP), one card per link ($n_{ij} = 1$), switching-on cost δ equal to 0.25, switching limit η_{on} equal to 1 and maximum utilization μ equal to 0.5 on each link. All values are obtained by averaging over the three stochastic realizations a, b, and c.

In each instance, whose description is given by the first block, for both exact formulation and EA-LG, we report the overall power consumption relative to the always-on reference solution, the gap from the best lower bound computed by CPLEX Gap_{opt}, or the gap between EA-LG and MILP objective Gap_{milp}, and the computing times in minutes. Note that:

$$Gap_{opt} = \frac{Energy_{milp} - Lower_{bound}}{Energy_{milp}} \tag{3.24}$$

$$Gap_{milp} = \frac{Energy_{EA-LG} - Energy_{milp}}{Energy_{milp}} \tag{3.25}$$

Note that EA-LG is run for 50 iterations (*Second Level Multi-Start*) and the size of the RCL is set to 5% of $|D|$.

Energy Consumption

Results reported in Table 3.9 clearly point out that both the exact formulation and EA-LG achieve similar levels of power savings, which are around 45% and 35% with `france` and

`nobel-eu`, respectively, when half of the nodes are of the edge type. Note that optimized network consumption grows up to 93.7% (instance 36) as the number of edge nodes increases. Since chassis are by far more consuming (see Table 3.2) than line cards, the impossibility of putting to sleep any of them (when we have only edge nodes) dramatically reduces power savings opportunities.

Heuristic Accuracy

We evaluate the accuracy of EA-LG by comparing its consumption values with those returned by CPLEX for the MILP formulation (see Figure 3.14). The gap between EA-LG and MILP solutions is lower than 5.81% and 2.89% with `france` (test 22) and `nobel-eu` (test 28), respectively. In two tests, namely 35 and 32, EA-LG even outperform the formulation. We expect the gap values to grow negatively as instance dimensions are further increased. Note in fact that the MILP gap from the best lower bound (remind that we impose a time-limit) naturally increases while passing from `france` to the slightly larger `nobel-eu` (3 more nodes and 70 more traffic demands): the gap range rises from 1.87-6.29% to 2.04-8.86%.

It is worth pointing out that Gap_{opt} is larger, and thus Gap_{milp} lower, when no core node is present, i.e., no chassis can be put to sleep. Switching-off very consuming elements such as the chassis of the core nodes helps the solver to drastically reduce the solution space to visit along the branch and cut procedure. Note that we expect to observe Gap_{milp} to decrease as Gap_{opt} increases because the performance of EA-LG should worsen slower w.r.t. that of the MILP formulation. From the complexity point of view, while the MILP always reaches the 6-hours time-limit, less than half an hour is typically required by EA-LG to terminate when half of nodes are edge, and less than 1 hour in case no core node is present.

Germany Network

To further evaluate the scalability of EA-LG, we have tested it with the 50-nodes `germany50` network. Results are reported in Table 3.10. Also in this case power consumption is substantially reduced (up to 61.8% in test 39) and computing times are maintained below 47.5 minutes. Note that to reduce computing times we have decreased EA-LG iterations from 50 to 20.

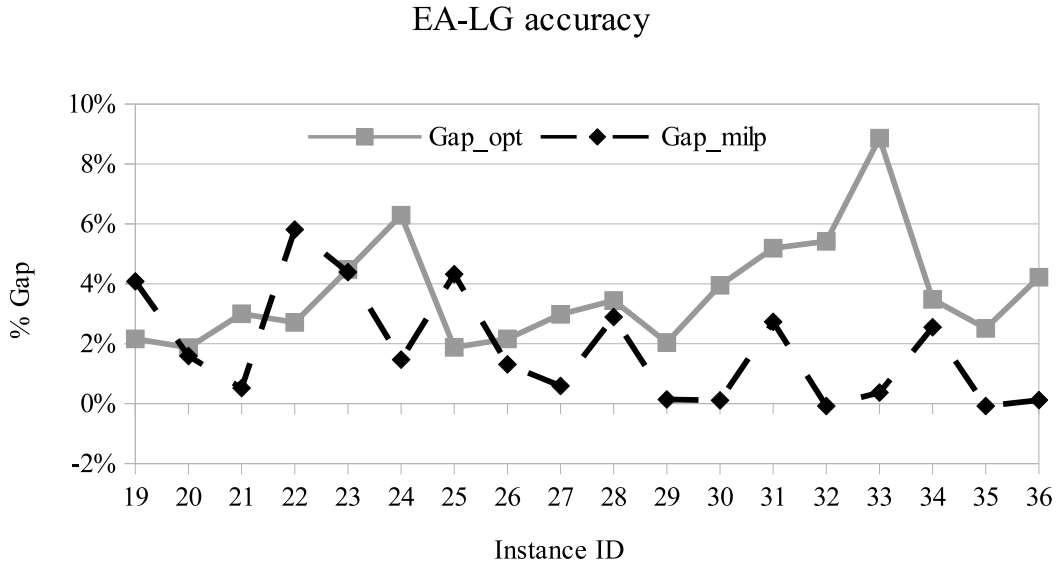


Figure 3.13 Analysis of the accuracy of PAVRP MILP with time limit of 6 hours (Gap_{opt}) and EA-LG run for 50 iterations (Gap_{milp}).

3.4.4 Economic Evaluation

According to the computational results just discussed, the energy power requirement of current backbone IP networks can be reduced from 30% up to 50% by implementing our energy-aware network management framework. From the perspective of the network operator, it is interesting to evaluate the consumption reduction in terms of cost cuts. Let `germany50` with B configuration be our reference network (this is the largest network of our test-bed in its most consuming configuration). When fully active, the network consumes

$$\text{Daily consumption} = 2\left(\sum_{i\in N}\bar{\pi}_i + \sum_{(i,j)\in A}\pi_{ij}n_{ij}\right)\sum_{\sigma\in S}h_\sigma. \tag{3.26}$$

Being $\bar{\pi}_i = 86.4$ W, $\pi_{ij} = 18.6$ W, $n_{ij} = 2$ and $\sum_{\sigma\in S} h_\sigma = 24$ h, the total daily consumption is equal to 521 KWh (the whole consumption is multiplied by two to take into account cooling equipment). Thus the yearly consumption is 521 KWh $\times$ 365 = 190.165 MWh, which, being the electricity cost equal to 0.20 € per KWh (see Cardona Restrepo *et al.* (2009)), results in an electricity bill of 38033 € . In that case, the energy-aware network management framework would allow to save up to about 20000 €.

Since in our experimental scenarios we considered network devices characterized by a limited capacity and, consequently, by a limited power consumption, it is worth evaluating

Table 3.9 Computational results for EA-LG with `france` and `nobel-eu`.

Variable routing (PAVRP)												
france						**PAVRP formulation**			**EA-LG**		
ID	$\|V\|$	$\|V^c\|$	$\|A\|$	$\|D\|$	dev	Energy	Gap_{opt}	t(min)	Energy	Gap_{milp}	t(min)
19	25	12	90	78	A	59.8%	2.16%	360	62.2%	4.08%	18.3
20	25	7	90	153	A	75.2%	1.87%	360	76.4%	1.59%	24.6
21	25	0	90	300	A	91.2%	3.00%	360	91.7%	0.52%	32.6
22	25	12	90	78	B	53.8%	2.71%	360	56.9%	5.81%	21.3
23	25	7	90	153	B	67.9%	4.48%	360	70.9%	4.39%	18.6
24	25	0	90	300	B	82.3%	6.29%	360	83.5%	1.47%	58.2
25	25	12	90	78	C	59.5%	1.88%	360	62.1%	4.32%	18.6
26	25	7	90	153	C	74.7%	2.16%	360	75.7%	1.31%	17.0
27	25	0	90	300	C	90.6%	2.98%	360	91.1%	0.59%	67.4
nobel-eu						**PAVRP formulation**			**EA-LG**		
ID	$\|V\|$	$\|V^c\|$	$\|A\|$	$\|D\|$	dev	Energy	Gap_{opt}	t(min)	Energy	Gap_{milp}	t(min)
28	28	14	82	91	A	65.9%	3.44%	360	67.8%	2.89%	26.5
29	28	7	82	210	A	73.9%	2.04%	360	74.0%	0.14%	34.4
30	28	0	82	377	A	94.1%	3.95%	360	94.2%	0.11%	45.2
31	28	14	82	91	B	61.6%	5.19%	360	63.3%	2.73%	32.1
32	28	7	82	210	B	69.2%	5.42%	360	69.2%	-0.08%	27.3
33	28	0	82	377	B	87.7%	8.86%	360	88.0%	0.37%	47.7
34	28	14	82	91	C	65.7%	3.48%	360	67.4%	2.55%	25.2
35	28	7	82	210	C	73.8%	2.51%	360	73.7%	-0.08%	26.0
36	28	0	82	377	C	93.7%	4.22%	360	93.8%	0.12%	46.4

Table 3.10 Computational results for `germany50`.

Variable routing (PAVRP)							
germany50 network						**EA-LG**	
ID	$\|V\|$	$\|V^c\|$	$\|A\|$	$\|D\|$	dev	Energy	t(min)
37	50	25	176	182	A	66.8%	29.8
38	50	12	176	397	A	79.0%	26.8
39	50	25	176	182	B	61.8%	47.5
40	50	12	176	397	B	73.0%	24.9
41	50	25	176	182	C	66.0%	26.3
42	50	12	176	397	C	79.0%	46.2

what would be the potential savings in case of more consuming equipment. Assume to equip each network node with a Juniper T640 chassis (hourly consumption of 1114 W) and each link $(i, j) \in A$ with n_{ij} Type-4 FPC, 40 Gbps full duplex line cards (hourly consumption of 394 W) Van Heddeghem *et al.* (2012b): in this case the yearly power consumption would amount to 3.4 GWh, and the resulting electricity bill would rise up to 681000 € per year.

Therefore, SEANM would allow to save around 300000 € per year for a single backbone network.

3.4.5 Single Period Heuristic

As done with EA-LG, we exploit `france` and `nobel-eu` instances to evaluate the effectiveness of our single period/online heuristic called EA-STH. In particular we are interested to quantify the performance degradation caused by the limited amount of information available when considering each time period separately.

During each run of EA-STH we start from a given time period, and then solve one by one 6 optimization problems, each one corresponding to a single time period. When the first time interval is considered we assume that all devices are powered on. Each time period is optimized within a time limit of 5 minutes. Note that such a reduced time-limit is necessary if we want to apply the algorithm in an online fashion. Since final results may vary as we select a different starting period, we repeat the optimization sequence six times, by changing from time to time the starting period.

In Table 3.11 and Figure 3.14 we report the worst results obtained by adjusting the starting time interval (the worst because if we apply the algorithm online we cannot choose the starting time interval). We consider variable routing (PAVRP), $\delta = 0.25$, $n_{ij} = 2$ and $\mu = 0.5$. In Table 3.11 we maintain the same notation used in previous tables, except for "# Card on" and "# Switch on", which represent, respectively, the overall (summed over all the six time periods) number of active cards and card switching-on.

Heuristic Accuracy

Results show that online EA-STH reduces power consumption by at least 40% and 38% in `france` and `nobel-eu`. The gap between online EA-STH and offline MILP solutions is on average around 5% (with a peak of 10% in instances 4 and 15). In online EA-STH solutions, 60 more line cards are kept activated, on average, w.r.t. MILP solutions.

Switching patterns

For what concerns the switching-on trend, we observe an interesting behaviour. With `france` we notice that the number of switching on is always lower in on-line EA-STH than in the off-line MILP. That means that the partial information used by the online algorithm is not sufficient to prevent the procedure from reaching the η_{on} limit too early for too many cards. This phenomenon restricts the algorithm choice along the last considered periods, wherein a significant amount of cards must be kept activated independently of their utilization level.

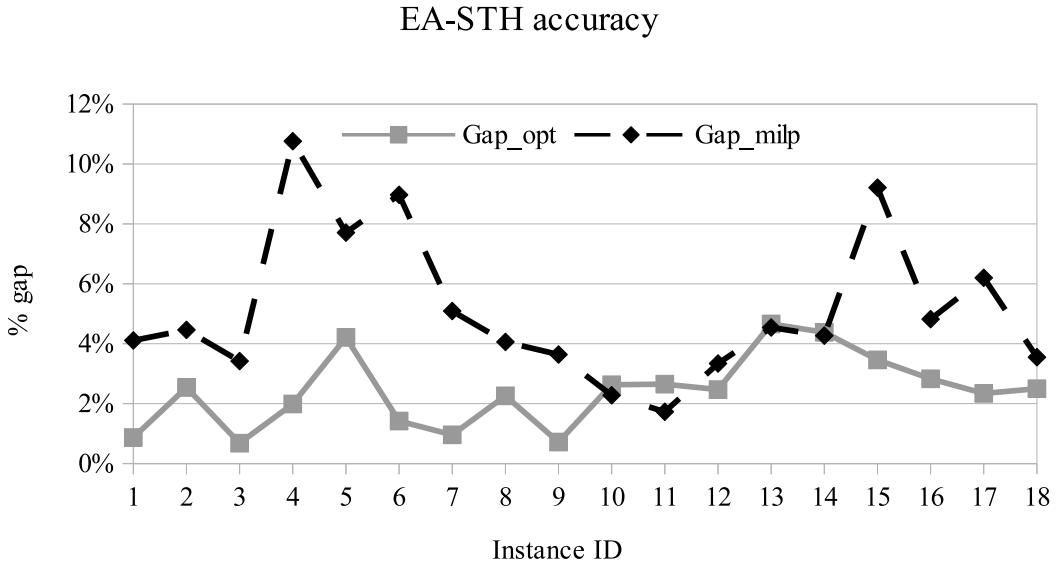


Figure 3.14 Analysis of the accuracy of PAVRP MILP with time limit of 6 hours (Gap_{opt}) and EA-STH with single period time limit of 5 minutes (Gap_{milp}).

A different trend is instead observed in `nobel-eu`, where switching-on are more numerous in on-line EA-STH solutions in 6 instances out of 9 (tests 10-11-15-16-17-18). In this case we believe that, due to `nobel-eu` topological features, on-line EA-STH may find additional cards to put to sleep in place of the blocked ones.

3.4.6 Evaluation with Real Traces

The validity of our novel energy-aware planning approach has been further investigated by conducting a group of tests involving the `geant` (Orlowski *et al.*, 2010) network (23 nodes ad 72 links) and a set of real traffic matrices (Uhlig, 2011). The traffic data-set includes traffic matrices computed every 15 minutes for a period of 6 months.

After a quantitative analysis of the traffic profiles, we split the single day among six time periods, namely (4:00a.m.-8:30a.m.), (8:30a.m.-11:00a.m.), (11:00a.m.-2:00p.m.), (2:00p.m.-6:00p.m.), (6:00p.m.-10:00p.m.) and (10:00p.m.-4:00a.m.). We report in Figure 3.15 the results obtained by applying the optimized network configuration over six consecutive days. In particular, the network configuration of each single day is computed by considering the traffic values observed along the previous day.

The traffic request of demand d during period σ, i.e., $r^{d\sigma}$, is computed as the average values over all the 15-minutes traffic matrices which belong to period σ. Note that this

Table 3.11 Performance comparison between online EA-STH and the offline MILP: normalized consumption (with respect to the reference case), number of daily card switching up and total number of cards powered on.

Instance			VAR, $\delta = 0.25$, $n_{ij} = 2$, $\mu = 0.5$							
			MILP				EA-STH			
ID	dev	rp	Energy	Gap_{opt}	# Card on	# Switch on	Energy	Gap_{milp}	# Card on	# Switch on
france network										
1	A	1	0.570	0.87%	344	76	0.593	4.11%	402	66
2	A	2	0.577	2.54%	342	80	0.603	4.46%	418	72
3	A	3	0.567	0.68%	336	72	0.586	3.42%	386	70
4	B	1	0.469	1.99%	344	74	0.519	10.76%	413	70
5	B	2	0.472	4.21%	344	80	0.508	7.71%	402	66
6	B	3	0.464	1.42%	336	72	0.506	8.97%	394	70
7	C	1	0.563	0.96%	344	72	0.592	5.09%	412	66
8	C	2	0.570	2.26%	340	80	0.593	4.06%	404	70
9	C	3	0.560	0.72%	336	72	0.581	3.64%	386	66
nobel-eu network										
10	A	1	0.594	2.63%	334	56	0.608	2.28%	382	62
11	A	2	0.592	2.65%	326	54	0.602	1.73%	362	56
12	A	3	0.593	2.47%	326	60	0.612	3.34%	372	58
13	B	1	0.495	4.66%	330	58	0.517	4.54%	364	58
14	B	2	0.489	4.38%	324	64	0.510	4.27%	368	56
15	B	3	0.490	3.46%	324	64	0.535	9.21%	378	78
16	C	1	0.587	2.83%	334	50	0.616	4.82%	394	60
17	C	2	0.585	2.34%	324	54	0.621	6.20%	398	66
18	C	3	0.586	2.50%	326	62	0.607	3.55%	372	62

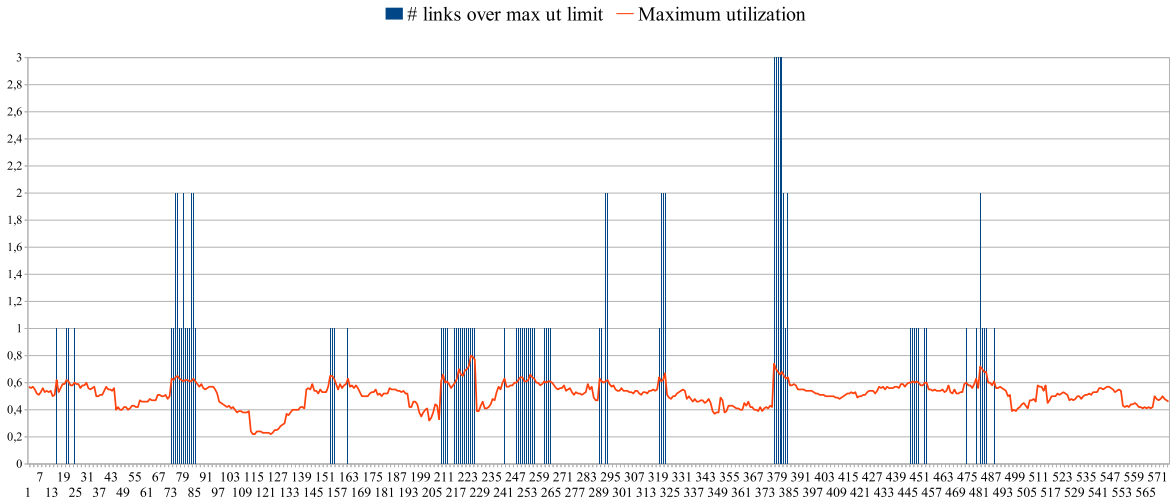


Figure 3.15 Maximum utilization values and number of links over the maximum utilization limit during the experimentation with the real traffic matrices.

prediction approach, according to which we predict the future traffic to be equal to the average values observed along the previous day, is extremely simple. Estimations made by network operators are typically much more sophisticated. We consider maximum utilization μ equal to 60%, $\delta = 0.25$, $n_{ij} = 1$ and $\eta_{on} = 1$.

Figure 3.15 shows both maximum link utilization and number of links violating μ observed for each 15-minutes traffic matrix when the MILP network configuration is applied. It is worth pointing out that, despite the simplicity of the prediction method, maximum utilization is maintained below 60% for most of the time and never above 80%. In addition, the number of links violating the μ is typically 0 or 1, with a maximum peak of three links observed for one hour during the fourth day.

3.5 Conclusions

In this chapter we have addressed the problem of minimizing the daily power consumption of IP networks operated with a flow-based routing protocol such as MPLS. We have formalized two different multi-period MILP formulations for centralized energy-aware network management, i.e., PAFRP with fixed routing and PAVRP with variable routing, and proposed both exact and heuristic approaches to handle them. Our approach exploits the temporal variations of the traffic demands to split a single day in a few time periods and smartly tailor the network configuration to the traffic conditions expected in each time interval.

We showed that, under realistic traffic conditions, it is possible to reduce the overall consumption by up to 50%. As expected, the possibility of adjusting the routing in each time period (PAVRP) increases the power saving by 15% with respect to the fixed routing case (PAFRP). We found that introducing the card reliability constraints, which limits the the number of times that a single line card can be switched on within the time-horizon, does not negatively affect the model performance in terms of energy consumption.

To handle instances up to 80 nodes, we presented a GRASP-based heuristic called EA-LG and the single-time period algorithm EA-STH. Results showed that both methods performed well, obtaining solutions with a very limited gap from the optimal solutions computed with the formulations. Furthermore, we showed that EA-STH could be potentially employed in an online fashion when traffic forecasts are not available or incomplete. Though the incomplete knowledge on traffic conditions and the impossibility of globally optimizing the whole daily configuration, the accuracy of EA-STH-online has proved to be satisfactory, with a gap from offline solutions typically lower than 10%. However, also if more time-consuming, the offline approach proved to be important to evaluate the best possible savings, and should be applied every time a long term prevision on the demand pattern is available.

CHAPTER 4

On Robustness and Survivability in SEANM with Flow-based Routing

Besides QoS and energy consumption, a crucial concern for network operators is represented by network survivability, which is intended as the network ability to efficiently react to unpredictable events such as device failures or unexpected peaks of traffic while preserving network performance.

It appears quite evident that the goal of energy-aware network management, which consists of adapting the set of active resources to the incoming traffic load, is openly in conflict with the necessity by the operator to reserve some spare capacity whose aim is to guarantee the correct network functioning whenever network conditions present an anomaly. Finding the most efficient trade-off between power reduction and survivability requirements represents an important open issue for the networking community.

Within the notion of network survivability, we identify two different sub-concepts, i.e., network resilience to failures and network robustness to traffic variations. Network resilience to failures can be provided in both an implicit and explicit fashion. To the first class belong those techniques which consider a failure probability measure for each device and aim at determining the network routing which minimizes the overall failure probabilities for the whole set of traffic demands. By contrast, in the second class we include the so-called *protection* techniques, whose aim is to explicitly reserve the backup capacity to be used whenever a failure affects a traffic demand. Backup resource allocation typically involves the definition, for each traffic demand, of a dedicated backup path on which a certain amount of spare capacity is reserved.

Being backup path definition the main step to provide explicit protection against failures, it comes natural to consider flow-based routing protocols such as MPLS to be the most appropriate to implement explicit protection techniques. Not surprisingly, MPLS offers in fact a functionality to define multiple backup paths to be used in case of problems detected on the primary one.

For what concerns robustness to traffic variations, the range of possible approaches is less diversified. If considered during the planning phase, providing robustness results in leaving some spare capacity on both links and nodes according to some criteria; if addressed on-line, it means relying on some mechanisms to dynamically adjust the network configuration.

In the reminder of the chapter, we quantify the energy cost of providing network survivability, we investigate the trade-off between being energy-aware and providing survivability,

and, finally, we answer to the question whether it is possible to design a green-survivable network. For this purpose we integrate different survivability constraints into the SEANM-FB methodologies previously presented in Chapter 3.

In Section 4.1 we provide an overview on our approaches to guarantee network resilience to failures and robustness to traffic variations. In Sections 4.2 and 4.3 we then show how modifying the PAVRP MILP formulation presented in Chapter 3 to explicitly consider protection against single-link failures and robustness to traffic variations, respectively. Finally, we discuss resolution methods in Section 4.4 and present computational results in Section 4.5.

4.1 Our Approach to Network Survivability

In this section we provide a general overview on how we integrate both energy-aware and survivability aspects into our optimization framework for SEANM-FB. To better point out the key points of our approach, in Section 4.1.3 we provide and thoroughly discuss a visual example which clearly highlights the impact of resilience and robustness on energy efficiency. For what concerns energy-aware network management, both general approaches, e.g., putting unused devices to sleep, and modeling assumptions are those described for PAVRP in Chapter 3, to which we refer the reader to get detailed information on the the energy-aware aspects.

4.1.1 Network Resilience

We guarantee network resilience to failures by considering two different protection schemes, namely *dedicated* and *shared* protection. Since node failures and multiple-link failures are typically very unlikely to occur, we implement both strategies to protect normal network operations w.r.t. single link-failure events.

To offer either *dedicated* or *shared* protection, during the planning phase a dedicated backup path must be assigned to each traffic demand. The backup path, which has to be link-disjoint (node disjoint if we consider node failures) with respect to the primary one, is meant to carry the primary traffic only when the primary path is not available because of a link failure. The two strategies differ on how spare capacity is allocated on the backup paths. According to *dedicated* protection the same amount of capacity is reserved on both primary and backup paths. With *shared* protection the allocation scheme is more sophisticated and allows the backup paths whose corresponding primary paths are link-disjoint to share the backup capacity on the common links. *Shared* protection exploits the observation that, in a scenario where only single-link failures occur, these backup paths will be never used simultaneously.

For what concerns backup resource allocation, *shared* protection requires a smaller amount of resources and is thus more efficient than *dedicated* protection. Under the energy-saving point of view, less backup capacity naturally results in higher energy savings potentially achievable. However, as we will see in Sections 4.2 and 4.5, modeling the resource allocation scheme of *shared* protection makes the problem significantly more complex.

4.1.2 Network Robustness

Although the regular daily/weekly behaviour of Internet traffic (Bolla *et al.*, 2012) is typically exploited by network providers to accurately predict the traffic matrices expected during future time-horizons (see Casas *et al.*, 2009, for state-of-the-art traffic matrix estimation methods), real traffic values naturally deviate within a certain range around the predicted values.

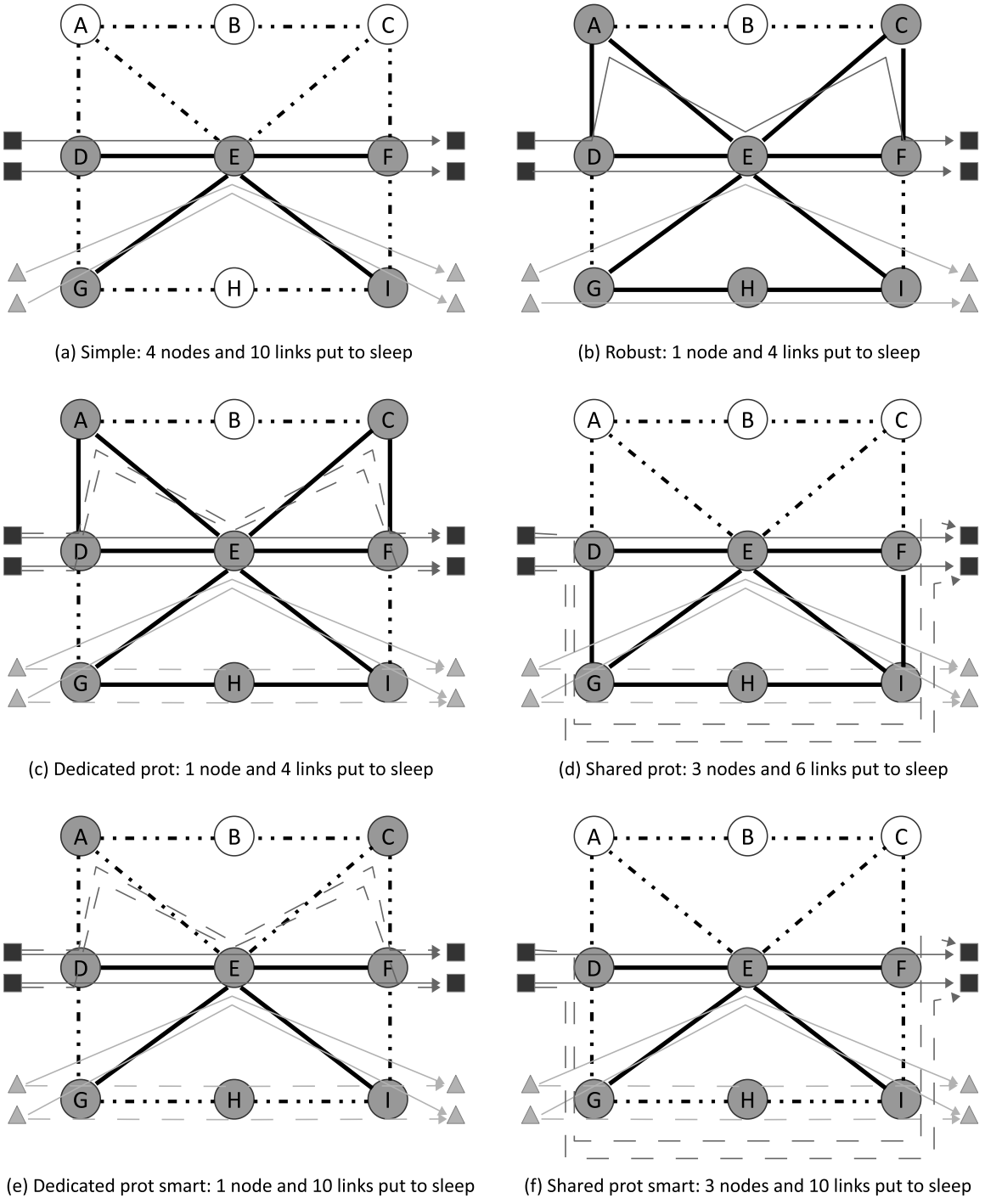
We exploit state-of-the-art robust optimization (RO) techniques to drive the optimization model to reserve enough spare capacity to cope with unpredictable traffic variations. In particular, we integrate our PAVRP MILP formulation with the modeling framework proposed in (Bertsimas *et al.*, 2011). According to (Bertsimas *et al.*, 2011), each traffic demand is assumed to be uncertain into a close symmetric interval centered on its predicted traffic value. The robustness degree of the computed solutions, i.e., the amount of spare resources used to accommodate traffic variations, is tuned by adjusting a set of input parameters representing the maximum total deviation assumed on each link.

4.1.3 A Visual Example

Let us analyze the visual example presented in Figure 4.1 to better comprehend how protection and robustness techniques influence the outcomes of the optimization framework.

In the network of Figure 4.1, each link has 2 units of capacity, while each one of the four traffic demands requests 1 unit of traffic. In Figure 4.1(a) we report the *simple* case (classic PAVRP) without any additional requirements. As expected, this scenario is the most energy-efficient, with even 4 nodes and 10 full-duplex links put to sleep. If traffic demand are considered uncertain (Figure 4.1(b)), i.e., each traffic request might oscillate around the predicted value of 1, no link can be used by more than one demand. Both additional links and nodes, more precisely 6 and 3, respectively, must be thus switched on w.r.t. to the *simple* case

If we implement *dedicated* protection (Figure 4.1(c)), 6 additional links and 3 more nodes are required to allocate the bandwidth on the backup paths. However, note that the number of active links and nodes can be substantially reduced by applying *shared* protection (Figure



(a) Simple: 4 nodes and 10 links put to sleep

(b) Robust: 1 node and 4 links put to sleep

(c) Dedicated prot: 1 node and 4 links put to sleep

(d) Shared prot: 3 nodes and 6 links put to sleep

(e) Dedicated prot smart: 1 node and 10 links put to sleep

(f) Shared prot smart: 3 nodes and 10 links put to sleep

Figure 4.1 Energy consumption minimization vs resilience requirements.

4.1(d)), according to which the backup paths of the two demands $G-I$ can share their backup capacity on links (G, H) and (H, I) with the backup paths of the two demands $D-F$. W.r.t. *dedicated* protection we put to sleep 2 more nodes and 2 more links. Note that the primary paths of both demand $G - I$ and $D - F$ are link disjoint.

To further reduce the energy-cost of network resilience, besides the protected approaches just mentioned to which we will refer as *classic*, we introduce a novel protected variant called *smart*, where we can put to sleep the line cards carrying backup paths only. Backup line cards, whose reactivation time are in the order of a few milliseconds (Hays, 2007), can be in fact activated only when a failure occurs without causing any service disruption. Note that the same is not valid for router chassis, which requires at least some seconds to complete to wake up. Since only-backup line cards stay active for a negligible amount of time corresponding to failure periods, their power consumption can be ignored within the SEANM-FB framework. Figures 4.1(e) and 4.1(f), clearly show that further power reductions are achieved by means of *smart* approaches: 6 and 4 more links can be put to sleep by, respectively, *smart dedicated* protection and *smart shared* protection. It is worth pointing out that *smart* techniques tend to split network links between two subsets, i.e., those carrying only primary paths, and those used by only backup paths.

Finally, although not reported in the visual example, we introduce a novel element which allows to further increase energy-savings. Since the occurrence of a link failure is a very unlikely event, we believe that, during failure periods, the network administrator might allow the network to operate under a higher maximum link utilization threshold $\mu_{ij}^{bck}\ \forall (i,j) \in A$ (larger than $\mu_{ij}\ \forall (i,j) \in A$). If carefully chosen, the new failure threshold should not cause an excessive degradation of the overall QoS. On link $(i,j) \in A$, μ_{ij}^{bck} accounts for both primary and backup paths, while the original threshold μ_{ij} is respected by primary traffic alone. Increasing the link maximum utilization limit during the very short failure periods substantially improves the energy efficiency of the network.

4.2 MILP Formulations for SEANM-FB with Network Resilience

We show in this section how to integrate path protection within the MILP formulation for multi-period SEANM-FB with variable routing presented in 3.2.2.

4.2.1 Variable Routing with Dedicated Protection

To explicitly consider *classic dedicated* protection, in addition to sets, parameters and variables already defined in Section 3.2 (to which we refer the reader), we introduce binary variables $\xi_{ij}^{d\sigma}$ which are equal to 1 if link $(i,j) \in A$ is used by the backup path of demand

$d \in D$ within time period $\sigma \in S$. We can thus express the following MILP formulation for PAVRP with *dedicated* protection:

$$\min \sum_{\sigma \in S} h_\sigma \sum_{j \in V} \pi_j y_j^\sigma + \sum_{\sigma \in S} h_\sigma \sum_{(i,j) \in A} \pi_{ij} w_{ij}^\sigma + \sum_{\sigma \in S} \sum_{j \in N} z_j^\sigma \tag{4.1}$$

s.t.

$$\sum_{\substack{j \in V: \\ (i,j) \in A}} x_{ij}^{d\sigma} - \sum_{\substack{j \in V: \\ (j,i) \in A}} x_{ji}^{d\sigma} = \begin{cases} 1 & \text{if } i = o_d, \\ -1 & \text{if } i = t_d, \\ 0 & \text{otherwise} \end{cases}, \qquad \forall i \in V,\, d \in D,\, \sigma \in S \tag{4.2}$$

$$\sum_{\substack{j \in V: \\ (i,j) \in A}} \xi_{ij}^{d\sigma} - \sum_{\substack{j \in V: \\ (j,i) \in A}} \xi_{ji}^{d\sigma} = \begin{cases} 1 & \text{if } i = o_d, \\ -1 & \text{if } i = t_d, \\ 0 & \text{otherwise} \end{cases}, \qquad \forall i \in V,\, d \in D,\, \sigma \in S \tag{4.3}$$

$$x_{ij}^{d\sigma} + \xi_{ij}^{d\sigma} \leq 1, \qquad \forall (i,j) \in A, d \in D,\, \sigma \in S \tag{4.4}$$

$$x_{ij}^{d\sigma} + \xi_{ji}^{d\sigma} \leq 1, \qquad \forall (i,j) \in A, d \in D,\, \sigma \in S \tag{4.5}$$

$$\sum_{\substack{i \in V: \\ (i,j) \in A}} \sum_{d \in D} r^{d\sigma} \left(x_{ij}^{d\sigma} + \xi_{ij}^{d\sigma} \right) + \sum_{\substack{i \in V: \\ (j,i) \in A}} \sum_{d \in D} r^{d\sigma} \left(x_{ji}^{d\sigma} + \xi_{ij}^{d\sigma} \right) \leq C_j y_j^\sigma, \qquad \forall j \in V,\, \sigma \in S \tag{4.6}$$

$$z_j^\sigma \geq \delta \pi_j \left(y_j^\sigma - y_j^{\sigma-1} \right), \qquad \forall j \in V,\, \sigma \in S \tag{4.7}$$

$$\sum_{d \in D} r^{d\sigma} x_{ij}^{d\sigma} \leq \mu_{ij} c_{ij} w_{ij}^\sigma, \qquad \forall (i,j) \in A,\, \sigma \in S \tag{4.8}$$

$$\sum_{d \in D} r^{d\sigma} \left(x_{ij}^{d\sigma} + \xi_{ij}^{d\sigma} \right) \leq \mu_{ij}^{bck} c_{ij} w_{ij}^\sigma, \qquad \forall (i,j) \in A,\, \sigma \in S \tag{4.9}$$

$$w_{ij}^\sigma = w_{ji}^\sigma, \qquad \forall (i,j) \in A,\, \sigma \in S :\, i < j \tag{4.10}$$

$$\sum_{k=1}^{n_{ij}} u_{ijk}^\sigma \geq w_{ij}^\sigma - w_{ij}^{\sigma-1}, \qquad \forall (i,j) \in A,\, \sigma \in S \tag{4.11}$$

$$\sum_{\sigma \in S} u_{ijk}^\sigma \leq \eta_{on}, \qquad \forall (i,j) \in A,\, k \in \{1, \ldots, n_{ij}\} \tag{4.12}$$

$$y_h^\sigma,\, x_{ij}^{d\sigma},\, \xi_{ij}^{d\sigma} \in \{0,1\}, \qquad \forall h \in V,\, (i,j) \in A,\, d \in D,\, \sigma \in S \tag{4.13}$$

$$u_{ijk}^\sigma \in \{0,1\}, \qquad \forall (i,j) \in A,\, \sigma \in S,\, k \in \{1, \ldots, n_{ij}\} \tag{4.14}$$

$$w_{ij}^\sigma \in \{0, \ldots, n_{ij}\}, \qquad \forall (i,j) \in A,\, \sigma \in S \tag{4.15}$$

$$z_j^\sigma \geq 0, \qquad \forall j \in V,\, \sigma \in S. \tag{4.16}$$

Objective function 4.1 and Constraints (4.2), (4.7), (4.10–4.12), (4.14–4.16) are kept unchanged from the general PAVRP formulation for pure energy-aware network management without any sort of protection. The single unsplittable backup path assigned to each traffic demand $d \in D$ is computed by means of flow conservation Constraints (4.3). Two new groups of constraints represented by Equations (4.4–4.5) are used to ensure that each backup path is link-disjoint from the corresponding primary path. Novel chassis and line card capacity Constraints, (4.6) and (4.8–4.9), respectively, are modified to consider both primary and backup traffic. For each demand $d \in D$, during time period $\sigma \in S$ we reserve $r^{d\sigma}$ units of traffic on the primary path as well as on the backup one. Note that there are both standard capacity Constraints (4.8) for normal network operations (no failure) which consider only primary traffic and a standard utilization limit μ_{ij}, and failure capacity Constraints (4.9) characterized by a higher utilization threshold μ_{ij}^{bck} which has to be respected by both concurrent primary and backup traffic during failure periods. The network operator may adjust both μ_{ij} and μ_{ij}^{bck} to reach the desired trade-off in terms of power reduction and QoS provided during both normal and failure conditions.

4.2.2 Variable Routing with Shared Protection

To implement *shared* protection we can take as reference the MILP formulation (4.1)–(4.16) for the *dedicated* case and introduce some modifications to correctly compute the amount of backup bandwidth allocated on each link. We know that according to *shared* protection, the amount of backup capacity reserved on a given link $(i, j) \in A$ must be large enough to support the worst case single-link failure: the latter is defined as the link breakdown which produces the largest shift of traffic from the failed link toward the backup link (i, j) considered. To correctly compute the backup capacity on a link $(i, j) \in A$, we need additional variables and constraints to evaluate the size of the traffic shifting induced by each link failure toward link (i, j) and then take the highest one. For this purpose, a new set of binary variables $g_{ijkl}^{d\sigma}$ is introduced: $g_{ijkl}^{d\sigma}$ is equal to 1 if, during time period $\sigma \in S$, both backup path and primary path of demand $d \in D$ are routed along link $(i, j) \in A$ and link $(k, l) \in A$, respectively. In that case we know that link (i, j) is used by demand d whenever link (k, l) fails. The following group of constraints is introduced to correctly compute the values of $g_{ijkl}^{d\sigma}$ variables:

$$g_{ijkl}^{d\sigma} \geq x_{ij}^{d\sigma} + \xi_{kl}^{d\sigma} - 1, \quad \forall (i, j), (k, l) \in A,\, d \in D,\, \sigma \in S. \tag{4.17}$$

Variables $g_{ijkl}^{d\sigma}$ are then exploited in a restated version of failure capacity constraints (4.9)

to reserve, on each link, the right amount of backup capacity to cope with each single failure:

$$\sum_{d\in D} r^{d\sigma}\left(x_{ij}^{d\sigma} + g_{klij}^{d\sigma}\right) \leq \mu_{ij}^{bck} c_{ij} w_{ij}^{\sigma}, \quad \forall (i,j),(k,l) \in A,\, \sigma \in S. \tag{4.18}$$

4.2.3 Variable Routing with Smart Protection

The *smart* protection variant allows network operators to put to sleep the link cards which carry exclusively backup paths. Thanks to the possibility of being awaken in a few milliseconds, these line cards will be activated only in case of unlikely failure occurrence. To introduce this behavior in our PAVRP model, it is sufficient to modify Constraints (4.9) (*dedicated* protection) and (4.18) (*shared* protection) as follows:

$$\sum_{d\in D} r^{d\sigma}\left(x_{ij}^{d\sigma} + \xi_{ij}^{d\sigma}\right) \leq \mu_{ij}^{bck} c_{ij} n_{ij} y_j^{\sigma}, \qquad \forall (i,j) \in A,\, \sigma \in S \tag{4.19}$$

$$\sum_{d\in D} r^{d\sigma}\left(x_{ij}^{d\sigma} + g_{klij}^{d\sigma}\right) \leq \mu_{ij}^{bck} c_{ij} n_{ij} y_j^{\sigma}, \quad \forall (i,j),(k,l) \in A,\, \sigma \in S. \tag{4.20}$$

Both Constraints (4.9) (*dedicated* protection) and (4.18) (*shared* protection) ensure that the sum of primary and backup traffic does not exceed the total capacity ($\mu_{ij}^{bck} c_{ij} n_{ij}$) available on each link when all the installed line cards are on. Note that during normal operations, chassis are kept activated also if used by backup paths only.

4.3 A MILP Formulation for Robust SEANM-FB

Traffic demands are uncertain if traffic parameters $r^{d\sigma}$ are subject to fluctuations around their predicted values. To cope with such variations, we introduce in our PAVRP model the cardinality-constrained framework proposed in (Bertsimas *et al.*, 2011). We assume that each uncertain parameter $r^{d\sigma}$ takes values within the symmetric interval $\left[\overline{r}^{d\sigma} - \hat{r}^{d\sigma}, \overline{r}^{d\sigma} + \hat{r}^{d\sigma}\right]$, where $\overline{r}^{d\sigma}$ is the average value of traffic of traffic demand $d \in D$ during time period $\sigma \in S$, and $\hat{r}^{d\sigma}$ is the maximum variation predicted for demand $d \in D$ during time period $\sigma \in S$.

