POLITECNICO DI MILANO

Faculty of Information Engineer
Degree Master Course
in Telecommunications Engineer
Electronic and Information Department

Models and Algorithms for Energy Saving in IP Networks

Supervisor: Prof. Antonio Capone

Graduation Thesis of:
Luca Giovanni Gianoli Matr. 734605

Academic Year 2009 - 2010

Contents

Abstract **9**

1 Introduction **11**

2 Energy Saving Techniques **15**

2.1 Energy Consumption Measurements and Models 17

2.1.1 An Internet Consumption Model 18

2.2 Energy Efficient Networks 21

2.2.1 Sleeping Strategies 21

2.2.2 Power Aware Network Design 25

2.2.3 Adaptive Link Rate (ALR) 27

2.3 Green Energy for Green ICT 29

3 Green IP Routing **33**

3.1 Heuristics for network global management 33

3.2 The Energy Aware Routing Algorithm 39

4 OSPES: Switching-Off Stage **43**

4.1 Problem Formulations 45

4.2 The Preprocessing Algorithm 49

5 OSPES: Feasible Routing Stage **57**

5.1 Optimal routing model 59

5.2 The IGP-WO Algorithm 60

5.2.1 Neighbourhood Structure 61

5.2.2 Hashing tables 61

5.3 Computational results 63

5.4 The Totem Toolbox 63

6 Generation of Data Input **65**
6.1 Network topologies 65
6.2 Traffic matrices 67
6.2.1 SLOM formulation 68
6.2.2 SNLM formulation 70
6.2.3 Low workload traffic matrices 70

7 Computational Results **71**
7.1 Switching-off Stage Computational Results 71
7.1.1 SLO formulation 71
7.1.2 SNL formulation 78
7.2 Feasible Routing Stage Computational Results 81
7.2.1 SLO formulation data input 83
7.2.2 SNL formulation data input 86

8 Conclusions **89**

Ringraziamenti **91**

Bibliography **93**

List of Figures

1.1 Estimated distribution of global CO_2 emissions from ICTs in 2007 . . 12
1.2 Example of the switching-off procedure 13

2.1 Internet schematic structure 20
2.2 Internet power consumption 100 Mb/s 22
2.3 Internet power consumption 1 Gb/s 22
2.4 Traffic behaviour in Internet 23
2.5 Inter-packet idle period 24
2.6 Predictive power management method 24
2.7 Cisco router power consumption 26
2.8 ALR handshake 28
2.9 Electricity price variations 30
2.10 Histograms of hour-to-hour changes in real-time hourly prices 31

3.1 Random topology 35
3.2 Heuristic [14] computational results first part 36
3.3 Heuristic [14] computational results second part 37
3.4 Heuristic [16] computational results first part 37
3.5 Heuristic [16] computational results second part 37
3.6 Calculation of the minimum number of links for a feasible routing . . 40
3.7 Example of EAR algorithm 41
3.8 EAR algorithm computational results 42

4.1 OSPES example 44
4.2 Comparison between different switching-off approaches 46
4.3 Fluid model routing 47
4.4 Examples of cuts in a graph 50
4.5 Building partition first stage 51
4.6 Building partition second stage 52
4.7 Candidate node that does not respect the full-connectivity constraint 52

4.8 The preprocessing procedure: the output 54

5.1 Cost function $\phi_{(i,j)}$ plot 58
5.2 Balancing flows neighbourhood 62

6.1 Exodus map 66
6.2 Ebone map 67
6.3 Example of trivial case with traffic demands equal to links capacity . 69

7.1 SLO$_{PRE}$: comparison between gap levels of the three networks considered 74
7.2 Exodus: SLO and SLO$_{PRE}$ formulations computational results 75
7.3 Ebone: SLO and SLO$_{PRE}$ formulations computational results 76
7.4 Telstra: SLO and SLO$_{PRE}$ formulations computational results 77
7.5 Exodus SNL performance 80
7.6 Default network: feasible routing stage performance 82
7.7 Default network topology 83
7.8 Feasible routing stage global performances comparison 84
7.9 Exodus: SNL feasible routing stage performance 86

List of Tables

2.1 Resume of energy saving techniques 16
2.2 Energy consumption in USA in 2000. 17
2.3 Cisco GSR 12008 configurations 25
2.4 Cisco 7507 configurations 26
2.5 ALR energy saving estimations 29
2.6 Electricity costs for large companies 31
2.7 Real-time market statistics 31

4.1 Traffic demands of figure (4.2) 46
4.2 Traffic demands of figure (4.8) 54

5.1 IGP-WO computational results 64

6.1 Rocketfuel real backbone topologies 66
6.2 Division between edge and core routers 66
6.3 Traffic demands of figure (6.2) 69

7.1 Exodus: SLO formulation computational results 72
7.2 Ebone: SLO formulation computational results 72
7.3 Telstra: SLO formulation computational results 72
7.4 Exodus: SLO_{PRE} formulation computational results 73
7.5 Ebone: SLO_{PRE} formulation computational results 73
7.6 Telstra: SLO_{PRE} formulation computational results 73
7.7 Exodus: SNL formulation computational results 79
7.8 Default network: feasible routing stage SLO computational results . . 81
7.9 SNO Feasible routing stage: global comparison between the three topologies 85
7.10 Exodus: SNL feasible routing stage computational results 87

Abstract

English

In the last years, the exponential expansion of the ICT sector, and in particular of Internet, has been consequently accompanied by a considerable increase of global energetic consumptions. One of the main problems consists into the fact that network infrastructures have been dimensioned considering the peak levels of traffic, without however developing any procedures able to reduce the performances, and therefore the power consumptions, during low workload periods. The approach that we propose has the aim to avoid unnecessary wastes of energy, adapting the network topology (switching off or switching on links and routers) to the different daily scenarios of workload. This approach is called OSPES (Optimized Shortest Path for Energy Saving) and is based on the main idea of exploiting the characteristics of the OSPF (Open Shortest Path First) routing protocol, in order to determine a general routing that lets to switch off the underutilized elements (links and routers). The work is composed of two distinct parts: the switching-off stage, where we try to switch off the greatest number of network elements, and the feasible routing stage, where we calculate a set of OSPF weights able to determine a feasible (no saturated links) shortest path routing. The computational results suggest that, if low workload conditions are considered, is possible to switch off up to 50% of links and nodes currently used, always keeping the same service quality.

Italiano

Negli ultimi anni, la crescita esponenziale del settore ICT, ed in particolare di Internet, é stata logicamente accompagnata da un considerevole incremento del consumo energetico globale. Uno dei problemi principali consiste nel fatto che le infrastrutture di rete sono state dimensionate considerando i livelli di traffico di picco, senza peró che venissero sviluppate procedure che consentissero di diminuire le prestazioni, e quindi di ridurre i consumi, al presentarsi di condizioni di basso carico. La pro-

cedura che presentiamo ha l'obiettivo di evitare inutili sprechi energetici, adattando la topologia di rete (spegnendo o accendendo i link e i router) ai diversi scenari di traffico. Il nostro approccio si chiama OSPES (Optimized Shortest Path for Energy Saving) ed é basato principalmente sull'idea di sfruttare le caratteristiche del protocollo di routing OSPF (Open Shortest Path First), in modo da determinare un instradamento delle domande di traffico che permetta di spegnere gli elementi sottoutilizzati. La procedura é composta da due parti distinte: lo stage di spegnimento, dove si cerca di spegnere il maggior numero possibile di router e link, e lo stage dell'instradamento ammissibile, dove si ricerca un insieme di pesi OSPF che determini un instradamento per cammini minimi ammissibile (senza link saturati). I risultati computazionali mostrano come, in condizioni di basso carico, sia possibile spegnere fino al 50% dei link e dei nodi attualmente utilizzati, sempre garantendo la stessa qualitá del servizio.

Chapter 1

Introduction

In the two last decades, the ICT sector (Information and Communication Technologies), has been protagonist of a rapid and constant growth. ICTs, through the implementation and the development of new tools and services, have been playing a really important role in the hard challenge against global warming and climate changes; for example, ICTs have been extensively used to reduce CO_2 emissions caused by the transport sector, offering services like email, phone calls, text messaging and video conferencing, able to reduce the requirement of physical travels.

The issue of exploiting ICTs in order to reduce emissions and electric consumptions of the different economic/social areas, is however accompanied by the necessity of reducing the emissions produced by the ICT sector itself, without at the same time penalizing the high growth rate. The ICT sector (in this definition internet, computing and telecommunications) is responsible for about 2.5% cent of global CO_2 emissions. Data reported in [1] show how in 2007, Internet had been responsible of 5.5% of the global energy consumption. This percentage is destined to grow rapidly because of the constant increase of the demand for high bandwidth services (the annual growth rate of IP traffic is estimated around 50%): at the moment the annual increment rate can be evaluated around 20-25%.

Therefore, the necessity of a next generation network (NGN) characterized by a strong reduction of energy consumption, has become one of the most important global issues. One of the main problems consists into the fact that network infrastructures have been dimensioned considering the peak levels of traffic, without however developing any procedures able to reduce the performances, and therefore the power consumptions, during low workload periods. The workload conditions on Internet, follow a periodic behaviour strictly connected with the daily time slot considered.

In order to better exploit network infrastructures and to avoid unnecessary wastes

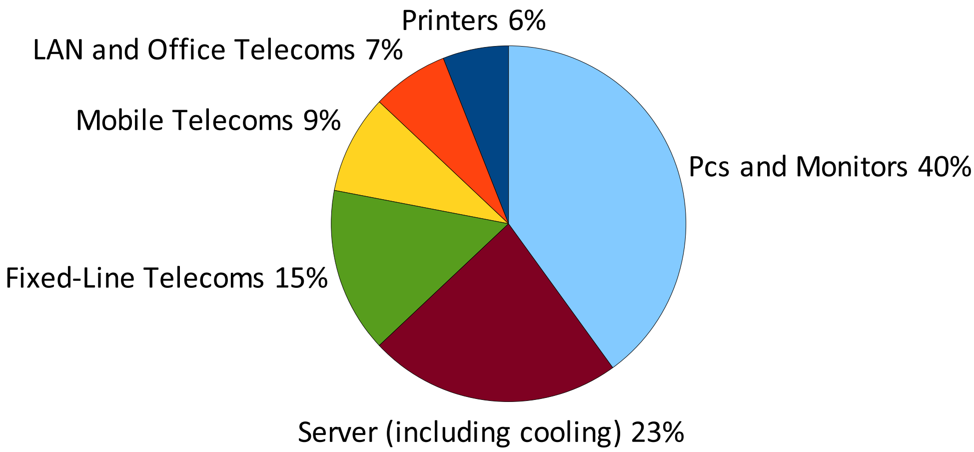


Figure 1.1: *Estimated distribution of global CO_2 emissions from ICTs in 2007. Source: Kumar, Rakesh and Mieritz, Lars (2007) Conceptualising Green IT and data centre powerand cooling issues, Gartner Research Paper No. G00150322.*

of energy, we propose a new approach to switch off network elements, that considers the different daily scenarios of workload. This approach is called OSPES (Optimized Shortest Path for Energy Saving) and is based on the main idea of exploiting the characteristics of the OSPF (Open Shortest Path First) routing protocol in order to determine a general shortest path routing that lets to switch off the underutilized elements (links and routers).

OSPF is the most commonly adopted intra-domain internet routing protocol. Each link has an assigned weight that is used by the network routers to calculate their own shortest path trees. The traffic demands are then routed along the shortest paths, being split at nodes where more outgoing links belong to the shortest paths in direction of the destination. Generally the configuration of the links weights has the aim to avoid network congestion, but, like in our case, it could be also exploited in order to obtain energy saving, by generating a routing that could involve only a part of the network elements.

The work is composed of two distinct parts, the switching-off stage and the feasible routing stage: initially, given a network topology and a traffic matrix as data input, we use an heuristic algorithm to switch off the greatest possible number of network elements; capacity and flow conservation constraints have to be always respected (see figure (1.2)). The OSPF weights of the links that have to be switched off, are set equal to a very big value, in order to exclude them from all the shortest path trees. In a second moment, during the feasible routing stage, we then exploit the algorithm proposed by Fortz and Thorup [2] to find the optimal configuration of OSPF weights. The new set of OSPF weights has the target to determine a feasible shortest path routing (no link with utilization over 100%) for the traffic matrix considered. When a feasible set of weights could not be found, it is necessary to switch on some elements.