The robust approach is mainly based on the idea that, although no information on the probability distribution of each parameter within its uncertainty interval is available, it is very unlikely for all traffic demands to assume, at the same time, the most pessimistic value. According to this principle, during time interval $\sigma \in S$ a link $(i,j) \in A$ is Γ_{ij}^{σ}-robust if and only if the active capacity supports the average traffic values $\overline{r}^{d\sigma}$, plus the worst case traffic variation produced by a total traffic deviation minor or equal than Γ_{ij}^{σ}. Note that a traffic

demand $d \in D$ assuming its most pessimistic value $\overline{r}^{d\sigma} + \hat{r}^{d\sigma}$ is meant to produce a deviation equal to 1, so that the total traffic deviation is computed as:

$$\sum_{d \in D} \frac{r^{d\sigma} - \overline{r}^{d\sigma}}{\hat{r}^{d\sigma} - \overline{r}^{d\sigma}}.$$

From this definition, Γ_{ij}^{σ} corresponds to the number of traffic demands flowing through link $(i,j) \in A$ during the time interval $\sigma \in S$ to be considered uncertain. Note that Γ has not to be necessarily integer. Each parameter $\Gamma_{ij}^{\sigma} \in [0, |D|]$ can be adjusted at will by the network operator to obtain solutions that are more or less robust. Since higher robustness means more active capacity in the network, a natural trade-off exists between energy-efficiency and robustness to traffic variations.

To make the original PAVRP model robust, we have to modify only link capacity Constraints (3.16) (which are equal to Constraints (4.8) used in the protected variants).[1] For each link capacity Constraints (3.16), let U_{ij}^{σ} be a subset of cardinality Γ_{ij}^{σ} of the demand set D. U_{ij}^{σ} contains all the traffic demands which are considered uncertain. The robust counterpart of constraints (3.16) is:

$$\sum_{d \in D} \overline{r}^{d\sigma} x_{ij}^{d\sigma} + \Theta_{ij}^{\sigma} \leq \mu_{ij} c_{ij} w_{ij}^{\sigma}, \quad \forall (i,j) \in A,\, \sigma \in S, \tag{4.21}$$

where Θ_{ij}^{σ} represents the worst case traffic variation occurred on link (i,j) during period σ when not more than Γ_{ij}^{σ} demands are uncertain. If Γ_{ij}^{σ} is integer, Θ_{ij}^{σ} can be defined as:

$$\Theta_{ij}^{\sigma} = \max_{\{U_{ij}^{\sigma} \subseteq D,\, |U_{ij}^{\sigma}| \leq \Gamma_{ij}^{\sigma}\}} \left\{ \sum_{d \in U_{ij}^{\sigma}} \hat{r}^{d\sigma} x_{ij}^{d\sigma} \right\}. \tag{4.22}$$

An alternative way to determine Θ_{ij}^{σ} is consider the dualization of (4.22). Let us see how: being $\overline{x}_{ij}^{d\sigma}$ the routing binary variables representing a certain solution, let us compute Θ_{ij}^{σ} by solving the following linear programming problem:

$$\Theta_{ij}^{\sigma} = \max_{v} \{ \sum_{d \in D} \hat{r}^{d\sigma} \overline{x}_{ij}^{d\sigma} v_{ij}^{d\sigma} \} \tag{4.23}$$

s.t.

1. Although uncertain parameters appear also in chassis capacity Constraints (3.15), the latter are not considered uncertain because their role does not consists in limiting chassis utilization, but int forcing chassis variables to assume the right status.

$$\sum_{d \in D} v_{ij}^{d\sigma} \leq \Gamma_{ij}^{\sigma}, \tag{4.24}$$

$$0 \leq v_{ij}^{d\sigma} \leq 1, \quad \forall d \in D, \tag{4.25}$$

where $v_{ij}^{d\sigma}$ are real variables in $[0,1]$ which quantify the deviation of demand $d \in D$ in time interval $\sigma \in S$. Due to the structure of the problem, it is not necessary to define $v_{ij}^{d\sigma}$ as binary.

Being $\epsilon_{ij}^{\sigma'}$ and $\epsilon_{ij}^{d\sigma''}$ the dual variables associated to, respectively, Constraints (4.24) and (4.25), we can express the dual formulation of (4.23–4.25) as:

$$\min \sum_{d \in D} \epsilon_{ij}^{d\sigma''} + \Gamma_{ij}^{\sigma} \epsilon_{ij}^{\sigma'} \tag{4.26}$$

s.t.

$$\epsilon_{ij}^{\sigma'} + \epsilon_{ij}^{d\sigma''} \geq \hat{r}^{d\sigma} \overline{x}_{ij}^{d\sigma}, \quad \forall d \in D \tag{4.27}$$

$$\epsilon_{ij}^{\sigma'} \geq 0,\ \epsilon_{ij}^{d\sigma''} \geq 0, \quad \forall d \in D \tag{4.28}$$

According to duality theory (strong duality theorem), the optimal values of primal objective function (4.23) and dual objective function (4.26) coincide. The dual problem (4.26–4.28) can be thus included in Constraints (4.21) to replace the original expression used for Θ_{ij}^{σ}:

$$\sum_{d \in D} \overline{r}^{d\sigma} x_{ij}^{d\sigma} + \sum_{d \in D} \epsilon_{ij}^{d\sigma''} + \Gamma_{ij}^{\sigma} \epsilon_{ij}^{\sigma'} \leq \mu_{ij} c_{ij} w_{ij}^{\sigma}, \qquad \forall (i,j) \in A,\ \sigma \in S \tag{4.29}$$

$$\epsilon_{ij}^{\sigma'} + \epsilon_{ij}^{d\sigma''} \geq \hat{r}^{d\sigma} x_{ij}^{d\sigma}, \quad \forall (i,j) \in A,\ d \in D,\ \sigma \in S \tag{4.30}$$

$$\epsilon_{ij}^{\sigma'} \geq 0,\ \epsilon_{ij}^{d\sigma''} \geq 0 \qquad \forall d \in D,\ \sigma \in S. \tag{4.31}$$

It is worth pointing out that the same strategy can be applied to the capacity constraints of the protected problems which consider both primary and backup traffic.

4.4 Heuristic Methods

To deal with medium sized instances not tractable with the exact MILP formulations, we adapted the single time period algorithm called EA-STH (see Section 3.3.2) to both protected and robust problems. The procedure remains the same, except for the overall formulation used to solve each single time period, which is derived, from time to time, by the corresponding protected or robust (also protected plus robust) multi-period formulation.

Furthermore, since in this case we do not use EA-STH as an online method, but simply as heuristic for the reference problem, instead of returning the worst solution computed by varying the starting period, we report the best one.

To further increase EA-STH scalability, we propose a novel version called energy-aware single time-period heuristic with restricted paths (EA-STH-RP), where the overall complexity is reduced by considering a limited set of pre-computed paths P^d for each single demand $d \in D$. Let $\chi_p^{d'}$ ($\chi_p^{d''}$) be the binary variables equal to 1 if path $p \in P^d$ is chosen as primary (backup) path of demand $d \in D$. The time period index σ is neglected since in EA-STH-RP we deal with single period problems.

Primary path and backup path Constraints (4.2–4.3) are replaced respectively, by

$$\sum_{p \in P^d} \chi_p^{d'} = 1, \quad \forall d \in D \tag{4.32}$$

$$\sum_{p \in P^d} \chi_p^{d''} = 1, \quad \forall d \in D, \tag{4.33}$$

stating that a single primary path and a single backup path must be chosen for each demand $d \in D$. Then all the other constraints are adjusted by considering that:

$$x_{ij}^d = \sum_{p \in P^d : (i,j) \subset p} \chi_p^{d'}, \quad \forall (i,j) \in A,\, d \in D \tag{4.34}$$

$$\xi_{ij}^d = \sum_{p \in P^d : (i,j) \subset p} \chi_p^{d''}, \quad \forall (i,j) \in A,\, d \in D. \tag{4.35}$$

For what concerns the generation of the path set P^d of each demand $d \in D$, we follow the procedure below:

1. For each demand $d \in D$ we compute the maximum flow m^d between nodes o^d and t^d when all links have unitary capacity.
2. We repeat Ω times the following operations:
   (a) We assign a random cost to each link.
   (b) for each demand $d \in D$ we compute, in a sequential manner, m_d shortest paths. To favor the generation of link-disjoint paths, each time a link is chosen by a path, we significantly increase its cost. Note that disjoint paths are required to favor load balancing and implement path protection.
   (c) We run the Kruskal algorithm to compute a minimum cost spanning tree.

(d) For each demand $d \in D$ we extract a single routing path from the spanning tree just computed. The paths extracted from the spanning tree are meant to favor energy efficiency, since they reduce the number of active elements.

(e) We store the Ωm_d (Point b) $+\Omega$ (Point d) paths computed for each demand $d \in D$.

4.4.1 Warm Starting

To reduce the computing time for the *shared* protection model, we warm start CPLEX with a feasible solution obtained by solving the model for *dedicated* protection within a limited time-limit. The warm-start is implemented in CPLEX using the option `send_statuses 2`. In both EA-STH and EA-STH-RP, warm starting is also exploited by considering the solution computed for the previous time period as the starting solution of the current one. The warm-start procedure is illustrated in Figures 4.4.1 and 4.4.1. Note that each feasible solution for the *dedicated* protection problem is naturally feasible for *shared* protection.

4.5 Computational Results

All the experiments are carried out on machines equipped with Intel i7 processors with 4 core and multi-thread 8x, and 8Gb of RAM. Mathematical formulations have been defined with AMPL and solved with CPLEX-12.5.

4.5.1 Test Instances

We consider four realistic networks topologies provided by the popular SND Library, i.e., SNDLib (Orlowski *et al.*, 2010): `polska`, `nobel-germany`, `nobel-eu` and `germany50`. The last two, which are the largest ones, are the same used in Chapter 3. Test instances are summarized in Table 4.1. The table notation is the same used in Chapter 3. Also in this case we experiment we three different equipment configurations, namely A, B and C (see Table 3.2), and three different random traffic realizations, i.e., a, b and c. The fourth traffic realization denoted by *aver* is deterministically computed according to the traffic profile of Figure 3.5, and represents the average traffic scenario used in input to the robust approaches. Network nodes are equally and randomly divided between core and edge routers.

Nominal traffic values are generated according to the same approach used in Chapter 3. The procedure is only slightly adjusted to account for backup traffic during the scaling phase. This time, the parameter used to scale SNDLib traffic matrices is taken as the largest one which allows to reserve both primary and backup resources (single path routing) while respecting both primary and overall utilization thresholds, i.e., μ and μ^{bck} (the same on each link). Where not otherwise specified we dimension the peak values to respect $\mu = 0.5$

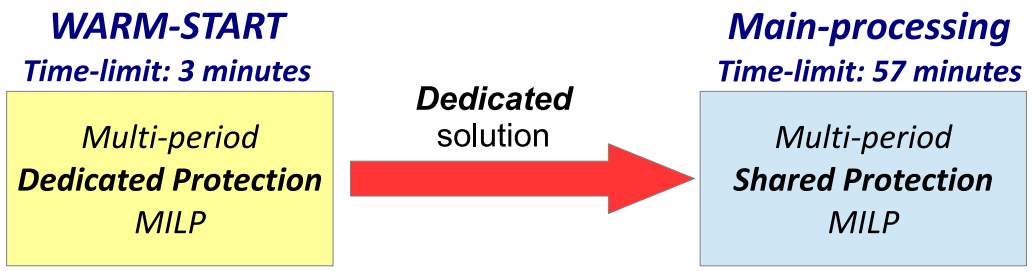


Figure 4.2 Warm-start of multi-period MILP with shared protection.

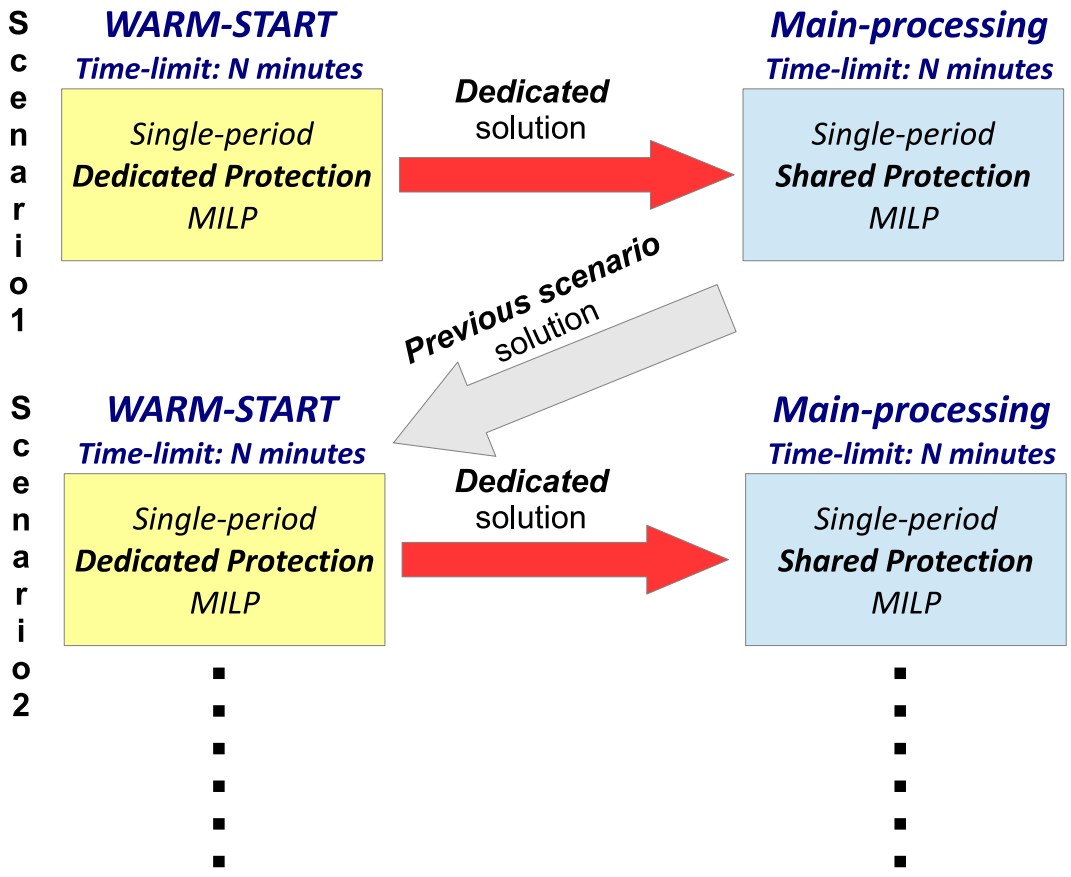


Figure 4.3 Warm-start of EA-STH with shared protection.

and $\mu^{bck} = 0.85$ with dedicated protection. The same configuration, i.e., $\mu = 0.5$ and $\mu^{bck} = 0.85$, is used in the optimization approach.

The day splitting is maintained the same (1) 8a.m.-11a.m., 2) 11a.m.-1p.m., 3) 1p.m.-2.30p.m., 4) 2.30p.m.-6.30p.m., 5) 6.30p.m.-10.30p.m., 6) 10.30p.m.-8a.m.). The robust approaches are evaluated by experimenting with uncertainty sets of different sizes, i.e., $\hat{r}^{d\sigma} =$

Table 4.1 Test instances.

polska					
ID	$\|V\|$-$\|V^c\|$	$\|A\|$	$\|D\|$	*dev*	*rp*
1	12-6	36	15	A	a
2	12-6	36	15	A	b
3	12-6	36	15	A	c
4	12-6	36	15	A	aver
5	12-6	36	15	B	a
6	12-6	36	15	B	b
7	12-6	36	15	B	c
8	12-6	36	15	B	aver
9	12-6	36	15	C	a
10	12-6	36	15	C	b
11	12-6	36	15	C	c
12	12-6	36	15	C	aver

nobel-eu					
ID	$\|V\|$-$\|V^c\|$	$\|A\|$	$\|D\|$	*dev*	*rp*
25	28-14	82	90	A	a
26	28-14	82	90	A	b
27	28-14	82	90	A	c
28	28-14	82	90	A	aver
29	28-14	82	90	B	a
30	28-14	82	90	B	b
31	28-14	82	90	B	c
32	28-14	82	90	B	aver
33	28-14	82	90	C	a
34	28-14	82	90	C	b
35	28-14	82	90	C	c
36	28-14	82	90	C	aver

nobel-germany					
ID	$\|V\|$-$\|V^c\|$	$\|A\|$	$\|D\|$	*dev*	*rp*
13	17-9	42	21	A	a
14	17-9	42	21	A	b
15	17-9	42	21	A	c
16	17-9	42	21	A	aver
17	17-9	42	21	B	a
18	17-9	42	21	B	b
19	17-9	42	21	B	c
20	17-9	42	21	B	aver
21	17-9	42	21	C	a
22	17-9	42	21	C	b
23	17-9	42	21	C	c
24	17-9	42	21	C	aver

germany50					
ID	$\|V\|$-$\|V^c\|$	$\|A\|$	$\|D\|$	*dev*	*rp*
37	50-25	176	182	A	a
38	50-25	176	182	A	b
39	50-25	176	182	A	c
40	50-25	176	182	A	aver
41	50-25	176	182	B	a
42	50-25	176	182	B	b
43	50-25	176	182	B	c
44	50-25	176	182	B	aver
45	50-25	176	182	C	a
46	50-25	176	182	C	b
47	50-25	176	182	C	c
48	50-25	176	182	C	aver

$0.05\rho^d$, $0.10\rho^d$, $0.15\rho^d$, $0.20\rho^d$. Furthermore, robustness parameters Γ_{ij}^{σ} are varied from 0 (no robustness) to 5 (high level of robustness) to evaluate the trade-off between energy savings and robustness. Finally, we consider $\delta = 0.25$ (chassis switching-on normalized consumption), $\eta_{on} = 1$ (switching-on limit), and $n_{ij} = 2$ (number of cards on link $(i,j) \in A$) for all links.

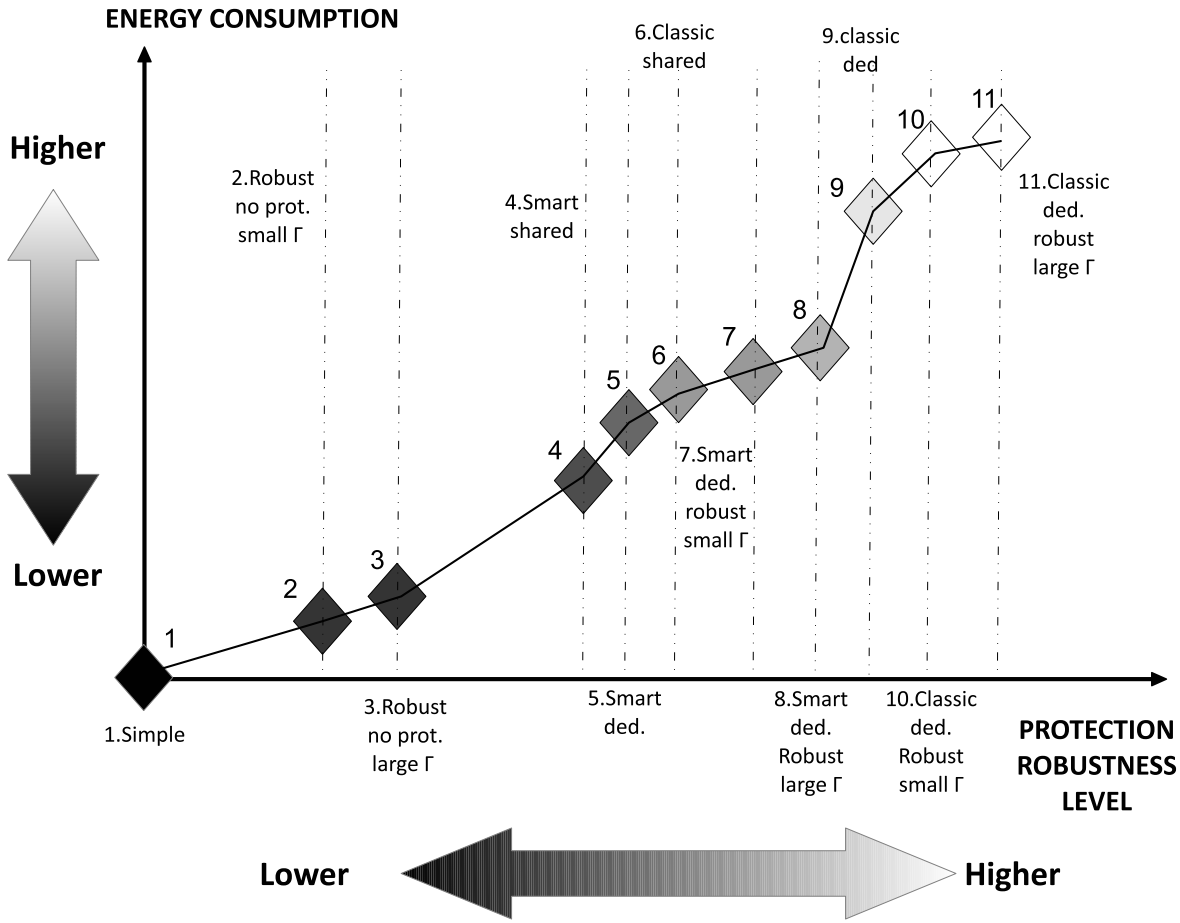


Figure 4.4 Trade-off between power consumption and survivability.

4.5.2 Savings vs. Protection/Robustness

One of the aim of our work is to evaluate the energy-cost paid to guarantee different levels of network survivability. To this purpose, we consider several protection/robustness strategies, each one obtained by including different survivability features in our general multi-period framework for SEANM-FB. We experiment with 8 different approaches, to which we refer as (i) *simple*, (ii) *robust*, (iii) *dedicated-classic*, (iv) *shared-classic*, (v) *dedicated-smart*, (vi) *shared-smart*, (vii) *robust plus dedicated-classic* and (viii) *robust plus dedicated-smart*. The energy-efficiency/survivability trade-off that we expect to observe in computational results is shown in Figure 4.4. This graph is purely qualitative and the gaps on both horizontal and vertical axis are not chosen in a deterministic manner. In fact, there is no standard way to jointly measure resilience to link failures and robustness to traffic variations; however we can consider both of them as two different sides of the same coin, i.e., the network capability to react to anomalous phenomena (network survivability). For instance, on the horizontal

axis, the survivability contribution assigned to link failure protection is much larger than that given to robustness to traffic variations (around the double): we know that the amount of additional bandwidth required to absorb a traffic variation of 10%-20% is considerably smaller than that typically reserved to establish a backup path for each demand. Thus, more active capacity naturally results in higher network survivability.

Moving from left to right along the graph, we first find, in the lower left corner, the solutions obtained for the *simple* case. Both energy consumption and survivability levels are expected to gradually grow once we consider robustness without protection (*robust* case) and we increase both robustness parameters $\Gamma_{ij}^{d\sigma}$ and uncertainty interval size ($\hat{r_d^{\sigma}}$). The larger the $\Gamma_{ij}^{d\sigma}$ and $\hat{r_d^{\sigma}}$, the larger the amount of the resources to be kept activated. Moving further to the right, we meet, in the following order, *shared-smart* and *dedicated-smart* approaches. Although both types of protection (*shared* and *dedicated*) theoretically ensure the same protection level, we consider *dedicated* protection as the most conservative because it typically leaves more spare capacity on links and routers. This additional capacity can be thus exploited by the network to passively react to other unexpected events not limited to pure link failures.

Continuing toward the right, we finally find *classic* protected strategies, and at last, the two *robust plus dedicated* approaches (first *smart* and then *classic*). We consider *classic* schemes more conservative than *smart* because of the higher amount of resources which is kept active.

Note that in term of consumptions, we believe that putting to sleep backup links brings more benefits than passing from *dedicated* to *shared* protection. We do not report *robust plus shared* strategies because too complex to be efficiently handled by our methodologies even with the smallest instances.

To verify whether the behaviour we expect is confirmed in realistic network instances, we solve the exact MILP formulation of each problem (time limit of one hour) by considering twelve instances from `polska` network. We also evaluate scalability, computing times, and solution optimality.

Robust Strategy

We start our analysis from the results obtained for the *robust* case (see Table 4.2). We denote with $\hat{r}$ the size of the uncertainty interval of each traffic demand (the same for all demands). For each instance we report the normalized power consumption with respect to the fully active network ($\%E_c$), plus other two quantities, i.e., $\%_{inf}$ and Max_{dv}, which requires additional explanations. While the first quantity is uniquely computed for each

Table 4.2 Robustness analysis: *robust* MILP with no protection and 1h time-limit on `polska` instances

	Exact model - Robust approach with no protection														
	$\Gamma = 0$			$\Gamma = 1$			$\Gamma = 2$			$\Gamma = 3$			$\Gamma = 4$		
ID-$\hat{r}$	$\%E_c$	$\%_{inf}$	Max_{dv}	$\%E_c$	$\%_{inf}$	Max_{dv}	$\%E_c$	$\%_{inf}$	Max_{dv}	$\%E_c$	$\%_{inf}$	Max_{dv}	$\%E_c$	$\%_{inf}$	Max_{dv}
4-0.05	60,6%	42,3%	6,6%	60,7%	9,9%	0,8%	60,9%	0,0%	0,0%	60,9%	0,0%	0,0%	60,9%	0,0%	0,0%
4-0.10	60,6%	81,7%	16,3%	60,9%	10,2%	2,6%	60,9%	0,1%	1,7%	60,9%	0,1%	1,7%	60,9%	0,0%	0,0%
4-0.15	60,6%	92,3%	24,6%	60,9%	15,3%	9,2%	61,4%	0,3%	4,8%	61,4%	0,3%	4,8%	61,4%	0,0%	0,0%
4-0.20	60,6%	95,6%	32,3%	60,9%	32,9%	19,2%	62,4%	0,1%	1,3%	62,4%	0,1%	1,3%	63,4%	0,0%	0,0%
8-0.05	50,6%	40,8%	6,2%	50,8%	10,5%	0,8%	51,0%	0,0%	0,0%	51,0%	0,0%	0,0%	51,0%	0,0%	0,0%
8-0.10	50,6%	78,4%	14,8%	51,0%	1,3%	0,6%	51,0%	0,0%	0,0%	51,0%	0,0%	0,0%	51,0%	0,0%	0,0%
8-0.15	50,6%	88,6%	24,2%	51,0%	9,5%	8,9%	51,5%	0,5%	4,9%	51,5%	0,5%	4,9%	51,6%	0,0%	0,0%
8-0.20	50,6%	93,0%	33,6%	51,0%	27,6%	19,4%	52,6%	0,1%	1,1%	52,6%	0,1%	1,1%	53,2%	0,0%	0,0%
12-0.05	60,0%	41,4%	7,1%	60,1%	4,7%	0,7%	60,3%	0,0%	0,0%	60,3%	0,0%	0,0%	60,3%	0,0%	0,0%
12-0.10	60,0%	78,7%	15,4%	60,3%	11,2%	2,3%	60,3%	0,0%	0,0%	60,3%	0,0%	0,00%	60,5%	0,0%	0,0%
12-0.15	60,0%	88,5%	24,4%	60,3%	6,9%	9,2%	60,7%	0,4%	4,9%	60,7%	0,4%	4,9%	60,8%	0,0%	0,0%
12-0.20	60,0%	94,1%	35,6%	60,3%	40,1%	18,5%	61,6%	0,0%	0,2%	61,6%	0,0%	0,2%	62,5%	0,0%	0,0%

Table 4.3 Robustness analysis: *robust plus dedicated classic* MILP with 1h time limit on `polska` instances

	Exact model - Robust approach with dedicated protection											
	$\Gamma = 0$			$\Gamma = 1$			$\Gamma = 3$			$\Gamma = 5$		
ID-$\hat{r}$	$\%E_c$	$\%_{inf}$	$\Delta^{classic}_{smart}$	$\%E_c$	$\%_{inf}$	$\Delta^{classic}_{smart}$	$\%E_c$	$\%_{inf}$	$\Delta^{classic}_{smart}$	$\%E_c$	$\%_{inf}$	$\Delta^{classic}_{smart}$
4-0.05	70,6%	96,7%	-3,1%	71,4%	17,9%	-3,4%	71,5%	0,5%	-3,4%	71,6%	0,0%	-3,5%
4-0.10	70,6%	98,7%	-3,1%	71,4%	61,7%	-3,2%	71,6%	1,2%	-3,3%	71,8%	0,0%	-3,4%
4-0.15	70,6%	99,8%	-3,1%	71,6%	63,2%	-3,4%	71,9%	0,7%	-3,3%	72,1%	0,0%	-3,3%
4-0.20	70,6%	99,7%	-3,1%	71,6%	64,1%	-3,2%	72,1%	3,5%	-3,1%	72,6%	0,0%	-3,4%
8-0.05	60,8%	95,7%	-5,5%	61,8%	31,9%	-6,2%	61,8%	0,4%	-5,7%	62,0%	0,0%	-6,0%
8-0.10	60,8%	99,1%	-5,5%	61,8%	38,6%	-5,7%	62,3%	1,9%	-5,4%	62,3%	0,0%	-5,9%
8-0.15	60,8%	99,0%	-5,5%	61,8%	63,8%	-5,8%	62,6%	1,8%	-5,8%	63,2%	0,0%	-6,2%
8-0.20	60,8%	99,8%	-5,5%	62,0%	55,4%	-5,5%	62,9%	2,8%	-5,3%	63,4%	0,0%	-5,5%
12-0.05	70,0%	91,4%	-3,2%	70,9%	21,4%	-3,7%	70,9%	0,9%	-3,4%	71,0%	0,0%	-3,7%
12-0.10	70,0%	97,4%	-3,2%	70,9%	32,3%	-3,5%	71,0%	1,0%	-3,4%	71,3%	0,0%	-3,7%
12-0.15	70,0%	99,0%	-3,2%	71,0%	56,2%	-3,5%	71,4%	1,5%	-3,4%	71,5%	0,0%	-3,5%
12-0.20	70,0%	99,4%	-3,2%	71,0%	71,3%	-3,4%	71,6%	2,1%	-3,3%	71,9%	0,0%	-3,5%

instance by solving the *robust* MILP, the last two are determined in post-processing and aim at quantifying the robustness degree of the solution returned by the MILP.

In the post-processing phase, we test the computed network configuration (that with a power consumption equal to $\%E_c$) over a set of 10000 traffic scenarios which are randomly generated. Each scenario contains a value of traffic for each demand $d \in D$ within each time period $\sigma \in S$, i.e., $r^{d\sigma}$. Each $r^{d\sigma}$ parameter is generated according to the uniform distribution $\mathcal{N}(\overline{r}^{\sigma}_{d} + \hat{r^{\sigma}_{d}}, \overline{r}^{\sigma}_{d} - \hat{r^{\sigma}_{d}})$. Once all scenarios have been generated, we test the computed solutions by applying the optimized routing configuration to each random scenario and later verifying whether maximum utilization constraints are violated or not. Column $\%_{inf}$ reports the percentage of random scenarios wherein at least one link maximum utilization constraint

is violated in one time period, and column Max_{dv} shows the largest violation (positive difference between the observed maximum utilization and the allowed maximum one) observed along the entire set of random scenarios. We consider a solution as completely robust if and only if $\%_{inf} = 0\%$.

Energy Cost of Robustness

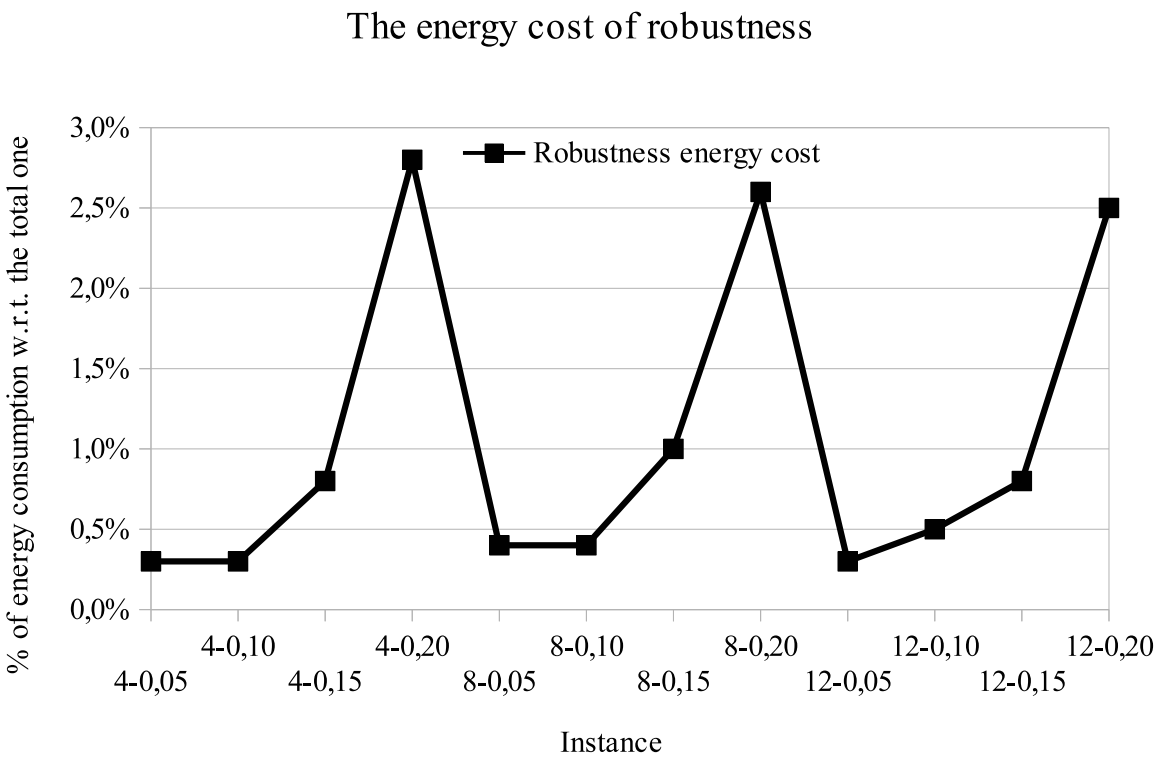


Figure 4.5 Energy cost of robustness as additional percentage of energy consumption w.r.t. that of the fully active network. Difference between $\%E_c$ obtained with Γ equal to 0 and $\%E_c$ obtained with Γ equal to 4.

Results clearly show that complete immunization to traffic variations can be achieved with Γ equal to 4 (four demands considered uncertain on each link within each scenario). Most importantly, as shown in Figure 4.5 we remark that the energy-cost of being robust is almost negligible (on average smaller than 1%), with a consumption increase of 2.8% observed in the worst case (instance 4 with $\bar{r} = 0.2$).

Robustness Trend

It is evident that the nominal solution ($\Gamma = 0$), which is equivalent to that of the *simple* case, is largely unreliable: infeasibility ratio $\%_{inf}$ is 95.6% and maximum deviation Max_{dv} is even 35.6% in the worst case (instance 12 with $\overline{r}_d^{\sigma} = 0.2$). As expected, robustness increases as Γ is incremented, and larger uncertainty intervals force the model to allocate more spare capacity to absorb the traffic variations.

Path Protection: the Energy Cost

Our first purpose is to compare classic protection methods, namely *dedicated classic* and *shared classic* strategies. We report in Table 4.4, the optimized power consumption $\%E_c$ for `polska` with both *simple*, *dedicated classic*, and *shared classic* models. We report in column Gap_{opt} the gap between the power consumption achieved within the time-limt and the best lower bound computed by CPLEX. Then, in column gap_{simple} we show the energy cost of implementing protection computed as $\left(E_c^{prot} - E_c^{simple}\right)/E_c^{simple}$. Quite intuitively, we denote with $\%E_c^{prot}$ and $\%E_c^{simple}$ the normalized power consumption for protected and unprotected cases, respectively.

Energy Cost of Protection

The energy-cost of protection is not negligible. From a power consumption range of $50.1\% - 60.6\%$ for the *simple* problem, we pass to $61.4\% - 71.4\%$ in the *dedicated classic* one. The absolute consumption increase is around 10% on average, while the relative one is close to 20%. As expected, if we allocate backup resources in a more efficient way adopting *shared* protection we can reduce the protection costs by about 5% w.r.t. the *dedicated classic* case.

Scalability of Shared Protection

While *dedicated classic* solutions are very close to optimality (Gap_{opt} usually lower than 1% and never above 3.5%), *shared classic* ones show a gap from optimality even larger than 15% in a few instances (Instances 6-7). This phenomenon, which explains why in some instances *shared* protection does not significantly outperform *dedicated* protection (Instances 6-7-11), suggests us that *shared classic* problems (obviously also the *shared smart* ones) do not scale well as soon as network dimensions begin to increase.

EA-STH Accuracy with Protection

This problem can be partially overcome by running EA-STH. Comparative results between the exact models and EA-STH are shown in Tables 4.5 by reporting the gap $Heur_{gap}$ of

EA-STH solutions (in terms of power consumption) from those returned by the exact models. $Heur_{gap}$ is computed as $\%E_c^{heur} - \%E_c^{model}$. Column TL represents the time limit to solve each single time period when we run EA-STH. Results clearly state that EA-STH executed with a single time period time-limit of 6 minutes (warm start of 3 minutes plus main processing of 3 minutes) reduces overall power consumption up to 5% (Instances 3-6-7-11) for the *shared classic* problem. Therefore, note that this time in instances 6-7, i.e., those with an optimality gap of 15% when solved with the exact model, the gap between *dedicated* and *shared* solutions is not negligible anymore. The validity of EA-STH is further confirmed by the small gaps (they vary from 0.6% to -0.2%) obtained with respect to exact model solutions for the *dedicated classic* problem, although the latter are known to be very close the optimal ones (optimality gap smaller than 1%).

Classic vs. Smart Protection

Being EA-STH solutions very close to those obtained by the exact formulations, in the remainder of the section we report only results achieved by EA-STH. In Table 4.6 we compare EA-STH solutions obtained with both *classic* and *smart* protection approaches to quantify how much putting to sleep backup links may concur to reduce the overall consumption. We report in column $\Delta_{smart}^{classic}$ the difference between the normalized consumption achieved by *smart* and *classic* solutions.

The impact of the *smart* approach is quite consistent: w.r.t. *classic* approaches, the *smart* ones achieve an average power consumption reduction of 5% and 3.9% for *dedicated*

Table 4.4 Comparison between *simple* and *protected* solutions computed by the exact MILP within a one hour time limit on `polska` instances.

	Exact model							
	simple case		***dedicated prot classic***			***shared prot classic***		
ID	$\%E_c$	Gap_{opt}	$\%E_c$	Gap_{opt}	gap_{simple}	$\%E_c$	Gap_{opt}	gap_{simple}
1	60,6%	1,3%	71,4%	1,4%	17,8%	66,9%	3,6%	10,3%
2	60,5%	0,9%	71,3%	0,9%	17,8%	66,3%	4,4%	9,6%
3	60,3%	0,6%	71,4%	0,7%	18,4%	70,4%	8,9%	16,7%
5	50,7%	2,4%	62,2%	2,6%	22,7%	59,3%	10,1%	17,0%
6	50,1%	0,8%	61,4%	3,3%	22,7%	60,3%	15,5%	20,5%
7	50,3%	0,4%	61,7%	2,7%	22,6%	61,7%	15,8%	22,6%
9	60,0%	1,4%	70,9%	0,9%	18,1%	66,2%	3,2%	10,3%
10	59,8%	0,7%	70,7%	0,8%	18,1%	65,7%	3,6%	9,8%
11	59,7%	0,0%	70,8%	0,5%	18,6%	70,9%	11,1%	18,8%

Table 4.5 Energy saving comparison between exact model and EA-STH with different types of protection.

	Exact model vs EA-STH					
	simple case		***dedicated prot classic***		***shared prot classic***	
ID	$Heur_{gap}$	TL	$Heur_{gap}$	TL	$Heur_{gap}$	TL
1	0,00%	60s	0,1%	30s	-0,3%	360s
2	0,25%	60s	0,1%	30s	-0,3%	360s
3	0,16%	60s	0,0%	30s	-4,0%	360s
5	0,41%	60s	0,6%	30s	-2,1%	360s
6	0,00%	60s	-0,1%	30s	-3,6%	360s
7	0,28%	60s	-0,2%	30s	-4,5%	360s
9	0,28%	60s	0,1%	30s	-0,3%	360s
10	0,28%	60s	0,2%	30s	-0,4%	360s
11	0,17%	60s	0,1%	30s	-4,9%	360s

Table 4.6 Comparison between the energy saving achieved by EA-STH with *classic* and *smart* protection schemes.

	EA-STH - Classic vs Smart			
	dedicated		***shared***	
ID	$\Delta_{smart}^{classic}$	TL	$\Delta_{smart}^{classic}$	TL
1	-3,9%	30s	-1,9%	360s
2	-3,5%	30s	-2,2%	360s
3	-3,2%	30s	-1,9%	360s
5	-7,1%	30s	-3,4%	360s
6	-5,9%	30s	-3,9%	360s
7	-5,5%	30s	-3,6%	360s
9	-4,2%	30s	-2,0%	360s
10	-3,8%	30s	-2,3%	360s
11	-3,5%	30s	-2,1%	360s

and *shared* cases, respectively. Larger improvements are observed with *dedicated* protection than in *shared* protection: being the allocation scheme of the first one more bandwidth hungry, it is natural that a higher amount of resources will be put to sleep once we shift to a *smart* scheme.

Furthermore, we believe that a very important result is that *dedicated smart* protection, which is considerably less computationally expensive than *shared classic* protection, can be more energy efficient too. This means that putting to sleep the backup elements has more impact than refining the way the backup bandwidth is computed.

Protection Strategies: Energy/Congestion Trade-off

Bandwidth Allocation Efficiency

We first aim to point out that, thanks to its sophisticated allocation scheme, *shared* protection allows the network operator to respect the desired maximum utilization thresholds while dealing with traffic levels otherwise not sustainable by *dedicated* protection. To highlight this point, we report in Table 4.7 the values of the parameters used to maximally scale SNDLib traffic matrices while respecting $\mu = 0.5$ and $\mu^{bck} = 0.85$. We distinguish between the two cases wherein *dedicated* or *shared* protection are alternatively considered. The peak traffic matrix supported with *shared* protection is, on average, 10% larger than in the *dedicated* cases. That is why *shared* protection is worth to be considered in spite of the additional computational efforts required to manage it.

Energy Consumption vs. Network Congestion

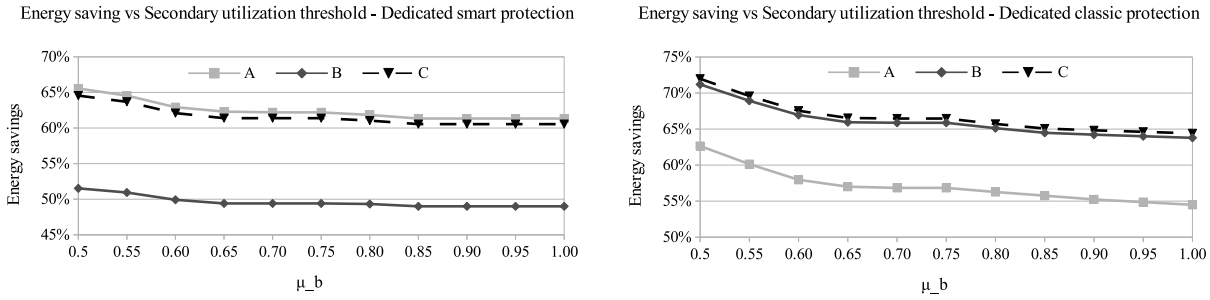


Figure 4.6 Trade-off between energy savings and network congestion with EA-STH. Secondary utilization threshold μ^{bck} is adjusted from 0.5 to 1.

To further investigate the balance between network congestion and energy savings, we show in Figure 4.6 how the overall network consumption is affected when the backup maximum utilization threshold μ^{bck} is varied from 0.5 to 1. In this specific set of tests, we consider

dedicated protection and use a nominal traffic matrix derived by imposing $\mu = 0.5$ and $\mu^{bck} = 0.5$ (instead of $\mu^{bck} = 0.85$). Otherwise, a nominal matrix dimensioned (as done in the rest of the chapter) to respect $\mu^{bck} = 0.85$ would make the problem infeasible for $\mu^{bck} < 0.85$. The plots show an energy gain ranging from 4% to 8% when μ^{bck} is adjusted from 0.5 to 1. It is up to the network operator to achieve the desired balance.

Joint Protection and Robustness

The results obtained combining *robust* and *protected* approaches are reported in Table 4.3. Specifically, we solve both *robust plus dedicated classic* and *robust plus dedicated smart* problems with the corresponding exact formulation. Solutions are completely immunized to traffic variations, i.e., $\%_{inf} = 0\%$, when $\Gamma^{\sigma}_{ij} = 5$. The energy cost of this practice is relatively small, ranging from an average value of 1% to a worst case cost of 2.6% (Instance 8, $\overline{r}^{\sigma}_{d} = 0.2$). Furthermore, as observed with non robust *protected* problems, the *smart* strategy further reduce network consumption from 4% to 6%.

To conclude, we summarize in Figure 4.7 the average energy savings (over the three random realization a, b and c) obtained with each different approach. It is worth pointing out that we observe the trade-off trend previously showed in Figure 4.4.

4.5.3 Larger Networks

In a second group of tests carried out on `nobel-germany`, `nobel-eu` and `germany50` networks, we verify whether trends previously observed with `polska` are confirmed within larger network domains, and evaluate the scalability of the proposed heuristic algorithms, i.e., EA-STH and EA-STH-RP. The time limits imposed to solve each single time period are shown in Table 4.8. Note that a / is used to represent the network instances which, due to the already large number of tests, have not been tested with the corresponding heuristic, e.g., `nobel-germany` instances are not solved with EA-STH-RP.

General Observations on Energy Consumption

The first interesting result emerged by running EA-STH with different protection/robustness levels, is that in both `nobel-germany` (Figure 4.8) and `nobel-eu` (Figure 4.9) we obtain the same power consumption trend observed with `polska` (Figure 4.7). The only substantial difference concerns the energy savings achieved by *dedicated smart* protection, which on this occasion outperform, on average, those of the *shared classic* case. This behaviour is explained by the large optimality gap obtain by CPLEX when solving each single time interval with *shared* protection within the time-limit (both *classic* and *smart*): in some instances warm-

Table 4.7 Comparison between the allocation scheme efficiency of shared and dedicated protection.

Scaling parameters		
ID	*Shared*	*Dedicated*
1-2-3-4	934.0	932.4
5-6-7-8	361.9	361.3
9-10-11-12	2335.0	2354.6
13-14-15-16	18040.0	14279.4
17-18-19-20	6990.5	6375.0
21-22-23-24	45099.9	36824.4

Table 4.8 CPLEX time limits for the single time period to solve `nobel-germany`, `nobel-eu` and `germany` instances with different types of protection.

	simple		*robust*	
Net	TL_{STPH}	$TL_{STPH-RP}$	TL_{STPH}	$TL_{STPH-RP}$
nobel-ger	60s	/	90s	/
nobel-eu	300s	/	300s	/
germany	/	600s	/	600s

	dedicated		*shared*		*robust-dedicated*	
Net	TL_{STPH}	$TL_{STPH-RP}$	TL_{STPH}	$TL_{STPH-RP}$	TL_{STPH}	$TL_{STPH-RP}$
nobel-ger	90s	/	360s	/	120s	/
nobel-eu	300s	300s	/	/	1200s	/
germany	/	600s	/	/	/	1200s

start solutions are not even improved by the solver. Furthermore, due to memory constraints (8GB of RAM), both `nobel-germany` and `germany50` instances cannot be even initialized by the solver when *shared* protection is considered (too many Constraints (4.17)). To overcome this issue, when *shared* protected problems are not manageable by the solver, we keep as solutions those obtained for the *dedicated* ones. The energy consumption difference between the *simple* problem and the most protected one, i.e., the *robust plus dedicated classic*, is on average close to 20% in both `nobel-germany` and `nobel-eu`.

Robust problem

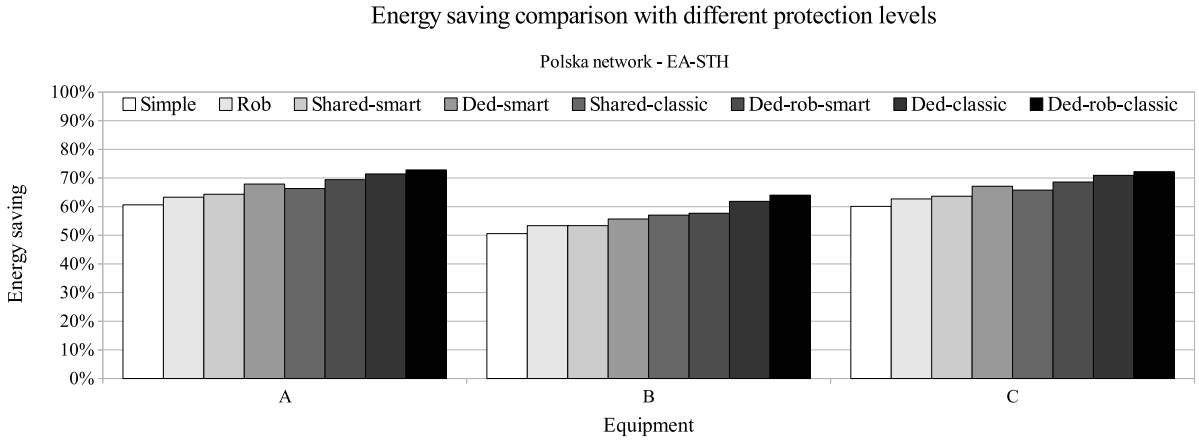


Figure 4.7 Energy savings achieved by EA-STH when implementing the different protection schemes on `polska` instances.

We report in Figure 4.10 the overall results, in terms of power consumption and robustness to traffic variations, obtained with EA-STH applied to the *robust* problem with no protection. The notation for both robustness degree ($\%_{inf}$) and maximum overrun ($\%_{inf}$) is the same used in Table 4.2. As observed in `polska`, results confirm that our robust approach protect the network configuration from traffic variations without significantly increasing the overall power consumption (consumption increase lower than 2% for robust solutions). Note that already with Γ equal to 1 the solutions result to be quite immunized, with $\%_{inf}$ improved from 90% (nominal case with $\Gamma = 0$) to around 15%. We do not report the plots for the equipment configuration C because very similar to those already shown.

Dedicated Protection Plus Robustness

For *robust plus dedicated* problems, due to complexity issues we solve the medium size `nobel-eu` instances by means of EA-STH-RP. The performance of EA-STH-RP are compared with that of the original EA-STH (Figure 4.12) by experimenting with `nobel-eu` and *dedicated classic* protection. For EA-STH-RP we consider $\Omega = 10$. The gap between normal and path restricted solutions is slightly lower than 10% in favour of the firsts (in terms of power consumption). This result confirms that the complexity required to consider all the possible paths is worth to be introduced as long as the problem itself remains tractable. Otherwise, path restriction represents a viable option to make the single time period heuristic more scalable: as shown in Figure 4.13, the normalized power consumption obtained on `germany50` instances by EA-STH-RP (with $\Omega = 5$) varies, on average (over the three equipment configurations A, B, C), between 50% for the *simple* problem and 80% for the *dedicated*

classic plus robust one.

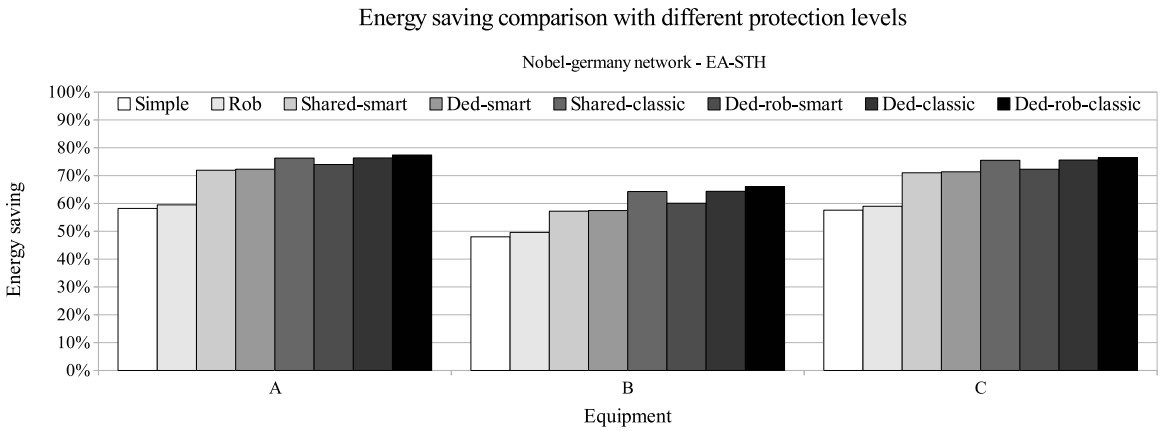


Figure 4.8 Energy savings achieved by EA-STH when implementing the different protection schemes on `nobel-germany` instances.

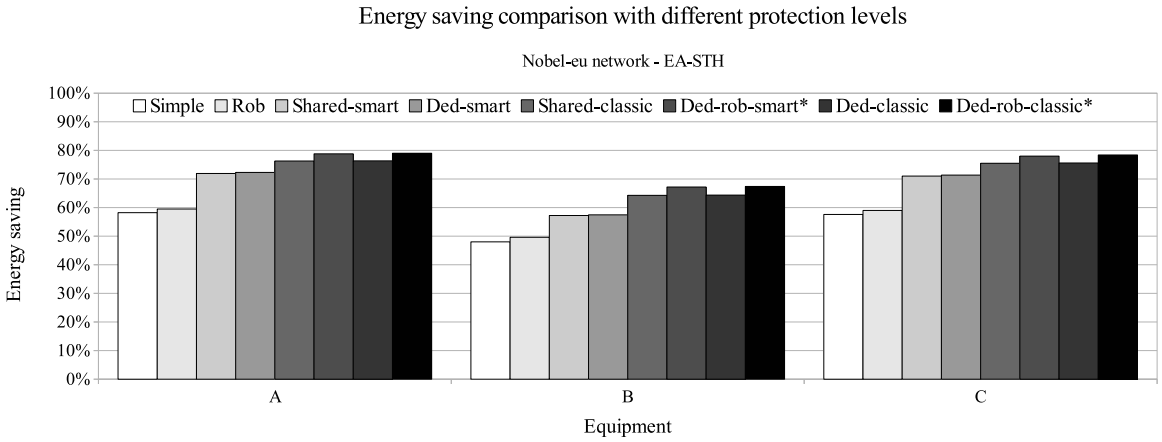


Figure 4.9 Energy savings achieved by EA-STH when implementing the different protection schemes on `nobel-eu` instances. The $*$ in the graph legend is used for the instances solved, due to complexity issues, with STPH-RP using $\Omega = 10$.

4.6 Conclusions

In this chapter, we have jointly handled both energy saving and network resilience issues in IP network management. To guarantee network resilience we have extended the methods presented in Chapter 3 to integrate the management of single link failures and unexpected traffic variations. We presented both exact and heuristic methods involving different survivability features.

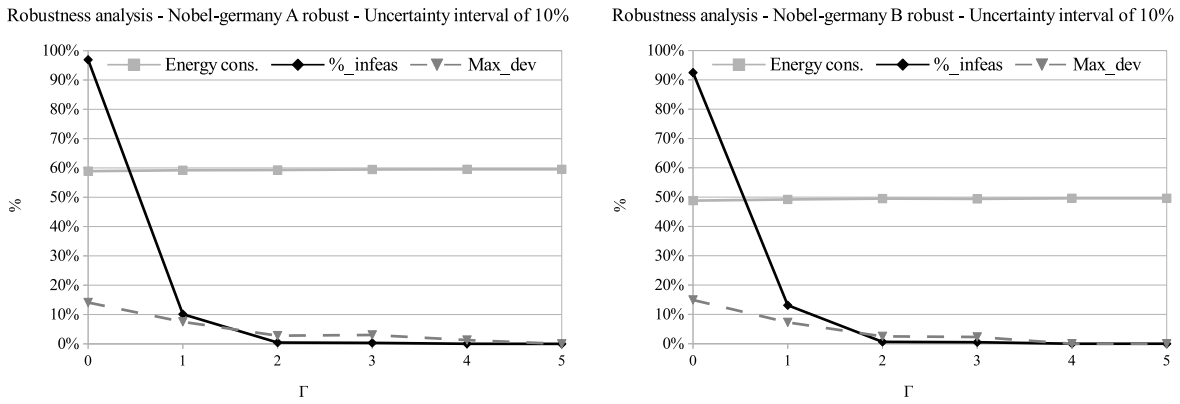


Figure 4.10 Energy savings achieved by EA-STH when implementing the robust scheme on `nobel-germany` instances.

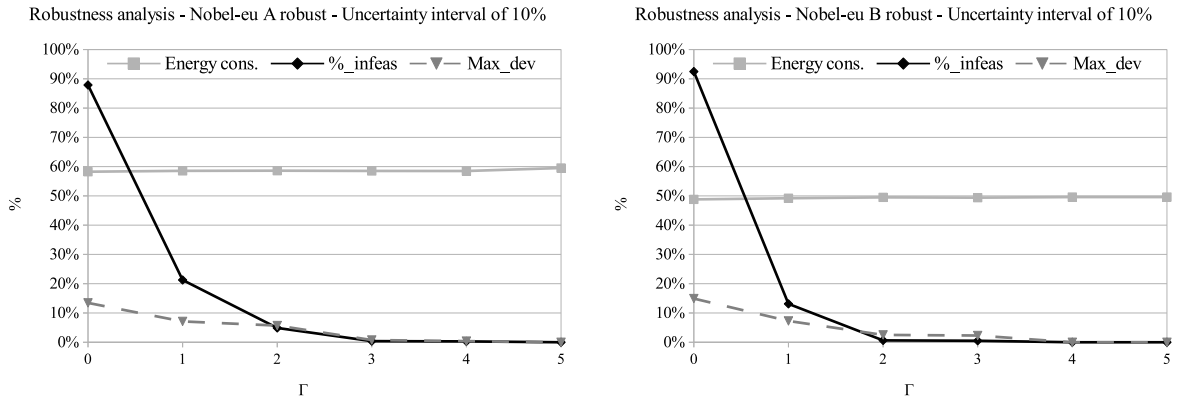


Figure 4.11 Energy savings achieved by EA-STH when implementing the robust scheme on `nobel-eu` instances.

Computational results showed a clear trade-off between the network energy requirements and the provided resilience level. We observed that robustness to traffic variations has a very limited impact in terms of power consumption, with an average consumption increment typically around 1%. Differently, protection to single link failures proved to be more energetically expensive, since it produces a saving reduction between 10% and 20%. Possible strategies to reduce the energy impact of failure protection include: (i) the implementation of *shared* protection rather than *dedicated* protection, (ii) the switching off of line cards carrying backup paths only and (iii) the use of a higher maximum link utilization threshold during link periods. Each of these strategies could further reduce the energy cost of link protection by up to 8%.

It is worth pointing out that when full protection is guaranteed (dedicated protection

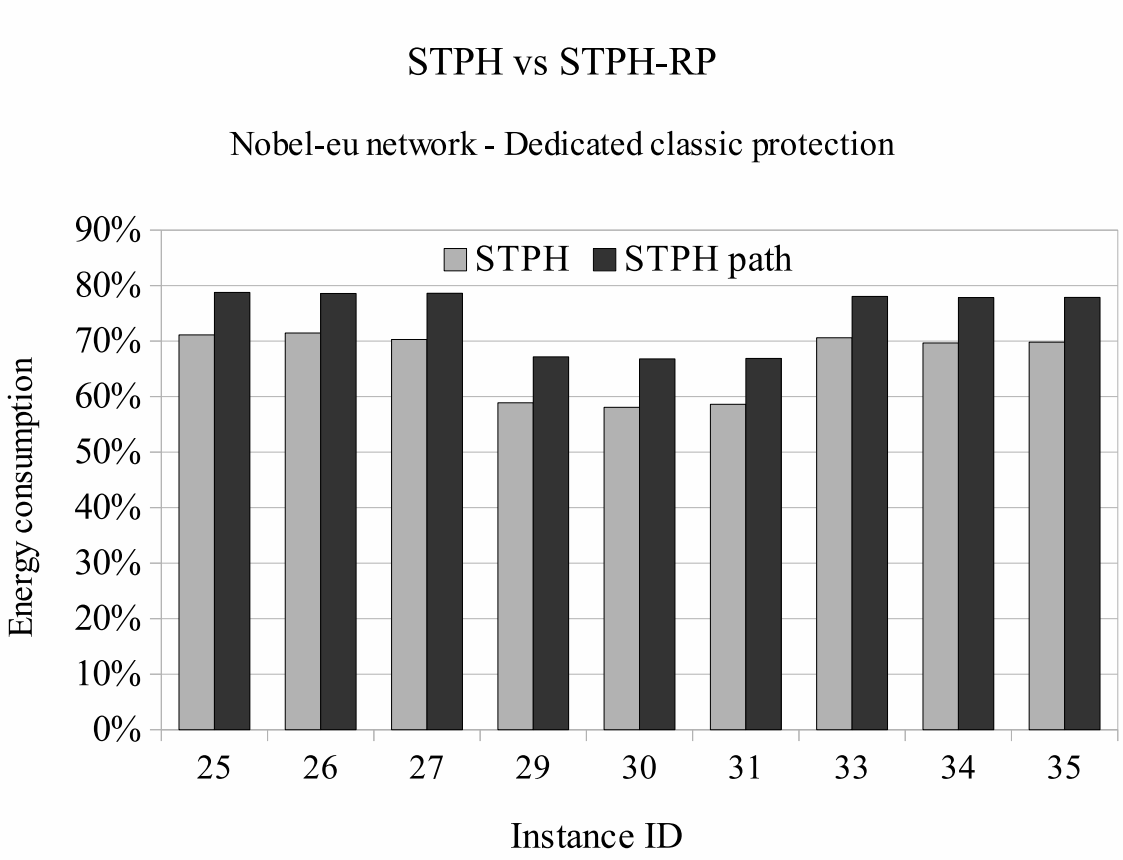


Figure 4.12 Energy saving comparison between EA-STH and EA-STH-RP on `nobel-eu` network with *dedicated classic* protection.

with robustness to traffic variations) our approaches are able to achieve a reduction of 30% of the daily network consumption.

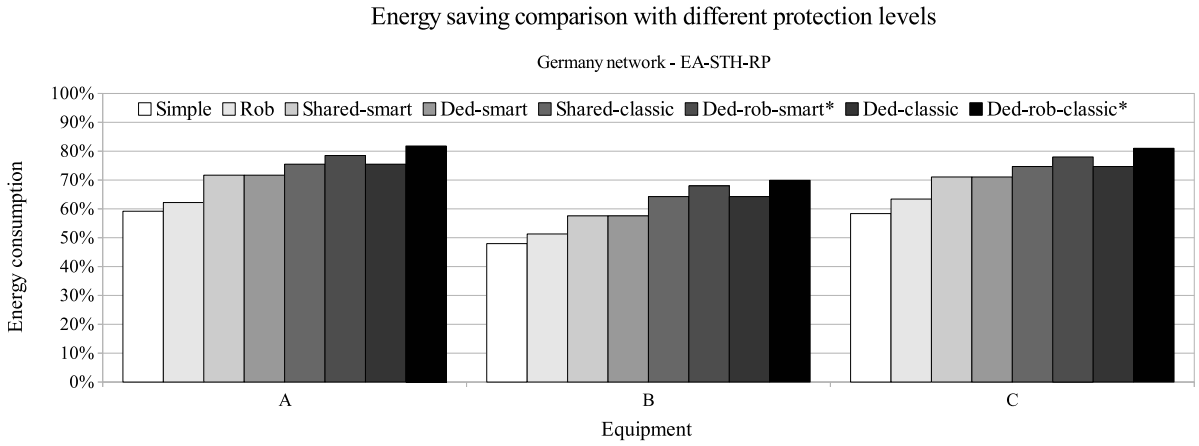


Figure 4.13 Energy savings achieved by EA-STH-RP with $\Omega = 5$ when implementing the different protection schemes on `germany50` instances.

CHAPTER 5

SEANM with Shortest Path Routing

In IP networks, routing the incoming traffic demands along the shortest paths between their sources and destinations is a very widespread practice. The most popular shortest path routing protocol is OSPF. With OSPF, an administrative link weight chosen by the network operator is assigned to each link, and traffic demands are routed on the shortest paths determined by the entire set of link weights. Each router builds its own routing table by referring to its own shortest path tree to determine which line card has to be used to forward the packets toward each specific destination; if two or more line cards lie on at least one shortest path toward a given destination, the router may opt to forward the packets on a single line card, e.g., that with the smallest IP address or that connected to the next-hop router with the lowest ID, or to equally split the interested traffic between all the shortest path line cards (equal cost multi-path (ECMP)). As suggested by Cisco guidelines on OSPF, link weights are typically set by network operators by using the inverse of link capacity. The shortest paths determined by this approach tend to use the link with the highest capacity. However, although this weight setting scheme may represent an acceptable solution, the link weight choice can be further optimized to adjust network routing (TE) and therefore improve network performance (see, e.g., Altin *et al.*, 2009; Fortz et Thorup, 2002; Bley, 2010).

With respect to classic per-flow (or flow-based) routing, performing TE while being constrained by the shortest path rule involves some complications: (i) the feasible set of routing paths is restricted because several combinations of paths are not compatible with the shortest-path scheme, and (ii) the strict bond between routing paths and link weights adds a new degree of complexity which makes the TE optimization problems bi-level. Nevertheless, it is worth pointing out that in large networks, the shortest path restriction does not cause a substantial degradation of the optimal TE solution (see Proposition 4.1 in Pióro et Medhi, 2004), while complexity issues have been successfully overcome in literature by means of very efficient heuristic approaches (Umit et Fortz, 2007; Altin *et al.*, 2009).

From the practical point of view, shortest path routing offers to network operators a substantial simplification of the routing management process, which does not require the definition of a dedicated path for each traffic demand (operation which is not very scalable as the number of demands rises), but involves only the configuration of a limited number of administrative weights. Furthermore, if we consider OSPF, the link states packet periodically exchanged by the network routers can be effectively exploited to disseminate real-time

information on network conditions.

Along with typical TE objectives such as delay or congestion minimization, it has been recently shown that link weight optimization can be effectively exploited to perform EANM as well as SEANM. Link weights can be adjusted to minimize the load proportional power consumption component as well as put to sleep the redundant network elements by means of very high link weights.

The reminder of the chapter is organized as follows. We discuss our general approach for SEANM with shortest path routing (SEANM-SP) and energy-aware link weight optimization in Section 5.1, while we describe the addressed SEANM-SP optimization problem and provide a MILP formulation in Section 5.2. Four heuristic methods are then presented in Section 5.3 and computational results are discussed in Section 5.4.

5.1 Our Approach for Energy-Aware Link Weight Optimization

Given an IP backbone networks operated with OSPF and ECMP (equal cost multi-path), our aim is to develop an optimization framework to maximally reduce network power consumption as well as network congestion, by effectively adjusting the OSPF administrative link weights.

Since static consumption components are largely predominant in current network hardware, where around 90% of the total energy is required to barely power on a device (Chabarek *et al.*, 2008; Mellah et Sansò, 2009), we focus on SEANM and opt to reduce the overall network consumption by putting to sleep the redundant network devices (both routers and links) which are not needed to guarantee the required QoS. Note that sleeping-based strategies will maintain their effectiveness until static and proportional energy components will be at least of the same order of magnitude (Chiaraviglio *et al.*, 2013b).

Network congestion is controlled by imposing hard constraints on maximum link utilization and minimizing a measure of network saturation which was first used in (Fortz et Thorup, 2002) (see Figure 5.1).

It is worth pointing out that in our bi-objective optimization approach, energy consumption and network congestion are minimized in a lexicographic way. That means that first, we minimize the network energy consumption while respecting constraints on link maximum utilization, and only in a second moment we consider the energy-optimal topology and optimize the link weights of the active elements to further reduce the network congestion. Practically speaking, we first guarantee an acceptable congestion level by imposing limitations on the link maximum utilization, and, once the reduced network topology (that composed of only the active devices) has been derived, we focus our efforts to directly decrease the overall

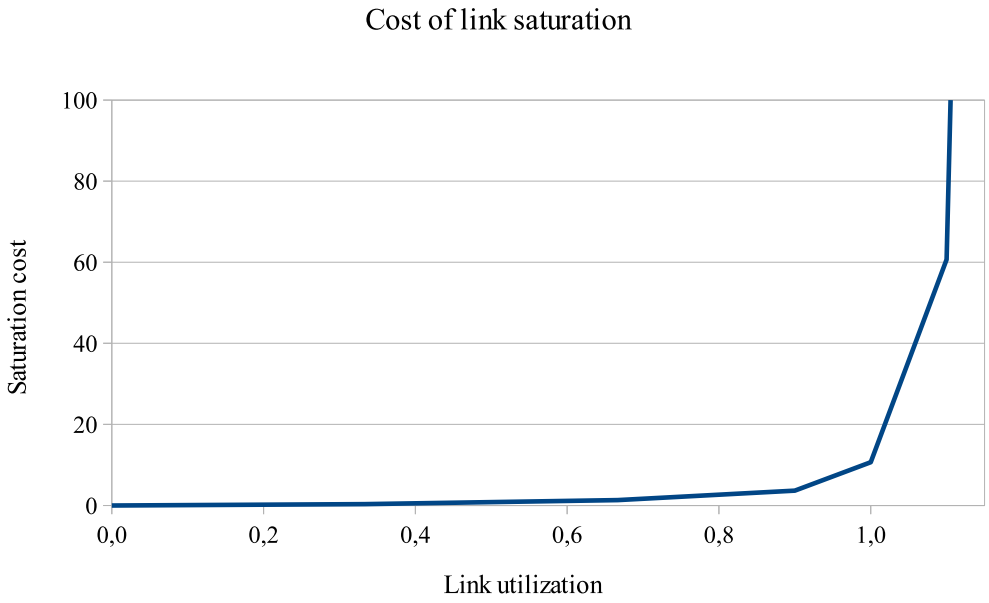


Figure 5.1 Piecewise-linear link saturation cost function.

congestion.

We consider a centralized off-line approach, according to which the link weights are optimized in a planning phase by a centralized network controller. The optimization is thus performed off-line, typically one day in advance of the considered time period if we are optimizing on a daily basis, or one week ahead in case of a weekly configuration. We assume that the slow and periodic dynamic of Internet traffic (see e.g., Mackarel *et al.*, 2011) is exploited by network operators to split a single day among a few time periods. Each time interval, which should present a quite constant level of traffic and should last for at least a few hours, is characterized by a predicted traffic matrix which can be accurately estimated by network operators by means of direct measurements (Cisco Systems, 2012) and state of the art estimation techniques (Casas *et al.*, 2009). Topology and traffic information is then processed by the optimization framework to derive a set of energy-aware link weights for each time interval.

An overall sketch of our approach is given in Figure 5.2, which helps us to highlight the most important aspects of the proposed framework. The day is divided into three time-periods, namely *morning*, *afternoon* and *night*, characterized by, respectively, *moderate*, *high* and *low* levels of traffic. All network links are bidirectional with one unit of capacity and 70% of maximum utilization allowed per direction, and a single demand d having node A as origin o^d and node E as destination t^d is considered. The predicted value of traffic r^d is

1 during *morning*, 2 during *afternoon* and 0.5 during *night*. The different link weight sets are applied in the network by a centralized network management platform (NMP), which can be instructed during the planning phase to automatically switch the OSPF configuration at a specific time, or, like we will later show in Chapter 6, can dynamically detect in real-time the best moment to perform the configuration switching. Note that link weights can be practically modified by means of standard network management protocols like simple network management protocol (SNMP), which is widely supported by the large majority of network devices.

In this trivial example (Figure 5.2), three different paths with 0.7 units of free capacity (remind the maximum utilization limit), namely $A-B-E$, $A-C-E$ and $A-D-E$ are available to carry traffic demand d. Keeping in mind that ECMP is enabled, we can quickly identify the optimal energy-aware configuration for each time slot. In the *morning* slot two paths $A-B-E$ and $A-C-E$ are needed to satisfy the single unit of traffic of d; the elements of the third path $A-D-E$, i.e., node C and its two incident links can be put to sleep by setting with a value large enough (in our example we use 100) the OSPF weights of links (A, D) and (D, E) so as to exclude them by all the shortest paths. Note that in our example 100 is not the only feasible choice for the sleeping weights, it would be enough to make the sleeping path longer than the active ones. By contrast, the weights of the remaining four links have to be configured so as to (i) respect the maximum utilization limit of 70% and (ii) minimize the overall congestion cost. The optimal solution, which consists into equally splitting d among the two active paths, is achieved by using the same weight value, in this example 1, for each active link. During the *afternoon* period the traffic grows up to 2 units, and all the three paths must be exploited to keep the maximum utilization under the desired value. No element is put to sleep and all link weights are set with the same value so as to guarantee the equal traffic splitting among all he three available paths. Finally, in the *night* period one path is enough to carry the 0.5 units of traffic of d and the all the network elements of the remaining two paths (4 links and 2 nodes) can be put to sleep.

Note that the activation of the sleeping state, which we do not directly consider in our work, can be automatically performed by some distributed (Nedevschi *et al.*, 2008) or centralized mechanisms (Bolla *et al.*, 2011a) able to detect the absence of traffic on the considered devices.

The choice of a centralized off-line approach is motivated by some crucial observations concerning solution optimality and network stability. Placing the core of the optimization in the planning off-line stage brings some significant pros in terms of (i) large time availability (typically in the order of several hours), which allows to handle more complex problems and enhances the possibility to compute global optimal solution, (ii) network stability and (iii)

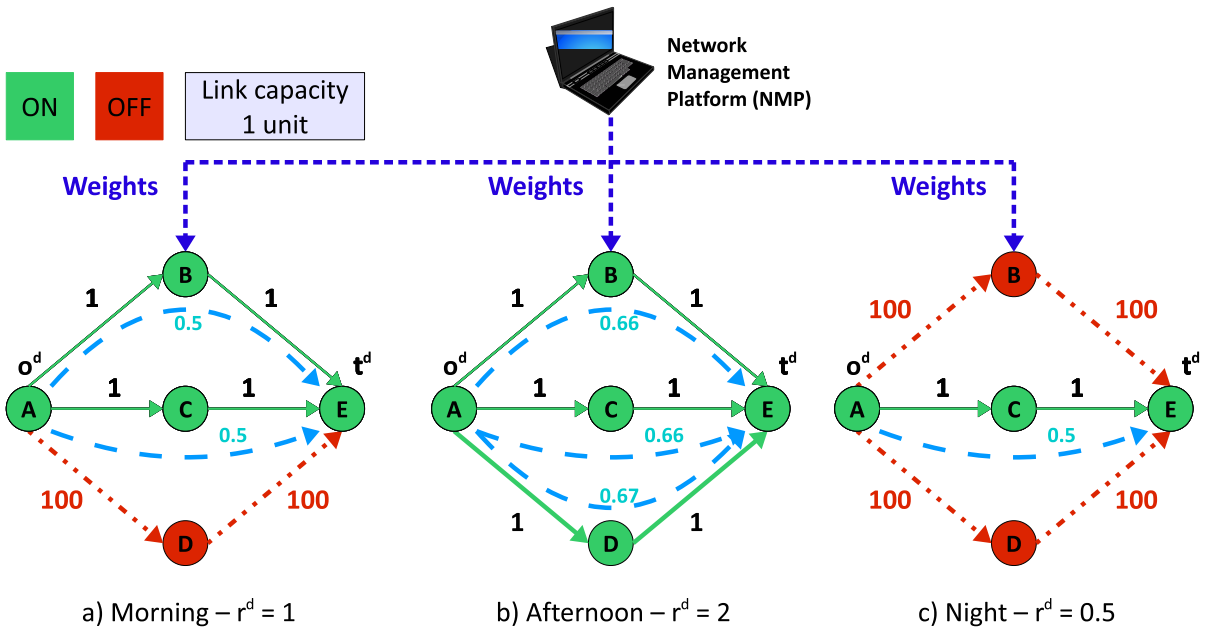


Figure 5.2 Centralized off-line approach for SEANM-SP.

complete control by the network operator on network dynamics. Especially the last two aspects are of great importance for network operators, which otherwise would be reluctant to dynamically adjust the network configuration.

To conclude this general overview, it is worth pointing out that, although operating off-line forces us to statically split the day into time intervals and makes the operator incapable of reacting in real-time to varying traffic conditions, all these issues have been demonstrated to not represent a real obstacle. First of all, since the time period splitting comes along with the necessity of network operators to avoid frequent reconfigurations that would cause network instability and performance degradation, also on-line approaches should limit the possibility of reconfiguring the network every time the conditions change. Secondly, it has been analytically shown that in IP networks, the use of a limited set of configurations (each one maintained for at least few hours) allows to reach quasi-optimal solutions (Chiaraviglio *et al.*, 2013a). Furthermore, the use of constraints on the maximum utilization can be effectively exploited to provide enough spare capacity to absorb unexpected traffic variations without the need of adjusting the routing or powering on some devices.

Finally, note that, as we will later show in Chapter 6, our off-line approach can be easily combined with on-line mechanisms to further refine the network configuration according to real time measurements. In this regard, it is worth emphasizing that, since we keep the OSPF protocol unchanged, we do not interfere with other procedures already adopted in current IP networks with fixed routing metrics to react to traffic variations or device failures on s short

time scale.

5.2 MILP formulation for SEANM-SP

Let us recall some of the notation already presented in Chapter 2. The network topology is represented by a directed graph $G = (V, A)$, where V and A are the sets of nodes and links, respectively. Nodes are further split between *edge* nodes V^e which generate and receive traffic, and *core* nodes V^c which simply carry the network traffic. Since we assume each link to be composed of a single line card, we consider n_{ij}, i.e., number of line cards on link (i, j), equal to 1 for each $(i, j) \in A$. D represent the traffic matrix, and each demand $d \in D$ is characterized by a source node o^d, a destination node t^d and a value of incoming traffic r^d. Each link $(i, j) \in A$ has capacity c_{ij}, while the link maximum utilization allowed on all the link is μ, with $0 \leq \mu \leq 1$ (the same for all links). π_i and π_{ij} represent the amount of energy required to keep active router $i \in V$ and link $(i, j) \in A$, respectively.

We consider the following extension of the basic weight optimization problem with ECMP for intra-domain TE (see, e.g., Pióro et Medhi, 2004), which is also a variant of the reference problem for SEANM-SP(2.1–2.19). We refer to it as energy-aware traffic engineering with shortest path routing (E-TESP).

E-TESP: Given as input a traffic matrix and an IP network topology with routers and capacitated links, select the network elements (both routers and links) that can be put to sleep and adjust the OSPF link weights so as to lexicographically optimize, in the following order, the overall network power consumption (primary objective) and the considered measure of network congestion (secondary objective). In the meantime all the traffic demands must be routed without violating the maximum link utilization allowed on each link.

Being y_k and w_{ij} the binary variables representing the power status (on/off) of, respectively, routers and links (remind that w are binary because $n_{ij} = 1\ \forall (i, j) \in A$), and being f_{ij}^t the non-negative real variables indicating the amount of flow carried by link $(i, j) \in A$ and destined to node $t \in V^e$, let us express the following MILP formulation for a special variant of the SEANM-SP problem, where only energy consumption (the primary objective) is directly minimized and network congestion is controlled by means of link maximum utilization constraints:

$$\min \left\{ \sum_{(i,j)\in A} \pi_{ij} w_{ij} + \sum_{i \in V} \pi_i y_i \right\} \tag{5.1}$$

s.t.

$$w_{ij} \leq n_{ij} y_i, \qquad \forall (i,j) \in A \tag{5.2}$$

$$w_{ij} \leq n_{ij} y_j, \qquad \forall (i,j) \in A \tag{5.3}$$

$$\sum_{(i,j)\in A} f_{ij}^t - \sum_{(j,i)\in A} f_{ji}^t = \begin{cases} -\sum_{\substack{d\in D:\\ t^d=t}} r^d & \text{if } i = t, \\ \sum_{\substack{d\in D:\\ (o^d,t^d)=(i,t)}} r^d & \text{otherwise} \end{cases}, \qquad \forall i,\, t \in V^e, \tag{5.4}$$

$$\sum_{t\in V^e} f_{ij}^t \leq \mu c_{ij} w_{ij}, \qquad \forall (i,j) \in A \tag{5.5}$$

$$0 \leq z_i^t - f_{ij}^t \leq (1 - u_{ij}^t) \sum_{\substack{d\in D:\\ t_d=t}} r^d, \qquad \forall t \in V^e,\, (i,j) \in A \tag{5.6}$$

$$f_{ij}^t \leq u_{ij}^t \sum_{\substack{d\in D:\\ t_d=t}} r^d, \qquad \forall t \in V^e,\, (i,j) \in A \tag{5.7}$$

$$0 \leq l_j^t + \omega_{ij} - l_i^t \leq (1 - u_{ij}^t) M, \qquad \forall t \in V^e,\, (i,j) \in A \tag{5.8}$$

$$1 - u_{ij}^t \leq l_j^t + \omega_{ij} - l_i^t, \qquad \forall t \in V^e,\, (i,j) \in A \tag{5.9}$$

$$u_{ij}^t \leq w_{ij}, \qquad \forall t \in V^e,\, (i,j) \in A \tag{5.10}$$

$$\omega_{ij} \geq (1 - w_{ij}) \omega_{max}, \qquad \forall (i,j) \in A \tag{5.11}$$

$$1 \leq \omega_{ij} \leq \omega_{max}, \qquad \forall (i,j) \in A \tag{5.12}$$

$$w_{ij} \in [0, \ldots, n_{ij}], \qquad \forall (i,j) \in A \tag{5.13}$$

$$u_{ij}^t,\, y_k \in \{0,1\}, \quad \forall k \in V,\, t \in V^e,\, (i,j) \in A \tag{5.14}$$

$$f_{ij}^t,\, l_h^t,\, z_h^t,\, \omega_{ij} \geq 0, \quad \forall h \in V,\, t \in V^e,\, (i,j) \in A. \tag{5.15}$$

The above formulation presents two important differences with respect to the reference model for SEANM-SP (2.1–2.19). The Objective function (5.1) minimizes the overall network consumption by focusing only on the static power component (ON/OFF consumption profile). Flow variables f_{ij}^t are not defined per-demand but per-destination; that allows to substantially reduce the problem complexity while preserving the routing feasibility. Let us quickly recap the meaning of each constraint: we have coherence Constraints (5.2–5.3) to ensure that active links are connected to active nodes (remind that only core nodes V^c can be put to sleep), flow conservation Constraints (5.4) to correctly route the traffic demands, maximum link utilization Constraints (5.5) to limit network congestion, ECMP Constraints (5.6–5.7) to correctly split the traffic at each node, shortest path Constraints (5.8–5.9) to define shortest paths which are compatible with the link weights, and Constraints (5.10–5.12) to correctly configure the weight of the sleeping links. Remind that binary variables u_{ij}^t are equal to 1 if link $(i,j) \in A$ is on a shortest path between node i and node t, non-negative real variables

z_i^t are the common value of flow assigned to all the outgoing links of i and belonging to the shortest paths from i to t, non-negative real variables l_i^t represent the length of the shortest path from node i to node t, and non-negative real variables ω_{ij} indicate the link weights. Note that M is a large enough constant.

Unfortunately, (2.1–5.15) turns out to be hardly tractable even for networks of very limited dimensions. We have tested the E-TESP formulation by considering a set of six networks described in (Orlowski *et al.*, 2010). We report in Table 5.1 the computational results obtained by solving (2.1–5.15) with CPLEX 12.2 running on a machine equipped with 8Gb of RAM and an Intel i7 processors with 4 core and multi-thread 8x. Columns *Instance* and *V-A-D* identify, respectively, the instance ID represented by the pair *network-traffic level* and the number of nodes-links-demands of the considered network. The last group of columns, namely *Non trivial*, *Optimum* and *t(s)* tells us if the final solution returned by the solver is trivial or not, if the solution itself is optimal, and how long the resolution has lasted. Note that a solution is considered *trivial* if all elements are active and all weights are equal to 1.

The results clearly point out that (2.1–5.15) can be solved in a reasonable amount of time only when the overall traffic level is very low, i.e., when the input traffic matrix can be

Table 5.1 Computational results for the E-TESP formulation solved with six different network topologies.

Instance	**V-A-D**	**Non trivial**	**Optimum**	**t (s)**
abilene-1%	12-15-132	yes	yes	0.6
abilene-10%	12-15-132	yes	yes	0.6
abilene-30%	12-15-132	yes	yes	339.5
dfn-bwin-1%	10-45-90	yes	yes	182.1
dfn-bwin-10%	10-45-90	yes	yes	257838.0
dfn-bwin-30%	10-45-90	no	no	648927.0
dfn-gwin-1%	11-47-110	yes	yes	348.4
dfn-gwin-10%	11-47-110	yes	yes	591561.0
dfn-gwin-30%	11-47-110	no	no	848884.0
di-yuan-1%	11-42-22	yes	yes	2551.5
di-yuan-10%	11-42-22	yes	yes	1867.1
di-yuan-30%	11-42-22	no	no	73835.9*
pdh-1%	11-34-24	yes	yes	11.1
pdh-10%	11-34-24	yes	yes	1630.6
pdh-30%	11-34-24	yes	no	25137.1*
polska-1%	12-18-66	yes	yes	6.8
polska-10%	12-18-66	yes	yes	16.9
polska-30%	12-18-66	yes	yes	29348.5
* Out of memory				

routed in the 100%-active network with fully splittable routing by maintaining the maximum utilization under 1% or 10% (instances *network-1%* and *network-10%*). In these cases the optimal solution corresponds to a simple steiner-tree connecting all the edge node V^e. When traffic is increased (instances *network-30%*), in three instances over six the solver has not been able to compute any feasible solution but a trivial one after one day of computation.