The problem of switching off network elements guaranteeing the feasible routing of all the traffic demands, is very complex; it is therefore impossible to find optimal solutions when big instances (networks with more than 50 nodes) are considered. We try to reduce the complexity of the problem by operating a preprocessing of the data input that helps to reduce the solution space: in this way, the time necessary to find a satisfactory feasible solution (less than 25% from the upper bound calculated)

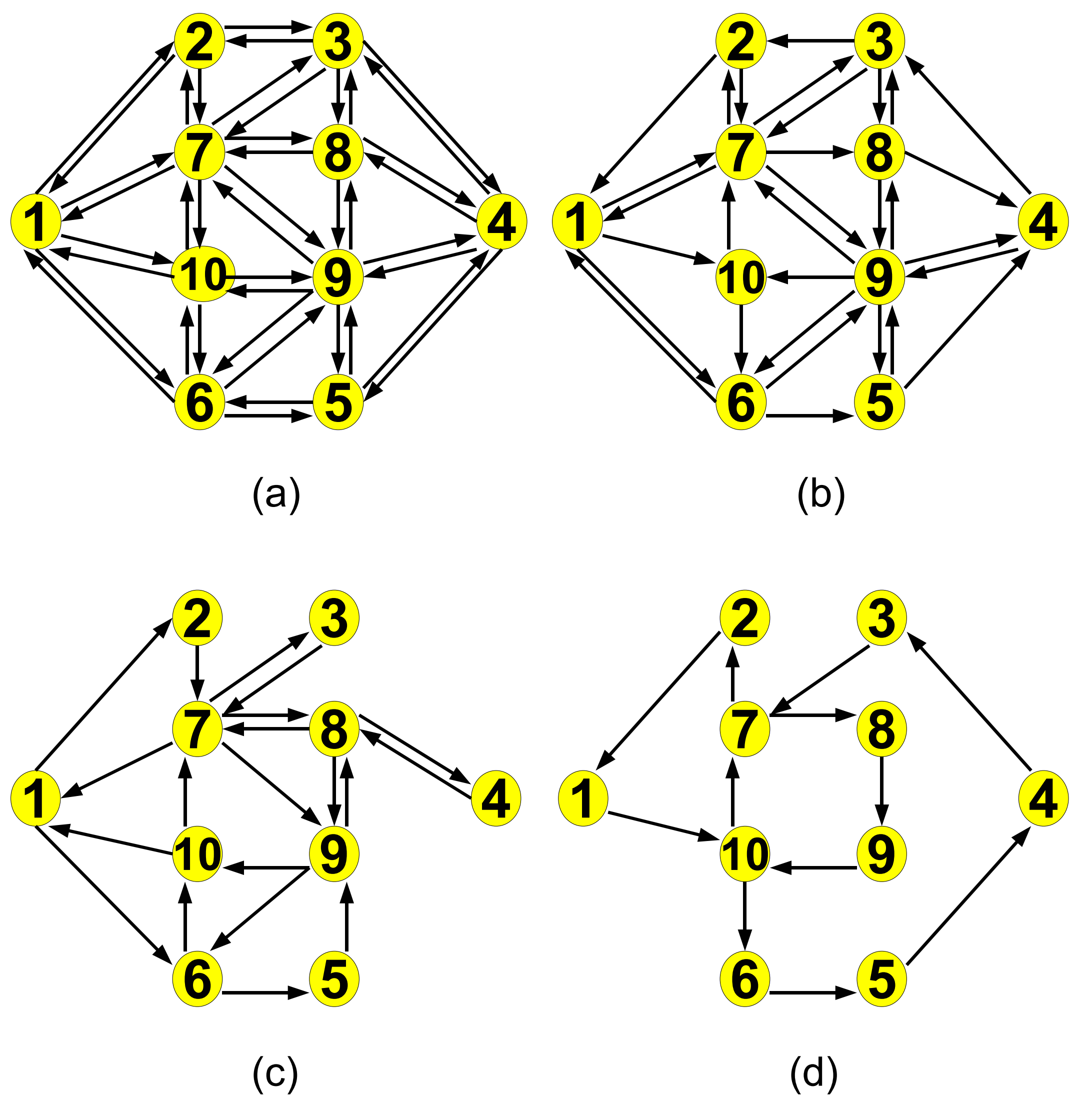


Figure 1.2: *The sequence represents the different reduced topologies obtained varying the workload on the network. In (a) all the links have to be powered on; in (b), (c), (d) the traffic matrix used in (a) is multiplied respectively for 0.6, 0.3, 0.1, and is therefore possible to switch off more and more links. In this case the procedure switches off only the links, ignoring the nodes.*

decreases significantly (above all for some type of topologies and traffic matrices), and also the bounds founded are better.

We have tested the procedure on three real network topologies with about 80/100 nodes and 300 unidirectional links: the computational results suggest that, if low workload conditions are considered, is possible to switch off up to 50% of links and nodes currently used, always keeping the same service quality.

In the second chapter of the thesis we present a general summary about objectives and SOTA (State Of The Art) of greening technologies. In the third chapter we illustrate some already published works concerning the green IP routing. In chapter 4 we present the switching-off stage, analysing in succession the MIP formulations of the switching-off problem and the preprocessing data algorithm. In chapter 5 we describe the feasible routing stage, with a particular attention for the IGP-WO (Interior Gateway Protocol-Weight Optimization) algorithm proposed in [2] by Fortz and Thorup, and for the role that it plays in our work. In chapter 6 we discuss the origins of the data input (network topologies and traffic matrix), presenting the hypothesis and the approximations that we have done about them. Finally we present the computational results in chapter 7, and the conclusions in chapter 8.

Chapter 2

Energy Saving Techniques

In the last years, the exponential expansion of ICT sector, and in particular of Internet, has been consequently accompanied by a considerable increase of global energetic consumptions. For these reasons, the themes of energy saving in IP networks and of power awareness in network design, have been recently become of greatest interest in the scientific community.

In the area of mobile networks and wireless sensor networks, the research has already reached some remarkable results for what concerns the energy efficiency. On the contrary, in the sector of wired networks (access and backbone networks), the research is still at the beginning. The energy saving issue could be handled at different levels (materials, devices, protocols, architectures, management algorithms and procedures etc.); in [3] the researchers propose an interesting categorization of the various works, based on the combination of three different aspects:

- **Target:** the target of the research is the element analysed to obtain the power consumption reduction. It is possible to subdivide these elements into communication mediums (optical fiber), components (routers, switches) and path; the latter is the union of the firsts two.

- **Technique:** energy saving could be obtained by the application of three principal categories of techniques. The first one consists into the study of new materials and fabrication techniques: an example could be the replacement of the old networks with the new all-optical networks. Then there are the procedures based on the concept of switching. The main idea is to switch off the components during periods of low utilization: our work belongs to this category. The last class consists into the integration of physical components.

- **Metric:** the metric plays a fundamental role in the definition of policies for the application of energy saving techniques. At the moment the most utilized

Table 2.1: *Resume of energy saving techniques. Source [3].*

Target	Technique	Metric
Medium	New Technology (Material, Fabrication etc.)	Traffic Pattern (time,space)
Component	Switching (ON/OFF, Duty, Frequency, ect.)	Energy Loss
Path	Aggregation	Semantic (time, space, address, application, packet, human behaviours)

metric is represented by the traffic pattern; other techniques are instead based on different metrics like for example the power loss.

The research about the energy aware design/management of telecommunications networks has been following five main research lines:

- **Energy consumption measurements and models:** the development of measurement methodologies and consumption models is fundamental, in order to provide reliable input data available for all the other researches. At the moment the manufacturers communicate only the peak power values.

- **Energy efficient computing centres:** the rapid growth of the number of service applications available, requires a consequent increase of the servers number. Typically, distinct applications run on different machines; in this way the servers utilization never exceeds 20%, causing therefore a considerable waste of energy. The target is to *virtualize* the applications, offering in this way the possibility to dynamically share between different services, the resources of a single machine.

- **Energy efficient networks:** the development of the next generation network NGN, is supported by huge investments from operators and public administrations. Beside the necessity of designing a next generation access network able to provide the ultra high bandwidth to all the users, it is fundamental to project the new network architecture considering all the possible ways to make Internet greener. The new networks will therefore require new devices, new protocols and new functionalities, adapt to a more dynamic and efficient management of network infrastructures.

- **Green clouds:** a new tendency looks at computing and networking components as strongly interconnected in the idea of "cloud". Cloud computing tries to optimize the management of service requests, operating directly at the Internet level.

- **Green energy for green ICT:** analysing the geographical distribution of the service centres, it is possible to operate significant optimizations concerning energy availabilities and energy costs in the different time zones.

Three of the last five research areas are of great interest for our work: measurements and models, energy efficient networks and green energy for green ICT. Therefore, in the next sections, we present an exhaustive summary about the state of the art of these three areas, illustrating the details of some significant works already published.

2.1 Energy Consumption Measurements and Models

Measurements and models concerning the network infrastructures behaviour, let to verify performances and feasibility of the different energy saving techniques. The energy consumptions analysis could be done at different levels: consumption of end systems, consumption of network devices, consumption of networks, benchmark and metrics.

In [4] the authors have presented a set of general observations and considerations about network energy consumptions, discussing the feasibility of some possible saving procedures based on the concept of sleeping. The firsts considerations concern the effective evaluation of the Internet electric consumption in USA.

The data illustrated in table (2.2) are obtained trough the average of the consumptions of the different device models utilized in USA; the different levels of

Table 2.2: *Estimations of the energy consumptions in USA of the different network devices. Data of the year 2000 (TW-h refers to Tera-Watt hours and AEC to Annual Electricity Consumption). Source [4].*

Device	**Approximate number deployed**	**Total AEC TW-h**
Hubs	93.5 Million	1.6 TW-h
LAN Switch	95000	3.2 TW-h
WAN Switch	50000	0.15 TW-h
Router	3,257	1.1 TW-h
Total		6.05 TW-h

consumptions are weighted considering the diffusion of each model. The energy costs for cooling and UPS (Uninterruptable Power Supplies) are not comprised in the table. The total consumption of 6.05 TW-h represents 0.07% of the total electric consumption in USA in the year 2000. This percentage is very low if compared with the total and it does not highlight significantly the reasons for an urgent necessity of a better energy management: it is therefore necessary a deeper analysis to point out the power inefficiencies of Internet. The first evidence of an energetic inefficiency comes out from the comparison between the energy cost for transmitting a byte through the Internet and the cost of the wireless transmission; in [4] it is showed that wireless transmission could be more efficient under an energetic profile, from 3 till even 24 times in case of directional antennas. The difference is caused by the great amount of energy wasted from network devices during he idle periods. With the growth of the rates the gap is obviously destined to greatly increase. A second reason for the necessity of new energy policies, concerns the possibility of a stronger deployment in areas with scarce availability of electricity; for example in India, differently from USA, in the year 2000 the percentage of Internet energetic consumption could be estimated as 5% of the total electric demand. In a similar scenario, the development of more efficient devices becomes of great interest and importance.

In [5] a network-based model for the measurements of Internet power consumption is presented. The details are reported in the following section.

2.1.1 An Internet Consumption Model

Consistently with the model just cited, the Internet power consumption in broadband countries (average access speed of 30 Mb/s) should be around 1% of the total. Power consumed by servers, data centres, redundant reliability infrastructures and home network equipments, is ignored by the model: therefore the final calculated power consumption represents a lower bound of the real value; the power consumed for cooling is estimated to be 50% of the total.

Network Structure

The Internet structure is represented and analysed after a logical division in three hierarchical sectors (as shown in figure (2.1): access network, metro network and core network.

- **Access Network:** the access network is the lower level of the structure; it is composed by the curb-side nodes, by the network units of each user, by the concentrators and, in the end, by the links utilized for the connection of all the previous network elements. The infrastructure is assumed to be all optical (PON - Passive Optical Network). In the curb-side nodes there are passive splitters able to split a single fiber from an OLT (Optical Line Termination of the edge nodes), into more fibers directed to a group of ONUs (Optical Network Units of each user). The ONUs that share the same connection with an OLT , exploit the time division multiplexing for the communications with the OLT itself.

- **Metro Network:** the metro network connects edge nodes and core nodes. The edge routers are usually connected with the OLTs by four Gigabit Ethernet lines; they adapt the IP packets to the SONET/SDH format and then they transmit them to the core routers. In the model each edge router (Cisco 12816 edge router) is connected to a core router through fourteen 10 Gb/s OC192c/STM-64c Packet over Sonet (POS) links, and to forty OLTs each through 4x1 Gb/links. It is not necessary to interconnect the edge routers because they perform only forwarding.

- **Core Network:** the core nodes consist into several multishelf core routers that perform all the network routing. The researchers have chosen to utilize in the model the Cisco CRS-1 Multishelf System, that, if equipped with 72 line card shelves and 8 fabric card shelves, is able to offer a maximum switching capacity of 92 Tb/s full/duplex. The CRS-1 routers are interconnected by 40 Gb/s links, and they connect to the edge routers trough 10 Gb/s links.

Finally, a last element that has to be considered is represented by the oversubscription rate. This rate is calculated as $(M - N)/N$, where M is the number of users connected and N is the number of users supportable simultaneously at the peak rate. The development of new applications like IP-TV, VOD and file sharing, imposes a low level of oversubscription.

Power Consumptions

The power consumption is calculated for each sector. The heat dissipation is used to estimate the power consumption of the different routers.