5.3 Heuristic Methods for SEANM-SP

In this section we present four novel heuristic algorithms to lexicographically minimize power consumption and network congestion by efficiently adjusting the OSPF link weights: (i) greedy algorithm for energy saving (GA-ES), (ii) greedy randomized algorithm for energy saving (GRA-ES), (iii) two-stage algorithm for energy saving (TA-ES) and (iv) MILP-based algorithm for energy-aware weight optimization (MILP-EWO). All these procedures, which are based on multiple optimization stages, include a link weight optimization phase operated by the open-source interior gateway protocol weight optimization (IGP-WO) algorithm presented in (Fortz et Thorup, 2002) and freely distributed inside the TOTEM toolbox (Leduc *et al.*, 2006). For this reason, before entering into the details of each algorithm, we first provide a quick overview on IGP-WO.

5.3.1 Congestion-Aware Link Weight Optimization

IGP-WO is one of the most efficient state-of-the-art algorithm for the congestion-aware optimization of the link weights in IP networks operated with OSPF and ECMP enabled (Altin *et al.*, 2009). Given an IP network topology and a traffic matrix, the tabu-search algorithm of IGP-WO looks for the set of link weights that minimizes a measure of network congestion computed as the summation over all the links of the cost function illustrated in Figure 5.1 and already described in Section 3.3.1.

The optimization problem implicitly solved by IGP-WO can be expressed as:

$$\min \left\{ \sum_{(i,j)\in A} \iota_{ij} \right\} \tag{5.16}$$

s.t.

$$(5.4),\ (5.6)-(5.10)$$

$$f_{ij} = \sum_{t \in V^e} f_{ij}^t, \qquad \forall (i,j) \in A \tag{5.17}$$

$$\iota_{ij} \geq \alpha_h f_{ij} - \beta_h c_{ij}, \quad \forall (i,j) \in A,\ h \in H \tag{5.18}$$

$$f_{ij}^t,\ f_{ij} \geq 0, \quad \forall (i,j) \in A,\ t \in V. \tag{5.19}$$

The Objective function (5.16) minimizes the overall congestion measure related to the link cost function $Cong_{ij}$, Constraints (5.4) and (5.6–5.10 are taken from the E-TESP formulation and aim at guaranteeing shortest path routing, while Constraints (5.17) compute the total amount of traffic carried by a link. Finally, Constraints (5.18) are required to model the piecewise link cost function composed by the set of pieces H, where each piece $h \in H$ is described by a slope α_h and an offset β_h. The non-negative real variables ι_{ij}^{σ} assume the right congestion cost for each link $(i,j) \in A$. α_h and β_h are properly set so that

$$Cong'_{(i,j)}(c_{ij}, f_{ij}) = \begin{cases} 1\,for & 0 \leq f_{ij}/c_{ij} < 1/3 \\ 3\,for & 1/3 \leq f_{ij}/c_{ij} < 2/3 \\ 10\,for & 2/3 \leq f_{ij}/c_{ij} < 9/10 \\ 70\,for & 9/10 \leq f_{ij}/c_{ij} < 1 \\ 500\,for & 1 \leq f_{ij}/c_{ij} < 11/10 \\ 5000\,for & 11/10 \leq f_{ij}/c_{ij} < \infty, \end{cases} \tag{5.20}$$

where $Cong'_{(i,j)}(c_{ij}, f_{ij})$ is the first derivative of the cost function. It is worth pointing out that no hard constraint on link maximum utilization/capacity is considered. It is in fact the responsibility of the convex link cost function to avoid saturated solutions, when possible.

IGP-WO starts the tabu search, which at each iteration adjusts a link weight to improve the objective function, by starting from an input set of link weights. The starting weight set can be randomly computed by IGP-WO itself, or provided in input by the network administrator. As showed in (Umit et Fortz, 2007), the smart choice of the input weights can substantially improve the algorithm performance and speed up the convergence toward a sub-optimal solution. The idea proposed in (Umit et Fortz, 2007) consists into warm-starting IGP-WO by taking as link weights the dual variable values corresponding to the optimal solution of the linear programming (LP) formulation for congestion minimization with multicommodity splittable flows which is composed by (5.16), (5.4), (5.17), (5.18) and (5.19).

According to Umit et Fortz (2007), the dual variable values corresponding to Constraints (5.17) are taken as initial link weights.

5.3.2 Greedy Algorithm for Energy Savings (GA-ES)

Adopting a greedy strategy represents a natural approach to rapidly address the E-TESP problem. The GA-ES algorithm we propose is composed of three different stages, of which the first and the last one are responsible for optimizing the link weights, while the middle-one is the actual greedy procedure aiming at putting to sleep the unnecessary network devices. The pseudo-code of GA-ES is reported in Algorithm 2.

Given the input network topology $G(V, A)$ and the predicted traffic matrix D, we first focus on computing a set of efficient link weights which minimizes the cost of network congestion in the full active network. This weight set is computed by means of IGP-WO warm started with the dual weights derived as in (Umit et Fortz, 2007), Note that in our tests, IGP-WO is typically run for 150 iterations by setting a maximum weight value of 100. Once congestion is minimized on the full active network, we keep the optimal link weights and proceed in a greedy way. We first sort the network elements (nodes and links) according to a specific criterion, and then try to switch them off one by one.

We distinguish between the sorting criteria used for network nodes and those applied to network links (see Table 5.2). We consider three node-sorting criteria, least-link (LL), least-flow (LF), sum-of-weights (SW) (LL and LF were already used in Chiaraviglio *et al.*, 2009, in a different greedy approach with flow-based routing), according to which nodes are sorted in non-decreasing order with respect to the number of incident links (LL), the amount of carried traffic (LF), and the sum of the incident link weights (SW). For what concerns link, we define two policies, of which the first one is the LF one already used to sort nodes, while the second one is called TE and is based on the weight value. Note that SW and TE policies are based on the idea that the network elements with highest link weights are less likely to be used to carry traffic. Due to their higher energy consumption, first we consider network nodes and then network links.

Input: G Topology, D Traffic Matrix,
$g(e)$ Sorting criterion

ComputeOSPFLinksWeights(D, G);
SortNetworkElements($g(e), G$);
foreach *Element* $e \in G$*, according to the sorting* **do**
  TurnOffElement(e, D, G);
  SortNetworkElements($g(e), G$);
end
ComputeOSPFLinksWeights(D, G);

Algorithm 2: GA-ES algorithm.

Each greedy iteration consists into considering the first element at the top of the sorted list and verifying whether its presence is crucial to maintain the maximum link utilization below the threshold μ. An element is therefore powered off if and only if the OSPF with ECMP routing determined by the weights of the active elements does not cause the violation of the maximum utilization limit μ. The greedy procedure is concluded once all the network elements have been tested. It is worth pointing out that the input set of link weights significantly influences the performance of the greedy stage: therefore, the weight optimization stage is crucial to maximize the amount of energy savings.

Finally, in the last part of GA-ES we take the reduced network topology derived by the greedy procedure and run IGP-WO to further reduce the network congestion in the energy-efficient sub-network (lexicographic optimization).

5.3.3 GRASP for Energy Savings (GRA-ES)

The GA-ES procedure is very light under the computational profile, but, as naturally expected in a greedy approach, does not guarantee an exhaustive exploration of the solution space. To overcome this limitation we propose GRA-ES, a novel algorithm obtained by integrating the greedy routine of GA-ES into a GRASP methodology.

A GRASP consists into a multi-start meta-sheuristic to find approximate solutions of combinatorial optimization problems (Resende et Ribeiro, 2003)(Resende et Ribeiro, 2005). GRA-ES is based on two new architectural elements, i.e., the RCL and the *Second-Level Multi-Start*. The implementation of RCL implies that, during the greedy stage, the algorithm is instructed to randomly selects one of the firsts $k\%$ elements of the sorted list (not only the first one). This practice produces a perturbation in the element ordering and allows (with a good probability) to generate different solutions whenever the greedy routine is run. To exploit this feature, the greedy procedure is thus repeated several times by considering the same sorting criterion (Second-Level Multi-start) and only the best solution is finally returned.

Table 5.2 Combinations of sorting criteria, the rows correspond to the link criteria and the columns to the router criteria.

	LF	**LL**	**SW**
LF	LF-LF	LL-LF	SW-LF
TE	LF-TE	LL-TE	SW-TE

5.3.4 Two-Stages Algorithm for Energy Savings (TA-ES)

Due to the complex problem structure, which obviously increases together with network dimensions, taking local decisions driven by some pre-defined criteria as we do in GA-ES and GRA-ES may negatively affect the effectiveness of the successive moves, and consequently reduce the overall performance. For instance, the powering off a certain node during the first iteration of GA-ES might undermine, in a second moment, the possibility of putting to sleep a group of other five nodes which highly contribute to the the overall network consumption.

For this reason, with the aim of improving the solution performance, we propose the novel and more complex TA-ES procedure, which takes routing and sleeping decisions by considering their global impact on the whole network. TA-ES, whose structure is illustrated in the flow chart of Figure 5.3, is composed of two main stages, namely the *Switching-off Stage* and the *Feasible Routing Stage*.

The *Switching-off Stage* receives as input the network topology $G(V, A)$ and the predicted

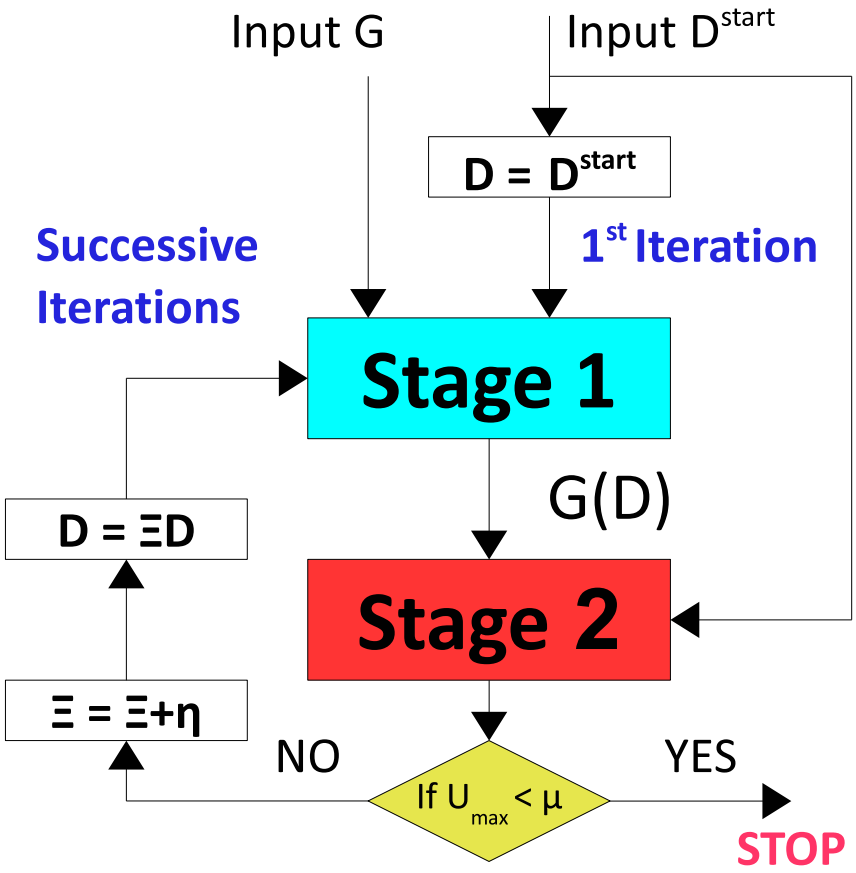


Figure 5.3 Two-stages algorithm for energy saving (TA-ES).

traffic matrix D. Its aim, which consists into determining which nodes and links should be put to sleep to minimize the power consumption, is pursued by means of the MILP formulation solved within a 3% gap from the best lower bound:

$$\min \left\{ \sum_{(i.j)\in A} \pi_{ij} w_{ij} + \sum_{i \in V} \pi_i y_i \right\} \tag{5.21}$$

s.t.

$$w_{ij} \leq n_{ij} y_i, \qquad \forall (i,j) \in A \tag{5.22}$$

$$w_{ij} \leq n_{ij} y_j, \qquad \forall (i,j) \in A \tag{5.23}$$

$$\sum_{(i,j)\in A} f_{ij}^t - \sum_{(j,i)\in A} f_{ji}^t = \begin{cases} -\sum_{\substack{d\in D: \\ t^d = t}} r^d & \text{if } i = t, \\ \sum_{\substack{d\in D: \\ (o^d,t^d)=(i,t)}} r^d & \text{otherwise} \end{cases}, \qquad \forall i,\, t \in V^e, \tag{5.24}$$

$$\sum_{t \in V^e} f_{ij}^t \leq \mu c_{ij} w_{ij}, \qquad \forall (i,j) \in A \tag{5.25}$$

$$w_{ij},\, y_k \in \{0,1\}, \quad \forall k \in V,\, (i,j) \in A \tag{5.26}$$

$$f_{ij}^t, \geq 0, \quad \forall t \in V^e,\, (i,j) \in A. \tag{5.27}$$

The above formulation is a relaxation of the main one (5.1–5.15) for E-TESP, obtained by removing all the shortest path constraints and variables. Routing is multi-path fully splittable (see Constraints (5.24)). This formulation falls within the well-known class of multi-commodity network design (MCND) problems Gendron *et al.* (1998). Despite this class of problems is known to be NP-hard, quasi optimal solutions (gap lower than 3%) to (5.21–5.27) for network up to 100 nodes and 300 links are typically obtained by a commercial solver like CPLEX in less than two hours of computing time.

Once a reduced network topology is computed by the *Switching-off Stage*, we then run the *Feasible Routing Stage* to determine the set of link weights which minimizes the overall cost of link utilization (overall congestion) and does not violate the maximum link utilization μ. Link weights are adjusted by running IGP-WO on the reduced topology $G(D)$ for 150 iterations and setting the maximum weight value to 100 (the same configuration used in GA-ES and GRA-ES). Also in this case IGP-WO is warm-started with the dual-weights derived as in (Umit et Fortz, 2007).

As expected, making (5.1–5.15) tractable by removing shortest path constraints has both pros and cons: it allows to a find reduced topology $G(D)$ in relatively small amount of time, but does not provide any assurance on the existence of a set of link weights able to maintain

the utilization below the desired limit μ. The fully splittable routing considered at the first stage might be in fact incompatible with the shortest path scheme considered by OSPF.

If IGP-WO fails to find a feasible set of OPSF weights for the reduced network $G(D)$ (either because it does not exist or it is simply not found by the tabu-search), we increase the traffic levels by scaling the original traffic matrix D with a fixed parameter Ξ, and repeat the *Switching-off Stage* by considering the traffic matrix ΞD. Ξ, which at the beginning of the procedure is equal to 1, is incremented by η after each run of IGP-WO wherein no feasible set of weights is found. If the maximum utilization threshold μ is violated within 1%, then η is 0.05, otherwise η is 0.1.

Scaling the traffic matrix naturally increases the power consumption of the reduced network $G(\Xi D)$, since more active elements are expected to be required to satisfy the increased matrix. Note that the *Feasible Routing Stage* takes always as input the original traffic matrix D. The *Feasible Routing Stage* ends when a feasible set of link weights is found.

5.3.5 MILP-based Algorithm for Energy-Aware Weight Optimization

Motivated by the aim of achieving quasi-optimal solutions while limiting the computational complexity of the resolution algorithm, we propose the novel MILP-EWO procedure, born by the effective combination of the core parts of GA-ES and TA-ES. Our purpose is to achieve a trade-off between the low complexity of GA-ES and the high performance of TA-ES in terms of maximum savings.

MILP-EWO, whose architecture is shown in Figure 5.4, is logically split into three sequential blocks: (i) the pre-processing stage , (ii) the MILP-based stage, and (iii) the post-processing stage (see the flow chart in Figure 5.4).

The pre-processing stage consists in running a slightly modified variant of GA-ES where, (i) the second run of IGP-WO is removed, and (ii) the maximum link utilization μ is decreased by 0.1. The aim of the pre-processing stage is to exploit very simple criteria to rapidly identify which routers and links may be immediately switched off without negatively affecting the overall performance. The adoption of a more restrictive maximum utilization threshold is specifically conceived so as to prevent the greedy routine from taking irreversible decisions that would require more global considerations. Let $G^{'}$ denote the reduced network topology resulting from the pre-processing (GA-ES) stage.

The MILP-based stage takes as input the reduced topology $G^{'}$ and includes a complete run of TA-ES. Since a consistent subset of network elements has been already powered off at the previous stage, in this case TA-ES is able to quickly determine (in the order of few minutes) the last elements, which are also the most crucial one, to be put to sleep.

To conclude the algorithm and further verify the possibility of putting to sleep some

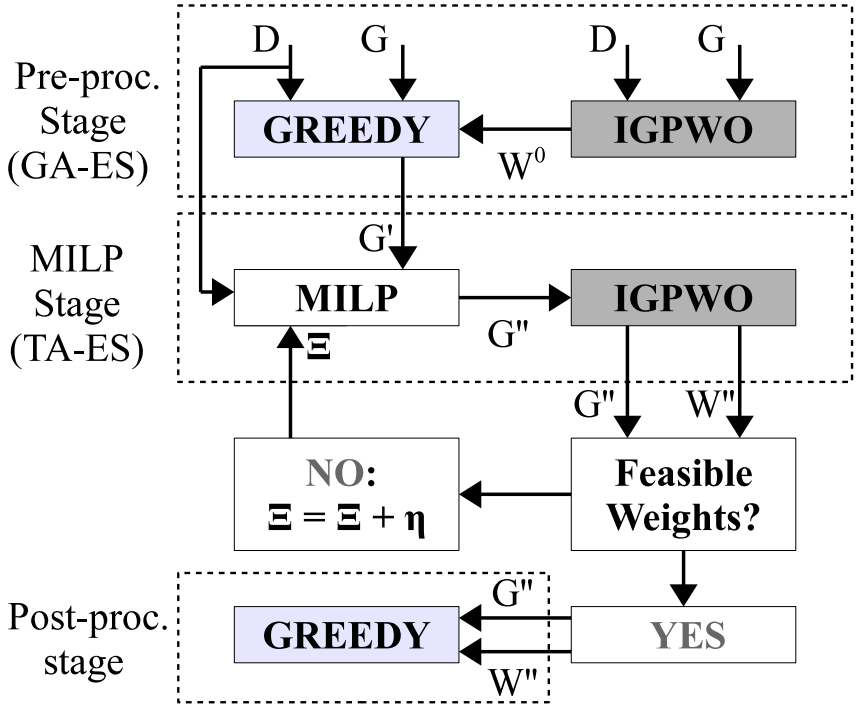


Figure 5.4 MILP-EWO flow chart.

additional elements, the post-processing stage runs a final round of the greedy routine of GA-ES by taking as input the reduced network G'' and the congestion-aware weights W'' returned by the MILP-based stage.

Table 5.3 Network topologies used in the tests. The first seven ones are real network topologies from the Rocketfuel project (Spring *et al.*, 2002), while the last one is used in (Lee *et al.*, 2012).

Network	**Type**	$\|V\|$	$\|A\|$	$\|V^e\|$	$\|V^c\|$	$\%V^c$
Ebone	Backbone	87	322	31	56	64.4
Exodus	Backbone	79	294	38	41	51.9
Sprint	Access	52	168	52	0	0
AT&T	Access	115	296	115	0	0
Abovenet	Access	19	68	68	0	0
Genuity	Access	42	110	42	0	0
Tiscali	Access	41	174	41	0	0
USA28	Access	28	90	28	0	0

5.4 Computational Results

5.4.1 Test Instances

Network Topologies

We have conducted an extensive test campaign by considering eight different real network topologies, of which seven are object of study of the Rocketfuel project (Spring *et al.*, 2002) and one, i.e., the USA28 network, has been kindly provided to us by the authors of (Lee *et al.*, 2012).

Since our approaches for SEANM-SP assume that both nodes and links can be potentially put to sleep, our tests mainly focus on backbone networks composed of both edge nodes and core nodes, where only the latter may be switched off. However, for comparison purposes with (Zhang *et al.*, 2010),(Vasić et Kostić, 2010) and (Lee *et al.*, 2012), also IP access networks containing only edge nodes (which cannot be put to sleep) are considered. The features of the eight network topologies are summarized in Table 5.3. Note that the five Rocketfuel access networks, Abovenet, AT&T, Genuity, Sprint and Tiscali, have been previously used in (Zhang *et al.*, 2010) and (Vasić et Kostić, 2010) to test other SEANM approaches (see Chapter 2).

Since Rocketfuel network topologies are not provided with any information concerning link capacity and network equipment, the network test-bed has been customized by us as follows: while experimenting with the backbone networks Ebone and Exodus, we consider all the links to be equipped with the same amount of capacity. Furthermore, we assume that each node comes with M10i routers (power consumption π_i of 86.4 W), and each link is equipped with a Gigabit Ethernet line card (power consumption pi_{ij} of 7.3 W and capacity c_{ij} of 2 Gbps). Since a single M10i router can support at most eight Gigabit Ethernet line cards, we consider each node to come with $\lceil \frac{g_i}{8} \rceil$ routers, where g_i is the number of link connected to node $i \in V$. The splitting between edge nodes V^e and core nodes V^c is randomly performed by considering all leaf nodes as edge ones and by keeping at least one edge node per city.

For what concerns the access networks we maintain both capacity values and equipment configurations used in the instances provided by the authors of (Zhang *et al.*, 2010; Vasić et Kostić, 2010; Lee *et al.*, 2012): link capacities are equal to (i) 9920 Gbps or 2480 Gbps in Sprint and AT&T instances of (Zhang *et al.*, 2010), (ii) to 100 Mbps or 52 Mbps in Abovenet, AT&T, Genuity, Sprint, and Tiscali instances adopted in (Vasić et Kostić, 2010) and (iii) to 100 Mbps in the USA28 network used in (Lee *et al.*, 2012).

Traffic Matrices

Access network instances have been provided by the authors of (Zhang *et al.*, 2010; Vasić et Kostić, 2010; Lee *et al.*, 2012) along with the traffic matrices used in their test campaigns. As for backbone networks Ebone and Exodus, we have personally generated two classes of traffic matrices with a nonzero demand between each pair of edge nodes:

1. *Constant* and *Poisson* matrices generated by means of the TOTEM toolbox (Leduc *et al.*, 2006). Being each traffic demand generated according to constant or Poisson distributions (the same for each demand), we consider the maximum matrices which can be supported by the fully active network with OSPF routing when all weights are equal to 1.
2. *LP multicommodity flow* matrices, which are obtained by scaling a maximum traffic matrix which represent the solution of a LP formulation aiming at maximizing the overall network throughput subject to lower and upper bounds constraints on the traffic value of each traffic demand.

For the sake of simplicity, in our tests we consider $\mu = 1$ (maximum utilization threshold).

5.4.2 Results

The computational experiments have been carried out on an Intel Pentium Duo 3.0GHz with 3.5GB of RAM. A first group of results is reported in Table 5.4 to evaluate the impact of the dual weight warm-start in term of overall performance. The extensive results obtained for backbone networks by running GA-ES, GRA-ES, TA-ES and MILP-EWO are then summarized in Tables 5.6, 5.7 and 5.8. To conclude, we report in Table 5.9 the results obtained by MILP-EWO with the Rocketfuel access network instances provided by (Vasić et Kostić, 2010; Zhang *et al.*, 2010), and compare in Figure 5.5 the energy savings achieved by MILP-EWO and those computed in (Lee *et al.*, 2012) with a state of the art method for SEANM-SP.

The first group of columns in Table 5.4, 5.6 and 5.9 describes the features of the considered network instances. The ID field is used to uniquely identify the corresponding instances, allowing in this way to save space in Tables 5.7 and 5.8, where instance features are not repeated for a second time. The Net field reports the pair network topology-traffic matrix of each instance, e.g., $Eb30$ states for Ebone network with and LP matrix responsible for maximum link utilization of 30%. Note that the abbreviations Ex, Eb, Spr, Abov, Gen and Tis, stand for, respectively, Exodus, Ebone, Sprint, Abovenet, Genuity and Tiscali. Placed after the network initials, letters C, P and numbers 30, 40, 50, 1, 2 and 3 represent the considered traffic matrix (see Table 5.5 for a detailed explanation of the abbreviations).

Table 5.4 Computational results: analysis of the impact of the dual weight warm start on GA-ES.

					GA-ES$_{no_warmstart}$				**GA-ES**			
ID	Net	$\|V^c\| - \|V^e\|$	$\|A\|$	$Cong_{min}$	$E_c\,(W)$	V_{off}	A_{off}	$Cong$	$E_c\,(W)$	V_{off}	A_{off}	$Cong$
1	Ex30	41-38	294	155052	5146.1	31	169	381870	5239.8	30	168	354534
2	Ex40	41-38	294	253240	5883.5	25	139	499842	5753.3	26	145	566605
3	Ex50	41-38	294	382600	6620.9	19	109	761000	6418.9	21	113	616497
4	ExC	41-38	294	147639	5130.3	30	183	388132	5058.5	31	181	386235
5	ExP	41-38	294	164764	5440.6	27	176	382164	5346.8	28	177	354508
6	Eb30	56-31	322	111265	5677.9	34	207	306940	5677.9	34	207	300163
7	Eb40	56-31	322	169811	6364.2	28	184	353066	6248.6	29	188	369041
8	Eb50	56-31	322	242613	7324.3	19	159	537422	7136.9	21	161	462120
9	EbC	56-31	322	179798	6616.1	25	185	310032	6313.1	28	191	379940
10	EbP	56-31	322	202439	6537.0	26	184	377949	6320.4	28	190	423698

Table 5.5 Notation used to represent traffic matrices in result tables.

Traffic matrix abbr.	Explanation	Type of network
30	LP maximum matrix scaled by 0.3	backbone
40	LP maximum matrix scaled by 0.4	backbone
50	LP maximum matrix scaled by 0.5	backbone
C	Maximum Constant distributed matrix	backbone
P	Maximum Poisson distributed matrix	backbone
1	Low traffic level matrix	access
2	Moderate traffic level matrix	access
3	High traffic level matrix	access

In addition, in a second block we provide the cardinality of the sets of both core and edge nodes $|V^c| - |V^e|$, the number of directed links $|A|$, the overall energy consumption of the original network $E_c^{tot}\,(W)$, and the measure of network congestion computed by IGP-WO on the complete network $Cong_{min}$. We then report the computational results concerning the energy consumption $E_c\,(W)$ of the optimized network, and the solution gap, i.e., *gap*, computed as the ratio $(E_c - E_c^b)/E_c^b$. E_c^b is the lower bound on the energy consumption obtained by solving to optimality the MILP formulation (5.21–5.27) of the *Feasible-routing stage* of TA-ES. Finally, we report the cost of network congestion $Cong$ (computed as in IGP-WO), the normalized congestion increase $Cong_{\%}$, i.e., the ratio $Cong/Cong_{min}$, the number of sleeping nodes V_{off}, that of sleeping links A_{off}, the computing times t and the normalized computing times t_{nor} (w.r.t. the resolution time of GA-ES).

Impact of weight optimization

Let us start the result analysis from Table 5.4, which reports the energy consumption achieved by both the original GA-ES and its variant GA-ES$_{no_warmstart}$, where IGP-WO is

Table 5.6 Computational results: comparison between the energy savings achieved by GA-ES, GRA-ES, TA-ES andMILP-EWO with backbone network instances.

Inst					**Bound**	**GA-ES**		**GRA-ES**		**TA-ES**		**MILP-EWO**	
ID	*Net*	$\|V^c\| - \|V^e\|$	$\|A\|$	$E_c^{tot}\,(W)$	$E_c\,(W)$	$E_c\,(W)$	*gap*	$E_c\,(W)$	*gap*	$E_c\,(W)$	*gap*	$E_c\,(W)$	*gap*
1	Ex30	41-38	294	9058.2	4546.2	5239.8	15.3%	5124.2	12.7%	4929.5	8.4%	4922.2	8.3%
2	Ex40	41-38	294	9058.2	5131.5	5753.3	12.1%	5738.7	11.8%	5342.0	4.1%	5399.2	5.2%
3	Ex50	41-38	294	9058.2	5536.7	6418.9	15.9%	6519.9	17.8%	6216.9	12.3%	6195.0	11.9%
4	ExC	41-38	294	9058.2	4537.7	5058.5	11.5%	4928.3	8.6%	4682.5	3.2%	4704.4	3.7%
5	ExP	41-38	294	9058.2	4653.3	5346.9	14.9%	5152.2	10.7%	4820.0	3.6%	4805.4	3.3%
6	Eb30	56-31	322	10126.6	5540.4	5677.9	2.5%	5670.6	2.4%	5670.6	2.4%	5569.6	0.5%
7	Eb40	56-31	322	10126.6	5872.6	6248.6	6.4%	6162.2	4.9%	6111.1	4.1%	6096.5	3.8%
8	Eb50	56-31	322	10126.6	6327.7	7136.9	12.8%	7107.7	12.3%	6689.1	5.7%	6667.2	5.4%
9	EbC	56-31	322	10126.6	5865.3	6313.1	7.6%	6298.5	7.4%	6002.8	2.3%	5931.0	1.1%
10	EbP	56-31	322	10126.6	5865.3	6320.4	7.8%	6226.7	6.2%	6096.5	3.9%	6010.1	2.5%

Table 5.7 Computational results: comparison between the number of sleeping elements which characterizes the solutions achieved by GA-ES, GRA-ES, TA-ES andMILP-EWO with backbone network instances.

Inst	**GA-ES**		**GRA-ES**		**TA-ES**		**MILP-EWO**	
ID	V_{off}	A_{off}	V_{off}	A_{off}	V_{off}	A_{off}	V_{off}	A_{off}
1	30 (73.2%)	168 (57.1%)	31 (75.6%)	172 (58.5%)	33 (80.5%)	175 (59.5%)	33 (80.5%)	176 (59.9%)
2	26 (63.4%)	145 (63.4%)	26 (63.4%)	147 (50.0%)	30 (73.2%)	154 (52.4%)	29 (70.7%)	158 (53.7%)
3	21 (51.2%)	113 (51.2%)	20 (48.8%)	111 (37.8%)	23 (56.1%)	117 (39.8%)	23 (56.1%)	120 (40.8%)
4	31 (75.6%)	181 (75.6%)	32 (78.0%)	187 (63.6%)	34 (82.9%)	197 (67.0%)	34 (82.9%)	194 (66.0%)
5	28 (68.3%)	177 (68.3%)	30 (73.2%)	180 (61.2%)	33 (80.5%)	190 (64.6%)	33 (80.5%)	192 (65.3%)
6	34 (60.7%)	207 (60.7%)	34 (60.7%)	208 (64.6%)	34 (60.7%)	208 (64.6%)	35 (62.5%)	210 (65.2%)
7	29 (51.8%)	188 (51.8%)	30 (53.6%)	188 (58.4%)	30 (53.6%)	195 (60.6%)	30 (53.6%)	197 (61.2%)
8	21 (37.5%)	161 (37.5%)	21 (37.5%)	165 (51.2%)	25 (44.6%)	175 (54.3%)	25 (44.6%)	178 (55.3%)
9	28 (50.0%)	191 (50.0%)	28 (50.0%)	193 (59.9%)	30 (53.6%)	198 (61.5%)	31 (55.4%)	198 (61.5%)
10	28 (50.0%)	190 (50.0%)	29 (51.8%)	191 (59.3%)	30 (53.6%)	197 (61.2%)	31 (55.4%)	197 (61.2%)

Table 5.8 Computational results: comparison between the computing times of GA-ES, GRA-ES, TA-ES andMILP-EWO with backbone network instances.

Inst	**GA-ES**		**GRA-ES**		**TA-ES**		**MILP-EWO**	
ID	*Cong*	$t\,(s)$ (t_{nor})	*Cong*	$t\,(s)$ (t_{nor})	*Cong*	$t\,(s)$ (t_{nor})	*Cong*	$t\,(s)$ (t_{nor})
1	229%	1159 (1)	264%	16224 (14.00)	318%	4732 (4.08)	300%	2189 (1.89)
2	224%	1146 (1)	239%	21152 (18.46)	263%	10509 (9.17)	352%	2024 (1.77)
3	161%	1252 (1)	162%	25323 (20.23)	174%	10020 (8.00)	171%	4983 (3.98)
4	262%	767 (1)	224%	14632 (19.08)	315%	1939 (2.53)	299%	1863 (2.43)
5	215%	957 (1)	185%	15329 (16.02)	243%	4369 (4.57)	253%	1798 (1.88)
6	270%	2217 (1)	300%	10121 (4.57)	281%	9062 (4.09)	302%	2674 (1.21)
7	217%	1897 (1)	194%	13053 (6.88)	269%	29808 (15.71)	279%	2606 (1.37)
8	190%	1818 (1)	182%	18218 (10.02)	202%	26303 (14.47)	213%	3663 (2.01)
9	211%	1624 (1)	196%	13075 (8.05)	241%	9701 (5.97)	280%	1868 (1.15)
10	209%	1434 (1)	201%	12163 (8.48)	242%	7657 (5.34)	209%	2648 (1.85)

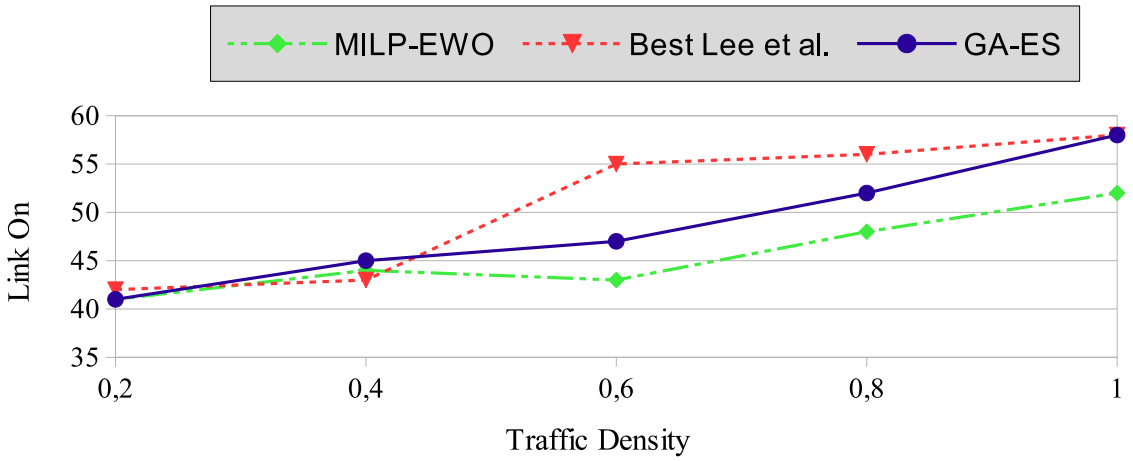


Figure 5.5 Comparison between the results obtained with our heuristics and the best algorithm proposed in (Lee *et al.*, 2012) for the USA28 network topology.

Table 5.9 Computational results: overall performance of MILP-EWO with the access network instances used in (Zhang *et al.*, 2010; Vasić et Kostić, 2010).

Instances used in (Zhang *et al.*, 2010) – Link capacities of 9920 *Gbps* or 2480 *Gbps*										
ID	*Net*	$\|V\|$	$\|A\|$	$E_c^{tot}(W)$	$Cong_{min}$	$E_c(W)$	V_{off}	A_{off}	*Cong*	$t(s)$
11	Spr1	52	168	24972	28785	11950 (47.9%)	0 (0.0%)	85 (50.6%)	160774 (558%)	578
12	Spr2	52	168	24972	59651	13339 (53.4%)	0 (0.0%)	76 (45.2%)	476098 (798%)	584
13	Spr3	52	168	24972	96214	13795 (55.2%)	0 (0.0%)	73 (43.5%)	412256 (428%)	1241
14	AT&T1	115	296	43344	38990	30504 (70.4%	0 (0.0%)	82 (27.7%)	215903 (553%)	1816
15	AT&T2	115	296	43344	77980	31026 (71.6%)	0 (0.0%)	79 (26.7%)	323802 (415%)	1854
16	AT&T3	115	296	43344	117347	32388 (74.7%)	0 (0.0%)	70 (23.6%)	616849 (525%)	1990
Instances used in (Vasić et Kostić, 2010) – Link capacities of 100 *Mbps* or 52 *Mbps*										
ID	*Net*	$\|V\|$	$\|A\|$	$E_c^{tot}(W)$	$Cong_{min}$	$E_c(W)$	V_{off}	A_{off}	*Cong*	$t(s)$
17	Abov1	19	68	7610	1079812	4910 (64%)	1 (5.0%)	35 (51.5%)	2458879 (228%)	59
18	Abov2	19	68	7610	1528712	5200 (68%)	0 (0.0%)	33 (48.5%)	3816638 (250%)	42
19	Abov3	19	68	7610	2118525	5580 (73%)	0 (0.0%)	29 (42.6%)	4659640 (219%)	73
20	AT&T1	115	296	37970	2251536	23130 (61%)	35 (30.4%)	137 (46.3%)	5388692 (240%)	662
21	AT&T2	115	296	37970	279125	23060 (60%)	35 (30.4%)	138 (46.6%)	6069666 (217%)	356
22	AT&T3	115	296	37970	3563969	23270 (61%)	35 (30.4%)	135 (45.6%)	5771727 (162%)	419
23	Gen1	42	110	14000	1841491	10670 (76%)	4 (9.5%)	39 (35.5%)	4960005 (270%)	182
24	Gen2	42	110	14000	2446328	10810 (77%)	4 (9.5%)	37 (33.6%)	4420327 (180%)	251
25	Gen3	42	110	14000	2972444	10810 (77%)	4 (9.5%)	37 (33.6%)	5691743 (191%)	115
26	Spr1	52	168	19560	2691568	13430 (68%)	4 (7.7%)	79 (47.0%)	7724987 (287%)	528
27	Spr2	52	168	19560	3351249	13780 (70%)	4 (7.7%)	74 (44.0%)	8319724 (248%)	298
28	Spr3	52	168	19560	4544579	14130 (72%)	4 (7.7%)	69 (41.1%)	4960005 (175%)	968
29	Tis1	41	174	18330	2534503	11660 (64%)	2 (4.9%)	91 (52.3%)	6459629 (255%)	368
30	Tis2	41	174	18330	2621918	11520 (63%)	2 (4.9%)	93 (53.4%)	4808477 (183%)	270
31	Tis3	41	174	18330	3426815	11590 (63%)	2 (4.9%)	92 (52.9%)	6334003 (184%)	346

run with random weight initialization. Results assess the positive impact of warm-starting IGP-WO with the dual weights: GA-ES outperforms GA-ES$_{no_warmstart}$ in nine cases out of ten. These data provide thus a clear insight on the importance of optimizing link weights

to improve the ability of the greedy routine to put to sleep network elements. Practically speaking, better are the weights in term of load balancing, higher are the possibilities for the greedy procedure to distribute the traffic of the sleeping link among an higher number of paths (if we use many paths it is more unlikely to incur in path saturation). Remind that enabling ECMP is thus crucial to increase load balancing.

Energy Consumption vs. Algorithm Complexity

Overall results on Exodus and Ebone networks are reported in Tables 5.6, 5.7,5.8 and Figures 5.7, 5.8 and 5.9.

We immediately note that all the four methods allow to substantially reduce network consumption (around 40%) also in case of moderated levels of traffic. As expected, GA-ES, which is the less time consuming approach, obtains the worse performance in terms of energy savings. Its GRASP evolution, i.e., GRA-ES, shows a slight energy saving improvement which is highly paid in terms of computing times: due to the multi-start architecture, according to which the greedy routine is repeated 100 times, the normalized times t_{nor} grow up to reach even 20.

Completely opposite insights are given by TA-ES results. In this case, the substantial higher complexity (in the worst cases $t_{nor} = 15$), brings clear power reduction improvements. Saving levels are typically 5% larger than in GRA-ES, with a gap from the lower bound smaller than 6% in 8 cases over 10. However, note that in some cases, the elevated computing times (up to 8 hours in instance 7) may represent a problem in term of algorithm scalability.

The scalability issues is effectively addressed by MILP-EWO, which seems to achieve the desired trade-off in terms of both power reduction and overall complexity. Compared to TA-ES, computing times are substantially smaller (t_{nor} smaller than 2 in eight instances over ten), while the saving levels are pretty much the same (except for instance 2). In absolute terms, computing times are generally less than half an hour: note that the amount of time required to solve the MILP formulation typically accounts for only half of the overall algorithm time. MILP-EWO might be thus exploited to to cope with network with up to 300 nodes and 600 links.

Network Congestion

Another important element to be evaluated is the congestion level of the energy-aware solution: all the four proposed algorithm produce an acceptable increase of the cost of link utilization (typically doubled or tripled), whose normalized value (see Fortz et Thorup, 2002)) does not usually exceed 0.3.

A visual example of the reduced network topologies computed by MILP-EWO for the Exodus network is shown in Figure 5.6. We show the solutions corresponding to three different instances, namely *Ex30*, *Ex40*, *Ex50*. As expected, a traffic decrease is translated into the possibility of putting to sleep a larger number of network elements.

Instances of Other Researchers

Instances from (Zhang et al., 2010; Vasić et Kostić, 2010)

Being clear from previous results that MILP-EWO is the most performing method, we have further tested it with the access network instances used in (Zhang *et al.*, 2010; Vasić et Kostić, 2010). Also in this case, with heterogeneous link capacities and no edge nodes, the algorithm performs very well, being able to power off up to 55% of links in some instances. Note that the saving levels achievable are strongly related to the link redundancy of the considered network topology. In the AT&T network, which is plenty of nodes with only one incident link, the power reduction is in fact about half w.r.t that achieved for Sprint network instances. Also in this case the computing times are in the order of 30 minutes. Note that some edge nodes of the instances used in (Vasić et Kostić, 2010) are put to sleep because traffic matrices do not contain traffic demands between all the possible origin-destination pairs. For what concerns the congestion levels, in a few instances, e.g., Spr2, the saturation increase is much larger w.r.t. backbone network instance (up to 8 times), but normalized values steadily remains below 0.4.

Instances from (Lee et al., 2012)

To conclude, we prove the validity of our approaches for energy-aware link weight optimization by comparing the performance of our methods with that of the best heuristic proposed in (Lee *et al.*, 2012). The latter is called Lagrangian Relaxation plus Harmonic Series (LR&HS) heuristic (Lee *et al.*, 2012) and aims at putting to sleep network links while respecting maximum link utilization constraints. In Figure 5.5 we plot the number of active links with respect to the traffic density (load level). For each traffic density level, i.e., 0.2, 0.4, 0.6, 0.8 and 1, we consider ten different randomly generated traffic matrices and report only the final average value of active links (the source code to generate the traffic matrices has been provided by the authors of Lee *et al.*, 2012). We observe that as soon as the traffic increases up to a non negligible level (when the final solution cannot be a simple tree), MILP-EWO clearly outperforms LR&HS by putting to sleep up to 20% more links. Also GA-ES, despite of its very low complexity, seems to compete very well with LR&HS, with up to 10% of more sleeping links for traffic density equal to 0.6.

5.5 Conclusions

In this chapter we have addressed the challenging problem of optimizing both energy consumption and network congestion in IP networks operated with a shortest path routing protocol such as OSPF. After formalizing an exact MILP formulation to model the problem, we have proposed four different heuristic algorithms, i.e., GA-ES, GRA-ES, TA-ES and MILP-EWO, to handle realistic network instances up to 150 nodes.