Each ONU and OLT consumes respectively 10W and 100W, while each Cisco 12816 edge router consumes approximately 4.21 kW. The splitters do not consume any power because they are passive elements. The power consumption P_{ma} of the metro-access network is therefore estimated to be:

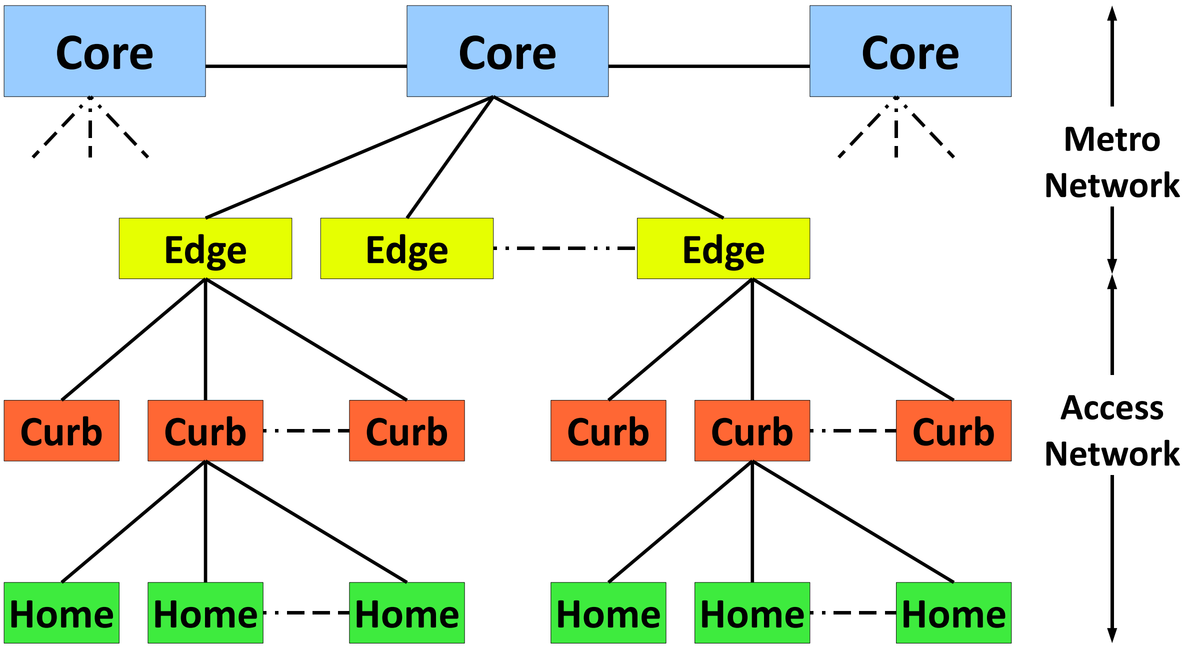


Figure 2.1: *Internet schematic structure.*

$$P_{ma} = 2 \times \frac{N}{S(\rho+1)}\left[\frac{4.21kW}{40} + 100W\right] + N \times 10W \quad (2.1)$$

ρ is the oversubscription rate, N is the number of users connected, and S is the number of ONUs connected to a single OLT when ρ is 0.

A fully equipped CRS-1 Multi-shelf System consumes 1020 kW (13.2 kW for each line card shelf and 8.97 kW for each fabric card shelf). Then it is assumed that the traffic managed by the router is 50% internal, 25% add/drop and 25% bypass: the traffic processed by the router is therefore 133% of the total metro traffic. Under these assumptions, and considering R the peak access rate in each direction, the power consumption P_c of a core router is:

$$P_c = 2 \times 2 \times 1.33 \times \frac{1020kW \times N \times 2R}{(\rho+1) \times 92Tb/s} \quad (2.2)$$

At last it is necessary to consider the consumption of the WDM links. A WDM terminal system connecting edge and core nodes, consumes about 1.5 kW for every 64 wavelengths. For each pair of terminal systems with a distance inferior than 200 km, it is necessary at most one multiple wavelength amplifier that consumes about 6 W per fiber. Instead, the WDM terminal systems connecting the core routers and the intermediate line amplifier (ILA), consume respectively 811 W and 622 W for every 176 channels. Two core nodes distant about 1500 km, need two terminal systems and 14 ILAs. Therefore the power P_{WDM} consumed by WDM links is:

$$P_{WDM} = \frac{16.44W \times N}{S(\rho+1)} + \frac{312.2W \times NR}{(\rho+1) \times 40Gb/s} \tag{2.3}$$

The plots of the network power consumptions are reported in figures (2.2) and (2.3) (the average access rate is given by $R/(\rho+1)$); these plots show how the total power consumption has to be principally attributable to the routers. This suggests that a good switch-off strategy, should focus the attention on the routers and not on the links.

2.2 Energy Efficient Networks

The green networking has the target to develop efficient procedures for the energy aware management/design of networks. The Internet structure is dimensioned to satisfy peak conditions; however, as it shown in [6], the traffic level is not constant, but ,to the contrary, it has a daily periodic behaviour (see figure (2.4)). This peculiarity could be therefore exploited in order to reduce the energetic consumptions during low traffic periods. In [4] the possibility of applying sleeping strategies to the different network devices is exhaustively discussed.

2.2.1 Sleeping Strategies

The sleeping strategies let to put network components into stand-by states. Focusing the attention on routers, it is possible to identify different components that could be switched off: memory, main processor, bus, line card processor/ASIC, switching fabric. The issues concerning the possibility of switching off each of these components, have to be investigated:

- **Line cards:** there are several models of line cards with different levels of complexity (interfaces with processors, memories, switch fabric interfaces). The possibility of switching off line cards or part of their components, depends from the rapidity of the transients from the sleeping states to the awake states. A too slow transient causes significant packet losses and high levels of latency. There are two possible classes of approaches to be adopted, coordinated, and uncoordinated sleeping approaches. The firsts are network-wide approaches where, during period of low load, the routing protocol aggregates the traffic into few routes, offering the possibility to put some interfaces to sleep. The seconds are link-layer approaches, where the management of the sleeping states is based on local decisions. Another strategy for energy saving could consist into slowing the clock of line card processors.

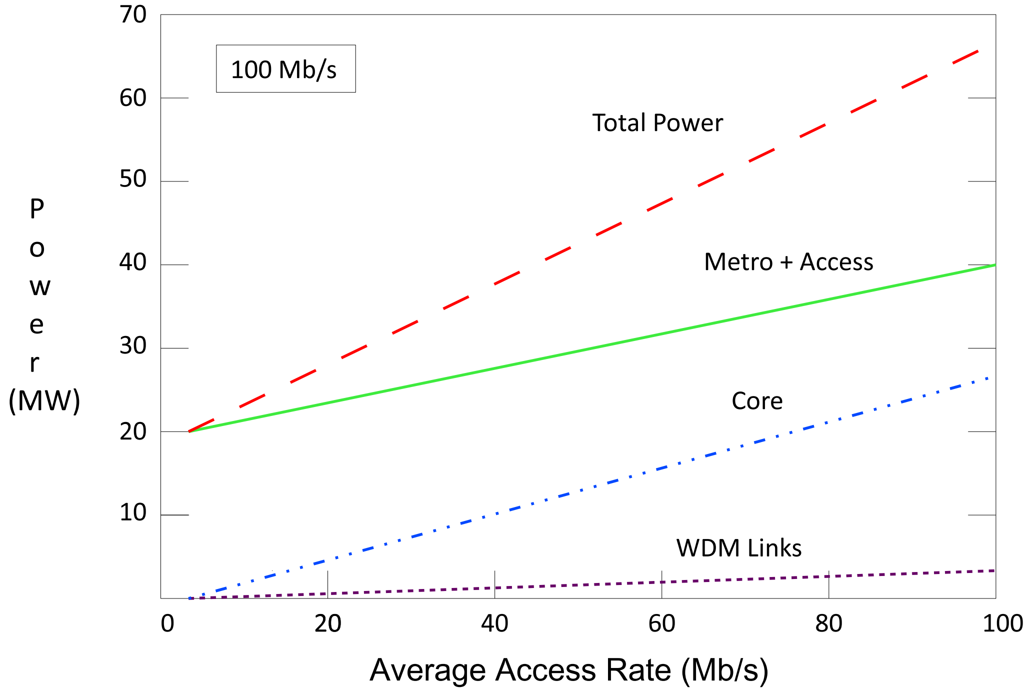


Figure 2.2: *Power consumption of the different Internet sectors for the full range of average access rates, with 2 million of users and peak rate of 100 Mb/s. Source [5].*

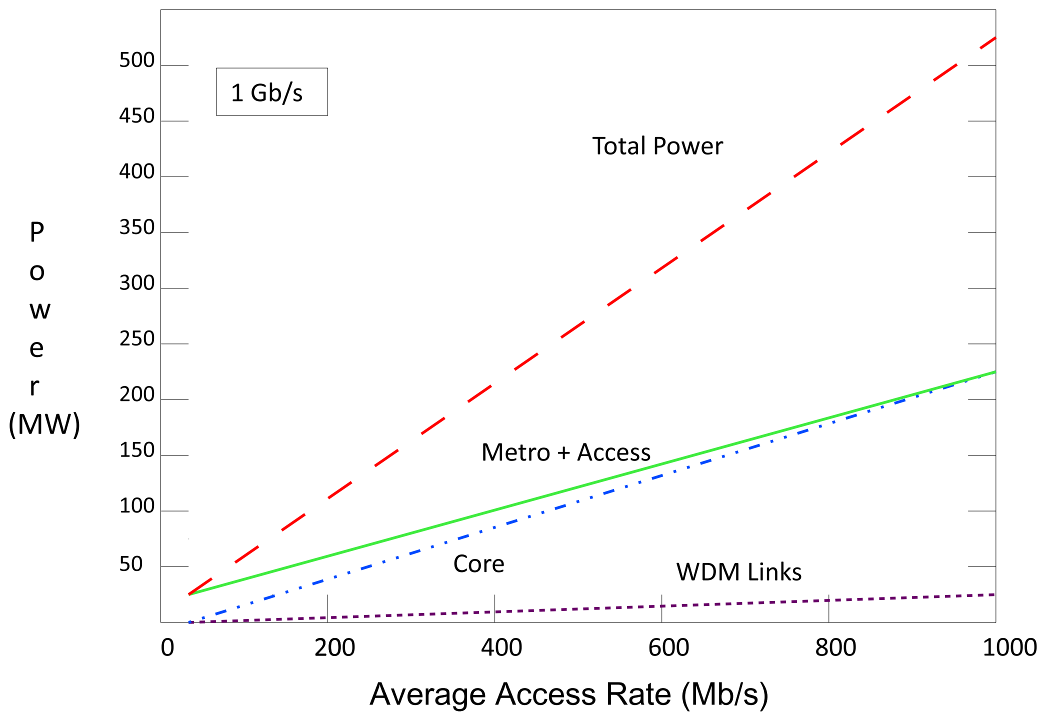


Figure 2.3: *Power consumption of the different Internet sectors for the full range of average access rates, with 2 million of users and peak rate of 1 Gb/s. Source [5].*

- **Crossbar:** Putting to sleep the crossbar together with the line card does not bring additional latency or loss. The problems could occur putting to sleep the crossbar alone, because in this case it would be necessary to greatly increase the line card buffer dimensions.
- **Main processor:** the main processor typically runs GHz rates. As just proposed for the line card processor, it is possible also in this case to slow down the processor clock.

The main issue concerning the sleeping procedures consists into when and how utilizing them. At first it is necessary to determine how long a particular component could remain in the sleeping state. The waking up procedure causes a peak in energy consumption; the components should therefore sleep long enough to compensate the additional energy drawn to wake up. In addition, a component should sleep for an interval longer than the period necessary to wake up. The maximum sleeping interval is conditioned by the traffic load conditions and by the protocol characteristics: for example, adopting an uncoordinated sleeping approach in conjunction with OSPF (Open Shortest Path First) protocol, the maximum sleep interval could not exceed the time scheduled between two consecutive OSPF Hello messages. The second issue concerns how taking the decision to enable the sleeping state. If a router/switch uses an uncoordinated approach, it can only monitor the traffic on all its interfaces, estimating the expected packets inter-arrival time. Instead, in case of coordinated approaches, the sleeping intervals are calculated through considerations based on traffic flow estimations at the AS (Autonomous System) level. Finally, the last problem is represented by which elements are the most suitable to be switched off. In this case there are a lot factors that have to be considered: routers location (backbone, access, AS boundaries, etc.), typology of the managed traffic (inter/intra

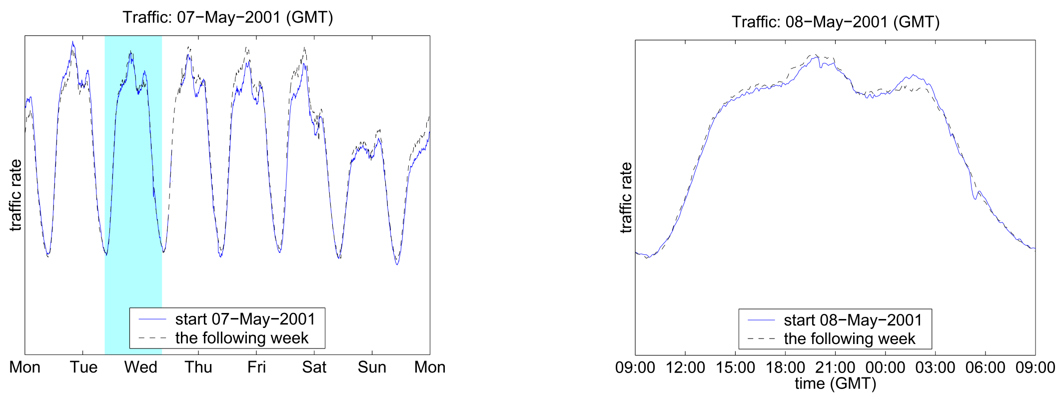


Figure 2.4: *Plot of the traffic into a region in two consecutive weeks. The second figure focuses on the highlighted region of the first. Source [6].*

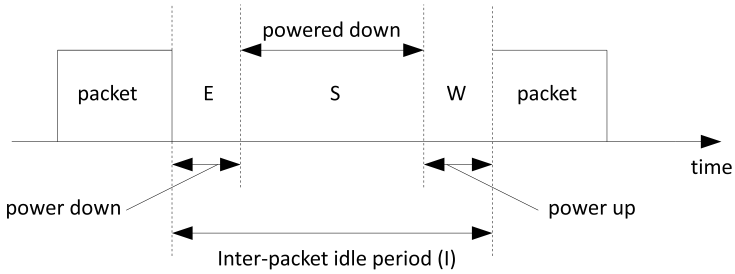


Figure 2.5: *Inter-packet idle period. Source [7].*

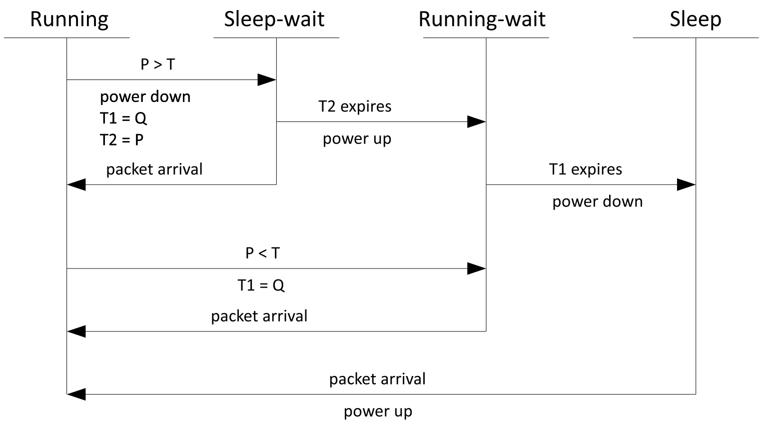


Figure 2.6: *Scheme of the sleeping predictive algorithm presented in [7].*

AS), number of routes inside the ASes.

Internet traffic has the characteristic to be bursty and self-similar. The self-similarity is translated in long successions of small idle periods, spaced out in few cases by very large idle periods. During these last ones, it is obviously convenient to activate sleeping procedures. In [7], Gunaratne and Christensen present a new sleeping procedure based on the prediction of the length of the idle periods. When the predicted idle period is greater than a previously calculated threshold, the device enters in a low power sleep state until an occurrence triggers a forced wake-up; if the idle period predicted is less than the threshold, the device stays powered on. The structure of an idle period is shown in figure (2.5). The threshold is $T = E+W$. The idle period length is calculated by the expression $P_{i+1} = P_i \times (1 - \alpha) + I_i \times \alpha$; P_{i+1} is the new prediction, P_i is the previous prediction, I_i is the length of the previous idle period and α is a constant ($0 \leq \alpha \leq 1$). The complete procedure is illustrated in figure (2.6) and is composed of four main states: running (device fully powered-up), running-wait (device fully powered up, counting a timer T1 and monitoring for packets arrival), sleep-wait (device in a sleep state, counting a timer T2 and monitoring for packets arrival) and sleep (device in a sleep state and monitoring for packets arrival). The Q value is calculated with a specific algorithm [8].

The characterization of the Internet traffic generated by personal computers (significant presence of idle time) is also exploited in [9]. The main idea consists into using a proxying Ethernet adapter that handles routine network tasks for a desktop computer that is in low-power sleep mode. In this way the desktop computers could be *on the network* all time, comprise during the low-power sleep mode periods.

The sleeping strategies just described focus their attention on the single devices. It is also possible, like in our research, to apply the sleeping strategies (switching-off strategies) into more global contexts, making considerations at level of the entire network. We have preferred to dedicate an entire chapter, the chapter 3, to the presentation of the works concerning these strategies, because of the strong connections with our research.

2.2.2 Power Aware Network Design

The effective efficiency of the energy saving procedures developed for the power aware management of the networks, could further be improved by a correct network design. In [10] the authors present a network design model that considers the consumption of different configurations of chassis and line cards for each point of presence. The target is to minimize the network consumption guaranteeing robustness and performances. The mixed-integer programming formulation presented in [10] needs data about the consumption of the different routers configurations available. The researchers have tested two platforms, the Cisco GSR 12008 and the Cisco 7507, trying different configurations of chassis and line cards (see tables 2.3,2.4 and figure (2.7)).

The test has highlighted that, under a power efficiency profile, it is convenient to

Chassis Slot	Line Card	Abbreviation
0	Empty	None
1	4 port GE line card	4GE
2	4 port OC-3/POS line card	OC-3
3	1 port OC-48/POS line card	OC-48
4	10 Gb/s Switching fabric	CSC
5	10 Gb/s Switching fabric	CSC
6	4 port OC-12/POS line card	OC-12
7	4 port GE line card	4GE
8	Route Processor	RP
9	Empty	None

Table 2.3: *Cisco GSR 12008 configurations used in power consumptions benchmarking experiments. Source [10].*

Table 2.4: *Cisco 7507 configurations used in power consumptions benchmarking experiments. Source [10].*

Chassis Slot	Line Card	Abbreviation
0	1 port GE line card	GE
1	1 port FE line card	FE
2	Route Processor	RP
3	Empty	None
4	1 port FE line card	FE
5	1 port DS1 line card	DS1
6	1 port FE line card	FE

minimize the number of chassis powered on at a given PoP, maximizing the number of line cards for chassis. The experimental results lead to the development of the following generalized model for routers power consumption PC:

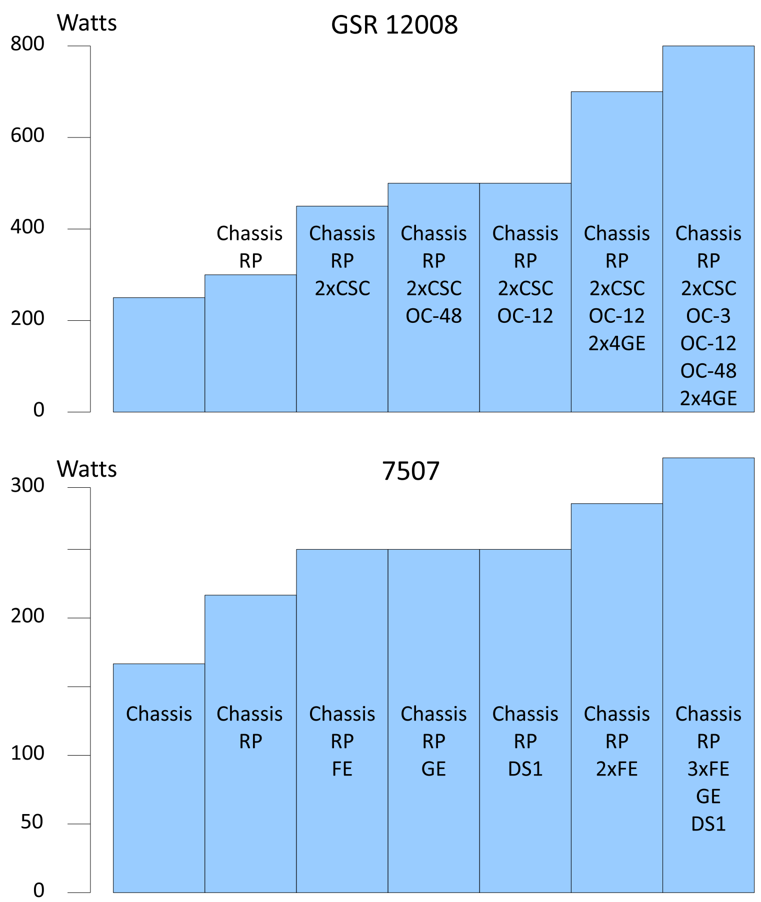


Figure 2.7: *Power consumption for different configurations of the GSR and of he 7507. See table (2.4)(2.3) for the abbreviations. Source [10].*

$$PC(X) = CC(x_0) + \sum_{i=0}^{N}(TP(x_{i0}, x_{i1}) + LCC(x_{i1})) \quad (2.4)$$

The vector X defines the router configurations (router type, chassis and line cards installed) and the traffic profile used. $CC(x_0)$ gives the power consumption of a particular chassis type, N is the number of active line cards, $(TP(x_{i0}, x_{i1})$ is a scaling factor corresponding to the traffic configuration on the router, and $LCC(x_{i1})$ returns the line card cost in base configuration. This model has been utilized in the optimization problem of network power aware design. The network design problem has been formulated as a multicommodity flow problem with design variables indicating the different configurations utilized at each node. The problem is NP-hard, but branch-and-bound and cut generation techniques let to solve many instances in a reasonable time. The results obtained show the possibility to save a percentage of energy over 50%.

2.2.3 Adaptive Link Rate (ALR)

In USA, Ethernet network interface controllers (NICs), consume hundreds of million of dollars of electricity per year. The majority of Ethernet links is underutilized, and it would be therefore possible to reduce the power consumption operating at lower data rate. Ethernet links operate at three different rates, 100 Mbps, 1 Gbps, 10 Gbps. Measurements show that 1 Gbps Ethernet links consume 4 W more than 100 Mbps links, and 10 Gbps links should consume from 10 to 20 W more.

In [11] and [12] Adaptive Link Rate (ALR) techniques for Ethernet links are discussed. ALR consists into adapting link data rate to link utilization. The ALR development presents two main issues to be handled: 1) definition of sufficiently quick switching-rate mechanisms and 2) creation of policies able to maximize the energy saving without excessively increasing the packets delay. In [11] the authors illustrate one switching mechanism (ALR MAC Frame Handshake Mechanism) and three different switching policies (ALR Dual-Threshold Policy, ALR Utilization-Threshold Policy, ALR Time-Out-Threshold Policy).

ALR switching mechanisms

The handshake mechanism has the necessity to be faster as possible, in order to avoid great packets delays. The ALR MAC Frame Handshake Mechanism is based on a two ways handshake ant it could be successfully completed in less than 100 microseconds. The details of the procedure are shown in figure (2.8).

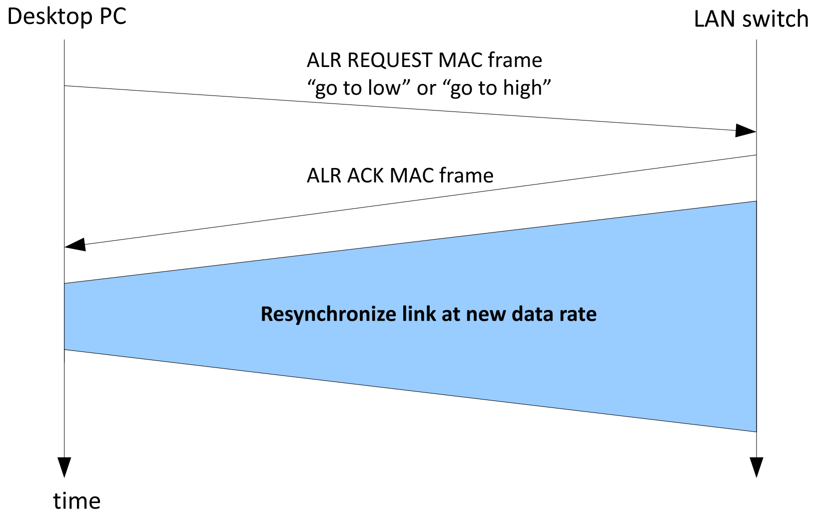


Figure 2.8: *Timing diagram for an ALR MAC frame handshake mechanism. Source [11].*

ALR policies

A good ALR policy has to guarantee robustness against rate oscillation. The ALR Dual-Threshold Policy, the simplest of the three policies proposed, is based on the output buffer occupancy; it handles the oscillation issue utilizing a system of two thresholds (qHigh and qLow). A rate reduction request has to be negotiated by the both sides (exchanging ACK or NACK messages), while a rate increase request has to be always accepted by the receiver side. The adoption of a two thresholds system does not guarantee the complete absence of undesired rate oscillations. These oscillations could occur if the packet arrival rate at the low link data rate, is high enough to cause the high threshold crossing, but not high enough to maintain the queue length above the low threshold at the high data rate. The ALR Utilization-Threshold Policy has been therefore developed in order to avoid the occurrence of these undesired oscillations. This policy is based on the addiction of utilization thresholds in conjunction with the occupancy thresholds (the condition $(qLen < qLow)$ is replaced by $((qLen < qLow) \wedge (uBytes < uThresh))$. The link utilization is directly monitored by counting the number of bytes transmitted in a given time period. Finally, the ALR Time-Out-Threshold Policy introduces two new timers, indicating the minimum time of mandatory permanence in high and low states after a rate switching.

Parameter	Assumed	Calculated
Total stock (desktop PCs only)	80 million	
Ethernet Connected fraction	90%	
Ethernet Connected Pcs		72 million
Number of NICs/product	1	
Work-time potential low-rate time	8 hrs/day	
Night and weekend full-on time	67%	
Weekly potential ALR low-rate time		about 70%
		6100 hrs/day
Annual savings per ALR PC link		12.2 kWh/yr
Annual savings per ALR PC link at $0.08/kWh		$0.98
ALR link penetration	100%	
Saving for 72 million ALR PC links		70 million/yr

Table 2.5: *ALR energy saving estimations: assumptions and calculations.*

Power Saving Estimations

The widespread deployment of ALR techniques in the commercial desktop sector, could bring significant energy saving results. Making the following assumptions, it is possible to calculate a conservative estimation of potential future energy saving in USA:

- Typical workday of 8 hours.
- 2/3 of PC are left powered on during nights and weekends
- The full 1 Gbps data rate is really necessary only one hour per workday.
- Power difference of 2 W between 100 Mbps and 1 Gbps data links.
- 72 million of Ethernet-connected commercial desktop PCs.

It has been calculated a potential energy saving of about $70 million per year. Table 2.5 summarizes the assumptions and the calculations done.