The computational results for eight real network topologies and different types of traffic matrices, some generated by us and other provided by other researchers, show that energy savings up to 50% are made possible if link weights are adjusted in an energy-aware fashion. In the meantime, the proposed procedures guarantee that congestion is kept under control even when a large portion of the network has been put to sleep.

Among our four algorithms, MILP-EWO proved to constitute the best trade-off between energy savings and computational complexity: the largest solutions (about 100 nodes) were solved in less than one hour while obtaining a small gap with respect to the best available lower bound.

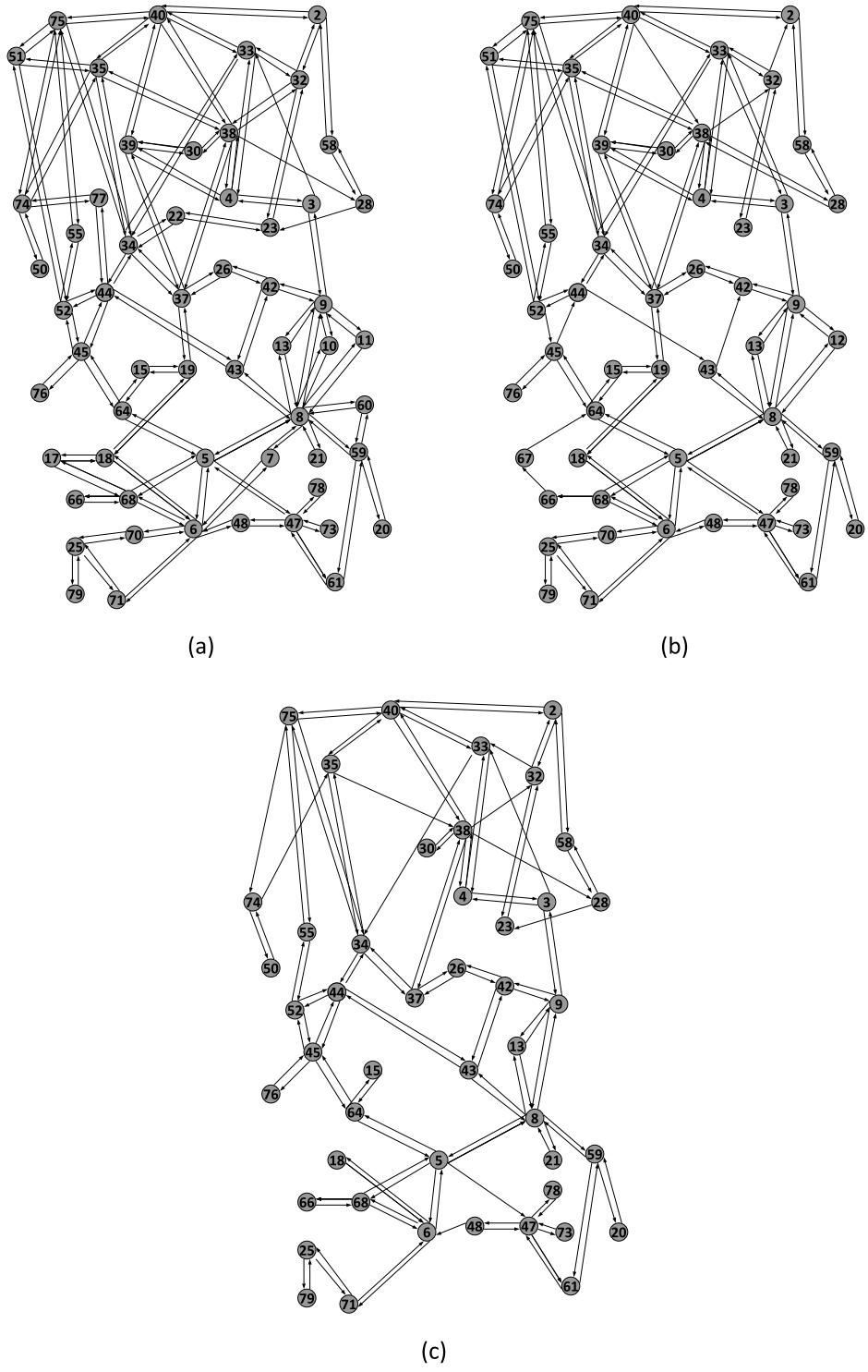


Figure 5.6 Visual representation of the sub-networks obtained with MILP-EWO by considering Exodus network with the LP traffic matrix scaled by (a) 0.5, (b) 0.4 and (c) 0.3.

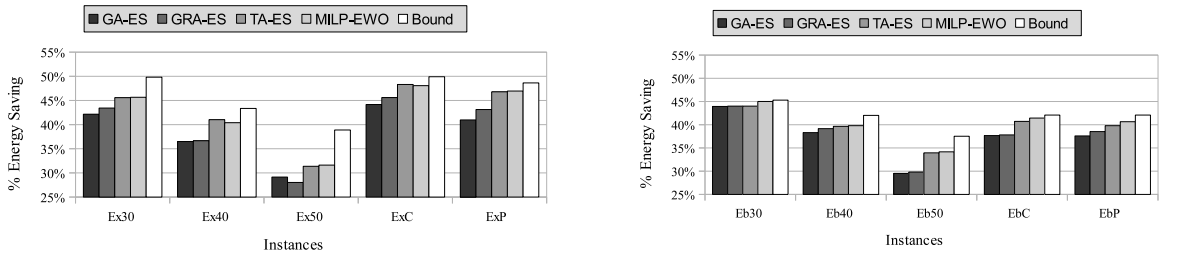


Figure 5.7 Computational results: percentage of energy saving achieved by the four proposed heuristics (GA-ES, GRA-ES, TA-ES and MILP-EWO) with Exodus and Ebone networks.

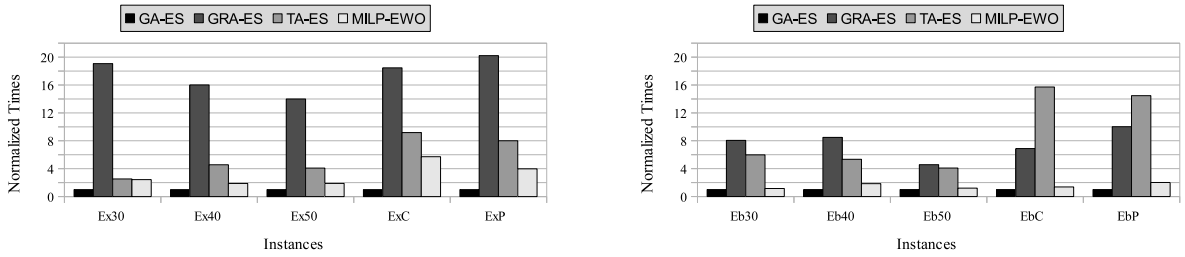


Figure 5.8 Computational results: computing times for the four proposed heuristics (GA-ES, GRA-ES, TA-ES and MILP-EWO) with Exodus and Ebone networks.

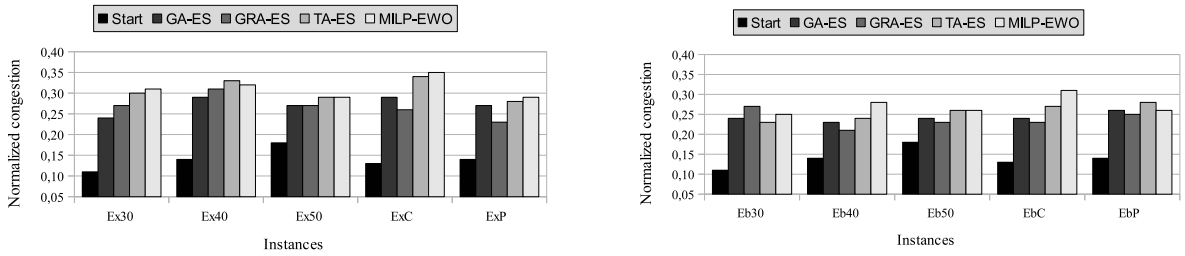


Figure 5.9 Computational results: normalized congestion values for the four proposed heuristics (GA-ES, GRA-ES, TA-ES and MILP-EWO) with Exodus and Ebone networks.

CHAPTER 6

Dynamic SEANM with Shortest Path Routing

In Chapter 5 we proposed different approaches for centralized off-line SEANM-SP. More specifically we focused on the following optimization problem: given an IP network operated with OSPF and ECMP, and a predicted traffic matrix, the aim is to find the optimal set of energy-aware link weights which lexicographically minimizes, in the following order, energy consumption and a measure of network congestion.

The main drawbacks concerning our off-line approach are that:

1. It strongly relies on the accuracy of the traffic matrices.
2. All the operations, such as the OSPF configuration switching, have to be planned in advance.
3. No on-line optimization can be directly performed to react to unexpected events, like link failures or peaks of traffic.

To overcome this issues, we propose a novel centralized approach for SEANM-SP which combines off-line and on-line optimization to guarantee both network stability and network responsiveness. Our aim is to dynamically adjust the OSPF link weights according to real-time traffic measurements collected in a distributed manner by the network routers, and successively sent to the centralized network controller which is responsible for efficiently managing the network configuration.

The link weights are not adjusted in total freedom by the network controller, but are chosen from a precomputed set of OSPF link weight configurations which are derived off-line during the planning phase by means of one of the algorithms presented in Chapter 5. OSPF configurations (each one corresponds to a set $|A|$ link weights) are applied dynamically according to a switching policy determined by the network operator.

Our management strategy has been practically implemented in a realistic network scenario by means of a novel open-source management framework called java-based network management platform (JNetMan). Along with several useful functions to efficiently manage the underlying network infrastructure, JNetMan offers a high level application program interface (API) to collect load measurements, adjust the link weights and perform any desired management operation through the well known SNMP protocol. Note that all the low level operations required to accomplish the desired routines are executed by JNetMan in a completely transparent way.

The proposed framework has been tested on different IP network topologies realized through an emulated network environment based on virtual machines. In the current version of the framework we do not explicitly manage the way network devices are practically put to sleep, but assume that sleeping activation and deactivation is autonomously managed by some centralized or distributed mechanisms able to detect the absence or presence of traffic on the considered device.

The reminder of the chapter is organized as follows. Our novel idea for joint off-line and on-line SEANM-SP is discussed in Section 6.1, detailed information on the architecture of our management framework is given in Section 6.2 and, to conclude, the most relevant computational results are presented and analyzed in Section 6.3.

6.1 Dynamic Energy-Aware OSPF Optimization

In the previous chapter, it has been shown that optimizing the OSPF link weights represents an effective strategy to adapt network power consumption to traffic levels without negatively affecting network performance.

In Figure 6.1, we illustrate our general idea to make SEANM-SP more agile, i.e., able to dynamically react to real-time events, while preserving stability and effectiveness typical of off-line planning operation. Our aim is to develop a management application for a centralized network management platform to dynamically select, from a closed precomputed set, the OSPF configuration (OC) that mostly fit the observed traffic levels. The OSPF configurations are computed off-line during the planning phase by considering a limited number of predicted traffic matrices. Each matrix ideally represents a specific level of traffic and the entire set of traffic matrices can be seen as a smart sampling of the daily traffic profile typically observed by the network operator (see Figure 6.2). Being the daily profile of Internet traffic typically characterized by a periodic behaviour and a very slow dynamic (Bolla *et al.*, 2012; Mackarel *et al.*, 2011), traffic matrices might be ideally sampled by splitting a single day into a few time intervals, and subsequently computing the matrices that best represent the overall traffic load of each period.

Let us look to the trivial example of Figure 6.1: the management platform first computes, and successively applies, two OSPF configurations corresponding to, respectively, the lowest and the highest level of traffic. The choice of switching the OSPF configuration is made in real-time by the management application according to a pre-defined policy properly designed by the network operator. The switching policy, whose aim is to adapt network configuration to traffic levels while avoiding frequent oscillations between different sets of link weights, may depend on several factors, e.g., maximum or average link utilization.

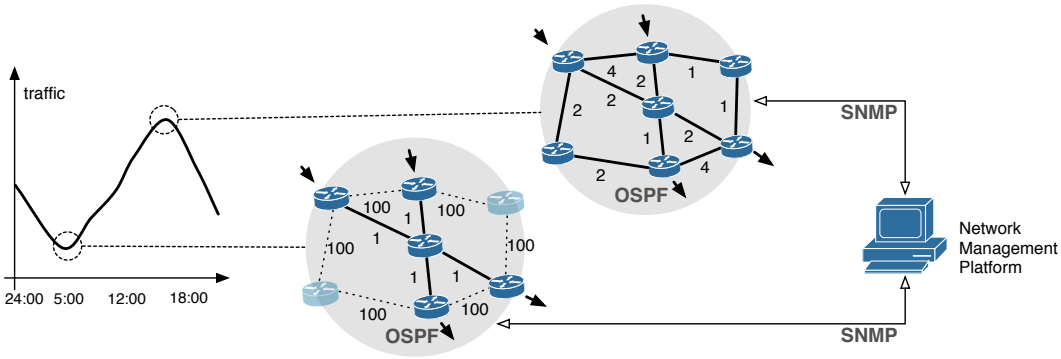


Figure 6.1 Our approach in brief: the network management platform dynamically adapts the network topology to the current traffic conditions by adjusting the OSPF link weights. SNMP is used to monitor the traffic on each link and practically modify the link weights.

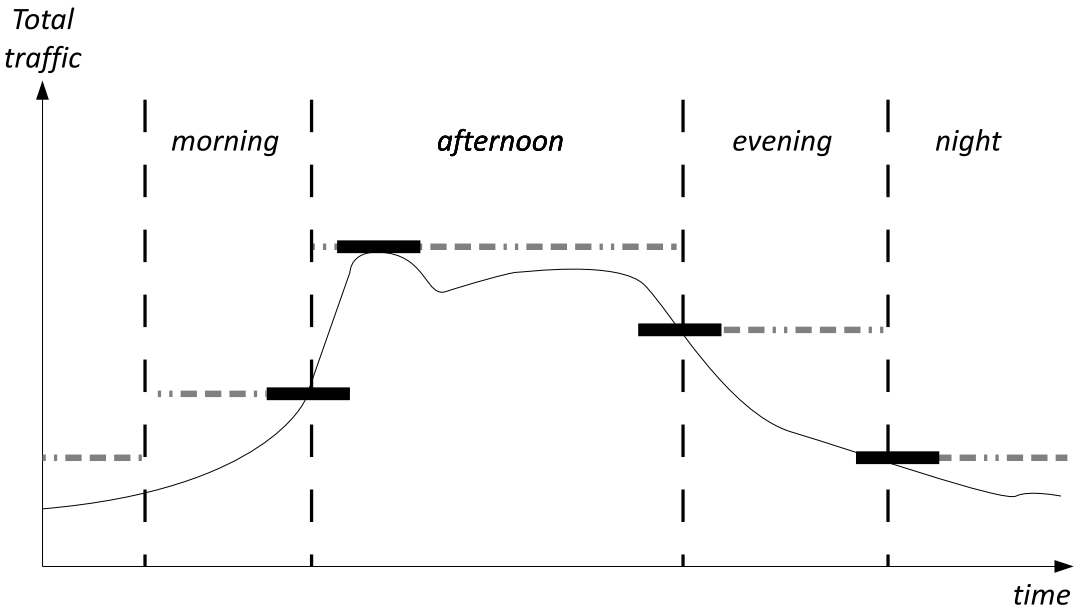


Figure 6.2 How to derive traffic matrices from a daily traffic profile: each bold black line represents an instant chosen to sample a traffic matrix.

In normal conditions, an OSPF configuration is meant to be ideally maintained for at least a few hours. As shown in (Chiaraviglio *et al.*, 2013a), quasi-optimal energy savings can be obtained through a limited set of network configurations to be updated every few hours. However, note that the number of traffic patterns, and consequently the switching frequency are strongly related to both the network scenario and the traffic figures.

From the practical point of view, all the low level management operations, such as measurement collection or link weight adjustments, are carried out by means of the very popular management protocol called SNMP, which is typically supported and implemented by all types of network devices.

Our novel management platform for dynamic and energy-aware management of OSPF weights is logically composed of three main building blocks (see Figure 6.3): (i) an off-line weight optimization algorithm, e.g., MILP-EWO, to compute in advance a limited set of OSPF configurations, (ii) a centralized network controller called energy-aware network intelligence (EANI) which is responsible for periodically monitoring network conditions and dynamically switching the OSPF configurations, and (iii) JNetMan, to practically implement all the instructions received by the controller (Cascone, 2013).

This novel management platform is exploited to integrate an off-line SEANM-SP algorithm such as MILP-EWO in a more sophisticated architecture specifically designed to overcome those issues inherent to off-line management approaches. The effective combination of off-line planning and real-time adaptation is meant to guarantee both network stability and network responsiveness and offer to network administrators a certain degree of control on the network operations.

Let us highlight the most important aspects of our novel platform. During the planning stage, besides the performance of the SEANM-SP algorithm, we need to pay great attention to traffic profile sampling and traffic matrix estimation, which strongly affect quality and applicability of OSPF configurations. Note that traffic matrices can be accurately predicted by means of both direct (Cisco Systems, 2012) and indirect measurements (Casas *et al.*, 2009).

The restricted set of OSPF configurations returned by MILP-EWO are then managed on-line by the network controller EANI, which in turn exploits the functions of the management framework JNetMan to perform the operations at the network level.

The architecture of the EANI, which is illustrated in Figure 6.4, presents four main operational blocks, i.e., (i) the *network monitor*, (ii) the *OC monitor*, the (iii) *OCs database*, and (iv) the *OC enabler*. The role of the *network monitor* is to periodically instruct the underlying management framework, i.e., JNetMan, to collect the utilization values observed on the connected link interfaces by each router. These data are then organized in the so-called

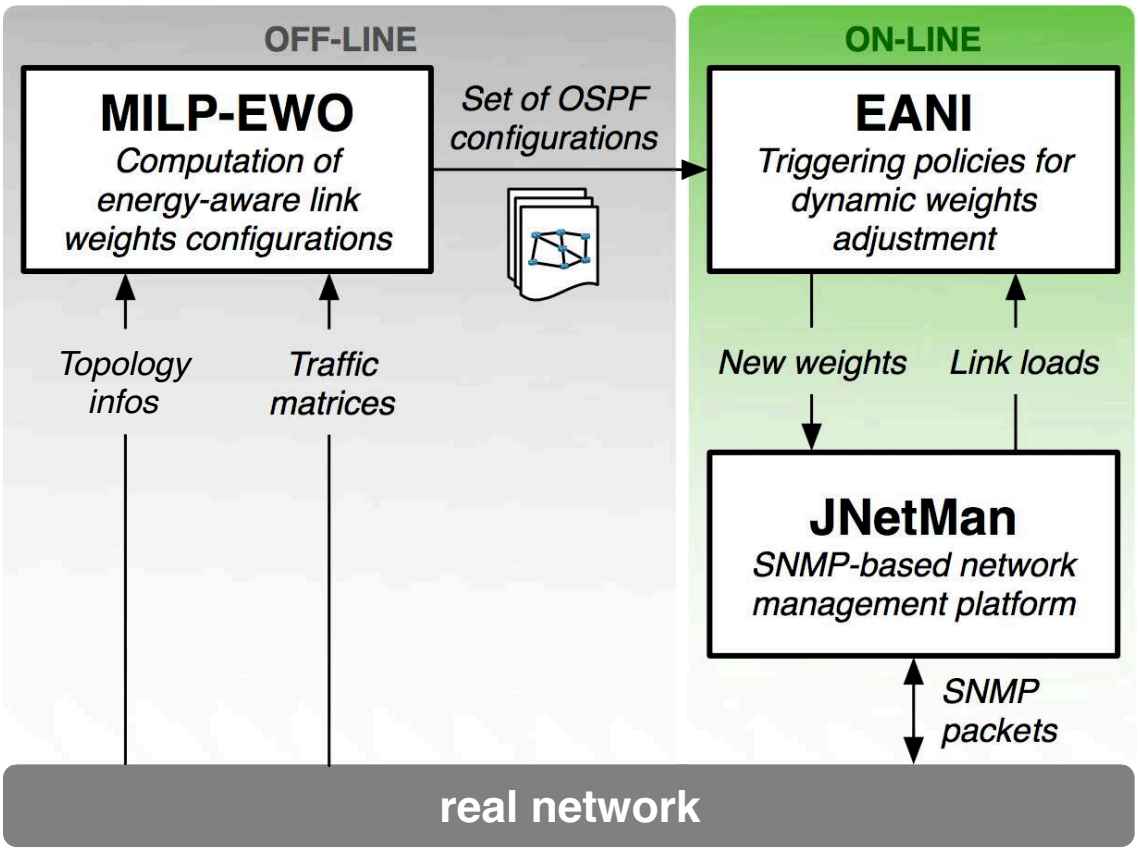


Figure 6.3 Flow chart of our network management framework.

bandwidth utilization report and passed by the *network monitor* to the *OC monitor*.

The *bandwidth utilization report* is analyzed by the *OC monitor*, which evaluates if current network conditions meet or not the desired requirements. If not, the *OC monitor* consults the "switching" policy configured by the network administrator and selects the new OSPF configuration from the *OCs database*. A complete switching policy should determine under which circumstances the current OSPF configuration must be updated, and which novel configuration stored in the *OCs database* must be applied.

The main role of the switching policy is to guarantee that the underlying network is neither congested nor underutilized. With this aim in mind, along our experimentation we opted for a switching policy defined by two average utilization thresholds ψ_{av}^{+} and ψ_{av}^{-}, and two maximum utilization thresholds ψ_{max}^{+} and ψ_{max}^{-}. The minuses and pluses used as apexes denote if a threshold should be considered to apply a more or less consuming OSPF configuration, respectively. The correct choice of the threshold values is crucial to prevent the management platform from continuously oscillating between different *OSPF configurations* in presence of normal conditions. Further details on switching policies are given in Section

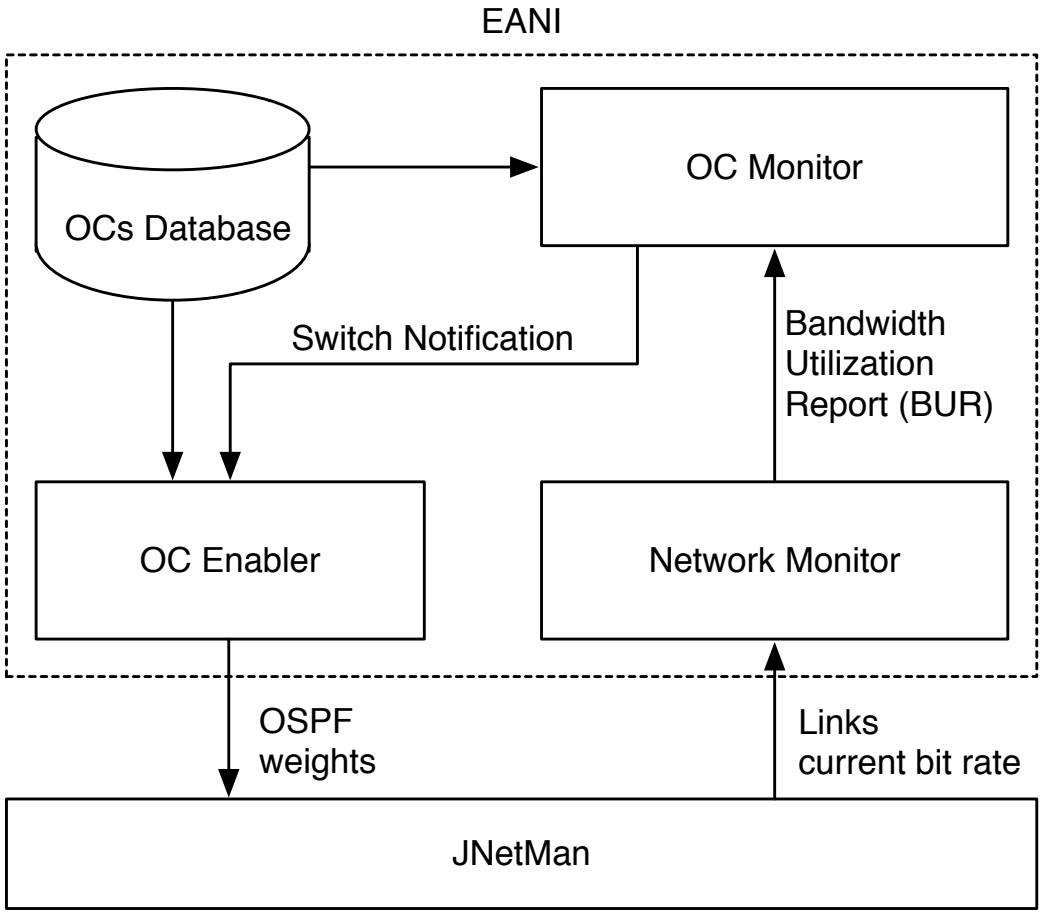


Figure 6.4 Architecture of EANI.

6.1.1.

Each *OSPF configuration* stored in the *OCs database* consists into an ASCII file containing the following information: (i) the OSPF weight of each link, (ii) the values of the four utilization thresholds which must be respected by the *OSPF configuration* itself, and (iii) the pointers stating which alternative *OSPF configuration* must be applied when utilization thresholds are exceeded (pointers define the switching policy).

A configuration switching involves the following operations: the switching request is initialized by the *OC monitor*, which sends to the *OC enabler* the *switch notification*. This includes the pointer to the novel OSPF configuration that will replace the current one. Then, the *OC enabler* interrogates the *OCs database* to obtain the ASCII file of the desired OSPF configuration and initializes a OSPF link weight update by means of the APIs of JNetMan.

It is worth pointing out that in large Autonomous Systems (ASs) composed by hundreds of routers and line cards, the link weight update may require a significant amount of computing power and the network itself may need an amount of time in the order of tens of seconds to

converge to a stable routing configuration. To speed up convergence, links whose weights are kept unchanged are ignored during the update process. With the aim of avoiding undesired temporary routing loops which may cause congestion and packet losses, we instruct the *OC enabler* to adjust the link weights one by one, with a changing frequency which is kept low enough so that each router has enough time to recalculate its routing table before a new weight is modified.

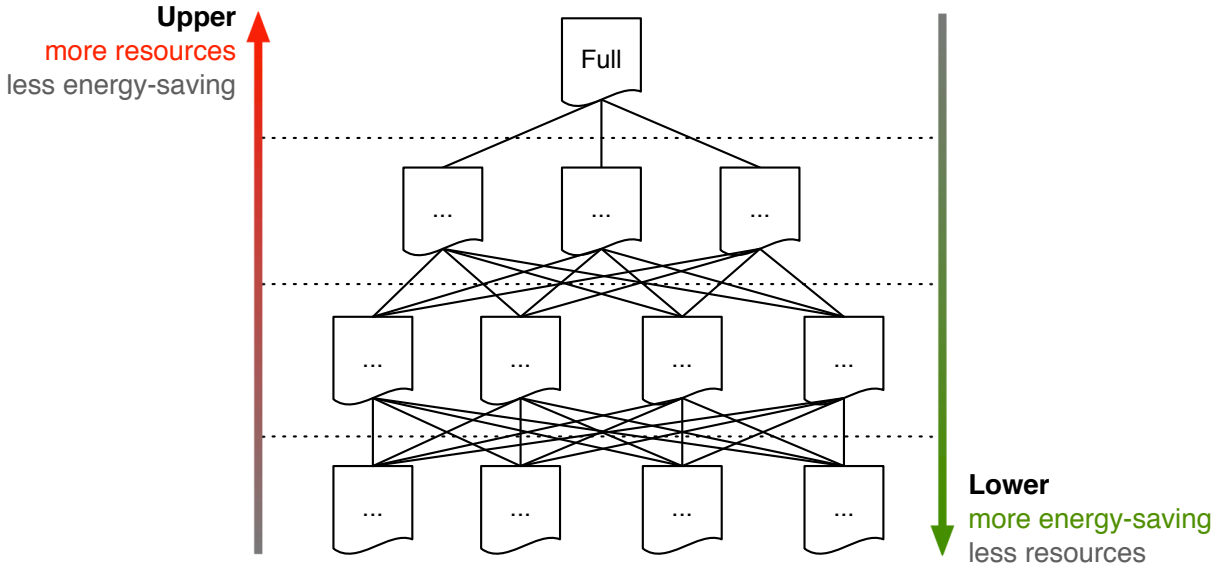


Figure 6.5 Example of OCs graph: OCs are sorted according to the amount of active network elements. "Full" represents the complete topology, with only active elements. Movements to different upper/lower OCs are made according to the observed traffic variations.

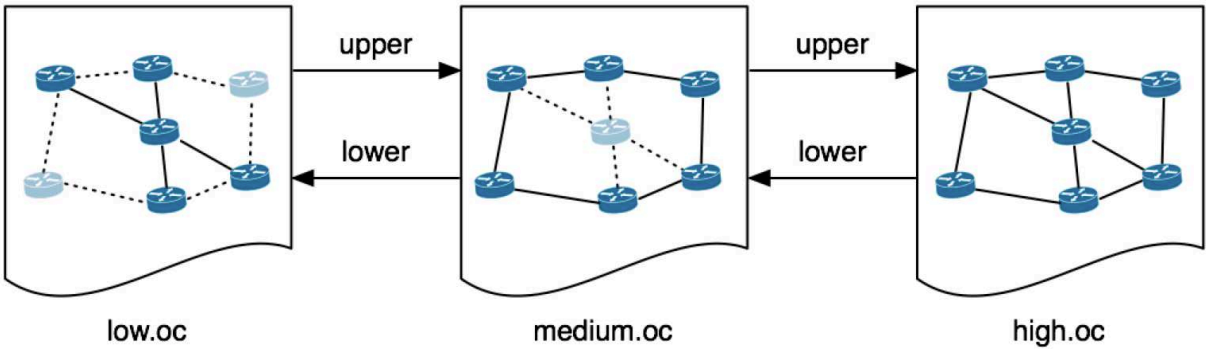


Figure 6.6 Example of *OCs chain*, which is a particular type of OCs graph. In this case, the most powerful configuration, i.e., `high.oc`, represents the full topology, while configuration `low.oc` embodies the less consuming topology.

6.1.1 OSPF Switching Policy

Besides effectiveness and accuracy of *OSPF configurations*, the crucial element of our novel energy-aware management approach is represented by the switching policy, which tells the management platform what to do with the *OSPF configurations* themselves.

The switching policy we have designed is based on the reasonable assumption that the higher the load of a traffic matrix, the higher the power consumption of the resulting OSPF configuration.

Under this assumption, a switching policy can be represented as a hierarchical structure wherein each OSPF configuration takes place according to its corresponding consumption level. A the top we find the full consumption configuration, while in the lowest part there are those configurations characterized by the minimal power consumption (see Figure 6.5). Arcs connecting different configurations are the visual representation of the *pointers*, which tell EANI which configuration switching is allowed when a specific condition is verified.

Motivated by the observation that in typical IP networks traffic load variations tend to preserve a constant ratio between different traffic demands (both absolute amounts of traffic and relative variations are strictly related to the number of clients served by the considered pair of nodes), we believe that a basic but effective policy can be defined by means of upper and lower thresholds on the maximum link utilization μ_{max}, and the average link utilization μ_{av}. In particular, in our experiments we use the following switching policy: an upper configuration (more consuming) is applied by means of the so-called *upper pointer* when $(\mu_{max} \geq \psi^{+}_{max}) \vee (\mu_{av} \geq \psi^{+}_{av})$, while a lower configuration (less consuming) is implemented through a *lower pointer* when $(\mu_{max} \leq \psi^{-}_{max}) \wedge (\mu_{av} \leq \psi^{-}_{av})$ (see Figure 6.7). Note that, according to this policy, each OSPF configuration is connected to both a more (upper pointer) and a less (lower pointer) consuming configuration. As shown in Figure 6.6, the general graph of Figure 6.5 is reduced to a simple *OCs chain*.

Note that, since in some occasions traffic demands may vary independently, i.e., some demands may vary much more than others, congestion could be observed only in some specific portions of the networks. To cope with this, in addition to traditional upper and lower pointers, there exists a second class of pointers defined as *link-specific*. These pointers introduce a finer level of granularity in terms of network conditions, and allow the management platform to better tailor the network configuration to the observed traffic scenario.

6.2 Java Framework for SNMP-based Network Management Applications

Our novel management platform completely relies on SNMP to practically perform the management operations requested by the controller. To correctly perform a management

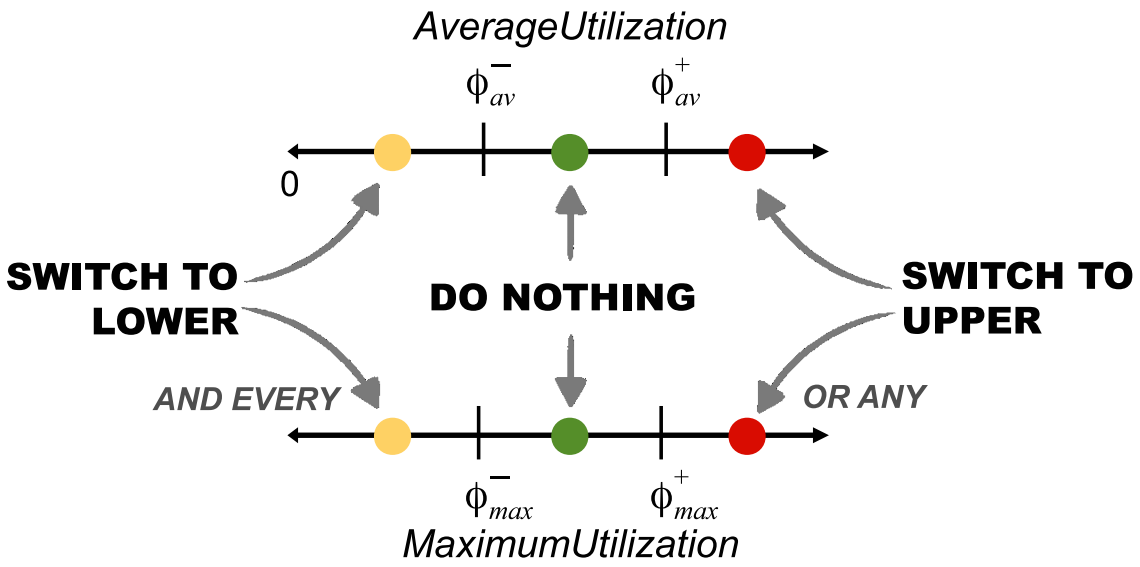


Figure 6.7 Switching policy.

task with SNMP, e.g., retrieving the load value of a specific link, it is necessary to explicitly manage a large set of complex low level operations, such as authentication, identification and management of the SNMP agent of each network element, retrieve of the object identifier (OID) which identifies the desired parameter inside the management information base (MIB) of a specific device, packet generation, time-out triggering, cast between different data types, and so on.

To avoid the necessity of explicitly managing all this complex operations whenever a new management task has to be implemented, we developed JNetMan, our novel open source Java-based framework for network management (Cascone, 2013). In its current version, JNetMan supports several SNMP operations, and by means of an extensive set of powerful APIs (primitives), offers a clear separation between the low level SNMP instructions and the specific high level management strategy. JNetMan, which is responsible for performing in a transparent manner all the low level management tasks required to successfully accomplish a management instruction, offers the user the possibility of implementing the desired management application by naturally exploiting a high-level abstraction of the network and a set of defined management methods.

JNetMan can be seen as a reusable software platform for rapid development, deployment and testing of novel management applications for IP networks. We show its architecture in Figure 6.8. Three main operational layers are defined, i.e., (i) the SNMP plug-in, (ii) the Topology Abstraction Manager and (iii) the set of Aspect Managers. Each Aspect Manager consists into a specific set of high level libraries destined to manage a particular aspect of

the network. Finally, note than on the top of JNetMan there is the management application, which exploits the APIs to perform the required management tasks.

The SNMP plug-in is composed by four different logical modules. At the lowest level stands the the communication interface, which is largely built on an open-source SNMP implementation for Java called SNMP4J SNMP4J.org (2003), and whose aim is to practically implement the low-level SNMP instructions required by a specific high level management operation, as for instance, setting up a SNMP communication channel between the management host and the SNMP agent of each network device. On the top of the communication layer, reside the SNMP client, the MIB helper and the structure of management information (SMI) helper. The first represents the entity which communicates with the SNMP agent of each network device. The last two are instead auxiliary modules aimed at speeding up the the whole management of SNMP operations by providing additional services such as, MIB data processing and Java-MIB format conversion in both the two directions (MIB formats are defined in the SMI).

Over the SNMP plug-in we put the topology abstraction manager (TAM), whose role is to offer a simplified and logical abstraction of the underlying network topology. To carry out this task, the TAM performs a mapping between the SNMP network abstraction, which is built by interconnecting all the SNMP agents, and a more natural and easily manageable network topology representation composed of the following network entities: *nodes*, *interfaces*, and *links*. Each *link* represents a connection between two *interfaces*, each one installed on a different *node*.

The entities which form the network topologies and their related attributes are defined by means of ad hoc methods provided by the TAM. Possible attributes include the interface ID, the nominal link transmitting rate and the interface IP address.

In Figure 6.10, we report the code required to define the simple two-node network topology illustrated in Figure 6.9 by using the APIs of the TAM. The TAM methods automatically organize the information into a high-level abstraction of the network topology, making it easier for the layers above, in our case for the Aspects Managers, to control each single network element by means of the network entities and manage the whole underlying network.

The network topology can be manually defined through plain text files, or automatically built by the management platform thanks to an auto-discovery module properly designed to independently collect topology information by directly interrogating the network devices.

Finally, at the top level of JNetMan we find the Aspect Managers. Each Aspect Manager is identified as an autonomous module and contains a library of high-level APIs which allow the user to interact with a specific aspect of the underlying network. Management information concerning different subsystems, e.g., the routing protocol or the power state, is collected

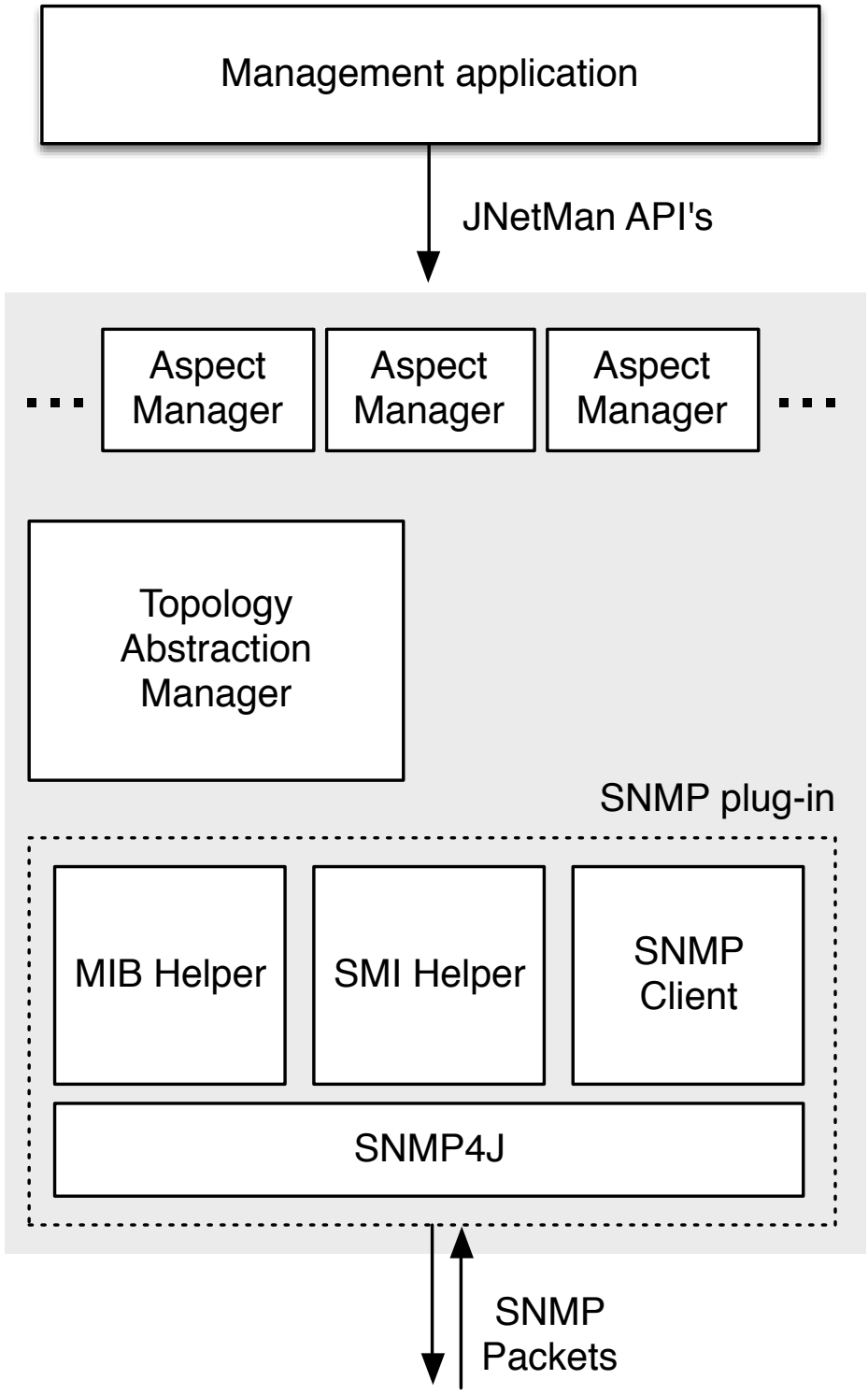


Figure 6.8 Architecture of JNetMan.

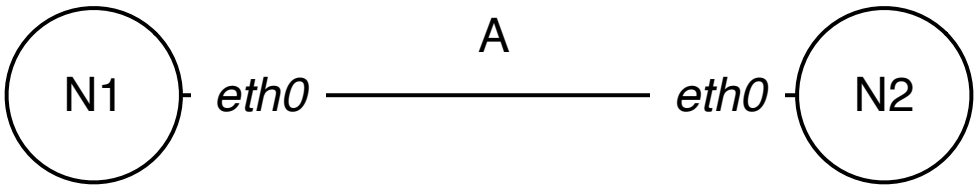


Figure 6.9 Example of network with 2 nodes and 1 link.

```
// First create a network object
Network network = new Network();
// Add two nodes, namely N1 and N2
Node n1 = network.createNode("N1");
Node n2 = network.createNode("N2");
// Add one interface per each node, named eth0
IfCard n1eth0 = n1.createInterfaceCard("eth0");
IfCard n2eth0 = n2.createInterfaceCard("eth0");
// Add one link, named A, and define it as connection
// between N1.eth0 and N2.eth0
Link linkA = network.createLink("A", n1eth0, n2eth0);
// Set the actual IP address of both nodes.
// This is used for SNMP communications.
n1.setIPAddress("172.16.10.1");
n2.setIPAddress("172.16.10.2");
```

Figure 6.10 Procedure to define the network topology of Figure 6.9 through TAM's APIs.

among multiple hierarchic databases called MIBs, which are maintained by each SNMP agent.

The APIs offered by the Aspect Mangers represent the interface between the management framework (JNetMan) and the top level management application. The desired management routine can be implemented by effectively combining the APIs themselves. In this way all the underlying low-level operations are executed in a transparent way w.r.t. to the user.