2.3 Green Energy for Green ICT

An efficient network management could concern, not only energy consumption, but also energy cost and availability. Energy producers and consumers are connected by an electric grid; the electricity storage is really complex, and it is therefore necessary to continuously balance the supply in relation with the demand. This situation

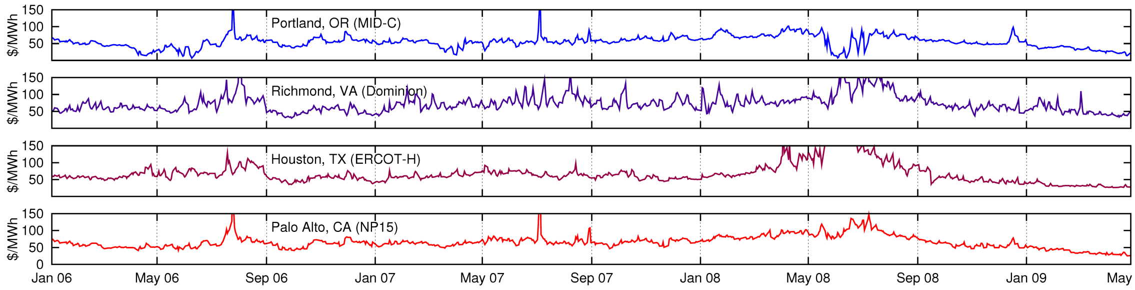


Figure 2.9: *Plot of the daily averages peak prices in four different regions in USA (day-ahead market). Observe how the peak of 2008 caused by the growth of natural gas prices, does not affect the hydroelectric dominated north-west region (first plot). In this region, there is also every year a strong decrease of the electric price around April, in correspondence with seasonal rainfall. Source [13].*

causes continuous oscillations, both temporal and geographic, of the electricity prices (see figure (2.9)).

Nowadays, a lot of Internet-based systems are geographically distributed, with servers and machines located in hundreds of sites around the world (see table 2.6): it would be therefore convenient to develop dynamics systems, able to mostly exploit the infrastructures deployed in regions with better energetic conditions (prices, availability, green level). In general there are four typologies of strategies that could be implemented for a cost-availability aware network management:

- **Implementation of joint optimization:** nowadays the network optimization is based on bandwidth costs, performances and reliability. Also dynamic energy costs should be considered for a more efficient analysis.

- **Regional Transmission Organization RTO interaction:** service operators could interact with electric consumers following different ways. It is important to investigate strategies for an optimal market participation.

- **Exploitation of weather differences:** a significant part of the energy used by network infrastructures is consumed by cooling systems; it is possible to reduce the energy requirements of the cooling systems, by mapping requests to location with lower ambient temperatures. In this way it would be possible to use external air. This typology of strategies lets to the reduce at the same time both costs and consumptions.

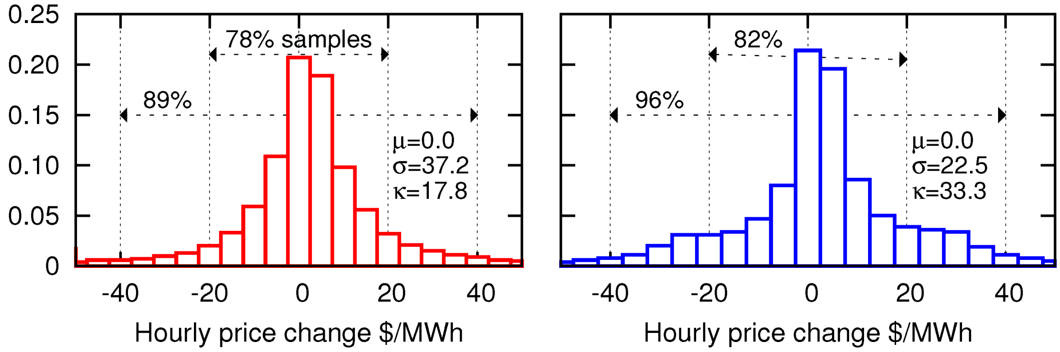
- **Exploitation of green energy:** the carbon footprint of the energy produced depends by the generating assets activated. It therefore could be socially responsible to develop mapping policies able to maximize the utilization of the clean energy available.

Table 2.6: *Estimated annual electricity costs for large companies (servers and infrastructures) @ \$60/MWh. Source [13].*

Company	**Servers**	**Electricity**	**Cost**
eBay	16K	$\sim 0.6 \times 10^5$ MWh	~\$3.7M
Akamai	40K	$\sim 1.7 \times 10^5$ MWh	~\$10M
Rackspace	50K	$\sim 2 \times 10^5$ MWh	~\$12M
Microsoft	>200K	$> 6 \times 10^5$ MWh	>\$36M
Google	>500K	$> 6.3 \times 10^5$ MWh	>\$38M
USA (2006)	10.9M	610×10^5 MWh	\$4.5B

Table 2.7: *Real-time market statistics, covering hourly prices form January 2006 to April 2009. Source [13].*

Location	**RTO**	**Mean**	**StDev**
Chicago, IL	PJM	40.6	26.9
Indianapolis, IN	MISO	44.0	28.3
Palo Alto, CA	CAISO	54.0	34.2
Richmond, VA	PJM	57.8	39.2
Boston, MA	ISONE	66.5	25.8
New York, NY	NYISO	77.9	40.26



(a) Palo Alto (NP15) **(b) Chicago (PJM)**

Figure 2.10: *Histograms of hour-to-hour changes in real-time hourly prices for two locations, over the 39-months period. Both distributions are zero-mean, Gaussian-like, with very long tails. Source [13].*

The effective efficiency of all the strategies previously presented, strongly depends by the energy elasticity of the network infrastructures. The energy elasticity represents the dependence between power consumptions and load levels; in the ideal case there would not be any consumptions in absence of load. The idle power of actual state-of-the-art systems is about 60% of the peak power.

[13] describes with a good accuracy the features of the electric market in US, presenting also a simple cost-aware routing policy that maps each request to the location with the cheapest electric prices: the aim is to provide some bounds concerning the possible electric costs reductions. The work is based on data about the electricity hourly prices in 29 US locations during a period of 39 months (from January 2006 to April 2009). These data confirm the great volatility of the hourly electricity prices (see table 2.7 and figure (2.10)), and their elaboration suggests a possible future potential saving near to 20%.

Chapter 3

Green IP Routing

Our research is focused on the development of efficient algorithms for the dynamic management of network resources, considering different traffic conditions. In this chapter we want to present some interesting already published works, [14],[15],[16] and [17] , where new switching-off algorithms are proposed: the aim is to provide the elements necessary to evaluate the validity of our work.

3.1 Heuristics for network global management

Given a physical network topology with routers and links, the goal of an efficient energy management procedure is to minimize the global power consumption, always maintaining the required quality of service QoS. The problem could be expressed as follows by a mixed integer linear programming formulation:

$$min \ \ P_{tot} = \sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij} PL_{ij} + \sum_{i=1}^{N} y_i PN_i \tag{3.1}$$

Subject to:

$$\sum_{j=1}^{N} f_{ij}^{sd} - \sum_{j=1}^{N} f_{ji}^{sd} = \begin{cases} t^{sd}, & \forall s,d,i = s \\ -t^{sd}, & \forall s,d,i = d \\ 0, & \forall s,d,i \neq s,d \end{cases} \tag{3.2}$$

$$f_{ij} = \sum_{s=1}^{N}\sum_{d=1}^{N} f_{ji}^{sd} \ \ \forall i,j \tag{3.3}$$

$$f_{ij} \leqslant \alpha c_{ij} x_{ij} \ \ \forall i,j \tag{3.4}$$

$$\sum_{j=1}^{N} x_{ij} + \sum_{j=1}^{N} x_{ji} \leqslant My_i \ \forall i \tag{3.5}$$

The network topology is represented by a graph $G = (V, E)$ where V and E are respectively the set of vertices and the set of edges; vertices represent network nodes (routers) and edges represent network links. N and L are the total number of nodes and links. Let c_{ij} be the parameter that indicates the capacity of the link between nodes i and j, let $\alpha \in \{0, 1\}$ be the maximum link utilization allowed and let t^{sd} be the average traffic demand between nodes s (source) and d (destination). Let $x_{ij} \in \{0, 1\},\ i = 1, \ldots, N,\ j = 1, \ldots, N$ and $y_i \in \{0, 1\},\ i = 1, \ldots, N$ be the binary variables that assume the value of 1 if respectively, the link between nodes i and j, and the node i, are powered on. Let $f_{ij}^{sd} \in \left[0, t^{sd}\right]$ be the the variables that point out the portion of the demand t^{sd} routed through the link between nodes i and j; f_{ij} represents instead the total amount of traffic routed through the arc from i to j. Finally PL_{ij} and PN_i indicate the power consumption of the link from i to j, and of node i, respectively.

The objective function 3.1 minimizes the total power consumption. Eq. 3.2 represents the classical flow conservation constraint and eq. 3.3 provides the total flow on each link. Eq. 3.4 is the capacity constraint, while eq. 3.5 forces, through the big-M method ($M \geqslant 2N$), to power off the links connected to a deactivated node.

This formulation falls in the class of capacitated multi-commodity minimum cost flow problems (CMCF); these problems are known to be NP-hard [18] , and it is impossible to use exact methods to solve real instances. The way that could be followed is therefore represented by the definition of heuristic methods. [14] describes some simple heuristic algorithms able to switch off nodes and links respecting traffic constraints. The heuristic algorithms proposed, consider at the beginning a fully powered on network; then they start iteratively to switch off nodes and links. Nodes consumptions are definitely higher than link consumptions: for this reason the algorithms try at the beginning to switch off all the possible nodes, checking the single links only in a second stage. The possibility of switching off an element, is verified by rerouting each demand t^{sd} on the shortest path between s and d, and then checking the validity of constraints 3.2 and 3.4; if no violation occurs, it is possible to switch off the candidate elements. The differences between the various algorithms presented, consist into the policy followed for choosing the elements to switch off at each iteration. These policies are:

- **Random**: nodes and links are chosen randomly.

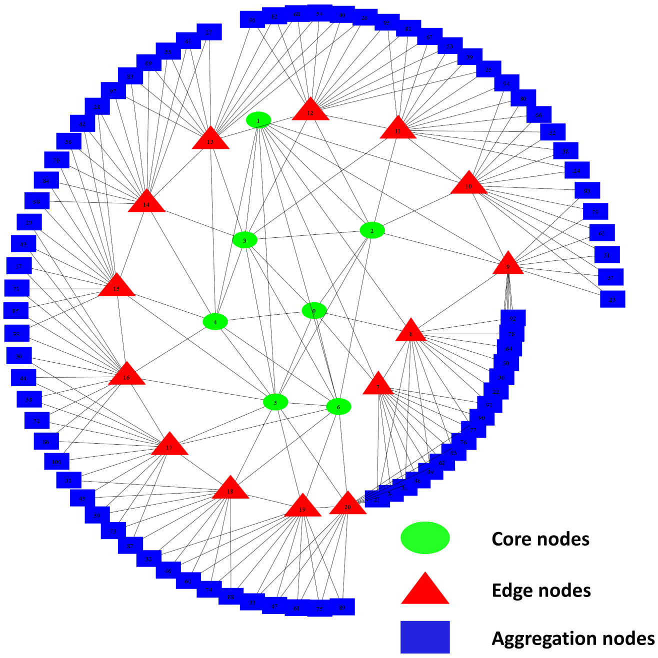


Figure 3.1: *Model of the random topologies utilized to test the heuristics in [14].*

- **Least-link**: nodes are sorted according to their grade (number of links connected)

$$X_i = \sum_{j=1}^{N} x_{ij} + \sum_{j=1}^{N} x_{ji}$$

Nodes with the minor grade are checked first. Obviously it has no sense to apply this policy to the links.

- **Least-flow:** nodes and links crossed by the minor amount of flow are checked first; for what concerns nodes, they are sorted in increasing value of

$$F_i = \sum_{j=1}^{N} f_{ij} + \sum_{j=1}^{N} f_{ji}$$

Links are instead sorted in increasing value of f_{ij}.

- **Opt-edge:** for protection purposes, in current Internet, aggregation nodes are connected each one to two different edge nodes (dual-homing). Because of this characteristic it is impossible to switch off simultaneously two edge nodes connected to the same aggregation node. Opt-edge policy orders the nodes selecting only one node of each pair.

The researchers have tested all the combinations of these policies on random generated networks with a hierarchical structure typical of WANs (see figure (3.1)).

There are three categories of nodes, core, edge and aggregation nodes, and three classes of links, high, middle-range and low capacity links. Each class of links has a minimum capacity c_{ij}^{min} (1 for low capacity links, 5 for middle-range links, 15 for high links); the inverse of the capacity is used as link routing weight, in order to force the traffic to be routed through edge and core nodes. The traffic demands are routed through the minimum cost path and then, if necessary, links capacities are adapted in order to maintain the utilization level of each link under 50%:

$$c_{ij} = min(\lceil f_{ij}/0.5 \rceil, c_{ij}^{min})$$

The networks generated and used for tests, have 160 nodes; in particular 10 core nodes, 30 edge nodes and 120 aggregation nodes. Core nodes are connected to other core nodes with a probability of 0.5. Edge nodes are connected to the two closest core nodes and to another edge node. Aggregation nodes are finally connected to the two closest edge nodes. Traffic demands could have as source and destination only the aggregation nodes: in this way is therefore possible to switch off edge and core nodes. The traffic pattern considered is uniform.