An example concerning the use of an API to retrieve the average bit rate of a link over a fixed time interval is given in Figure 6.11. Note that the *getAvgBitrate* primitive belonging to the *Monitoring* Aspect Manager exploits the *link* entity defined by the TAM.

The Aspect Managers are further split among *Basic Managers*, i.e *Monitoring Manager*, *Interface Manager*, *IP Manager*, and *Advanced Managers* like the *OSPF Manager*. The firsts offer primitives to control basic configuration elements typically common to all network devices, while the latter focus on the management of more sophisticated domains, such as

```
/*
 * We want to measure and print the average
 * bit rate (bit/s) of each link defined in
 * the network over an interval of 5000 ms.
 */
for (Link link : network.getLinks()) {
  long br = Monitoring.getAvgBitrate(link, 5000);
  // Print the result, e.g. "linkA: 638000 bit/s"
  System.out.printf("%s: %d bit/s%n", link.getName(), br);
}
```

Figure 6.11 Procedure to retrieve the link load through the APIs of the *Monitoring* aspect manager.

routing or signaling protocols.

As for the particular case of EANI, the *Monitoring Manager* and the *OSPF Manager* are exploited, respectively, to collect link load measurements and update the link weights.

6.3 Computational Results

6.3.1 Test Instances

To test our management platform in a realistic network scenario we built an emulated network environment by means of Netkit (Rimondini, 2007) (Pizzonia et Rimondini, 2008), an open-source platform based on User-Mode-Linux which allows to initialize and interconnect multiple virtual routers. To integrate the emulated environment with the required functionalities, we used other networking modules like the OSPF routing daemon Quagga (Jakma *et al.*, 2012), the SNMP agent Net-Snmp (NET-SNMP, s.d.), and the Distributed Internet Traffic Generator (D-ITG) (Botta *et al.*, 2012).

We conducted experiments on two network topologies, `polska` and `abilene`, provided by the widely known SNDLib (Orlowski *et al.*, 2010) and whose features are reported in Table 6.1). The splitting among edge nodes V^e and core nodes V^c has been manually done by us so as to equally distribute them in each portion of the corresponding network. Due to

Table 6.1 Network topologies from SNDLib (Orlowski *et al.*, 2010) used in our tests.

Network	**Nodes**	**Links**	**Edge**$_{nodes}$	**Core**$_{nodes}$
abilene	11	14	6	5
polska	12	18	6	6

constraints on the available computing power, we were not able to implement larger network topologies with Netkit.

The considered traffic matrices were derived from those provided by the SNDLib: we first removed all the traffic demands involving core nodes, and then scaled the resulting matrix with the largest parameter (the same for each demand) which allowed to successfully route all demands with fully splittable routing while not violating a 70% maximum utilization threshold. The parameter was computed by means of a LP formulation.

The peak traffic matrix of each network was used as reference for the traffic generator to simulate the simple, but also quite realistic *fade-in fade-out* profile shown in Figure 6.12. The peak values are those described in the peak traffic matrix. To add the uncertainty typical of Internet traffic we generated packets by varying both inter departure time and packet size according to some normal distributions centered on the average values required to reproduce the desired rate (traffic matrix). To experiment with different degrees of variability, we considered variances of 10%, 20% and 30% for both inter departure time and packet size.

Before running the management application on our platform we computed off-line the OSPF configuration. The energy and congestion features of each configuration are reported in Table 6.2, while the resulting configuration chains are entirely shown in Figure 6.13. To briefly explain the notation, note that a configuration identified as c01 is computed by running MILP-EWO with the peak traffic matrix scaled to obtain a maximum link utilization of 10%, e.g., c03 $\rightarrow$ maximum utilization of 30% and so on. The maximum utilization limit imposed to MILP-EWO has been 70%. Differently from the other configurations, the *full* one has

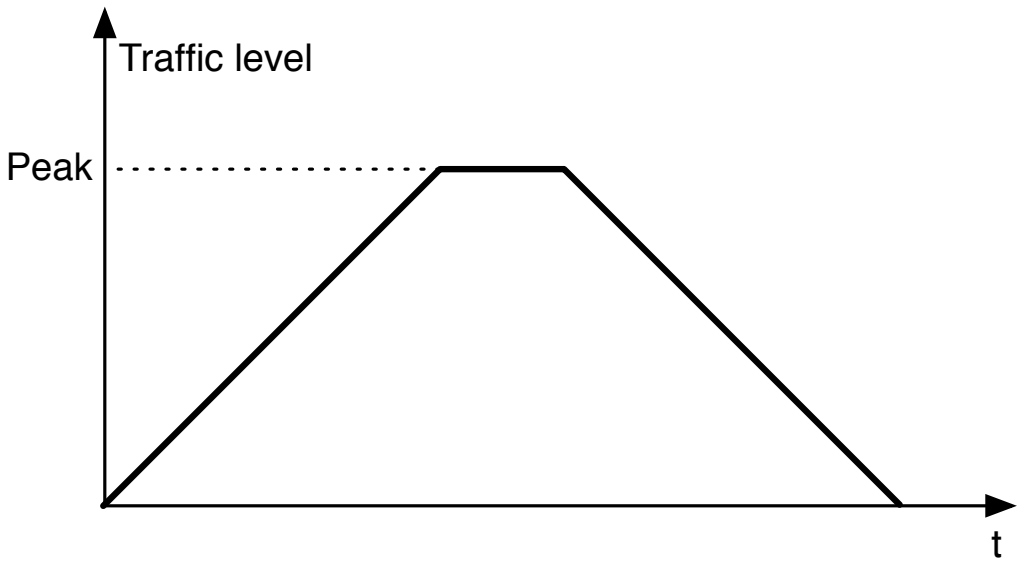
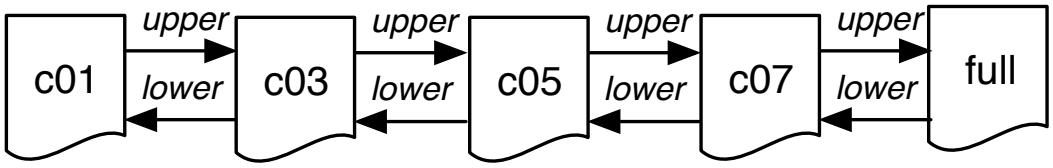


Figure 6.12 Fade-in fade-out profile used to generate traffic during tests.

Table 6.2 Description of the *OSPF configurations* computed by MILP-EWO.

Network	ψ_{max}	**OC**	A_{on}	V_{on}
polska	10%	c01	7 (39%)	8 (67%)
	30%	c03	9 (50%)	9 (75%)
	50%	c05	13 (72%)	11 (92%)
	70%	c07	14 (78%)	12 (100%)
	100%	full	18 (100%)	12 (100%)
abilene	50%	c05	9 (82%)	10 (71%)
	70%	c07	10 (91%)	12 (86%)
	100%	full	11 (100%)	14 (100%)

(a) Polska



(b) Abilene

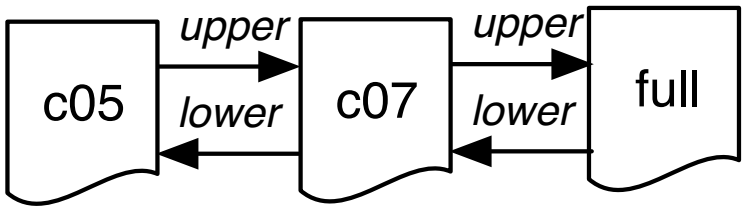


Figure 6.13 *OCs chains* used in tests.

Table 6.3 Switching policies.

Network	**Switching policy**	ψ^+_{av}	ψ^-_{av}	ψ^+_{max}	ψ^-_{max}
polska	*restrictive*	60%	40%	80%	50%
	permissive	60%	40%	80%	60%
abilene	*restrictive*	60%	40%	80%	50%
	permissive	60%	40%	80%	65%

been obtained by running only IGP-WO so as to minimize network congestion in presence of peak traffic levels; c01 and c03 are not considered in `abilene` because completely equivalent to c05.

We tested two different switching policies, namely *restrictive* and *permissive*, whose thresholds values are reported in Table 6.3. The *permissive* one, which is characterized by an increased value of ψ^{-}_{max} (up to 60% for `polska` and 65% for `abilene`), is meant to give the platform the possibility to adopt a more aggressive strategy to reduce consumption.

Simulations last 30 minutes with `polska` and 20 minutes with `abilene`. Load measurements are collected every 20 (`polska`) or 15 (`abilene`) seconds.

6.3.2 Experimentation

To summarize, for each network we tested two switching policies, i.e., *restrictive* and *permissive*, and considered three different levels of variance for packet generation parameters.

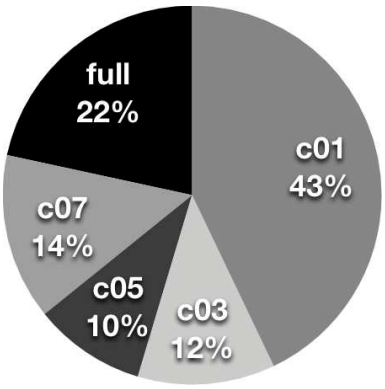
The first interesting result concerns the time distribution among the different OSPF configurations shown in Figure 6.14. The values observed with each policy are averaged over three different instances (each one with a different variance). Both policies allow to substantially reduce the application time of the *full* configuration. Note that the less-consuming policy is always the most used. The *permissive* policy further reduces the use of the *full* configuration by about 10% of the total simulation time. This amount of time seems to be fairly redistributed between all the other configurations.

Besides energy consumption levels and configuration distribution, the most important aspect concerns the congestion levels observed during the simulation when our dynamic SEANM-SP strategy is applied. We report in Figure 6.16 both average and maximum utilization levels observed in the network when our energy-aware strategy is applied and when the network is kept fully activated with the congestion-aware link weights of the *full* configuration. The plots represent the values obtained with inter-arrival departure time and packet size variance of 20%.

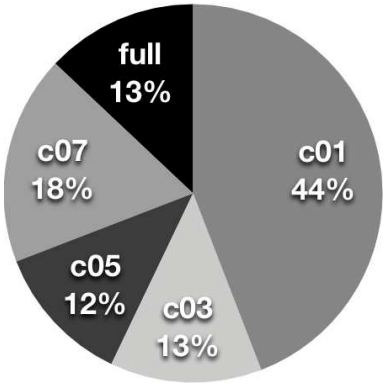
As expected, the sequence of configurations applied by EANI along the simulation naturally reflects the *fade-in fade-out* traffic profile. Starting from the less consuming configuration, the configuration chain is first progressively applied until the *full* configuration is implemented in correspondence of the traffic peak, and then gradually re-visited up to the less consuming one as soon the traffic begins to decrease.

It is important to point out how the threshold-based switching policy allows to successfully absorb the traffic variability induced by the packet generation variance, preventing in this way the network from oscillating between upper and lower configurations (same behaviour observed also with variances of 30%).

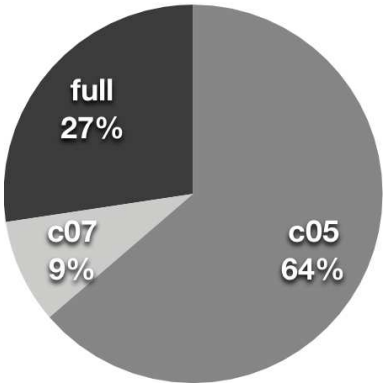
(a) Polska *restrictive*



(b) Polska *permissive*



(c) Abilene *restrictive*



(d) Abilene *permissive*

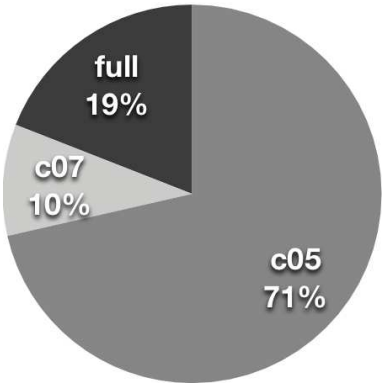
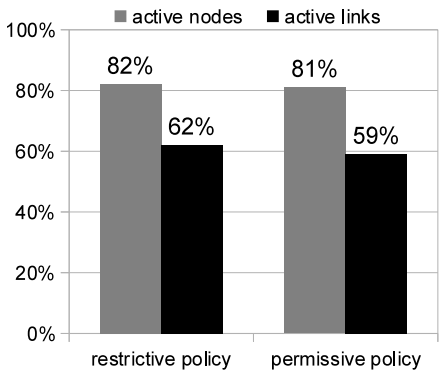


Figure 6.14 Distribution of OCs as percentage of time w.r.t the total duration of the simulations.

(a) Polska



(b) Abilene

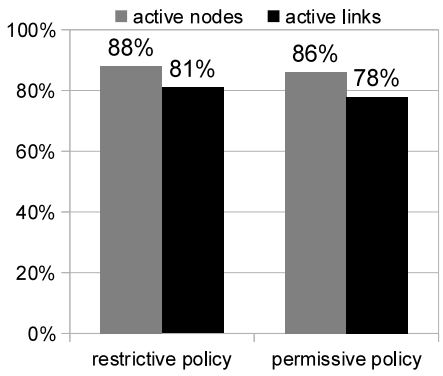


Figure 6.15 Relative resource consumption against an *always full topology* situation.

Once traffic load is high enough, the energy-aware optimization of the OSPF configurations keeps both maximum and average link utilization quite constant: once congestion has increased too much we push it down by applying a more consuming configuration, while in the opposite case we prevent it form decreasing excessively by switching to a greener configuration. Though higher utilization levels are observed w.r.t. the *full* configuration approach (around 10% with `abilene`, and up to 30% with `polska`), congestion is always kept under control and maintained under the desired thresholds.

A second interesting observation concerns how the utilization threshold values may influence both performance and stability of the proposed approach: in our simulation, adopting the *permissive* policy, i.e., increasing ψ^{-}_{max}, allows, during the traffic decremental phase, to better distribute the application time of the different configurations. The problem with the *restrictive* policy is that green configurations are applied when the network has become too unloaded.

Finally, we report in Figure 6.15 data concerning the average resource/energy consumption. Since we experimented with virtual devices, instead of reporting some consumption values taken from a datasheet of some possible real routers and cards, we refer to energy consumption as the percentage of active routers and links along the entire simulation.

The level of savings strictly depends on the topology structure of the considered network. In `polska` (`abilene`) 20% (15%) of network routers and 40% (20%) of network links are put to sleep, on average. The further savings obtained with the *permissive* policy are in the order of 3% for what concerns network links, and 1% as for network nodes. We interpret this result as a positive news, since it states that overall energy savings are preserved also when a higher priority is given to network congestion minimization.

6.4 Conclusions

In this chapter we presented a novel centralized network management platform to implement, in an on-line fashion, the link weight configurations computed by MILP-EWO during the planning phase. The management platform, which is built on our novel open-source network management framework called JNetMan, allows to combine both off-line and on-line optimization to offer both network stability and reactivity to real-time events.

Real-time link load data are exploited to determine, by means of a utilization-based policy, if the current link weight configuration must be replaced by a novel set of weights. All the applicable weight configurations are contained in a database which is built during the planning phase. The very popular management protocol called SNMP is exploited to periodically collect link load measurements and practically adjust the desired link weights.

Tests carried on emulated network environments showed that JNetMan can be efficiently exploited to dynamically adjust the link weights to adapt the network consumption to the incoming levels of traffic. We have showed that, to avoid too frequent oscillations between different configurations, it is crucial to carefully design the switching policy. The policy we have proposed, which is based on four utilization thresholds, has proved to be stable when coping with classic fade-in fade-out traffic profiles which well approximate realistic Internet traffic. From the energy point of view, tests showed that the adoption of the green platform allows to put to sleep, in average, 20% of nodes and 40% of links.

Note that JNetMan can be easily extended to introduce additional management functions and could be used by other researchers to practically implement novel energy-management proposals in a realistic network environment.

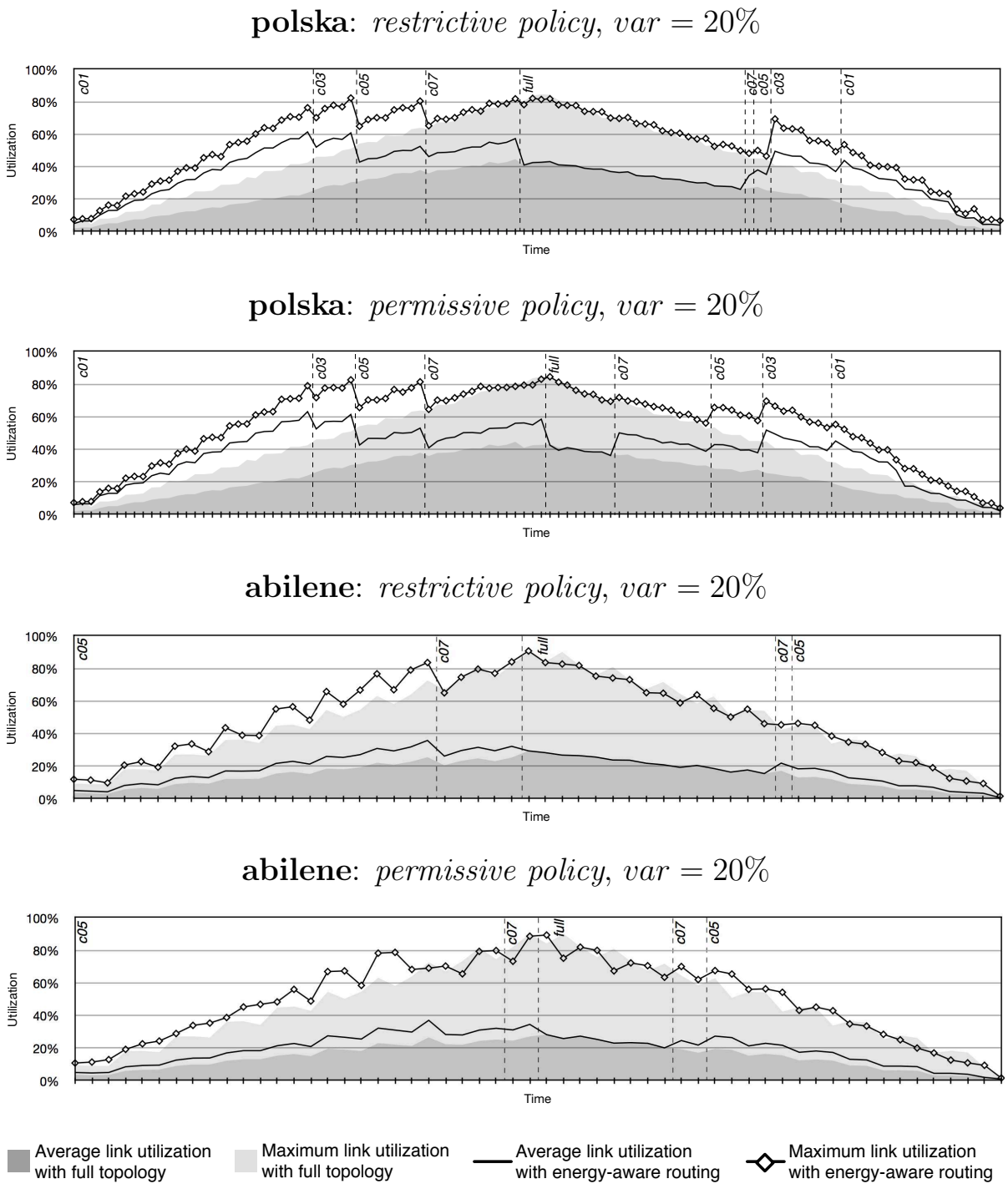


Figure 6.16 Link utilization charts obtained during four test instances.

CHAPTER 7

SEANM with Elastic Traffic

In current IP networks each connection is served at the transport layer (layer 4 according to the open systems interconnection (OSI) stack) by one of the two following end-to-end protocols, namely transport control protocol (TCP) and user datagram protocol (UDP). These protocols deeply differ on the service model offered to each connection: UDP provides a minimal *best effort* service where packets are transmitted at a fixed rate determined by the top layer application and no delivery assurance or performance guarantee is offered. The TCP service model offers multiple functions to guarantee reliable connections (packet reordering, lost packet retransmission, error-free transmission) and performs both flow and congestion control. The main peculiarity of TCP flows is that their transmission rate is not known a priori, but dynamically determined by some distributed congestion control mechanisms which allow each flow to maximally exploit the available resources along its path. TCP traffic is commonly defined as *elastic*, while UDP one is known as *inelastic*, i.e., with a fixed rate known a priori.

Nowadays, almost 70% of Internet traffic is carried by TCP (Finamore *et al.*, 2011) and elastic traffic largely outnumbers the inelastic. However, in the backbone all elastic demands appear to be inelastic with a fixed transmission rate upper bounded by the capacity available at the access level. This phenomenon is briefly illustrated in Figure 7.1, where we represent a communication between two users who are connected to their access network by a 10 MB/s link, e.g., a DSL connection. Since all backbone (black) and access-backbone (red) links have a capacity of 10 GB/s, when the backbone network is not completely saturated, the communication will appear to the backbone administrator as inelastic with a bandwidth requirement of 10 MB/s. In a few words, all elastic demands whose bottleneck is outside the network domain are assumed to be inelastic with a fixed rate determined by their bottleneck arcs.

This situation might change soon: the deployment of next-generation optical access (NGOA) networks is expected to increase the access rate of Internet users up to 2.4 Gbps in both download and upload (itu, 2008), with peak and average per-user rate close to 1 and 0.3 Gbps, respectively (Breuer *et al.*, 2011). This may shift some bottlenecks from the access to the backbone, making thus inappropriate to consider all demands as inelastic to simplify network management procedures. Elastic demands will compete with each other to increase their transmission rate, which will vary according to the routing path.

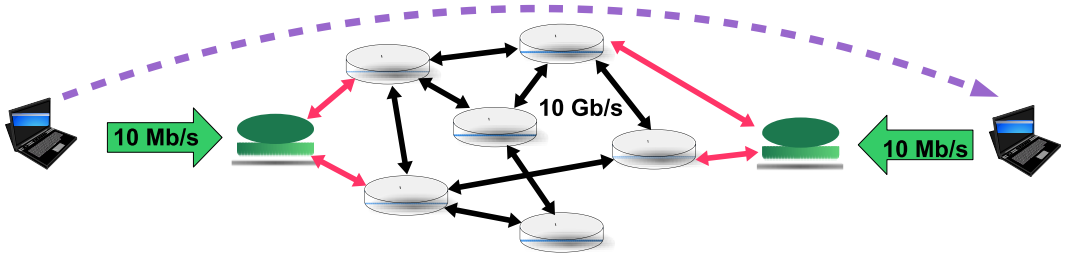


Figure 7.1 Why elastic demands may appear as inelastic.

The explicit management of elastic demands may revolutionize the way energy-aware network management is currently addressed. In classic SEANM, link utilization is the parameter typically monitored to detect network congestion or determine if a certain subset of underutilized network elements has to be put to sleep. However, if traffic is elastic, absolute utilization values are not in themselves enough to determine if a network is underutilized or congested: it is instead necessary to distinguish between the resources destined to serve inelastic demands, and those used by elastic traffic. While the first must be guaranteed, the latter should be handled from a different perspective. We illustrate this concept in Figure 7.2: two elastic traffic demands flow through a trivial 2-nodes network with two parallel 10 GB/s links. In the left configuration each demand is routed along a different link, the demand transmission rate is 10 GB/s and both links are saturated. Typical SEANM approaches based on inelastic traffic would see the network as fully saturated by two connections demanding 10 GB/s of bandwidth each one: no link would be allowed to go to sleep. However, if we consider that both demands are elastic, we realize that their 10 GB/s transmission rate is not strictly requested and is determined by the congestion control mechanisms of TCP. Thus, if we believe that 5 GB/s are enough to provide the desired QoS to both elastic demands, we can instruct our SEANM framework to put to sleep one link and route all the traffic on the active one (right part of Figure 7.2).

We show how to effectively manage elastic traffic demands into a general SEANM framework. In Section 7.1 we first present our novel bi-level approach for SEANM with elastic traffic, i.e., SEANM with elastic traffic (SEANM-ET). Then in Section 7.2 and Section 7.3 we show how to model resource allocation for elastic demands by means of two different notions, i.e., max-min-fairness (MMF) and proportional fairness (PF). Finally, in Section 7.4 we discuss a visual example which points out the novelties of our novel bi-level approach w.r.t. state of the art, while in Section 7.6 we report the computational results.

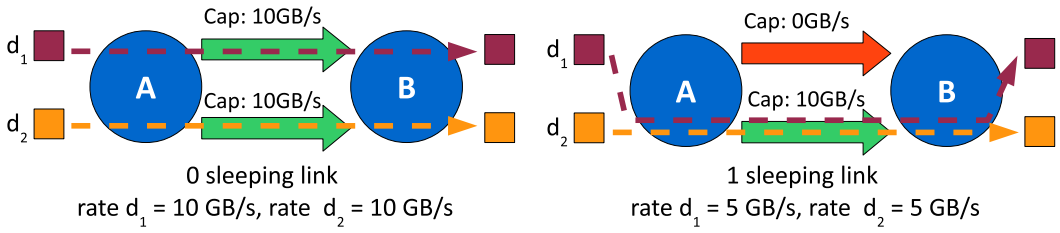


Figure 7.2 SEANM with elastic traffic.

7.1 A Novel Bi-Level Network Management Problem

In classic SEANM problems, the routing paths (single or multiple) used by each traffic demand are optimized so as to put to sleep the largest possible number of network elements. The portion of traffic carried by each path is explicitly determined by the operator, and each traffic demand is considered satisfied if and only if the link maximum utilization stays below the desired threshold on each link of the corresponding paths.

However, the explicit management of elastic demands completely overturns the way traffic demands are typically handled. First of all, while the network operator maintains the control on the network routing, the resource allocation along the routing paths is in this case independently determined by additive increase and multiplicative decrease mechanism of TCP. This means that each different routing configuration leads to a specific transmission rate (allocation) vector. A second element of novelty concerns how the operator determines if the amount of resource allocated to a certain demand is satisfactory or not. With inelastic demands, the required bandwidth is given and we must find the resources to satisfy it. When traffic is elastic, we adopt a different perspective: properly designed utility functions are used to quantify the satisfaction degree of each elastic demand from the point of view of both clients and network operators (Shi *et al.*, 2008).

Driven by these considerations we formulate a novel bi-level network management approach to jointly optimize network utility and overall power consumption in presence of elastic traffic. At the upper level, the network operator (the *leader*) is responsible for optimizing the routing paths, while, at the lower one, the transport protocol (the *follower*) allocates the resources among the elastic demands according to some specific criteria. The goal of the network operator is to determine the routing configuration whose resulting allocation vector maximizes a measure of network utility related to both transmitting rates and network energy consumption.

The key point consists in how to model the resource allocation scheme followed by the transport protocol. Note that for simplicity, we use the expression *transport protocol* to refer

to the entire set of distributed congestion mechanisms implemented by the network.

The first observation is that congestion control algorithms are typically designed to favor fairness between competing flows. Practically speaking, link capacity is shared in a fair way if no special treatment, i.e., any sort of priority, is provided to any elastic flow. In the literature, two different definitions of fairness are largely employed, i.e., *max-min-fairness* (MMF) (details given in Section 7.2) and *proportional-fairness* (PF) (details given in Section 7.3). Informally, the resource allocation is *max-min-fair* when we cannot increment the transmission rate of any elastic demand without decreasing the bandwidth assigned to another demand with a smaller rate. This is equivalent to finding the allocation vector which is lexicographically maximized. *proportional fairness* is instead achieved if the summation of the logarithm of the rate assigned to each elastic demand is maximized. Differently from *max-min-fairness*, *proportional fairness* implicitly gives higher priority to the elastic demands whose routing paths are shorter (in terms of number of hops).

In 1998, the seminal work of Kelly et al. (Kelly *et al.*, 1998) showed that each distributed algorithm to balance network congestion implicitly solves a properly designed utility maximization problem subject to link capacity constraints. In particular, it has been demonstrated that TCP tends to achieve *Proportional Fairness*. The framework proposed in (Kelly *et al.*, 1998) was later exploited to compute the utility function maximized by different versions of TCP, e.g., Reno or Vegas (Low, 2003), and by HTTP traffic over TCP (Chang *et al.*, 2004). Following a different approach, Massoulié et al. (Massoulie et Roberts, 2002) showed how different queuing policies allow to achieve different types of fairness: in particular, *first-in-first-out* (FIFO) queuing achieves PF, while *per-flow fair* queuing tends to *max-min-fairness*.

To correctly handle resource allocation for elastic traffic demands once routing paths have been selected, it is thus necessary, at the lower level, to solve at optimality the right fairness/utility maximization problem. This approach greatly differs from state-of-the-art approaches on overall fairness/utility maximization, whose aim is, given a network topology and set of elastic connections, to find the joint routing-allocation setting which maximizes a measure of fairness (utility) related to demand rates (see, e.g., Nace et Pioro, 2008). In our case, maximizing fairness on the paths selected by the upper level, i.e., the network operator, is a hard network constraint which forces the optimization model to fairly share network resources among the competing elastic demands as done by the transport protocol. Let us show in the following section how to formally define our novel bi-level problem for energy-aware network management with elastic demands (SEANM-ET).

7.1.1 A General MILP for SEANM with Elastic Traffic

Given an IP network represented by the directed graph $G(V, A)$, let V and A bet the sets of network nodes and network links, respectively. Each link $(i, j) \in A$ is equipped with c_{ij} units of capacity and consume π_{ij} Watt. The consumption of node $i \in V$ is instead denoted by π_i. Let D^e be the set of elastic demands and let o^d (t^d) be the source (destination) node of each demand $d \in D^e$. Furthermore, note that each elastic demand $d \in D^e$ is composed by Δ^d TCP connections: this parameter is crucial because congestion control mechanisms discriminate between each single TCP connections. A utility function $U^d\left(\phi^d/\Delta^d\right)$ is used to quantify the benefit of assigning ϕ^d/Δ^d units of bandwidth (ϕ^d is a non-negative real variable) to each TCP connection carried by demand $d \in D^e$. The goal of the network operator is to determine the single path of each demand which maximizes the overall network utility computed as $\sum_{d \in D^e} \Delta^d U^d\left(\phi^d/\Delta^d\right)$. The routing is kept single path unsplittable due to modeling constraints which are crucial, as we will show later, to correctly compute the behaviour of the congestion control mechanisms: like in MPLS, which is the most popular per-flow routing protocol (see Section 3), a single routing path is explicitly assigned to each elastic traffic demand and to all its TCP connection.

In the meanwhile the operator has to put to sleep enough network elements to respect a constraint on the maximum energy consumption allowed denoted by B. Due to the preliminary nature of our studies on the management of elastic traffic, instead of introducing a complex bi-objective function to jointly account for both utility and consumption optimization, we prefer this solution based on utility maximization subject to energy budget constraints, which allows to find the desired trade-off in a straightforward fashion by simply adjusting the consumption budget B. Our aim is to clearly highlight to network operators the relationship between power consumption and flow utility when elastic traffic demands are at stake. Note that in the same way, it would be possible to overturn the problem so as to minimize the overall network consumption while being subject to constraints on the minimum utility required by each flow (or on the minimum aggregated utility of all flows). To summarize, we do not consider the formulation that we are going to present as the definitive one for SEANM-ET, but rather as a preparatory tool which allows to better catch the new dynamics of network management in presence of elastic traffic.

Problem variables are represented with the same notation used in previous chapters: let x_{ij}^d be the binary variables which are equal to 1 if elastic demand $d \in D^e$ is routed along link $(i, j) \in A$, let f_{ij}^d be the non-negative real variables representing the amount of flow generated by demand $d \in D^e$ and carried on link $(i, j) \in A$. Being w_{ij} and y_i the binary variables equal to 1 if, respectively, link $(i, j) \in A$ or node $i \in V$ is active, we can formulate our bi-level problem as follows:

Upper level

$$\max_{x} \left\{ \sum_{d \in D^e} \Delta^d U^d \left(\phi^d / \Delta^d \right) \right\} \tag{7.1}$$

s.t.

$$\sum_{\substack{j \in V: \\ (i,j) \in A}} x_{ij}^d - \sum_{\substack{j \in V: \\ (j,i) \in A}} x_{ji}^d = \begin{cases} 1 & \text{if } i = o_d, \\ -1 & \text{if } i = t_d, \\ 0 & \text{otherwise} \end{cases}, \qquad \forall i \in V,\, d \in D^e \tag{7.2}$$

$$x_{ij}^{st} \leq w_{ij}, \qquad \forall (i,j) \in A,\, d \in D^e \tag{7.3}$$

$$w_{ij} \leq y_i, \qquad \forall (i,j) \in A \tag{7.4}$$

$$w_{ij} \leq y_j, \qquad \forall (i,j) \in A \tag{7.5}$$

$$\sum_{(i,j) \in A} \pi_{ij} w_{ij} + \sum_{i \in V} \pi_i y_i \leq B, \tag{7.6}$$

$$x_{ij}^d,\, w_{ij},\, y_h \in \{0,1\}, \quad \forall h \in V,\, (i,j) \in A,\, d \in D^e. \tag{7.7}$$

Objective function (7.1) maximizes the overall network utility while flow conservation Constraints (7.2) are used to define a single unsplittable path for each demand. Constraints (7.3) keep active all the network links crossed by at least one routing path, and Constraints (7.4–7.5) allow to put to sleep a network node only if all the incident links are sleeping too. Finally, the total energy budget B is respected by means of Constraint (7.6), while Constraints (7.7) define the variable domains.

Note that the role of the upper layer consists exclusively in determining the single routing path of each demand. Therefore, though the goal of the operator is to maximize a utility function related to the rate of each demand, he has no direct control on the rate assignment operation itself. This is where the lower level comes into play: to determine how network resources are allocated by the transport protocol among the elastic demands (on the paths chosen by the network operator), we further constrain the single level problem (7.1–7.7) to the following lower level fairness maximization problem:

Lower level

$$\max \{\text{fairness}\} \tag{7.8}$$

s.t.

$$\sum_{\substack{j\in V:\\(i,j)\in A}} f_{ij}^d - \sum_{\substack{j\in V:\\(j,i)\in A}} f_{ji}^d = \begin{cases} \phi^d & \text{if } i = o^d \\ -\phi^d & \text{if } i = t^d \\ 0 & \text{else} \end{cases}, \qquad \forall i \in V,\, d \in D^e \tag{7.9}$$

$$\sum_{d\in D^e} f_{ij}^d \leq c_{ij}, \qquad \forall (i,j) \in A \tag{7.10}$$

$$f_{ij}^d \leq c_{ij} x_{ij}^d, \quad \forall (i,j) \in A,\, d \in D^e \tag{7.11}$$

$$f_{ij}^d,\, \phi^d \geq 0, \quad \forall (i,j) \in A,\, d \in D. \tag{7.12}$$

Objective function (7.8), which for the moment is not formalized in details, maximizes a certain measure of fairness related to how resources are shared by the whole set of elastic traffic demands. The specific form of (7.8) has to be determined by the network operator according to the congestion control mechanisms implemented in the considered network domain. Let us recall that each combination of congestion control mechanisms tends to maximize a specific utility function (Kelly *et al.*, 1998). Since in existing IP networks resource allocation can be well approximated by MMF and PF (due to the interaction of TCP with different queuing policies), we later show in Sections 7.2 and 7.3 how to formalize Objective function (7.8) for these two specific forms of fairness maximization.

Flow conservation Constraints (7.9), capacity Constraints (7.10) and coherence Constraints (7.11) are required to correctly compute the rate ϕ^d assigned to each elastic demand $d \in D^e$. Link capacity cannot be exceeded and no unit of flow can be routed along the links not selected by the network operator at the upper layer.

7.2 Max-Min-Fairness

It is widely known that in IP networks, resource allocation among elastic demands is quite well approximated by the MMF paradigm. Deviations from MMF allocation are mainly due to the non-uniformity of round-trip-time (RTT) for different TCP connections. The speed at which the transmission window of each TCP connection grows is in fact determined by the time required by a packet to reach the destination node plus that taken by the successive ACK to reach the transmitting node: connections with the smallest RTT will thus occupy the available bandwidth at a faster rate than all the others. These deviations can be limited by applying per-flow fair queuing (Massoulie et Roberts, 2002) and by introducing other congestion avoidance mechanisms such as *random early detection* (RED) commonly implemented in IP routers (Altman *et al.*, 2000).

The allocation vector $\underline{\phi}$ is MMF if it is lexicographically maximized (see, e.g., Nace et Pioro, 2008). Being $\sigma(\underline{\phi})_i$ the i-th element of the allocation vector $\underline{\phi}$ sorted in a non-decreasing

order by means of the σ operator, $\underline{\phi}$ is MMF if and only if it any element $\sigma(\underline{\phi})_i$ for $i \in \{1, \ldots, |\underline{\phi}|\}$ cannot be increased without reducing the amount of flow of at least one element $\sigma(\underline{\phi})_j$ with $j < i$.

To instruct the lower level to allocate the network resources in a MMF way, we replace the general fairness objective function (7.8) with the following expression:

$$\max \min \operatorname{lex} \left\{ \underline{\phi} \right\}, \tag{7.13}$$

where $\underline{\phi}$ is the allocation vector containing all the values of $\phi^d \; \forall \; d \in D^e$. This objective function is clearly hardly tractable, but the following characterization of MMF can be exploited to replace it with a group of integer linear constraints: an allocation vector $\underline{\phi}$ is MMF if and only if each elastic demand $d \in D^e$ is routed through at least one *bottleneck* link (Massoulie et Roberts, 2002; Nace et Pioro, 2008). A link $(i, j) \in A$ is considered bottleneck for an elastic demand $d \in D^e$ if it has no spare capacity available and if $\phi^d \geq \phi^h$ for each other $h \in D^e$ which is routed on link $(i, j) \in A$ too.

7.2.1 Exact MILP Formulation

The bottleneck characterization just provided can be efficiently exploited to transform the MMF lower level problem in the following feasibility problem:

$$(7.9) - (7.12)$$

$$\sum_{(i,j) \in A} \varpi_{ij}^d \geq 1, \qquad \forall d \in D^e \tag{7.14}$$

$$\sum_{h \in D^e} f_{ij}^h \geq c_{ij} \varpi_{ij}^d, \quad \forall (i,j) \in A, \, d \in D^e \tag{7.15}$$

$$\vartheta_{ij} \geq f_{ij}^d / \Delta^d, \quad \forall (i,j) \in A, \, d \in D^e \tag{7.16}$$

$$f_{ij}^d / \Delta_d \geq \vartheta_{ij} - c_{ij}(1 - \varpi_{ij}^d), \quad \forall (i,j) \in A, \, d \in D^e \tag{7.17}$$

$$\varpi_{ij}^d \in \{0, 1\}, \quad \forall (i,j) \in A, \, d \in D^e \tag{7.18}$$

$$\vartheta_{ij} \geq 0, \qquad \forall (i,j) \in A. \tag{7.19}$$

Binary variables ϖ_{ij}^d are equal to 1 if link $(i, j) \in A$ is the bottleneck of demand $d \in D^e$, while non-negative real variables ϑ_{ij} represent the largest amount of flow which belongs to a single TCP connection carried by link $(i, j) \in A$. Constraints (7.14) ensure the presence of at least one bottleneck link for each elastic demand, Constraints (7.15) impose the bottleneck

link saturation, and Constraints (7.16–7.17) allow to choose arc $(i,j) \in A$ as bottleneck link of a certain demand $d \in D^e$ if and only if the flow ϕ^d/Δ^d assigned to each TCP connection of demand d is at least as large as the flow assigned to each other connection carried by link (i,j). Note that by means of Δ^d, if we reasonably assume that, like in our case, all the TCP connections flowing between the same pair of origin-destination nodes are routed on the same path, we can manage a demand set of cardinality $\sum_{d \in D^e} \Delta^d$ without dramatically increasing the problem complexity.

The above feasibility problem can be integrated within the upper level formulation (7.1–7.7) to obtain a comprehensive single level MILP formulation for SEANM-ET with MMF allocation. However, due to the particular structure of the MMF constraints, a further change has to be made to the upper level problem (7.1–7.7) to guarantee that the routing paths chosen by the network operator contain no subtours. To this purpose we introduce the following group of constraints:

$$\gamma^d_{ijh} \leq x^d_{ij}, \quad \forall h \in V^{o^d}, d \in D^e, (i,j) \in A \tag{7.20}$$

$$\sum_{(i,h) \in A} x^d_{ih} = \psi^d_h, \qquad \forall h \in V^{o^d}, \, d \in D^e \tag{7.21}$$

$$\sum_{(i,j) \in A} \gamma^d_{ijh} - \sum_{(j,i) \in A} \gamma^d_{jih} = \begin{cases} \psi^d_h & \text{if } i = o^d \\ -\psi^d_h & \text{if } i = h \\ 0 & \text{else} \end{cases}, \quad \forall h \in V^{o^d}, \, i \in V, \, d \in D^e, \tag{7.22}$$

where binary variables ψ^d_h are equal to 1 if demand $d \in D^e$ flows through node $h \in V^{o^d}$, where V^{o^d} represents the node set from which source node o^d of demand $d \in D^e$ has been excluded. Then binary variables γ^d_{ijh} are equal to 1 when link $(i,j) \in A$ lies on the portion of the routing path used by demand $d \in D^e$ which connects node o^d to node $h \in V$. Constraints (7.20–7.22) prevents the creation of subtours by forcing the source node o^d of each demand $d \in D^e$ to send an auxiliary unit of traffic to each intermediate node of the corresponding routing path.

To conclude, note that the two following groups of valid inequalities can be added to speed up the convergence of the MILP solver to the optimal solution:

$$\gamma^d_{ij} \leq x^{st}_{ij}, \quad \forall (i,j) \in A, \, d \in D^e \tag{7.23}$$

$$\frac{\phi^d}{\Delta_d} \geq \frac{\min_{(i,j) \in A}\{c_{ij}\}}{\sum_{h \in D^e} \Delta_h} \qquad \forall d \in D^e. \tag{7.24}$$

7.3 Proportional-Fair Allocation

Both empirical and analytical studies (see, e.g., Kelly *et al.*, 1998; Low, 2003) showed that, mainly due to the heterogeneity which characterizes both connection RTT and link capacity, in current IP network resource allocation is better approximated by PF. PF can be introduced in the lower level allocation problem by replacing (7.8) with:

$$\max \left\{ \sum_{d \in D^e} \Delta^d \log \left(\frac{\phi^d}{\Delta^d} \right) \right\}. \tag{7.25}$$

Maximizing the summation of the logarithm of each elastic connection (recall that Δ^d TCP connections are aggregated within each elastic demand $d \in D^e$) allows to favor the increase of the smallest demands, while, at the same time, giving an higher priority to the short connections. Practically speaking, differently from MMF, with PF we are not necessarily forced to increase the rate of a certain demand $d \in D^e$ if that would means reducing the bandwidth of two or more other flows larger than d.

In the next section we show how to translate the lower level Objective function (7.25) to achieve PF, into a group of integer linear constraints which can be integrated in the upper level formulation (7.1-7.7) (as done for MMF) to obtain a compact single level MILP for SEANM-ET with PF allocation. However, this operation requires the introduction of some approximations which make the resulting formulation heuristic.