[14] and [16] report the computational results obtained from tests executed with two different traffic conditions. In [14] the algorithms work with the full traffic matrix (peak workload conditions) and links utilization has to not exceed the 80% (in eq. 3.4 $\alpha = 0.8$). Instead, in [16], the traffic matrix utilized for the tests is scaled by a factor of 0.2, in order to simulate the low traffic conditions typical of nocturne hours; moreover the maximum links utilization allowed cannot exceed 50%.

Instead in [15], differently from the two works just cited, the researchers have tested the algorithms utilizing realistic traffic profiles, and realistic figures of power

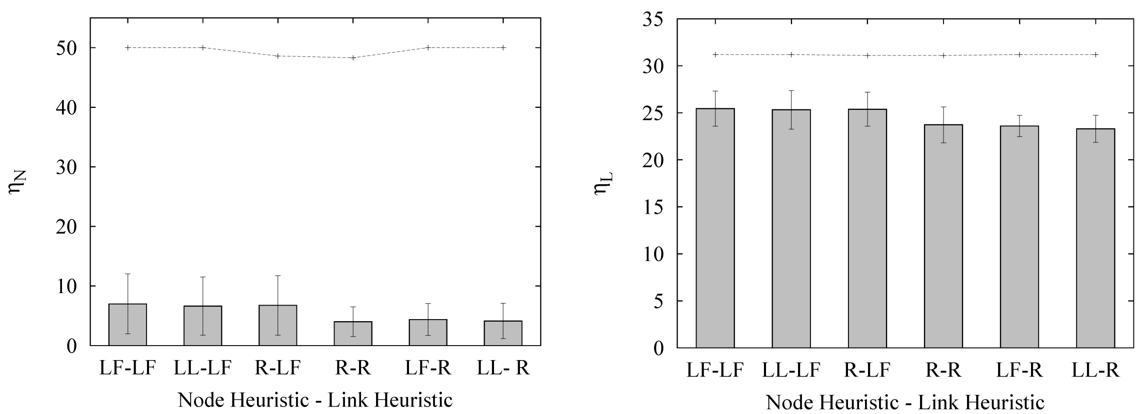


Figure 3.2: *Comparison of the percentage of links η_L and nodes η_N switched off with the different algorithms in high workload conditions. Source [14].*

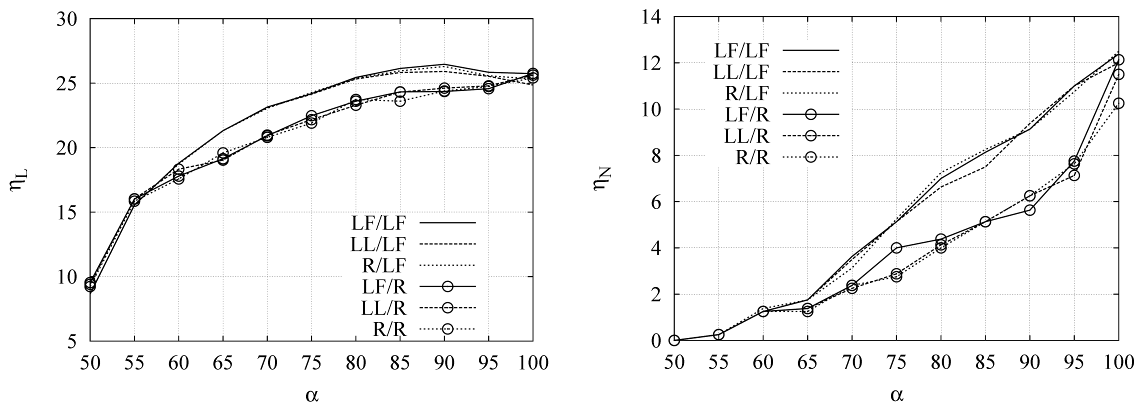


Figure 3.3: *Percentage off links η_L and nodes η_N switched off versus α in high workload conditions. Source [14].*

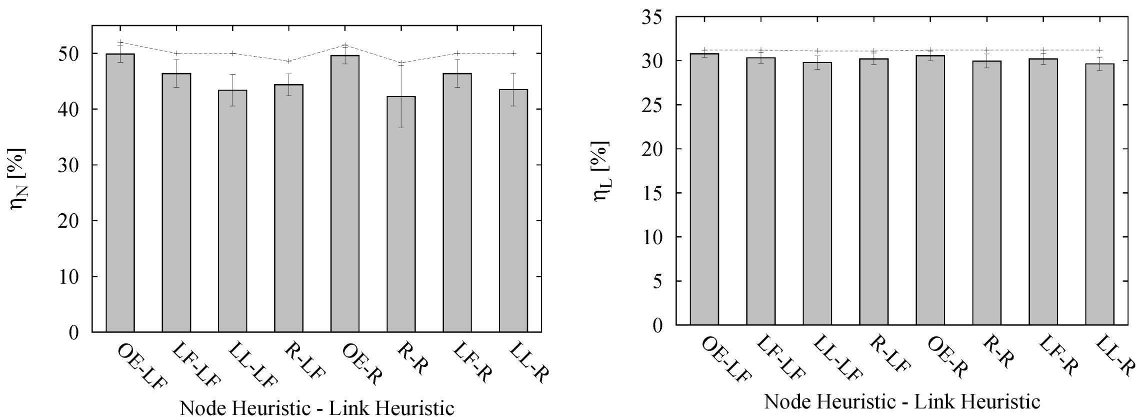


Figure 3.4: *Comparison of the percentage of link η_L and nodes η_N switched off with the different algorithms in low workload conditions. Source [16].*

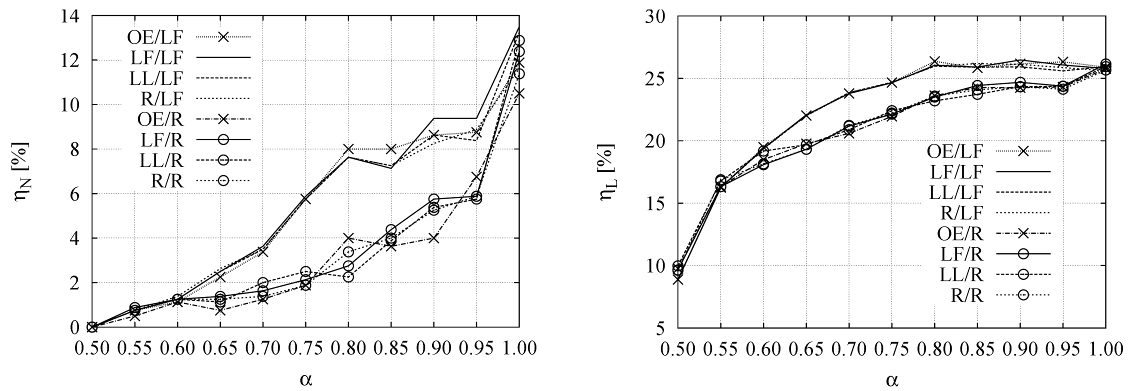


Figure 3.5: *Percentage off links η_L and nodes η_N switched off versus α in low workload conditions. Source [16].*

consumptions derived from available products; the goal is to quantify the efficiency improvements in terms of the amount of energy saved.

Figure (3.2) and (3.4) shows the percentage of links η_L and nodes η_N switched off with the various algorithms, under peak and low traffic conditions, respectively. The data presented have been collected after tests on 20 randomly generated topologies and traffic patterns. The figures reports both mean and standard deviation values. Tests suggest how it would be possible, under high workload conditions, to switch off about 25% of links and 5-10% of nodes; during low workload periods it is instead possible to powered off about 50% of nodes and 30% of links. In general, random link selection policies show the worse results, if compared with the least flow and opt-edge selection policies. Figures (3.2) and (3.4) report also an upper bound, calculated by relaxing the capacity costraint 3.4; in this way it is possible to obtain the minimum set of nodes and links that guarantees the full connectivity. The percentage of switched off nodes with high workload conditions, is the only that remains very distant from the bound. Finally, figures (3.3) and (3.5) show the relation between the utilization parameter α and the algorithm performances. All the heuristics present significant improvements for α up to 0.8.

The results presented seem at a first sight to be really impressive: the percentage of network elements switched off is very high and, above all, very close to the bounds. In the lecture of these results it is however necessary to do some important observations:

- Scarce complexity of the network topology adopted: the adopted topologies are very simple, with a strongly symmetric and hierarchical structure. These factors strongly influence the algorithms performances. It would be interesting to evaluate the algorithms with more complex topologies, like the real ISP topologies proposed in [19].

- Too low traffic demands: the traffic demands used to simulate heavy workload conditions don't seem to be enough high. The fact is aggravated by the capacity adaptation procedure that force the initial utilization of each link to be under 50%.

In the end we think that the algorithms proposed would have a lot of difficult to converge to good solutions, if tested with more realistic instances. In our opinion the risk of finding inefficient local optimums without the possibility to escape, is very high.

3.2 The Energy Aware Routing Algorithm

OSPF (Open Shortest Path First) is, with MPLS (Multi Protocol Label Switching), the most utilized routing protocol in IP networks. According to the OSPF protocol, each router, through the flooding mechanism, transmits a Link State Advertisement (LSA) packet to all the other network routers. The LSA packet describes all router links reporting also the OSPF weights. At this point each router is able to exploit the information received by the LSA packets, and uses the Djikstra algorithm to calculate its own Shortest Path Three (SPT). The traffic packets will be therefore routed through the links that belong to the SPT. The multipath could be utilized when more equivalent shortest paths are detected.

A link can be utilized only if it belongs to at least one SPT. In [17] this characteristic is exploited in order to switch off network links. The researchers have proposed a new algorithm called EAR (Energy-Aware Routing) algorithm, that performs packet routing exploiting only a subset of SPTs: the number of links used is thus reduced. Let analyse this point more deeply: given a graph $G = (V, E, W)$, V,E and W represent, respectively, the set, of nodes, unidirectional links, and links weights. Let R be the cardinality of the set V and let L be the cardinality of the set E. If we denote with SPT_k the SPT computed by k-th router, then the subgraph of G obtained by the superposition of each SPT is

$$SPG(V, E_s, W_s) = \cup_{1=1..R} SPT_i$$

E_s represents the set of links that belong to at least one SPT, while L_D is the cardinality of E_s. It can be easily demonstrated that

$$L_D \leq L$$

In [17] the minimum value of L_D is calculated as:

$$L_{Dmin} = 2(R - 1)$$

It can be obtained when all the routers compute the same SPT. This parameter is very important because it is used to evaluate the efficiency of EAR algorithm. However a correct analysis shows how the value of L_{Dmin} has been overestimated: the correct value (as shown in figure(3.6)) should in fact be:

$$L_{Dmin} = R$$

The overestimation of L_{Dmin} causes a better evaluation of the computational results reported in the paper.

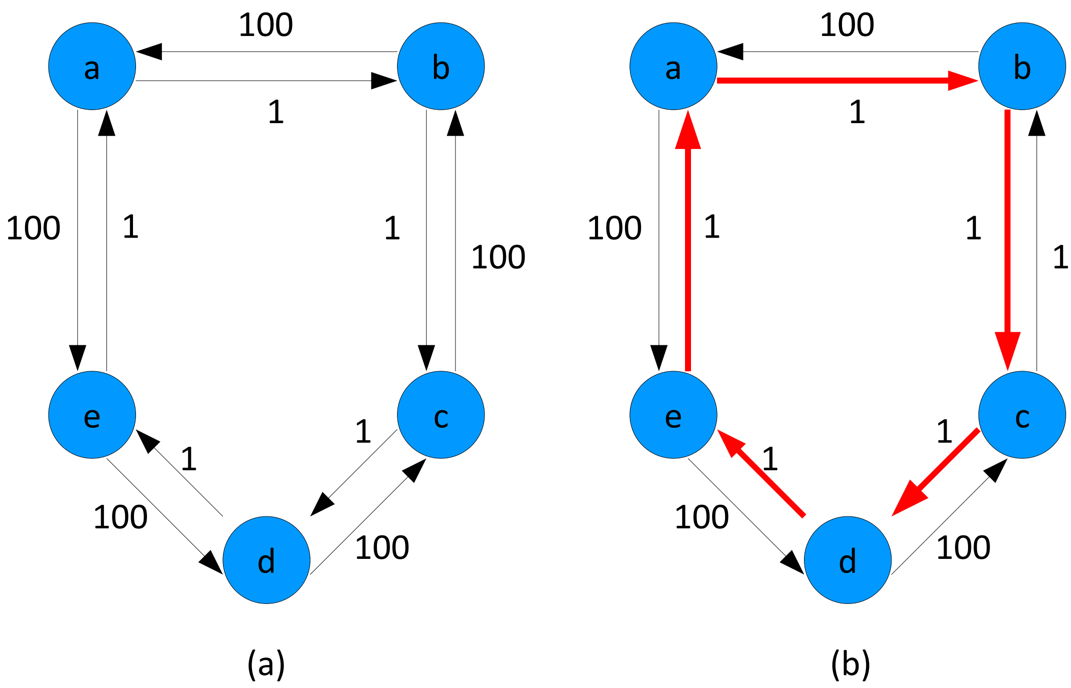


Figure 3.6: *a)Circular topology with links weights b) SPG calculated by the routers (red links).* $L_{Dmin} = 5 = R$ *and* $L_{Dmin} \neq 8 = 2(R-1)$.