7.3.1 Heuristic MILP for PF

Let us a briefly recall the complete lower level allocation problem with PF:

$$\max_{f,\,\phi} \left\{ \sum_{d \in D^e} \Delta^d \log \left(\frac{\phi^d}{\Delta^d} \right) \right\}. \tag{7.25}$$

s.t.

$$\sum_{\substack{j\in V:\\(i,j)\in A}} f_{ij}^d - \sum_{\substack{j\in V:\\(j,i)\in A}} f_{ji}^d = \begin{cases} \phi^d & \text{if } i = o^d \\ -\phi^d & \text{if } i = t^d \\ 0 & \text{else} \end{cases}, \quad \forall i \in V,\, d \in D^e \tag{7.9}$$

$$\sum_{d\in D^e} f_{ij}^d \le c_{ij}, \quad \forall (i,j) \in A \tag{7.10}$$

$$f_{ij}^d \le c_{ij} x_{ij}^d, \quad \forall (i,j) \in A,\, d \in D^e \tag{7.11}$$

$$f_{ij}^d,\, \phi^d \ge 0, \quad \forall (i,j) \in A,\, d \in D^e. \tag{7.12}$$

To make (7.25), (7.9–7.12) tractable, we first approximate the non-linear Objective function (7.25) with a piece-wise linear function. As in Chapter 5, let H bet the set of linear pieces which form the piece-wise linear log-function: each piece $h \in H$ is described by both its slope α_h and its offset β_h. The logarithmic Objective function (7.25) is replaced by

$$\max \left\{ \sum_{d\in D^e} \Delta^d \varrho^d \right\} \tag{7.26}$$

s.t.

$$\varrho^d \le \alpha_h \frac{\phi^d}{\Delta^d} + \beta_h, \quad \forall d \in D^e,\, h \in H, \tag{7.27}$$

$$\varrho^d \in \mathcal{R}, \qquad \forall d \in D^e, \tag{7.28}$$

where ϱ^d represents the real variables used to compute the piecewise log-utility of each TCP connection carried by the aggregate elastic demand $d \in D^e$.

We propose to exploit the strong duality theorem to transform the lower level problem (7.26–7.28) plus (7.9–7.12) into a feasibility problem which can be integrated in the upper level formulation (7.1–7.7). According to this theorem, if a linear program is feasible and bounded, the optimal objective of a primal maximization problem is equivalent to the optimal one of the corresponding dual minimization problem: the maximization of (7.26) can be thus obtained by forcing the primal Objective function (7.26) to be equal to the corresponding dual objective function, subject to all the constraints of both primal and dual problem. Let us formulate the dual problem of (7.26–7.28) plus (7.9–7.12):

$$\min \left\{ \sum_{d\in D^e} \sum_{h\in H} \beta_h \lambda_h^d + \sum_{(i,j)\in A} c_{ij}\mu_{ij} + \sum_{(i,j)\in A} \sum_{d\in D^e} c_{ij} x_{ij}^d \theta_{ij}^d \right\} \tag{7.29}$$

s.t.

$$\sum_{h\in H} \lambda_h^d = \Delta^d, \qquad \forall d \in D^e \tag{7.30}$$

$$-\sum_{h\in H} \alpha_h \frac{\lambda_h^d}{\Delta^d} - \kappa_{o^d}^d + \kappa_{t^d}^d \geq 0, \qquad \forall d \in D^e \tag{7.31}$$

$$\kappa_i^d - \kappa_j^d + \mu_{ij} + \theta_{ij}^d \geq 0, \qquad \forall (i,j) \in A,\, d \in D^e \tag{7.32}$$

$$\lambda_h^d,\, \mu_{ij},\, \theta_{ij}^d \geq 0, \quad \forall (i,j) \in A,\, d \in D^e,\, h \in H \tag{7.33}$$

$$\kappa_i^d \in \mathbf{R}, \qquad \forall d \in D^e. \tag{7.34}$$

Below follows the description of the dual variables:

λ_h^d Non-negative dual variables corresponding to Constraints (7.27).

κ_i^d Real dual variables corresponding to Constraints (7.9).

μ_{ij} Non-negative dual variables corresponding to Constraints (7.10).

θ_{ij}^d Non-negative dual variables corresponding to Constraints (7.11).

Due to the non linearity of $\sum_{(i,j)\in A} \sum_{d\in D^e} c_{ij} x_{ij}^d \theta_{ij}^d$, which is the last term of dual Objective function (7.29), a last change is made to the dual formulation to eliminate any non-linearity. Following a classic approach to linearize the product of a non-negative variable with a binary one, we replace $\sum_{(i,j)\in A} \sum_{d\in D^e} c_{ij} x_{ij}^d \theta_{ij}^d$ with the following MILP formulation:

$$\sum_{(i,j)\in A} \sum_{d\in D} c_{ij} \upsilon_{ij}^d, \quad \forall (i,j) \in A,\, d \in D^e \tag{7.35}$$

$$0 \leq \upsilon_{ij}^d \leq M x_{ij}^d, \quad \forall (i,j) \in A,\, d \in D^e \tag{7.36}$$

$$\theta_{ij}^d - M\left(1 - x_{ij}^d\right) \leq \upsilon_{ij}^d \leq \theta_{ij}^d, \quad \forall (i,j) \in A,\, d \in D^e \tag{7.37}$$

$$\upsilon_{ij}^d \geq 0, \quad \forall (i,j) \in A,\, d \in D^e. \tag{7.38}$$

Note that M, which represents a very big parameter whose aim is to let υ_{ij}^d to assume the right value, is the second element which, together with the piece-wise approximation, makes the final single level MILP not exact. In fact, being θ_{ij}^d not upper bounded (from

duality theory), there exists no value of M to account for the entire feasibility region of θ_{ij}^d. Practically speaking it means that if M is not high enough, the constraint which matches primary and dual objective may not be satisfied for a certain choice of paths made at the upper level (duality gap). However, we reasonably assume that setting M with a very large value, e.g., 10000, should drastically minimize this phenomenon.

Comprehensive MILP for Lower Level Feasibility Problem

Thanks to the previous steps, we can formalize the lower level problem by means of the following feasibility MILP formulation:

$$\sum_{d\in D^e} \Delta^d \varrho^d \geq \sum_{d\in D^e}\sum_{h\in H} \beta_h \lambda_h^d + \sum_{(i,j)\in A} c_{ij}\mu_{ij} + \sum_{(i,j)\in A}\sum_{d\in D^e} c_{ij}\upsilon_{ij}^d \tag{7.39}$$

$$\varrho^d \leq \alpha_h \frac{\phi^d}{\Delta^d} + \beta_h, \quad \forall d \in D^e,\, h \in H \tag{7.27}$$

$$\sum_{\substack{j\in V:\\(i,j)\in A}} f_{ij}^d - \sum_{\substack{j\in V:\\(j,i)\in A}} f_{ji}^d = \begin{cases} \phi^d & \text{if } i = o^d \\ -\phi^d & \text{if } i = t^d \\ 0 & \text{else} \end{cases}, \quad \forall i \in V,\, d \in D^e \tag{7.9}$$

$$\sum_{d\in D^e} f_{ij}^d \leq c_{ij}, \quad \forall (i,j) \in A \tag{7.10}$$

$$f_{ij}^d \leq c_{ij} x_{ij}^d, \quad \forall (i,j) \in A,\, d \in D^e \tag{7.11}$$

$$\sum_{h\in H} \lambda_h^d = \Delta^d, \quad \forall d \in D^e \tag{7.30}$$

$$-\sum_{h\in H} \alpha_h \frac{\lambda_h^d}{\Delta^d} - \kappa_{o^d}^d + \kappa_{t^d}^d \geq 0, \quad \forall d \in D^e \tag{7.31}$$

$$\kappa_i^d - \kappa_j^d + \mu_{ij} + \theta_{ij}^d \geq 0, \quad \forall (i,j) \in A,\, d \in D^e \tag{7.32}$$

$$0 \leq \upsilon_{ij}^d \leq M x_{ij}^d, \quad \forall (i,j) \in A,\, d \in D^e \tag{7.36}$$

$$\theta_{ij}^d - M\left(1 - x_{ij}^d\right) \leq \upsilon_{ij}^d \leq \theta_{ij}^d, \quad \forall (i,j) \in A,\, d \in D^e \tag{7.37}$$

$$\varrho^d \in \mathcal{R}, \quad \forall d \in D^e \tag{7.28}$$

$$f_{ij}^d,\, \phi^d, \geq 0, \quad \forall (i,j) \in A,\, d \in D^e \tag{7.12}$$

$$\lambda_h^d,\, \mu_{ij},\, \theta_{ij}^d \geq 0, \quad \forall (i,j) \in A,\, d \in D^e,\, h \in H \tag{7.33}$$

$$\kappa_i^d \in \mathbf{R} \quad \forall d \in D^e \tag{7.34}$$

$$\upsilon_{ij}^d \geq 0, \quad \forall (i,j) \in A,\, d \in D^e. \tag{7.38}$$

By adding the above formulation to (7.1–7.7) (upper level MILP) we obtain a compre-

hensive MILP for SEANM-ET with piece-wise approximated PF.

7.4 A Visual Example

To better point out the bi-level nature of our novel approach and remark the fundamental differences with state of the art methods for overall fairness maximization, let us discuss the following example. Note that we consider MMF allocation at the lower level, and, for the sake of simplicity, we set the energy budget B to infinite (no element is put to sleep).

Example:
Consider the network in Figure 7.3 with five origin-destination pairs $(o_1, t_1), \ldots, (o_5, t_5)$, where the dashed lines represent paths with a capacity greater or equal to 10, the link (a_1, b_1) has a capacity of 10, and the links (a_i, b_i), with $i = 2, \ldots, 5$, have a capacity of 1. We consider two solutions where each elastic demand is routed along a different path, and bandwidth is allocated according to the MMF paradigm.

In a) the objective function of the upper layer (network operator) is the maximization of the total throughput, i.e., $\max \sum_{d=1}^{5} \phi^d$. It is easy to demonstrate that in the optimal solution, each traffic demand d is routed along a dedicated path $o_d - a_d - b_d - t_d$. All paths are link disjoint and MMF is naturally achieved by sending the maximum amount of traffic on each path. This amounts to obtain the sorted allocation vector $\sigma(\underline{\phi}) = (1, 1, 1, 1, 10)$ (σ is the sorting operator) characterized by a total throughput of 14.

In b), the aim of the network operator is to maximize the amount of bandwidth assigned to the smallest elastic demand, i.e., $\max q : \ q \leq \phi^d, \forall d \in \{1, 2, 3, 4, 5\}$. The optimal solution is very different from the previous case: all demands (o_d, t_d) are routed through the link (a_1, b_1) (along the paths o_d, a_1, b_1, t_d) and the resulting MMF allocation vector (sorted) $\sigma(\underline{\phi})$ is $(2, 2, 2, 2, 2)$. Note that the smallest demand has a transmission rate of 2, while the total throughput is 10.

These two trivial examples clearly point out the meaning of being fair only locally, i.e., on the path selected by the upper level, to model how link capacity bandwidth is shared among competing flows by the transport protocol. The aim of the upper level is to adjust the routing configuration so as to drive the lower level to maximize not only the local fairness, but also the overall network utility, which corresponds to $\max \sum_{d=1}^{5} \phi^d$ and $\max q : \ q \leq \phi^d, \forall d \in \{1, 2, 3, 4, 5\}$ in a) and b), respectively. For the sake of completeness, note that in b), the optimal solution of the bi-level problem is MMF not only locally, but also globally.

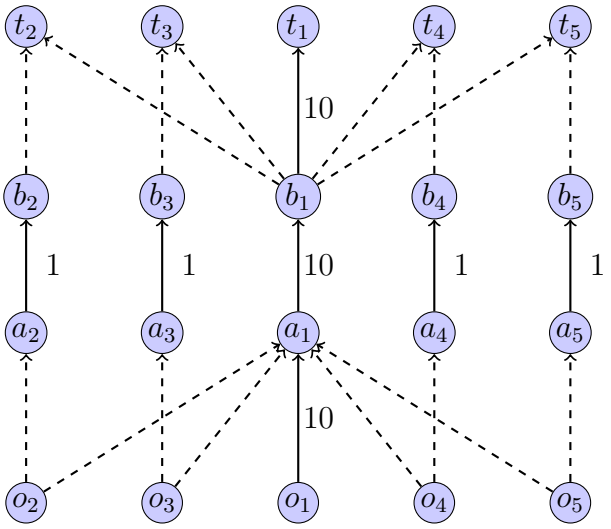


Figure 7.3 Representation of the capacitated network and five origin-destination pairs discussed in the example.

7.5 Restricted Path Heuristic

Both formulations with MMF and PF allocation turns out to be very challenging as soon as the number $|D^e|$ of demands is incremented. To partially overcome this problem and make both formulations more scalable, we propose to constrain the network routing configured by the upper level to a restricted set of pre-computed paths (as done in Section 4.4 for survivable SEANM-FB).

Being P^d bet the set of precomputed paths of demand $d \in D^e$, let $\chi_p^{d'}$ be the binary variables equal to 1 if path $p \in P^d$ is chosen to route the traffic of elastic demand $d \in D^e$. The following group of constraints is added to ensure that a single path is used by each demand

$$\sum_{p \in P^d} \chi_p^{d'} = 1, \quad \forall d \in D. \tag{7.40}$$

Then, we discard Constraints (7.2) (upper level path selection problem) and Constraints (7.20–7.22) (lower level MMF allocation problem) and adjust all the other constraints by replacing the x variables as follows:

$$x_{ij}^{d} = \sum_{p \in P^{d}:(i,j) \subset p} \chi_{p}^{d'}, \quad \forall (i,j) \in A,\, d \in D^{e} \tag{7.41}$$

The restricted set of precomputed paths is almost entirely generated by means of the same procedure described in Section 4.4 (to which we refer the reader). With respect to that method, we only compute one more path for each demand to guarantee the existence of a feasible solution in case the power budget B parameter is very low. More specifically, these paths are extracted from the directed Steiner tree which minimizes the network energy consumption while connecting all the nodes which generate or receive traffic (edge nodes V^{e}). The Steiner tree is computed with a state of the art MILP formulation.

7.6 Computational Results

7.6.1 Test Instances

In Section 7.1 we did not precisely formalized the utility function $\Delta^{d} U^{d}\left(\phi^{d}/\Delta^{d}\right)$ assigned to each aggregate elastic demand $d \in D^{e}$ and used to compute the overall network utility (7.1) of the upper problem. Along our test campaign we consider two different versions of (7.1):

$$U_{1} = \max \sum_{d \in D^{e}} \phi^{d} \tag{7.42}$$

$$U_{2} = \max \sum_{d \in D^{e}} \Delta_{d} a \left(1 - e^{\frac{1}{b}\frac{\phi^{d}}{\Delta_{d}}}\right). \tag{7.43}$$

Objective function (7.42) maximizes the overall network throughput, while Objective function (7.43), if a, $b > 0$ are properly chosen, tends to favour the smallest demands by introducing a saturating effect for large ϕ^{d}. Being (7.43) a concave non-linear function, to avoid non-linearities we experiment with a piecewise-affine approximation (with 6 pieces) (see Figure 7.4, obtained for $a = 1000$ and $b = 200$).

We take four realistic network topologies from the widely used SNDLib (Orlowski *et al.*, 2010), i.e., `polska`, `abilene`, `geant` and `nobel-eu`. Each network node is equipped with one ore more M10i chassis (according to the number of links connected to the node), while each link is equipped with a line card randomly chosen from Table 7.1. For each topology we consider two random equipment configurations.

To experiment with demand sets D^{e} of different cardinality we vary, from time to time,

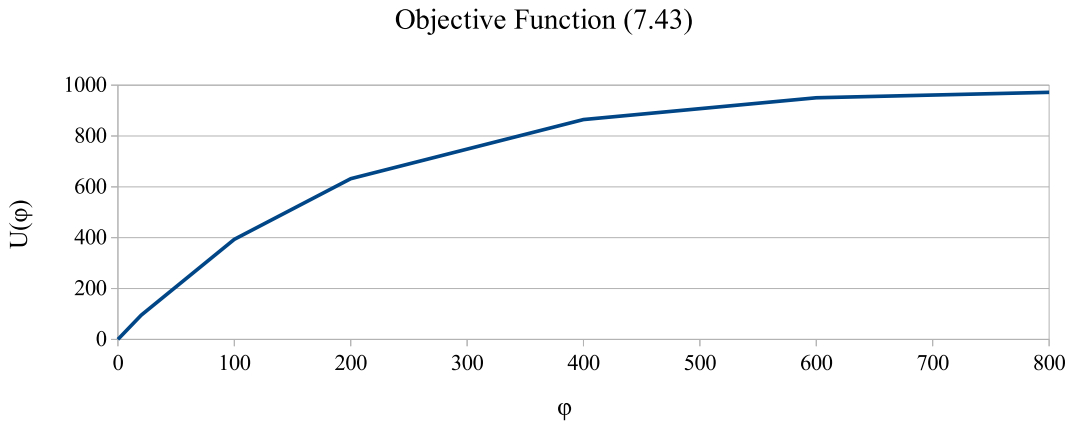


Figure 7.4 Piecewise linear approximation of Objective function (7.43).

the number of edge nodes V^e of the network (which are manually selected by us). The commodity set D^e is built by taking the SNDLib traffic matrices and then discarding all the demands generated between non-edge nodes. Furthermore, for each demand set D^e we generate two different traffic scenarios by randomly (with a uniform distribution) choosing Δ_d in the closed interval $[0, 10]$. The whole set of network instances is summarized in Table 7.2.

Table 7.1 Router chassis and line Cards

case	device	capacity	hourly power cons.
all	Chassis Juniper M10i	16Gbps	86.4 W
1	Gigabit-Eth 1 port	2 Gbps	7.3 W
2	Fast-Eth 12 ports	2.4 Gbps	18.6 W
3	Gigabit-Eth 4 ports	8 Gbps	31 W
4	SONET/SDH OC-48c	5 Gbps	41.4 W

Table 7.2 Network data.

Net			**1**		**2**		**3**		**4**		**5**	
Net	$\|V\|$	$\|A\|$	$\|V^e\|$	$\|D^e\|$	$\|V^e\|$	$\|D^e\|$	$\|V^e\|$	$\|D^e\|$	$\|V^e\|$	$\|D^e\|$	$\|V^e\|$	$\|D^e\|$
polska	12	36	4	6	5	10	6	21	7	28	8	36
abilene	15	44	4	12	5	20	6	30	7	42	8	56
geant	22	72	5	20	6	30	7	42	8	56	9	72
nobel-eu	28	82	9	36	10	45	11	55	12	66	13	78

7.6.2 Numerical Results

In this group of tests we experiment with both exact and restricted path MILPs for SEANM-ET with MMF allocation. The experiments on SEANM-ET with PF allocation are left to future work. All formulations are solved with CPLEX 12.4.0.1 running on a machine equipped with 8Gb of RAM and an Intel i7 processors with 4 core and multi-thread 8x. CPLEX is run by means of the AMPL modeling language.

In the first set of experiments we consider the two smallest networks, i.e., `polska` and `abilene`, and both network utility functions (7.42) and (7.43). Results obtained with the exact MILP formulation for SEANM-ET with MMF allocation are illustrated in Figure 7.5. Each plot shows the network utility (Y-axis)/energy consumption (X-axis) trend observed by varying the total power budget B. Each trace represents the normalized utility values obtained by considering a specific number of edge nodes (remind that more edge nodes means more elastic demands).

For each instance we solve the exact MILP five times, by increasing from time to time the value of B. The smallest B corresponds to the minimum energy consumption required to connect all the edge nodes, while the largest one is equal to the overall consumption of the full active network. Note that both energy and utility values are normalized w.r.t. to those computed, for each edge node configuration, with the smallest B. That means that each trace starts from the point $(1, 1)$. Furthermore, all values are computed by averaging over the four instances obtained by combining the two equipment configurations with the two traffic scenarios generated for each network.

As expected, the normalized utility grows as soon as the power budget is increased. The marginal beneficial effects (in terms of overall utility) observed when B is incremented, tends to decrease as the maximum budget is approached. With Objective function (7.42) (overall throughput maximization), the normalized utility is increased by 2.5 (`polska`) and 2.8 times (`abilene`) w.r.t. the minimum budget instances. Due to its saturating effect (higher concavity), with Objective function (7.43) we observe a lower increment, with a normalized utility which in the best case is close to 1.8 and 1.65. Note that the X-extension of the plots tends to decrease as the number of edge nodes is increased: in fact, while the maximum budget B^{MAX} of the full active network remains the same for each edge node configuration, it is not the same for the minimum budget B^{MIN}, which results to be higher when more edge nodes (they cannot be put to sleep) are considered.

Finally, let us observe the last point of each curve, which represents the normalized network utility achieved by applying the so-called *standard configuration*, according to which (i) all network elements are active, (ii) each link has an administrative weight equal to $1/c_{ij}$ (the weight setting typically used by the operators), (iii) and each elastic traffic demand

is routed along a single shortest path determined by the whole set of link weights. It is worth pointing out that, thanks to our optimized approach, the normalized network utility of the *standard configuration* can be achieved by drastically reducing the energy budget B. Furthermore, when the network is completely activated, the optimization framework produces a utility increase between 10% and 30%.

In Table 7.3, we report other interesting results concerning both the exact formulation and the restricted path heuristic with $\Omega = 4$. Note that Ω determines the number of pre-computed paths considered for each demand, and the higher Ω the higher the number of paths (we refer the reader to Section 4.4 for the detailed description of Ω). In columns Gap_{opt}, Gap_{milp}, t_m and t_h we report, respectively, the gap between the solution computed within the time-limit and the best upper bound provided by CPLEX, the gap between the restricted-path utility and that of the exact MILP, the computing time in seconds used by CPLEX to solve the exact formulation, and that spent, always by CPLEX, to solve the restricted-path. Since we impose a time-limit of 1 hour, it happens that Gap_{opt} can be larger than 0.

We observe that when the overall throughput is optimized (Objective (7.42)), Gap_{opt} is on average around 2.5%, while it 90^{th} percentile values is up to 7% (with both `polska` and `abilene`). Figures are significantly better with the concave utility function (7.43), for which the 90^{th} percentile optimality gap is even 0.4% with `polska` and 3.9% with `abilene`. We believe that this performance improvement can be attributed to the concavity of Objective function (7.43), which naturally drives the branch-and-cut algorithm of CPLEX to allocate resources in a MMF way.

For what concerns the performance of the the restricted-path formulation, the 90^{th} percentile Gap_{milp} is around 6% and 3.2% with, respectively, Objective function (7.42) and Objective function (7.43). This result is quite satisfactory, since it confirms that the utility degradation caused by path restriction remains acceptable, with much smaller computing times (on average by 1000 seconds).

In the last set of tests, we solve the restricted path MILP for SEANM-ET with MMF allocation by considering the the two largest networks, namely `geant` and `nobel-eu`. These instances cannot be efficiently solved within a reasonable time limit by the exact MILP. The set of pre-computed paths is generated by considering Ω equal to 2 and 4, and a time limit of one hour is imposed to CPLEX. In Table 7.4 we report the gap between the utility obtained with $\Omega = 4$ and that achieved with $\Omega = 2$, i.e., $g^{\Omega 2}_{\Omega 4}$, the gap between the *standard configuration* utility and that obtained with $\Omega = 2$, i.e., $g^{\Omega 2}_{standard}$, and $\Omega = 4$, i.e., $g^{\Omega 4}_{standard}$ (with the full active network).

In `geant`, varying Ω from 2 to 4, i.e. doubling the number of pre-computed paths, improves the average network utility by 2.5% (Objective (7.42)) and 1.3% (Objective (7.43)).

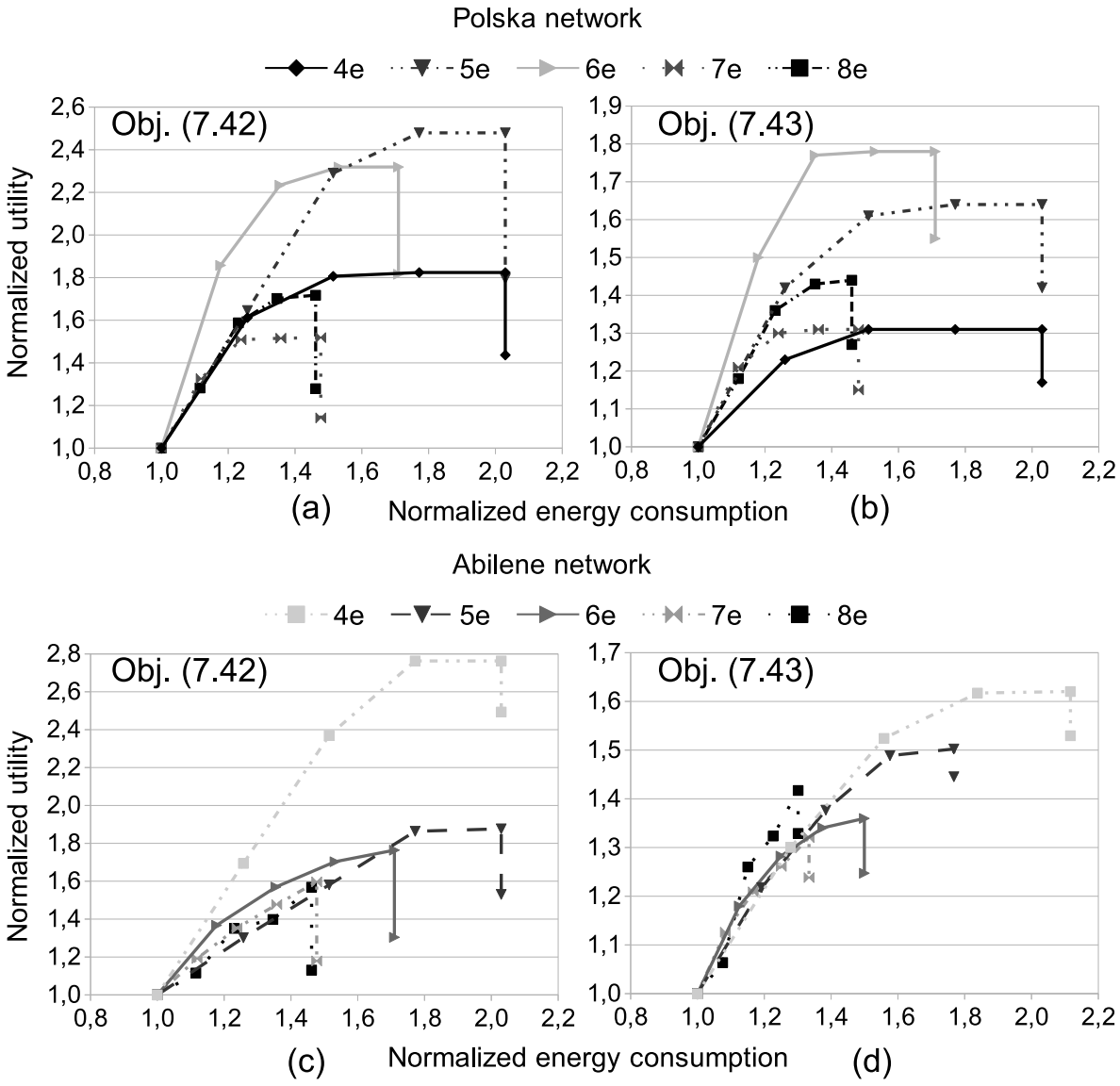


Figure 7.5 Results with `polska` and `abilene` networks: MILP for SEANM-ET with MMF allocation.

It is instead quite surprising that, in `nobel-eu`, doubling the path set causes an average utility reduction of 0.5% with both Objective functions. According to the time availability (time limit), it is therefore crucial to find the right trade-off between solution optimality and overall complexity: additional paths mean more possibilities to improve the final solutions, but also more variables and constraints that must be managed by the solver.

For what concerns the comparison with *standard configuration* solutions (with fully activated networks), the optimized framework improves the average utility by about 33% (Objective (7.42)) and 18% (Objective (7.43)).

To conclude, let us analyze in Figure 7.6 the plots representing the normalized utility/-

Table 7.3 Comparison between the solutions found by the exact MILP for MMF SEANM-ET and the restricted-path heuristic: time-limit of 1 hour, $\Omega = 4$ (see Section 4.4 for the meaning Ω).

	polska				**abilene**			
obj-(7.42)	Gap_{opt}	Gap_{milp}	$t_m\,(s)$	$t_h\,(s)$	Gap_{opt}	Gap_{milp}	t_m	t_h
mean	2.8%	1.8%	2.5k	0.3k	1.6%	2.4%	1.3k	0.7k
st.dev	4.9 %	3.9 %	2.8k	0.3k	2.9%	3.2%	1.6k	1.2k
90perc	6.6%	5.3%	tl	0.8k	6.3%	7.3%	tl	tl
obj-(7.43)	Gap_{opt}	Gap_{milp}	$t_m\,(s)a$	$t_h\,(s)$	Gap_{opt}	Gap_{milp}	t_m	t_h
mean	0.2%	0.7%	1.2k	70	1.0%	1.0%	1.3k	0.8k
st.dev	0.6 %	1.3 %	1.7k	0.4k	2.1%	2.7%	1.6k	1.4k
90perc	0.4%	3.2%	tl	0.2k	3.9%	3.1%	tl	tl

Table 7.4 Comparative results between the restricted-path formulation with $\Omega = 2$, $\Omega = 4$, and *standard configuration* solutions.

	geant			**nobel-eu**		
obj-(7.42)	$g^{\omega 2}_{\omega 4}$	$g^{\omega 2}_{standard}$	$g^{\omega 4}_{standard}$	$g^{\omega 4}_{\omega 4}$	$g^{\omega 2}_{standard}$	$g^{\omega 4}_{standard}$
mean	-2.5%	30.9%	33.4%	-0.4%	31.0%	30.8%
st.dev	2.8%	5.0 %	5.4%	5.5%	9.5%	10.0%
obj-(7.43)	$g^{\omega 2}_{\omega 4}$	$g^{\omega 2}_{standard}$	$g^{\omega 4}_{standard}$	$g^{\omega 2}_{\omega 4}$	$g^{\omega 2}_{standard}$	$g^{\omega 4}_{standard}$
mean	-1.3%	17.4%	18.7%	0.6%	15.9%	14.7%
st.dev	2.8%	7.4%	7.3%	3.6%	7.4%	9.2%

consumption trend obtained by the restricted path formulation ($\Omega = 4$) with `geant` and `nobel-eu`. The plot behaviour is the same observed for the smallest networks `polska` and `abilene`.

7.7 Conclusions

We have formulated a novel bi-level optimization problem to jointly optimize network utility and network power consumption in presence of elastic traffic demands (SEANM-ET). At the upper level, the network operator adjusts the single routing path used by each elastic demand and puts to sleep the unused network devices, while, at lower level, the transport protocol is responsible for allocating the contended network resources among the competing elastic demands. Resource allocation is modelled as a utility/fairness maximization problem subject to link capacity constraints. To efficiently approximate how bandwidth is shared in current IP networks, we consider at the second level both max-min-fairness and proportional-fairness.

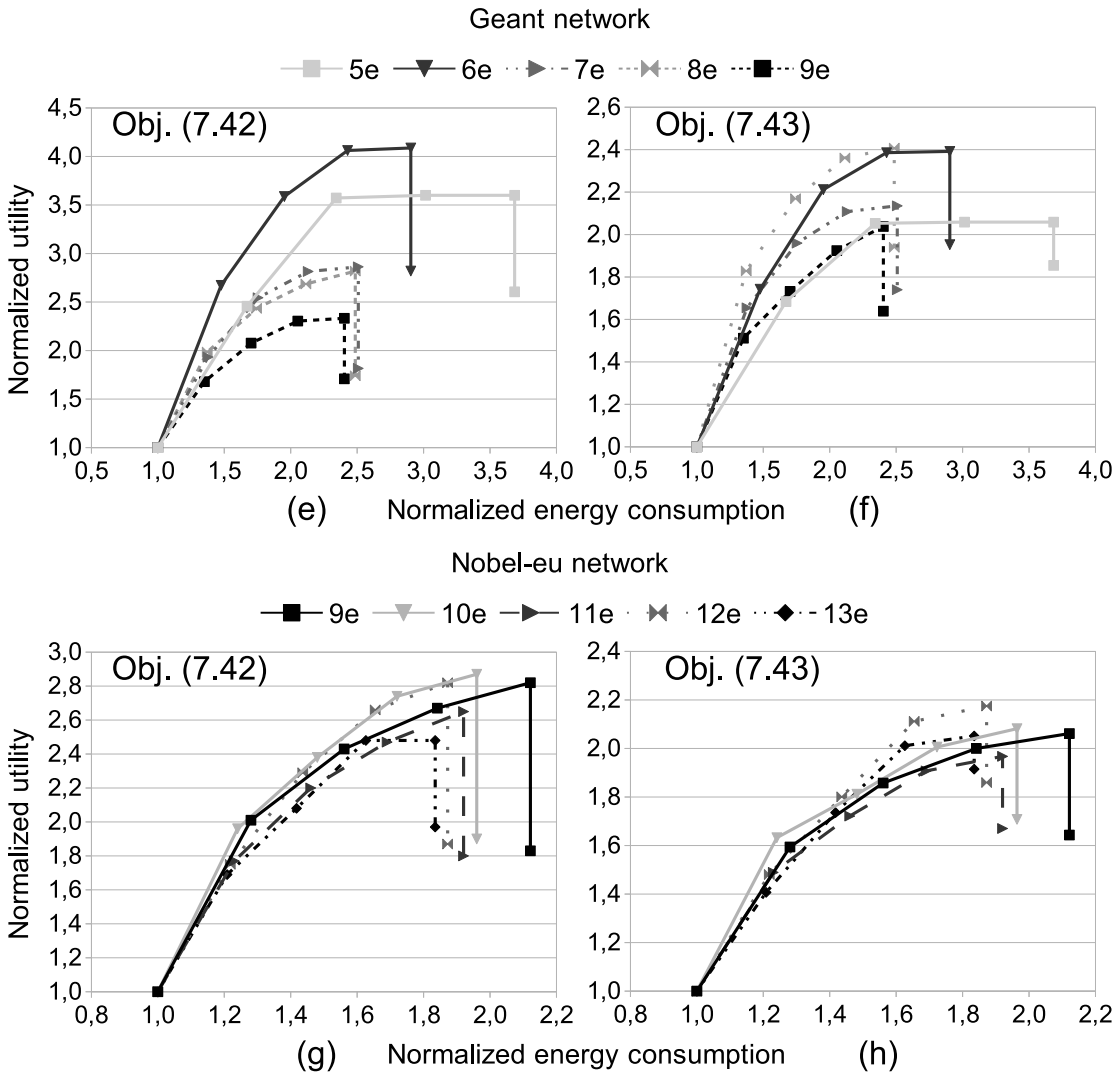


Figure 7.6 Computational results obtained with the restricted path MILP formulation for `geant` and `nobel-eu`.

We have proposed two single level MILP formulations for SEANM-ET subject to, respectively, MMF and PF allocation. While the MMF MILP is exact, some approximations required to suppress non-linearities make the PF MILP heuristic. The exact MMF MILP and its restricted-path variants have been tested on several realistic network instances.

Results have shown that the optimization framework substantially reduces the network power consumption by up to 40% while obtaining the same utility levels achieved by standard network configurations when no element is put to sleep. Due to its considerable complexity, the MMF formulation can be efficiently solved with networks composed by no more than 20 nodes and 50 links.

CHAPTER 8

Conclusions

A promising strategy to drastically reduce the power consumption of IP networks is to manage the whole network configuration in a coordinated manner so as to put to sleep the redundant network devices and minimize the network congestion on the active elements (SEANM). Within our research project we have focused our efforts to develop centralized SEANM approaches for IP networks operated with different configurations and protocols: we have first dealt with networks operated with both flow-based (SEANM-FB) and shortest path (SEANM-SP) routing protocols, while successively we have considered the management of different types of traffic, i.e., elastic or inelastic. To fully guarantee the correct network functioning and evaluate the energy cost of network survivability, we have enhanced the proposed methods for SEANM-FB to explicitly consider additional network constraints regarding resilience to single link failures and robustness to traffic variations. Finally, to practically implement SEANM-SP in a realistic network environment, we have proposed a network management platform built on our novel open-source network management framework called JNetMan.

8.1 Summary of Work

Along our whole research project on green IP networks we have accomplished the following deliverables:

8.1.1 Flow-Based Routing

We have developed an extensive centralized optimization framework to optimize the energy consumption of IP networks operated with a flow-based routing protocol like MPLS (SEANM-FB). Our framework considers a multi-period planning problem, whose aim is to optimize the future network configuration while respecting intra-period constraints on link maximum utilization and single-path routing, and different groups of inter-period constraints. The latter include limitations on both routing variation and line card switching along different time periods.

We propose two exact MILP formulations for both fixed (PAFRP) and variable (PAVRP) routing variants of the reference problem. To efficiently manage instances with up to 80 nodes and 250 links we present also two mathematical-programming heuristic algorithms,

i.e., EA-LG and EA-STH. The first one looks for near optimal solutions by routing all traffic demands one by one, while the second one decomposes the whole multi-period problem among several single period problems (one for each time period) which are solved in sequence. Note that EA-STH works only if routing is variable (PAVRP).

Tests with realistic networks have showed that the daily energy consumption can be reduced by up to 50% without excessively increasing network congestion and degrading the QoS.

8.1.2 Network Survivability

To investigate the energy cost of network resilience, we have enhanced our centralized off-line framework for classic SEANM-FB to explicitly guarantee, along with power consumption minimization, both protection to single link failures and robustness to unexpected traffic variations.

To protect the network against single link failures we consider two different protection schemes, namely *dedicated* and *shared* protection, according to which an additional backup path is assigned to each traffic demand and a certain amount of spare capacity (it depends on the protection scheme) is reserved on each link to cope with the possible failures. Furthermore, we consider a *smart* approach which allows to put to sleep those line cards used by only backup paths, and include a second utilization threshold (higher than the main one) to be respected when a failure occurs.

For what concerns robustness to traffic variations, we adapt to our optimization model a state of the art cardinality-constrained approach (Bertsimas *et al.*, 2011) which considers uncertain parameters, in our case the traffic request of each demand, that vary into a close symmetric interval. Both protected and robust problems can be solved by adapting both PAVRP exact formulation and EA-STH to accounts for the additional network constraints.

Results obtained from an extensive campaign of tests on realistic network instances have showed that the daily network consumption can be still reduced by up to 30% while guaranteeing the highest level of survivability. The energy cost of guaranteeing network survivability has been quantified around 20% of the power consumption produced by the full active network.

8.1.3 Shortest Path Routing

We have developed a centralized optimization framework to optimize the energy consumption of IP networks operated with a shortest path routing protocol like OSPF, i.e., SEANM-SP. We propose to split a single day among different time periods characterized

by a quite constant and predictable level of traffic, and then compute the optimized set of link weights which lexicographically minimizes, in the following order, the network energy consumption and a measure of network congestion.

We have proposed four different heuristic approaches, i.e., GA-ES, GRA-ES, TA-ES and MILP-EWO, each one characterized by a different complexity level.

Extensive tests conducted on realistic network instances have shown that, thanks to the energy-aware optimization of the link weights, network energy consumption can be reduced by up to 40% in presence of moderate traffic levels.

8.1.4 Practical Implementation

We have developed a centralized network management platform to implement, in an on-line fashion, the link weight configurations computed by MILP-EWO during the planning phase. The management platform, which is built on our novel open-source network management framework called JNetMan, allows to combine both off-line and on-line optimization to offer both network stability and reactivity to real-time events. Real-time link load data are exploited to determine, by means of a utilization-based policy, if the current link weight configuration must be replaced by a novel set of weights. All the applicable weight configurations are contained in a database which is built during the planning phase. The very popular management protocol called SNMP is exploited to periodically collect link load measurements and practically adjust the desired link weights. Tests carried on emulated network environments showed that JNetMan can be efficiently exploited to dynamically adjust the link weights without incurring in undesirable oscillations between different weight configurations.

8.1.5 Elastic Traffic

We have formulated a novel bi-level optimization problem to jointly optimize network utility and network power consumption in presence of elastic traffic demands (SEANM-ET). At the upper level, the network operator adjusts the single routing path used by each elastic demand and puts to sleep the unused network devices, while, at lower level, the transport protocol is responsible for allocating the contended network resources among the competing elastic demands. Resource allocation is modelled as a utility/fairness maximization problem subject to link capacity constraints. To efficiently approximate how bandwidth is shared in current IP networks, we consider at the second level both Max-Min-Fairness and Proportional-Fairness.

We have proposed two single level MILP formulations for SEANM-ET subject to, respectively, MMF and PF allocation. While the MMF MILP is exact, some approximations

required to suppress non-linearities make the PF MILP heuristic. The exact MMF MILP and its restricted-path variants have been tested on several realistic network instances.

Results have shown that the optimization framework allows to substantially reduce the network power consumption by up to 40% while obtaining the same utility levels achieved by standard network configurations when no element is put to sleep.

8.2 Limitations of the Proposed Solutions

From the optimization point of view, the whole suite of proposed algorithms allows to efficiently handle network instances up to 150 nodes and 600 links for SEANM-SP, 80 nodes and 300 links for SEANM-FB, and 40 nodes and 150 links for SEANM-ET. Novel approaches should be designed to further increase the instance dimensions.

The most evident limitation of our entire work concerns the practical implementation of the proposed approaches, which by now has been partially realized only for SEANM-SP. Testing the proposed method in a real network environment would allow to:

1. Evaluate network performance during transition periods (when the configuration is modified).
2. Empirically measure the power consumption reduction.
3. Practically implement sleeping transitions.
4. Evaluate the reliability of network configurations based on traffic predictions.

8.3 Future Developments

We leave as future work the improvement of JNetMan to:

1. Integrate in our management platform a new functionality to adjust the MPLS routing paths. This would allow to realistically implement our approaches for SEANM-FB and SEANM-ET.
2. Include the possibility of explicitly putting to sleep and awake both chassis and line cards.

For what concerns SEANM-ET, in future we aim (i) to extensively test the PF based approach and (ii) to develop a novel highly scalable resolution method for both MMF and PF problems to cope with larger instances. Furthermore we believe that tests in real network scenarios will allow to asses how well MMF and PF approximate the allocation of elastic demands.

REFERENCES

(2008). ITU-T G.984.1 Gigabit-capable passive optical networks (GPON): General characteristics. http://www.itu.int/rec/T-REC-G.984.1-200803-I/en.

ADDIS, B., CAPONE, A., CARELLO, G., GIANOLI, L. et SANSÒ, B. (2012a). Energy aware management of resilient networks with shared protection. *Sustainable Internet and ICT for Sustainability (SustainIT), 2012*. IEEE, Pisa, 1–9.

ADDIS, B., CAPONE, A., CARELLO, G., GIANOLI, L. et SANSO, B. (2012b). Energy-aware multiperiod traffic engineering with flow-based routing. *2012 International Conference on Communications (ICC)*. IEEE, 5957–5961.

ADDIS, B., CAPONE, A., CARELLO, G., GIANOLI, L. et SANSO, B. (2012c). Multi-period traffic engineering of resilient networks for energy efficiency. *2012 Online Conference on Green Communications (GreenCom)*. IEEE, 14–19.