In general it is possible to switch off an higher number of links when the routers SPTs are similar. The EAR algorithm reduces SPTs differences, forcing a class of nodes (import routers IR) to compute their SPT on the base of the SPT received by a second class of routers (exporters routers ER). The selection of the exporters routers represents the key stage of the algorithm: the number of ER (R_e) and the selection policy strongly influence EAR performances. The selection policy adopted in [17] consists into the choice of the routers with the highest degree values. In general exporters routers normally calculate their SPT, and then transmit it to their neighbour import routers. The import routers have therefore only to modify the received SPT, by inverting the sense of the SPT links placed on the path between them and the export router (see figure(3.7)); the IRs could then switch off the links than don't belong to the modified SPT. The modified tree is called Modified Path Tree (MPT). Finally, each router performs a normal OSPF routing on the modified topology obtained by the union of all the MPTs.

Considering L_e as the number of active links when EAR is performed, and certifying the validity of the following expression

$$L_{Dmin} \leq L_e \leq L_D \leq L$$

it is possible to evaluate EAR performances considering the ratio (η_e) between the

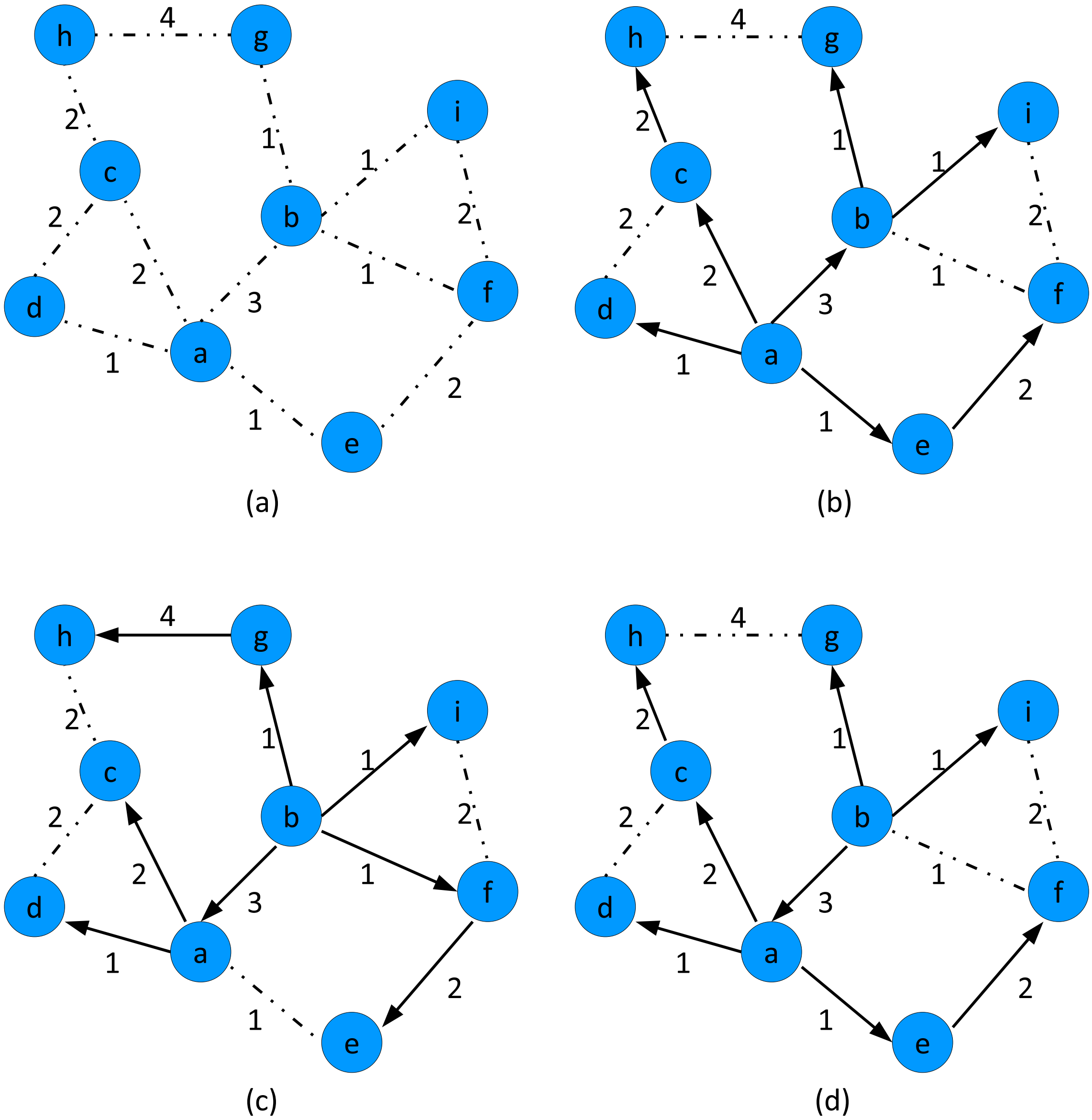


Figure 3.7: *a)Example of network graph with OSPF weights. b) SPT computed by a. c) SPT computed by b. d) EAR algorithm performed by b if a is the exporter router.*

links switched off by EAR and the maximum number of links that could be powered off, i.e.

$$\eta_e = (L_D - L_e)/(L_D - L_{Dmin})$$

It is necessary to observe that EAR does not consider both traffic matrices and QoS constraints. The researchers considerate it as an improvement to the OSPF protocol that could be used during nocturne hours, when the traffic is significantly decreased.

The EAR algorithm has been tested on a real network topology [19] composed of 244 routers and 1080 unidirectional links. OSPF links weights are not known: the researchers have therefore fixed the weights assuming a uniform distribution among a mean value of 10, with three different values of standard deviation σ (0, 5, 9). The computational results reported in figure (3.8) highlight the possibility to reach levels of η_e around 60%. However, evaluating these results, it is necessary to not forget that η_e has been calculated with $L_{Dmin} = 2(R-1)$. As expected, the algorithm performances improve when the ER number increases.

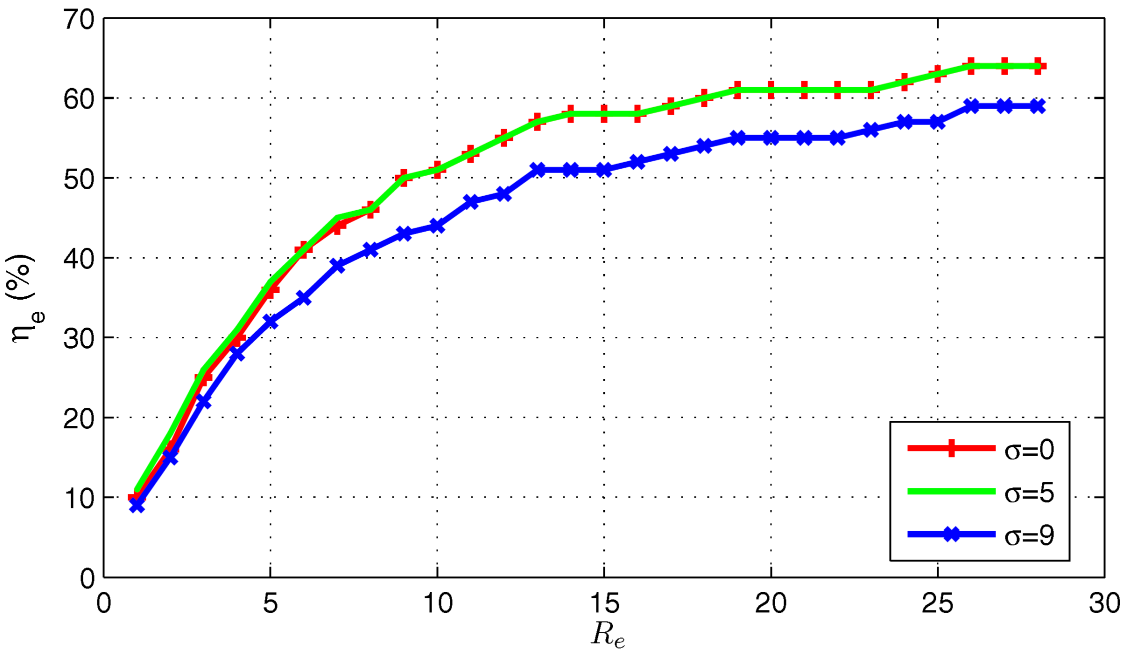


Figure 3.8: *Levels of η_e reachable performing the EAR algorithm. Source [17]*

Chapter 4

Optimized Shortest Path for Energy Saving: switching-off stage

OSPF is the most commonly adopted intra-domain internet routing protocol. Traffic demands are routed along the shortest paths, being split at nodes where more outgoing links belong to the shortest paths in direction of the destination. Links weights are managed by network operators, that have the possibility to modify them in order to avoid congestion. The goal of the Optimized Shortest Path for Energy Saving (OSPES) procedure is to exploit the configuration of OSPF weights, not only to avoid congestion, but also to perform energy saving. This purpose could be achieved by configuring the OSPF links weights, in order to determine a general feasible routing (the capacity constraint has to be respected for each link) that would be able to involve only a part of the network elements (links and routers). The unnecessary routers and links could be in fact excluded by each shortest path tree, by configuring the correspondent links weights with very big values (see figure (4.1)).

The problem of finding a *low traffic* version of a network together with a new feasible set of OSPF links weights, is very complex and it is therefore necessary to split the resolution of the problem in two distinct stages. The first one, the switching-off stage, is described in this chapter; it has the goal, given a network topology and different levels of traffic, to provide the reduced versions of the network. The second one, the feasible routing stage, will be instead discussed exhaustively in the next chapter. It receives as input the network topologies obtained by the first stage, and it has the aim to find a set of links weights able to define a feasible shortest path routing for all the traffic demands. However, given a traffic matrix and a network topology, there is no guarantee concerning the existence of a feasible set of weights. The infeasibility of the solution could be overcome adjusting the traffic

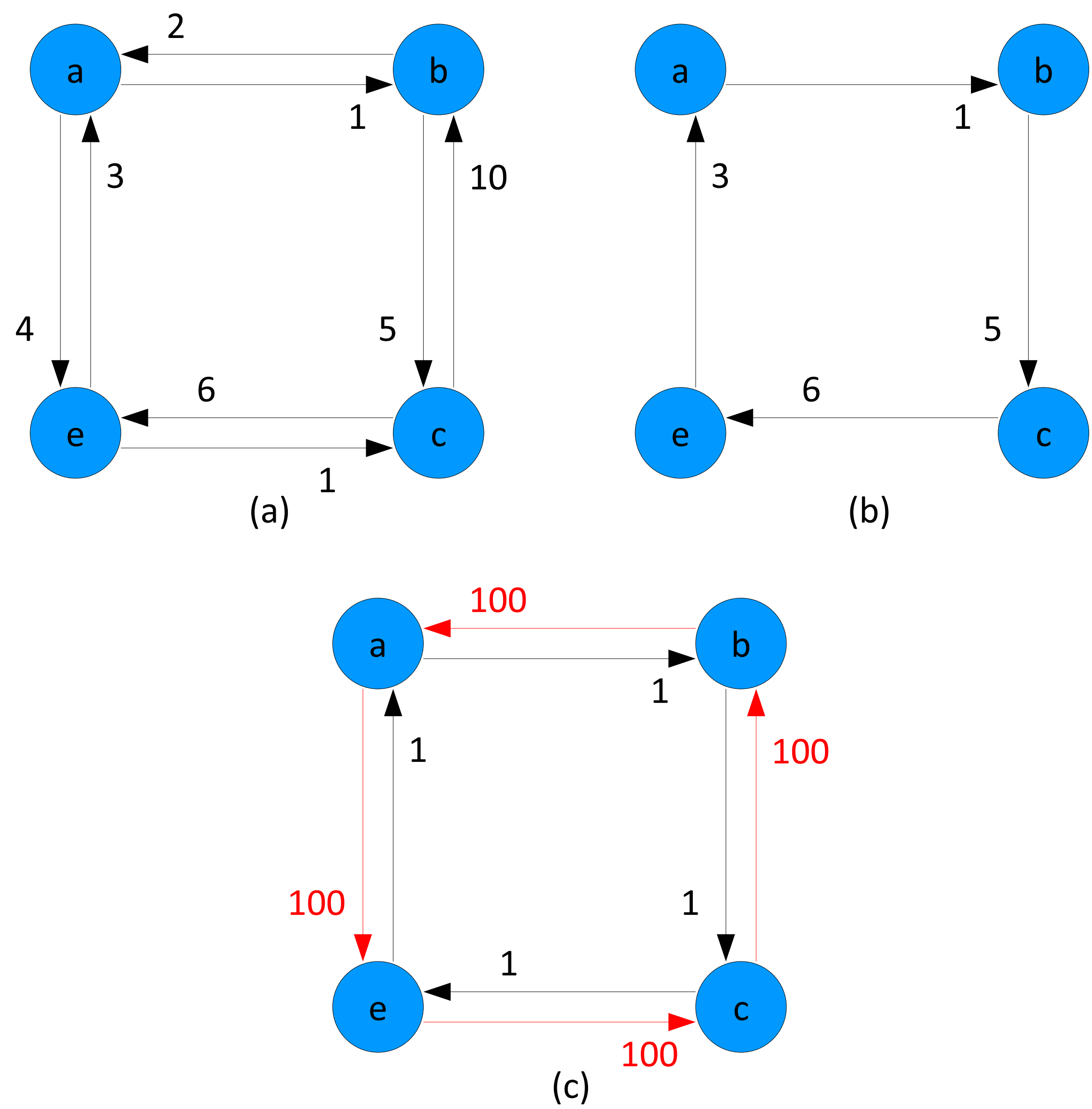


Figure 4.1: *Example of the Optimized Shortest Path for Energy Saving (OSPES) procedure . a) Network topology with links weights. The capacity of each link is assumed to be of 1000 units; a traffic demand of 30 units is considered for each different couple source-destination. b) Reduced network topology obtained by the first stage. c) Final solution obtained by the feasible routing stage: the weights of the deactivated links (red links) are very big and the remaining links weights guarantee the feasible routing of all the traffic demands.*

levels considered.