ADDIS, B., CAPONE, A., CARELLO, G., GIANOLI, L. et SANSO, B. (2013). A robust optimization approach for energy-aware routing in MPLS networks. *2013 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 567–572.

ADDIS, B., CAPONE, A., CARELLO, G., GIANOLI, L. et SANSO, B. (2014a). Energy management through optimized routing and device powering for greener communication networks. *IEEE/ACM Transactions on Networking*, 22, 313–325.

ADDIS, B., CAPONE, A., CARELLO, G., GIANOLI, L. et SANSO, B. (2014b). On the energy cost of robustness and resiliency in ip networks. *submitted for publications to Elsevier Computer Networks*.

ADELIN, A., OWEZARSKI, P. et GAYRAUD, T. (2010). On the impact of monitoring router energy consumption for greening the internet. *2010 11th IEEE/ACM International Conference on Grid Computing*. IEEE, 298–304.

AHMAD, A., BIANCO, A., BONETTO, E., CHIARAVIGLIO, L. et IDZIKOWSKI, F. (2013). Energy-aware design of multilayer core networks [invited]. *Journal of Optical Communications and Networking*, 5, A127.

AJMONE MARSAN, M., CHIARAVIGLIO, L., CIULLO, D. et MEO, M. (2009). Optimal energy savings in cellular access networks. *2009 International Conference on Communications Workshops*. IEEE, 1–5.

ALDRAHO, A. et KIST, A. (2012). Enabling energy efficient and resilient networks using dynamic topologies. *Sustainable Internet and ICT for Sustainability (SustainIT), 2012*. Pisa, 1–8.

ALTIN, A., FORTZ, B., THORUP, M. et ÜMIT, H. (2009). Intra-domain traffic engineering with shortest path routing protocols. *4OR*, 7, 301–335.

ALTMAN, E., BARAKAT, C., LABORDE, E., ANTIPOLIS, S., ALTMAN, E., BROWN, P., COLLANGE, D., CNET, F. et BROWN, E. (2000). Fairness analysis of tcp/ip. *Proceedings of the 39th IEEE Conference on Decision and Control*. 61–66.

AMALDI, E., CAPONE, A., CONIGLIO, S. et GIANOLI, L. (2013a). Energy-aware traffic engineering with elastic demands and mmf bandwidth allocation. *2013 18th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, 169–174.

AMALDI, E., CAPONE, A., CONIGLIO, S. et GIANOLI, L. (2013b). Network optimization problems subject to max-min fair flow allocation. *IEEE Communications Letters*, 17, 1463–1466.

AMALDI, E., CAPONE, A. et GIANOLI, L. (2013c). Energy-aware ip traffic engineering with shortest path routing. *Computer Networks*, 57, 1503–1517.

AMALDI, E., CAPONE, A., GIANOLI, L. et MASCETTI, L. (2011a). Energy management in ip traffic engineering with shortest path routing. *2011 International Symposium on a World of Wireless, Mobile and Multimedia Networks*. IEEE, 1–6.

AMALDI, E., CAPONE, A., GIANOLI, L. et MASCETTI, L. (2011b). A milp-based heuristic for energy-aware traffic engineering with shortest path routing. *Network Optimization*, 464–477.

AMALDI, E., CONIGLIO, S., GIANOLI, L. et ILERI, C. (2013d). On single-path network routing subject to max-min fair flow allocation. *Electronic Notes in Discrete Mathematics*, 41, 543–550.

ARABAS, P., MALINOWSKI, K. et SIKORA, A. (2012). On formulation of a network energy saving optimization problem. *2012 Fourth International Conference on Communications and Electronics (ICCE)*. IEEE, 227–232.

AVALLONE, S. et VENTRE, G. (2012). Energy efficient online routing of flows with additive constraints. *Computer Networks*, 56, 2368–2382.

BALDI, M. et OFEK, Y. (2009). Time for a "greener" internet. *2009 International Conference on Communications Workshops*. IEEE, 1–6.

BALIGA, J., AYRE, R., HINTON, K. et TUCKER, R. (2011). Energy consumption in wired and wireless access networks. *IEEE Communications Magazine*, 49, 70–77.

BAO, N., LI, L., YU, H., ZHANG, Z. et LUO, H. (2012). Power-aware provisioning strategy with shared path protection in optical wdm networks. *Optical Fiber Technology*, 18, 81–87.

BEN-TAL, A., GHAOUI, L. E. et NEMIROVSKI, A. (2009). *Robust optimization*. Princeton University Press.

BERTSEKAS, D., GALLAGER, R. et HUMBLET, P. (1987). *Data networks*, vol. 2. Prentice-hall Englewood Cliffs, NJ.

BERTSIMAS, D., BROWN, D. et CARAMANIS, C. (2011). Theory and applications of robust optimization. *SIAM review*, 53, 464–501.

BIANCO, A., DEBELE, F. et GIRAUDO, L. (2012). Energy saving in distributed router architectures. *2012 International Conference on Communications (ICC)*. IEEE, 2951–2955.

BIANZINO, A., CHAUDET, C., LARROCA, F., ROSSI, D. et ROUGIER, J. (2010). Energy-aware routing: A reality check. *2010 Globecom Workshops*. IEEE, Miami, 1422–1427.

BIANZINO, A., CHAUDET, C., MORETTI, S., ROUGIER, J., CHIARAVIGLIO, L. et LE ROUZIC, E. (2012a). Enabling sleep mode in backbone ip-networks: A criticality-driven tradeoff. *2012 International Conference on Communications (ICC)*. IEEE, 5946–5950.

BIANZINO, A., CHAUDET, C., ROSSI, D. et ROUGIER, J. (2012b). A survey of green networking research. *IEEE Communications Surveys & Tutorials*, 14, 3–20.

BIANZINO, A., CHIARAVIGLIO, L. et MELLIA, M. (2012c). Distributed algorithms for green ip networks. *2012 Proceedings IEEE INFOCOM Workshops*. IEEE, 121–126.

BIANZINO, A., CHIARAVIGLIO, L., MELLIA, M. et ROUGIER, J. (2012d). Grida: Green distributed algorithm for energy-efficient ip backbone networks. *Computer Networks*, 56, 3219–3232.

BILAL, K., KHAN, S., MADANI, S., HAYAT, K., KHAN, M., MIN-ALLAH, N., KOLODZIEJ, J., WANG, L., ZEADALLY, S. et CHEN, D. (2012). A survey on green communications using adaptive link rate. *Cluster Computing*, 16, 575–589.

BLEY, A. (2010). An integer programming algorithm for routing optimization in ip networks. *Algorithmica*, 60, 21–45.

BOLLA, R., BRUSCHI, R., CARREGA, A., DAVOLI, F., SUINO, D., VASSILAKIS, C. et ZAFEIROPOULOS, A. (2012). Cutting the energy bills of internet service providers and telecoms through power management: An impact analysis. *Computer Networks*, 56, 2320–2342.

BOLLA, R., BRUSCHI, R., CIANFRANI, A. et LISTANTI, M. (2010). Introducing standby capabilities into next-generation network devices. *Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow - PRESTO '10*. ACM Press, New York, New York, USA, vol. 17, 1.

BOLLA, R., BRUSCHI, R., CIANFRANI, A. et LISTANTI, M. (2011a). Enabling backbone networks to sleep. *IEEE Network*, 25, 26–31.

BOLLA, R., BRUSCHI, R., DAVOLI, F. et CUCCHIETTI, F. (2011b). Energy efficiency in the future internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures. *IEEE Communications Surveys & Tutorials*, 13, 223–244.

BOLLA, R., BRUSCHI, R. et LAGO, P. (2013). The hidden cost of network low power idle. *2013 International Conference on Communications (ICC)*. IEEE, 4148–4153.

BOLLA, R., DAVOLI, F., BRUSCHI, R., CHRISTENSEN, K., CUCCHIETTI, F. et SINGH, S. (2011c). The potential impact of green technologies in next-generation wireline networks: Is there room for energy saving optimization? *IEEE Communications Magazine*, 49, 80–86.

BONETTO, E., CHIARAVIGLIO, L., IDZIKOWSKI, F. et LE ROUZIC, E. (2013). Algorithms for the multi-period power-aware logical topology design with reconfiguration costs. *Journal of Optical Communications and Networking*, 5, 394.

BONETTO, E., MELLIA, M. et MEO, M. (2012). Energy profiling of isp points of presence. *2012 IEEE International Conference on Communications (ICC)*. IEEE, 5973–5977.

BOTERO, J. et HESSELBACH, X. (2013). Greener networking in a network virtualization environment. *Computer Networks*, 57, 2021–2039.

BOTTA, A., DAINOTTI, A. et PESCAPÉ, A. (2012). A tool for the generation of realistic network workload for emerging networking scenarios. *Computer Networks*, 56, 3531–3547.

BREUER, D., GEILHARDT, F., HÜLSERMAN, R., KIND, M., LANGE, C., MONATH, T., WEIS, E. *ET AL.* (2011). Opportunities for next-generation optical access. *IEEE Communications Magazine*, 49, 16–24.

CAO, Z. et ZEGURA, E. (1999). Utility max-min: An application-oriented bandwidth allocation scheme. *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. IEEE, vol. 2, 793–801.

CAPONE, A., CASCONE, C., GIANOLI, L. et SANSO, B. (2013). Ospf optimization via dynamic network management for green ip networks. *2013 Sustainable Internet and ICT for Sustainability (SustainIT)*. IEEE, 1–9.

CAPONE, A., CASCONE, C., GIANOLI, L. et SANSO, B. (2014). Jnetman: An open-source platform for easy implementation and testing of network management approaches. *Working draft*.

CAPONE, A., CORTI, D., GIANOLI, L. et SANSÓ, B. (2012). An optimization framework for the energy management of carrier ethernet networks with multiple spanning trees. *Computer Networks*, 56, 3666–3681.

CARDONA RESTREPO, J., GRUBER, C. et MAS MACHUCA, C. (2009). Energy profile aware routing. *2009 IEEE International Conference on Communications Workshops*. IEEE, 1–5.

CASAS, P., VATON, S., FILLATRE, L. et CHONAVEL, L. (2009). Efficient methods for traffic matrix modeling and on-line estimation in large-scale ip networks. *Proc. of 21st International Teletraffic Congress, ITC 2009*. IEEE, 1–8.

CASCONE, C. (2013). JNetMan, Java framework for effortless development of SNMP-based, proactive, network management applications [Online]. `http://www.jnetman.org/`.

CAVDAR, C., BUZLUCA, F. et WOSINSKA, L. (2010). Energy-efficient design of survivable wdm networks with shared backup. *2010 Global Telecommunications Conference GLOBECOM 2010*. IEEE, 1–5.

CHABAREK, J., SOMMERS, J., BARFORD, P., ESTAN, C., TSIANG, D. et WRIGHT, S. (2008). Power awareness in network design and routing. *2008 INFOCOM - The 27th Conference on Computer Communications*. IEEE, 457–465.

CHANG, C., LIU, Z. et MEMBER, S. (2004). A bandwidth sharing theory for a large number of http-like connections. *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*. vol. 2, 667–675.

CHARALAMBIDES, M., TUNCER, D., MAMATAS, L. et PAVLOU, G. (2013). Energy-aware adaptive network resource management. *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*. Ghent, 369–377.

CHIARAVIGLIO, L., CIANFRANI, A., ROUZIC, E. et POLVERINI, M. (2013a). Sleep modes effectiveness in backbone networks with limited configurations. *Computer Networks*, 57, 2931–2948.

CHIARAVIGLIO, L., CIULLO, D., MELLIA, M. et MEO, M. (2011). Modeling sleep modes gains with random graphs. *2011 Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 355–360.

CHIARAVIGLIO, L., CIULLO, D., MELLIA, M. et MEO, M. (2013b). Modeling sleep mode gains in energy-aware networks. *Computer Networks*, 57, 3051–3066.

CHIARAVIGLIO, L., MELLIA, M. et NERI, F. (2009). Reducing power consumption in backbone networks. *2009 International Conference on Communications*. IEEE, 1–6.

CHIARAVIGLIO, L., MELLIA, M. et NERI, F. (2012). Minimizing ISP Network Energy Cost: Formulation and Solutions. *IEEE/ACM Transactions on Networking*, 20, 463–476.

CHO, J. et CHONG, S. (2007). Utility max-min flow control using slope-restricted utility functions. *IEEE Transactions on Communications*, 55, 963–972.

CHRISTENSEN, K., REVIRIEGO, P., NORDMAN, B., BENNETT, M., MOSTOWFI, M. et MAESTRO, J. (2010). Ieee 802.3az: the road to energy efficient ethernet. *IEEE Communications Magazine*, 48, 50–56.

CIANFRANI, A., ERAMO, V., LISTANTI, M., MARAZZA, M. et VITTORINI, E. (2010). An energy saving routing algorithm for a green ospf protocol. *2010 INFOCOM Conference on Computer Communications Workshops*. IEEE, 1–5.

CIANFRANI, A., ERAMO, V., LISTANTI, M. et POLVERINI, M. (2011). An ospf enhancement for energy saving in ip networks. *2011 Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 325–330.

CIANFRANI, A., ERAMO, V., LISTANTI, M. et POLVERINI, M. (2012a). Introducing routing standby in network nodes to improve energy savings techniques. *Proceedings of the 3rd International Conference on Future Energy Systems Where Energy, Computing and Communication Meet - e-Energy '12*. ACM Press, New York, New York, USA, 1–7.

CIANFRANI, A., ERAMO, V., LISTANTI, M., POLVERINI, M. et VASILAKOS, A. (2012b). An ospf-integrated routing strategy for qos-aware energy saving in ip backbone networks. *IEEE Transactions on Network and Service Management*, 9, 254–267.

CISCO SYSTEMS (2012). Introduction to Cisco IOS NetFlow - A Technical Overview. White paper., Cisco Systems.

COIRO, A., CHIARAVIGLIO, L., CIANFRANI, A., LISTANTI, M. et POLVERINI, M. (2014). Reducing Power Consumption in Backbone IP Networks through Table Lookup Bypass. *Computer Networks*, 64, 125–142.

COIRO, A., LISTANTI, M., VALENTI, A. et MATERA, F. (2013a). Energy-aware traffic engineering: A routing-based distributed solution for connection-oriented ip networks. *Computer Networks*, 57, 2004–2020.

COIRO, A., POLVERINI, M., CIANFRANI, A. et LISTANTI, M. (2013b). Energy saving improvements in ip networks through table lookup bypass in router line cards. *2013 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 560–566.

COLUCCIA, A., D'ALCONZO, A. et RICCIATO, F. (2011). On the optimality of max–min fairness in resource allocation. *Annals of Telecommunications - Annales Des Télécommunications*, 67, 15–26.

COUDERT, D., KOSTER, A., PHAN, T. et TIEVES, M. (2013). Robust redundancy elimination for energy-aware routing. *2013 International Conference on Green Computing and Communications and Internet of Things and Cyber, Physical and Social Computing*. IEEE, no. GreenCom, 179–186.

CUOMO, F., ABBAGNALE, A., CIANFRANI, A. et POLVERINI, M. (2011a). Keeping the connectivity and saving the energy in the internet. *2011 Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 319–324.

CUOMO, F., ABBAGNALE, A. et PAPAGNA, S. (2011b). Esol: Energy saving in the internet based on occurrence of links in routing paths. *2011 International Symposium on a World of Wireless, Mobile and Multimedia Networks*. IEEE, 1–6.

CUOMO, F., CIANFRANI, A., POLVERINI, M. et MANGIONE, D. (2012). Network pruning for energy saving in the internet. *Computer Networks*, 56, 2355–2367.

DANNA, E., HASSIDIM, A., KAPLAN, H. et KUMAR, A. (2012a). Upward max min fairness. *IEEE INFOCOM*. 1–9.

DANNA, E., MANDAL, S. et SINGH, A. (2012b). A practical algorithm for balancing the max-min fairness and throughput objectives in traffic engineering. *Proceedings Ieee Infocom*, 846–854.

DI GREGORIO, L. (2013). Energy budget simulation for deep packet inspection. *2013 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 585–589.

FINAMORE, A., MELLIA, M., MEO, M., MUNAFO, M., TORINO, P. et ROSSI, D. (2011). Experiences of internet traffic monitoring with tstat. *IEEE Network*, 25, 8–14.

FISHER, W., SUCHARA, M. et REXFORD, J. (2010). Greening backbone networks: Reducing energy consumption by shutting off cables in bundled links. *Proceedings of the first ACM SIGCOMM workshop on Green networking - Green Networking '10*. ACM Press, New York, New York, USA, 29.

FORSTER, C., DICKIE, I., MAILE, G., SMITH, H., CRISP, M., CLEMO, G. et OZDEMIROGLU, E. (2009). Understanding the environmental impact of communication systems final report. Rapport technique April.

FORTZ, B. et THORUP, M. (2002). Internet traffic engineering by optimizing OSPF weights. *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*. IEEE, vol. 2, 519–528.

FRALEIGH, C., MOON, S., LYLES, B., COTTON, C., KHAN, M., MOLL, D., ROCKELL, R., SEELY, T. et DIOT, S. (2003). Packet-level traffic measurements from the Sprint IP backbone. *Network*, 17, 6–16.

FRANCOIS, F., MOESSNER, K. et GEORGOULAS, S. (2013a). Optimizing link sleeping reconfigurations in isp networks with off-peak time failure protection. *IEEE Transactions on Network and Service Management*, 10, 176–188.

FRANCOIS, F., WANG, N. et MOESSNER, K. (2013b). Green IGP Link Weights for Energy-efficiency and Load-balancing in IP Backbone Networks. *IFIP Networking Conference, 2013*. Brooklyn, 1–9.

FRATTA, L., GERLA, M. et KLEINROCK, L. (1973). The flow deviation method: An approach to store-and-forward communication network design. *Networks*, 3, 97–133.

GALÁN-JIMÉNEZ, J. et GAZO-CERVERO, A. (2013a). Elee: Energy levels-energy efficiency tradeoff in wired communication networks. *Communications Letters, IEEE*, 17, 166–168.

GALÁN-JIMÉNEZ, J. et GAZO-CERVERO, A. (2013b). Using bio-inspired algorithms for energy levels assessment in energy efficient wired communication networks. *Journal of Network and Computer Applications*.

GAN, L., WALID, A. et LOW, S. (2012). Energy-efficient congestion control. *ACM SIGMETRICS Performance Evaluation Review*, 40, 89.

GARROPPO, R., GIORDANO, S., NENCIONI, G., PAGANO, M., PISA, U., SCUTELLÀ, M. et INFORMATICA, D. (2012). Energy saving heuristics in backbone networks. *Sustainable Internet and ICT for Sustainability (SustainIT), 2012*. Pisa, 1–9.

GARROPPO, R., GIORDANO, S., NENCIONI, G. et SCUTELLA, M. (2011). Network power management: Models and heuristic approaches. *2011 Global Telecommunications Conference - GLOBECOM 2011*. IEEE, 1–5.

GARROPPO, R., GIORDANO, S., NENCIONI, G. et SCUTELLÀ, M. (2013a). Mixed integer non-linear programming models for green network design. *Computers & Operations Research*, 40, 273–281.

GARROPPO, R., NENCIONI, G., TAVANTI, L. et SCUTELLA, M. (2013b). Does traffic consolidation always lead to network energy savings? *IEEE Communications Letters*, 17, 1852–1855.

GELENBE, E. (1993). Learning in the Recurrent Random Neural Network. *Neural Computation*, 5, 154–164.

GELENBE, E., LENT, R. et NUNEZ, A. (2004). Self-aware networks and QoS. *Proceedings of the IEEE*, 92, 1478–1489.

GELENBE, E. et MAHMOODI, T. (2011). Energy-aware routing in the cognitive packet network. *International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies (Energy'11)*.

GENDRON, B., CRAINIC, T. et FRANGIONI, A. (1998). Multicommodity capacitated network design. *Telecommunications Network Planning*, 1–19.

GIROIRE, F., MAZAURIC, D., MOULIERAC, J. et ONFROY, B. (2010). Minimizing routing energy consumption: From theoretical to practical results. *2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*. IEEE, 252–259.

GIROIRE, F., MOULIERAC, J., PHAN, T. et ROUDAUT, F. (2012). Minimization of network power consumption with redundancy elimination. *NETWORKING 2012*, 247–258.

GREENTOUCH CONSORTIUM (s.d.). www.greentouch.org.

GROUP, T. C. (2008). SMART 2020: Enabling the low carbon economy in the information age. *2010 State of Green Business*.

GUNARATNE, C. et CHRISTENSEN, K. (2006). Ethernet Adaptive Link Rate: System Design and Performance Evaluation. *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, 28–35.

GUNARATNE, C., CHRISTENSEN, K., NORDMAN, B. et SUEN, S. (2008). Reducing the energy consumption of ethernet with adaptive link rate (alr). *IEEE Transactions on Computers*, 57, 448–461.

GUNARATNE, C., CHRISTENSEN, K. et SUEN, S. (2006). NGL02-2: Ethernet Adaptive Link Rate (ALR): Analysis of a Buffer Threshold Policy. *IEEE Globecom 2006*. IEEE, 1–6.

GUNARATNE, P. et CHRISTENSEN, K. (2005). Predictive power management method for network devices. *Electronics Letters*, 41, 775.

GUPTA, M. et SINGH, S. (2003). Greening of the internet. *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications*. 19–26.

HAN, H., SHAKKOTTAI, S., HOLLOT, C., SRIKANT, R. et TOWSLEY, D. (2006). Multi-path tcp: A joint congestion control and routing scheme to exploit path diversity in the internet. *IEEE/ACM Transactions on Networking*, 14, 1260–1271.

HANNAN, A., BAILEY, B. et NI, L. (2000). Traffic engineering with MPLS in the Internet. *IEEE Network*, 14, 28–33.

HARKS, T. et POSCHWATTA, T. (2005). Utility fair congestion control for real-time traffic. *Proceedings 24th Annual Joint Conference of the IEEE Computer and Communications Societies.* IEEE, vol. 4, 2786–2791.

HAYS, R. (2007). Active/idle toggling with 0base-x for energy efficient ethernet. presentation to the IEEE 802.3az Task Force, Nov. 2007.

HE, J., BRESLER, M., CHIANG, M. et REXFORD, J. (2007). Towards robust multilayer traffic engineering: Optimization of congestion control and routing. *IEEE Journal on Selected Areas in Communications*, 25, 868–880.

HE, J., CHIANG, M. et REXFORD, J. (2006). Tcp/ip interaction based on congestion price: Stability and optimality. *2006 International Conference on Communications*. IEEE, vol. 00, 1032–1039.

HE, R. et LIN, B. (2013). Dynamic power-aware shared path protection algorithms in wdm mesh networks. *Journal of Communications*, 8, 55–65.

HINTON, K., BALIGA, J., FENG, M., AYRE, R. et TUCKER, R. (2011). Power consumption and energy efficiency in the internet. *IEEE Network*, 25, 6–12.

HO, K. et CHEUNG, C. (2010). Green distributed routing protocol for sleep coordination in wired core networks. *Networked Computing (INC), 2010 6th International Conference on*. IEEE, Gyeongju, 1–6.

HOU, C., ZHANG, F., ANTA, A., WANG, L. et LIU, Z. (2014). A hop-by-hop energy efficient distributed routing scheme. *ACM SIGMETRICS Performance Evaluation Review*, 41, 101–106.

IDZIKOWSKI, F., BONETTO, E., CHIARAVIGLIO, L., CIANFRANI, A., COIRO, A., DUQUE, R., JIMÉNEZ, F., ROUZIC, E., MUSUMECI, F., HEDDEGHEM, W., VIZCAíNO, J. et YE, Y. (2013a). Trend in energy-aware adaptive routing solutions. *IEEE Communications Magazine*, 51, 94–104.

IDZIKOWSKI, F., CHIARAVIGLIO, L. et BONETTO, E. (2012). Ewa: An adaptive algorithm for energy saving in ip-over-wdm networks. *2012 17th European Conference on Networks and Optical Communications*, 1–6.

IDZIKOWSKI, F., CHIARAVIGLIO, L., DUQUE, R., JIMENEZ, F. et LE ROUZIC, E. (2013b). Green horizon: Looking at backbone networks in 2020 from the perspective of network operators. *2013 IEEE International Conference on Communications (ICC)*. IEEE, vol. 2013, 4455–4460.

IDZIKOWSKI, F., ORLOWSKI, S., RAACK, C., WOESNER, H. et WOLISZ, A. (2010). Saving energy in ip-over-wdm networks by switching off line cards in low-demand scenarios. *2010 14th Conference on Optical Network Design and Modeling (ONDM)*. IEEE, 1–6.

IDZIKOWSKI, F., ORLOWSKI, S., RAACK, C., WOESNER, H. et WOLISZ, A. (2011). Dynamic routing at different layers in ip-over-wdm networks — maximizing energy savings. *Optical Switching and Networking*, 8, 181–200.

J. WANG, S. L. et DOYLE, J. (2005). Cross-layer optimization in tcp/ip networks. *IEEE/ACM Transactions on Networking*, 13, 582–595.

JAKMA, P., JARDIN, V., D.LAMPARTER, SCHORR, A., TEJBLUM, D. et TROXEL, G. (2012). Quagga routing software suite [online]. `http://www.nongnu.org/quagga/`.

JIRATTIGALACHOTE, A., CAVDAR, C., MONTI, P., WOSINSKA, L. et TZANAKAKI, A. (2011). Dynamic provisioning strategies for energy efficient wdm networks with dedicated path protection. *Optical Switching and Networking*, 8, 201–213.

KELLY, F., MAULLOO, A. et TAN, D. (1998). Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49, 237–252.

KIST, A. et ALDRAHO, A. (2011). Dynamic topologies for sustainable and energy efficient traffic routing. *Computer Networks*, 55, 2271–2288.

KOOMEY, J. (2007). Estimating total power consumption by servers in the us and the world. Rapport technique.

KOSTER, A., PHAN, T. et TIEVES, M. (2013). Extended Cutset Inequalities for the Network Power Consumption Problem. vol. 41, 69–76.

LAMBERT, S., VAN HEDDEGHEM, W., VEREECKEN, W., LANNOO, B., COLLE, D. et PICKAVET, M. (2012). Worldwide electricity consumption of communication networks. *Optics express*, 20, B513–24.

LANGE, C. (2009). Energy-related aspects in backbone networks. *Proceedings of 35th European Conference on Optical Communication (ECOC 2009)*.

LANGE, C., KOSIANKOWSKI, D., WEIDMANN, R. et GLADISCH, A. (2011). Energy consumption of telecommunication networks and related improvement options. *IEEE Journal of Selected Topics in Quantum Electronics*, 17, 285–295.

LEDUC, G., ABRAHAMSSON, H., BALON, S., BESSLER, S., D'ARIENZO, M., DELCOURT, O., DOMINGO-PASCUAL, J., CERAV-ERBAS, S., GOJMERAC, I., MASIP, X., PESCAPH, A., QUOITIN, B., ROMANO, S., SALVATORI, E., SKIVÈ, F., TRAN, H., UHLIG, S. et MIT, H. (2006). An open source traffic engineering toolbox. *Computer Communications*, 29, 593–610.

LEE, S., LI, K. et CHEN, A. (2013). Survivable green active topology design and link weight assignment for ip networks with notvia fast failure reroute. *2013 International Conference on Communications (ICC)*. IEEE, 3575–3580.

LEE, S., TSENG, P. et CHEN, A. (2012). Link weight assignment and loop-free routing table update for link state routing protocols in energy-aware internet. *Future Generation Computer Systems*, 28, 437–445.

LEE, Y. et REDDY, A. (2012). Multipath routing for reducing network energy. *Online Conference on Green Communications (GreenCom), 2012 IEEE*. 44–49.

LIN, G., SOH, S., CHIN, K. et LAZARESCU, M. (2013). Efficient heuristics for energy-aware routing in networks with bundled links. *Computer Networks*, 57, 1774–1788.

LIN, X. et SHROFF, N. (2006). Utility maximization for communication networks with multipath routing. *IEEE Transactions on Automatic Control*, 51, 766–781.

LOW, S. (2003). A duality model of tcp and queue management algorithms. *IEEE/ACM Transactions on Networking*, 11, 525–536.

LUBRITTO, C., PETRAGLIA, A., VETROMILE, C., CATERINA, F., D'ONOFRIO, A., LOGORELLI, M., MARSICO, G. et CURCURUTO, S. (2008). Telecommunication power systems: Energy saving, renewable sources and environmental monitoring. *INTELEC 2008 - 2008 IEEE 30th International Telecommunications Energy Conference*. IEEE, 1–4.

LUO, B., LIU, W. et AL-ANBUKY, A. (2013). Energy aware survivable routing approaches for next generation networks. *2013 Australasian Telecommunication Networks and Applications Conference (ATNAC)*. Ieee, 160–165.

MACKAREL, A., MARAY, T., MOTH, J., NORRIS, M., PEKAL, R., SOWINSKI, R., VASSILAKIS, C. et ZAFEIROPOULOS, A. (2011). Geant deliverable dn3.5.2: Study of environmental impact. Rapport technique 238875.

MAHADEVAN, P., SHARMA, P., BANERJEE, S. et RANGANATHAN, P. (2009). A power benchmarking framework for network devices. *NETWORKING 2009*, 795–808.

MALMODIN, J., MOBERG, S., LUNDÉN, D., FINNVEDEN, G. et LÖVEHAGEN, N. (2010). Greenhouse gas emissions and operational electricity use in the ict and entertainment & media sectors. *Journal of Industrial Ecology*, 14, 770–790.

MASSOULIE, L. et ROBERTS, J. (2002). Bandwidth sharing: objectives and algorithms. *IEEE/ACM Transactions on Networking*, 10, 320–328.

MELLAH, H. et SANSÒ, B. (2009). Review of facts, data and proposals for a greener internet. *Proceedings of the 6th International ICST Conference on Broadband Communications, Networks and Systems*. IEEE.

MINAMI, M. et MORIKAWA, H. (2008). Some open challenges for improving the energy efficiency of the internet. *Proc. 3rd International Conference on Future Internet (CFI 2008)*.

MONTI, P., MUHAMMAD, A., CERUTTI, I., CAVDAR, C., WOSINSKA, L., CASTOLDI, P. et TZANAKAKI, A. (2011). Energy-efficient lightpath provisioning in a static wdm network with dedicated path protection. *2011 13th International Conference on Transparent Optical Networks*. IEEE, 1–5.

MUHAMMAD, A., MONTI, P., CERUTTI, I., WOSINSKA, L. et CASTOLDI, P. (2013). Reliability differentiation in energy efficient optical networks with shared path protection. *2013 Online Conference on Green Communications (OnlineGreenComm)*. IEEE, 64–69.

MUHAMMAD, A., MONTI, P., CERUTTI, I., WOSINSKA, L., CASTOLDI, P. et TZANAKAKI, A. (2010). Energy-efficient wdm network planning with dedicated protection resources in sleep mode. *2010 Global Telecommunications Conference GLOBECOM 2010*. IEEE, 1–5.

MUKHERJEE, B. (2000). WDM optical communication networks: progress and challenges. *IEEE Journal on Selected Areas in Communications*, 18, 1810–1824.

MUKHERJEE, B. (2002). Traffic grooming in an optical WDM mesh network. *IEEE Journal on Selected Areas in Communications*, 20, 122–133.

MUMEY, B., TANG, J. et HASHIMOTO, S. (2012). Enabling green networking with a power down approach. *2012 International Conference on Communications (ICC)*. IEEE, 2867–2871.

NACE, D. et PIORO, M. (2008). Max-min fairness and its applications to routing and load-balancing in communication networks: a tutorial. *IEEE Communications Surveys & Tutorials*, 10, 5–17.

NEDEVSCHI, S., CHANDRASHEKAR, J., LIU, J., NORDMAN, B., RATNASAMY, S. et TAFT, N. (2009). Skilled in the art of being idle: reducing energy waste in networked systems. *Proceedings of the 6th symposium on Networked systems design and implementation*. USENIX Association, 381–394.

NEDEVSCHI, S., POPA, L., IANNACCONE, G. et RATNASAMY, S. (2008). Reducing network energy consumption via sleeping and rate-adaptation. *NSDI'08 Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. vol. 8, 323–336.

NET-SNMP (s.d.). Net-SNMP suite [Online]. `http://www.net-snmp.org`.

NIEWIADOMSKA-SZYNKIEWICZ, E., SIKORA, A., ARABAS, P. et KOŁODZIEJ, J. (2013). Control system for reducing energy consumption in backbone computer network. *Concurrency and Computation: Practice and Experience*, 25, 1738–1754.

NILSSON, P. (2006). *Fairness in communication and computer network design*. Thèse de doctorat, Lund University.

OGRYCZAK, W., PIORO, M. et TOMASZEWSKI, A. (2005). Telecommunications network design and max-min optimization problem. *Journal of telecommunications and informaton technology*, 3, 1–14.

ORLOWSKI, S., WESSÄLY, R., PIÓRO, M. et TOMASZEWSKI, A. (2010). SNDlib 1.0-Survivable Network Design Library. *Networks*, 55, NA–NA.

P. CHARALAMPOU, A. ZAFEIROPOULOS, C. VASSILAKIS, C. TZIOUVARAS, V. GIANNIKOPOULOU et N. LAOUTARIS (2013). Empirical evaluation of energy saving margins in backbone networks. *2013 19th Workshop on Local & Metropolitan Area Networks (LANMAN)*. IEEE, 1–6.

PHILLIPS, C., GAZO-CERVERO, A., GALÀN-JIMÈNEZ, J. et CHEN, X. (2011). Proactive energy management for wide area networks. *Communication Technology and Application (ICCTA 2011), IET International Conference on*. 317–322.

PIORO, M. (2007). Fair routing and related optimization problems. *15th International Conference on Advanced Computing and Communications (ADCOM 2007)*, 1, 229–235.

PIÓRO, M. et MEDHI, D. (2004). *Routing, Flow and Capacity Design in Communication and Computer Networks*. Morgan Kaufman.

PIZZONIA, M. et RIMONDINI, M. (2008). Netkit: easy emulation of complex networks on inexpensive hardware. *Proc. of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities*. ICST, 1–10.

RADUNOVIC, B. et LE BOUDEC, J. (2007). A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Transactions on Networking*, 15, 1073–1083.

RAHMAN, M., SAHA, S., CHENGAN, U. et ALFA, A. (2006). IP Traffic Matrix Estimation Methods: Comparisons and Improvements. *2006 International Conference on Communications*. IEEE, vol. 00, 90–96.

RAMAN, S., VENKAT, B. et RAINA, G. (2012). Using bgp to reduce power consumption in core and edge networks: A metric-based approach. *Journal On Advances in Networks*, 5, 367–376.

RESENDE, M. et RIBEIRO, C. (2003). Greedy randomized adaptive search procedures. *Handbook of metaheuristics*, 219–249.

RESENDE, M. et RIBEIRO, C. (2005). GRASP with path-relinking: Recent advances and applications. *Metaheuristics: Progress as real problem solvers*, 29–63.

RIMONDINI, M. (2007). Emulation of computer networks with netkit. *Dipartimento di Informatica e Automazione, Roma Tre University, http://www. netkit. org/, RT-DIA-113-2007*.

RIZZELLI, G., MOREA, A., TORNATORE, M. et RIVAL, O. (2012). Energy efficient traffic-aware design of on–off multi-layer translucent optical networks. *Computer Networks*, 56, 2443–2455.

SAKELLARI, G., MORFOPOULOU, C. et GELENBE, E. (2013). Investigating the trade-offs between power consumption and quality of service in a backbone network. *Future Internet*, 5, 268–281.

SANSO, B. et MELLAH, H. (2009). On reliability, performance and internet power consumption. *2009 7th International Workshop on Design of Reliable Communication Networks*. IEEE, 259–264.

SARKAR, S. et TASSIULAS, L. (2002). Fair allocation of utilities in multirate multicast networks: a framework for unifying diverse fairness objectives. *IEEE Transactions on Automatic Control*, 47, 931–944.

SHEN, G. et TUCKER, R. (2009). Energy-minimized design for ip over wdm networks. *Journal of Optical Communications and Networking*, 1, 176.

SHEN, M., LIU, H., XU, K. et WANG, N. (2012). Routing on demand: Toward the energy-aware traffic engineering with ospf. *NETWORKING 2012*, 232–246.

SHENKER, S. (1995). Fundamental design issues for the future internet. *IEEE Journal on Selected Areas in Communications*, 13, 1176–1188.

SHI, L., LIU, C. et LIU, B. (2008). Network utility maximization for triple-play services. *Computer Communications*, 31, 2257–2269.

SNMP4J.ORG (2003). The SNMP API for Java (SNMP4J) [Online]. `http://www.snmp4j.org`.

SOH, S. et LAZARESCU, M. (2013). Energy-aware two link-disjoint paths routing. *2013 14th International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 103–108.

SONG, Y. et LIU, M. (2012). Understanding the basic principles implied in green traffic routing. *2012 Symposium on Computers and Communications (ISCC)*. IEEE, 454–459.

SPRING, N., MAHAJAN, R. et WETHERALL, D. (2002). Measuring isp topologies with rocketfuel. *ACM SIGCOMM Computer Communication Review*, 32, 133–145.

SUBRAMANIAN, M. (2000). *Network Management - Principles and Practice*. Addison-Wesley.

TOMASZEWSKI, A. (2005). A polynomial algorithm for solving a general max-min fairness problem. *European Transactions on Telecommunications*, 16, 233–240.

TOURE, H. (2008). Icts and climate change-the itu perspective. *Climate Action*.

UHLIG, S. (2011). private communication.

UMIT, H. et FORTZ, B. (2007). Fast heuristic techniques for intra-domain routing metric optimization. *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium*. vol. 6, 1–6.

VAN HEDDEGHEM, W. et IDZIKOWSKI, F. (2012). Equipment power consumption in optical multilayer networks-source data. Rapport technique, Technical Report IBCN-12-001-01 (January 2012).

VAN HEDDEGHEM, W., IDZIKOWSKI, F., LE ROUZIC, E., MAZEAS, J., POIGNANT, H., SALAUN, S., LANNOO, B. et COLLE, D. (2012a). Evaluation of power rating of core network equipment in practical deployments. *2012 Online Conference on Green Communications (GreenCom)*. IEEE, no. GreenCom, 126–132.

VAN HEDDEGHEM, W., IDZIKOWSKI, F., VEREECKEN, W., COLLE, D., PICKAVET, M. et DEMEESTER, P. (2012b). Power consumption modeling in optical multilayer networks. *Photonic Network Communications*, 24, 86–102.

VASIC, N., BHURAT, P. et NOVAKOVIC, D. (2011). Identifying and using energy-critical paths. *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies on - CoNEXT '11*. ACM Press, 1–12.

VASIĆ, N. et KOSTIĆ, D. (2010). Energy-aware traffic engineering. *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking - e-Energy '10*. ACM Press, New York, New York, USA, 169.

VASSEUR, J., PICKAVET, M. et DEMEESTER, P. (2004). *Network recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Elsevier.

VEREECKEN, W., DEBOOSERE, L., COLLE, D. et VERMEULEN, B. (2008). Energy efficiency in telecommunication networks. *Proceedings of NOC2008, the 13th European Conference on Networks and Optical Communications*. 44–51.

VEREECKEN, W., HEDDEGHEM, W., DERUYCK, M., PUYPE, B., LANNOO, B., JOSEPH, W., COLLE, D., MARTENS, L. et DEMEESTER, P. (2011). Power consumption in telecommunication networks: overview and reduction strategies. *IEEE Communications Magazine*, 49, 62–69.

VETTER, P., LEFEVRE, L., GASCA, L., KANONAKIS, K., KAZOVSKY, L., LANNOO, B., LEE, A., MONNEY, C., QIU, X., SALIOU, F. et WONFOR, A. (2012). Research roadmap for green wireline access. *2012 IEEE International Conference on Communications (ICC)*. IEEE, 5941–5945.

WANG, N., HO, K., PAVLOU, G. et HOWARTH, M. (2008). An overview of routing optimization for internet traffic engineering. *IEEE Communications Surveys & Tutorials*, 10, 36–56.

WU, G. et MOHAN, G. (2012). Power efficient integrated routing with reliability constraints in ip over wdm networks. *2012 International Conference on Communication Systems (ICCS)*. IEEE, 275–279.

WU, Y., GUO, B., SHEN, Y., WANG, J. et LIU, X. (2013). Toward energy-proportional Internet core networks: an energy-minimized routing and virtual topology design for Internet protocol layer. *International Journal of Communication Systems*.

XIA, M., TORNATORE, M., ZHANG, Y., CHOWDHURY, P., MARTEL, C. et MUKHERJEE, B. (2011). Green Provisioning for Optical WDM Networks. *IEEE Journal of Selected Topics in Quantum Electronics*, 17, 437–445.

XU, M., HOU, M., WANG, D. et YANG, J. (2013). An efficient critical protection scheme for intra-domain routing using link characteristics. *Computer Networks*, 57, 117–133.

YAYIMLI, A. et CAVDAR, C. (2012). Energy-aware virtual topology reconfiguration under dynamic traffic. *2012 14th International Conference on Transparent Optical Networks (ICTON)*, 1–4.

ZEADALLY, S., KHAN, S. et CHILAMKURTI, N. (2011). Energy-efficient networking: past, present and future. *The Journal of Supercomputing*, 62, 1093–1118.

ZHANG, M., YI, C., LIU, B. et ZHANG, B. (2010). Greente: Power-aware traffic engineering. *The 18th International Conference on Network Protocols*. IEEE, 21–30.

ZHANG, X., WANG, H. et ZHANG, Z. (2014). Survivable green IP over WDM networks against double-link failures. *Computer Networks*, 59, 62–76.

ZHANG, Y., TORNATORE, M., CHOWDHURY, P. et MUKHERJEE, B. (2011). Energy optimization in ip-over-wdm networks. *Optical Switching and Networking*, 8, 171–180.

ZHAO, Y., WANG, S., XU, S., WANG, X., GAO, X. et QIAO, C. (2013). Load balance vs energy efficiency in traffic engineering: A game theoretical perspective. *2013 Proceedings INFOCOM*. IEEE, 530–534.

ANNEX A

Network congestion measure

For each scenario $\sigma \in S$, we evaluate the average arc congestion ν_{ij}^{σ} as follows. The single arc congestion is calculated as

$$\nu_{ij}^{\sigma} = \frac{1}{c_{ij}w_{ij} - f_{ij}^{\sigma}}, \tag{A.1}$$

where $f_{ij}^{\sigma} = \sum_{d \in D} r^{d\sigma} x_{ij}^{d\sigma}$ is the flow on link $(i,j) \in A$ and $c_{ij}w_{ij}^{\sigma}$ is the capacity available on link $(i,j) \in A$ during period $\sigma \in S$. The average congestion, which is evaluated with respect to active cards only, is calculated as follows:

$$\nu^{\sigma} \frac{1}{\sum_{d \in D} f^{d\sigma}} \sum_{(i,j) \in A} \frac{f_{ij}^{\sigma}}{(c_{ij}w_{ij} - f_{ij}^{\sigma})}, \tag{A.2}$$

where $f^{d\sigma} = r^{d\sigma}$ is the flow associated with demand $d \in D$.