Given a network topology and a maximum sustainable traffic matrix (the biggest traffic matrix sustainable by the network without exceeding the links capacity), the first stage has the aim to calculate the different reduced topologies, derived utilizing the traffic matrix versions obtained by multiplying the full matrix with a parameter comprise between 0 (minimum traffic) and 1 (maximum traffic)(see figure

(1.2)). Unfortunately the problem of switching off network elements guaranteeing the feasible routing of all the traffic demands, falls in the class of the capacitated multi-commodity minimum cost flow problems (CMCF), i.e., the problem in which multiple commodities have to be routed over a graph with capacity constraints. CMCF problems are known to be NP-hard [18], so exact methods can only used to solve trivial cases. We consequently force our efforts into the development off heuristic methods.

We have already seen in [14] that greedy algorithms could be utilized in order to achieve our goal. However, we think that greedy algorithms could present a series of problems: at first, the greedy structure does not assure satisfactory performances in case of complex network topologies; this occurs because of the lack of a robust element-selection policy, and because of the strong influence that each iteration has on the successive (the inefficient choice of a link or node could affect negatively the entire algorithm proceeding). In addition, we think that routing constraints should play a different role in the switching-off procedure: in [14] the OSPF routing is performed at each iteration, in order to confirm or not the possibility of switching off a certain node/link; the current OSPF routing therefore actively influences the algorithm operations. Our idea, has been instead to develop a switching-off stage where, also for a reason of minor complexity, the routing is consider completely free: it has only to respect capacity and flow conservation constraints. In this way we can operate in a second moment (the second stage) on the OSPF weights in order to find a feasible routing for all the commodities. In conclusion, we have chosen to adapt the routing to the reduced network, rather than to determine the reduced network, by being conditioned by the current routing (see figure(4.2)).

The switching-off stage that we have developed, consists into two mixed integer programming formulations (one switches off only links, one both nodes and links,) that are processed by a solver (CPLEX) through its branch-and-cut algorithm. Obviously it's impossible to reach the optimum solution for big instances, but, how we will show, the found solutions are enough close to the calculated bounds (around 20%). In addition, we have defined a pre-processing data procedure that lets to add to the formulation a great number of constraints: these constraints help to reduce the solutions space and , in some cases, let to find better solutions in a fewer time.

4.1 Problem Formulations

The problem handled at the first stage can be described as follows: given a traffic matrix and a network topology, including routers, links and links capacity, the goal

Table 4.1: *Traffic demands of figure 4.2.*

Source / Destination	**A**	**B**	**C**	**D**	**E**
A	/	70	2	0	9
B	0	/	70	0	0
C	0	0	/	0	0
D	0	0	0	/	0
E	0	0	9	0	/

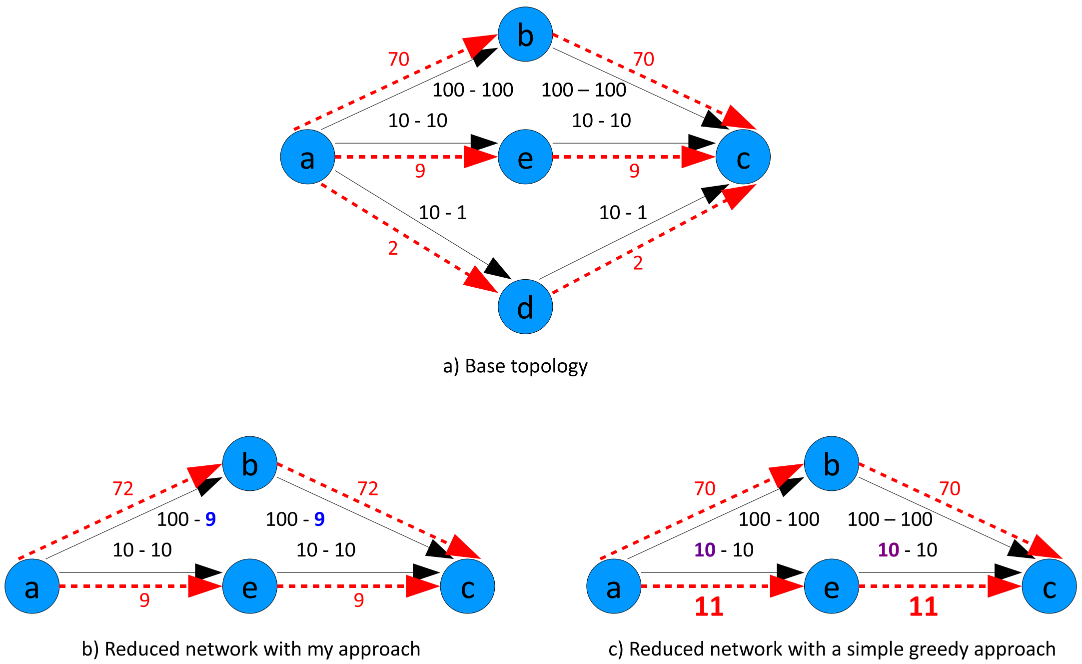


Figure 4.2: *a) Network topology with links capacity, links weights and links flows. Links capacities and weights are represented by the couples (capacity-weight). Links flow is represented by the red dashed arrows. Traffic demands are routed with the classic OSPF protocol (through the shortest path); traffic demands are reported in table 4.1. b) Switching off node d and its links, it is then possible to route the traffic demand between a and c only using the path $a-b-c$; path $a-e-c$ would be in fact saturated. It is therefore necessary to change the weights of links $a-b$ and $b-c$ in order to force the OSPF routing to use the path $a-b-c$ rather than the path $a-e-c$. c) A normal greedy algorithm like those presented in [14], wouldn't be able to switch off any elements because of the impossibility of routing the traffic demand between a and c through the saturated path $a-e-c$. $a-e-c$ represnts in fact the shortest path between a and c.*

is to find the maximum set of links to be powered off, guaranteeing at the same time a feasible routing. That means that feasible solutions have to respect flow conservation and capacity constraints. Traffic demands are freely divisible through the network (each demand is routed following the fluid/splittable multicommodity flow model, see figure (4.3)).

The network topology is represented by a set off nodes V of cardinality N, and a set of binary parameters e_{ij}, $i = 1, \ldots, N, j = 1, \ldots, N$ that are equal to 1 when a directional link from node i to node j exists ; Let A be the total number of links. Let c_{ij} be the capacity of the link from node i to node j, and let d_{ij} be the traffic demand between nodes $i = 1, \ldots, N$ and $j = 1, \ldots, N$. PN_i and PL_{ij} represent, respectively, the power consumption of node i and the power consumption of the link that connects node i and node j. PL_{ij} is assumed to be equal to 1 for each link (i, j), while PN_i is calculated as

$$PN_i = \lceil \tfrac{3g_i}{2} \rceil$$

where g_i represents the degree of node i. In this way, considering the set composed by a router and its links, the router is responsible for about 66% of the total power

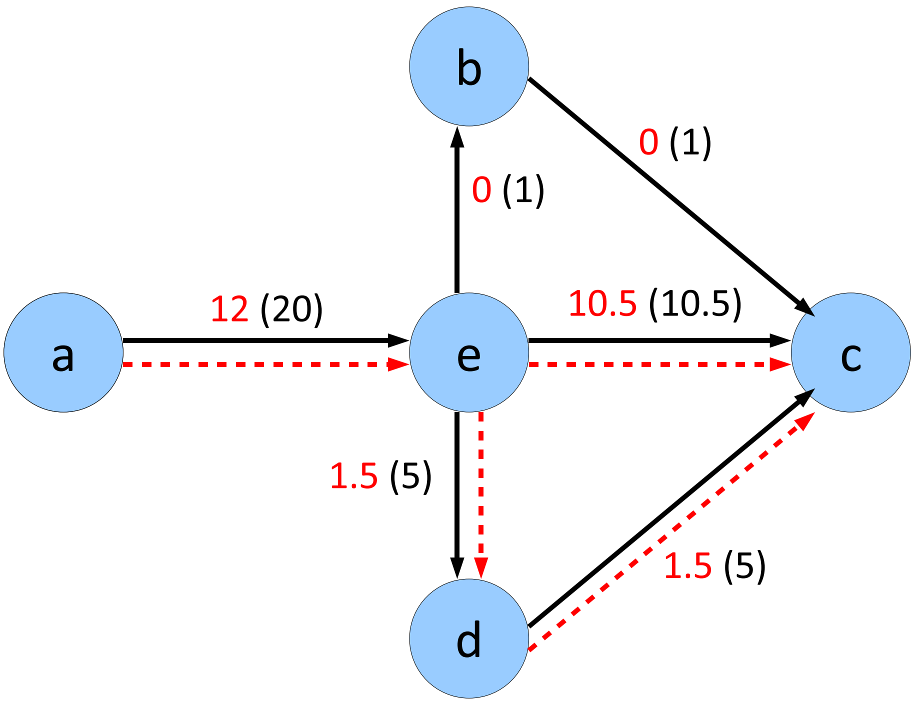


Figure 4.3: *Example of how it is possible to route traffic demands following the fluid model: at node e, because of the insufficient capacity of the link $(e - c)$,the traffic demand could be easily split on two different paths; it should be possible to use three different paths too. Links are represented by black arrows, while flows are represented by red dashed arrows; near each link, link flow (red) and link capacity (black between parenthesis) are reported.*

consumption. Let $y_{ij} \in \{0,1\}, i = 1,\ldots,N, j = 1,\ldots,N$ be binary variables that assume the value of 1 when the the link from node i to node j is powered off. Let $z_i \in \{0,1\}, i = 1,\ldots,N$ be binary variables that assume the value of 1 when node i is powered off. Let $x_{ijs} \in [0, \sum_{d=1}^{N} d_{sd}]$ be continue positive variables that represent the amount of traffic originated in node s, routed through the arc from i to j. Notice that variable x_{ij} is real; it is therefore possible to split each traffic demands as in the fluid model, in order to exploit a multipath routing. Differently from the formulation presented in section 3.1, we have used flow variables that don't highlight the traffic demands which the flow on each link belongs to; in fact we know only the sources and not the destinations, of the flows routed trough each link. In this way we drastically reduce the number of variables, decreasing therefore the complexity of the multi-commodity routing sub-problem. As disadvantage it is not possible, if desired, to put some limitations on the traffic demand routing: this because we don'n have the control of the single traffic demand.

With the previous definitions, we can we can formalize the two formulations used, SLO (Switching Links Only) and SNL (Switching Nodes and Links).

SLO Formulation

$$max \sum_{i=1}^{N}\sum_{j=1}^{N} PL_{ij}y_{ij} \tag{4.1}$$

Subject to:

$$\sum_{s=1}^{N} x_{ijs} \leqslant c_{ij}(1-y_{ij})e_{ij} \quad \forall\, i,j \in V \tag{4.2}$$

$$\sum_{j=1}^{N} x_{iji} = \sum_{d=1}^{N} d_{id} \quad \forall\, i \in V \tag{4.3}$$

$$\sum_{j=1}^{N} x_{jis} - \sum_{j=1}^{N} x_{ijs} = d_{si} \quad \forall\, i,s \in V : s \neq i \tag{4.4}$$

$$y_{ij} \leq e_{ij} \quad \forall\, i,j \in N \tag{4.5}$$

Eq. (4.1) represents the objective function that maximizes the number of links powered off; nodes couldn't be powered off. Eq. (4.2) is the capacity constraint that forces the total flow routed through each link, to not exceed the link capacity. Eq. (4.3) and (4.4), describe the classical flow conservation constraints, and eq. (4.5) forbids to switch off a link that doesn't exist.

SNL Formulation

The SNL formulation presents only few differences if compared with (4.1-4.5). The objective function is modified in order to consider nodes, and two new typologies of constraints are added:

$$max \sum_{i=1}^{N}\sum_{j=1}^{N} PL_{ij}y_{ij} + \sum_{i=1}^{N} PN_i z_i \tag{4.6}$$

$$y_{ij} \geq z_i \quad \forall\, i, j \in N \tag{4.7}$$

$$y_{ji} \geq z_i \quad \forall\, i, j \in N \tag{4.8}$$

Eq. 4.6 has been modified in order to consider the energy saved by switching off network nodes. Eq. 4.7 and 4.8 force to switch off all the links connected with deactivated nodes. Obviously a nodes can be switched off only if there are no traffic demands having it as source or destination (edge or core node).

4.2 The Preprocessing Algorithm

The solutions space grows exponentially with network dimensions (number of nodes and links). We therefore have thought to a preliminary procedure, with the aim, at first, to speed up the branch and cut algorithm used by the solver (CPLEX), and, at second, to improve the bound values.

The procedure is based on the analysis of the network cuts. We use an algorithm to find the cuts, and then, analysing the total traffic that has to be routed trough each cut, we calculate the new constraints that will be added to the basic formulation (SLO). A cut is detected when it is possible to split a graph in two fully connected partitions (see figure (4.4)). The number of cuts, and consequently of fully connected bi-partitions, in a graph of about 100 nodes is extremely big, and it would be practically impossible to find them all. That is why we are content with finding only a good part of the existing cuts. The procedure begins to build the partitions, starting from each single node and adding new connected nodes at each iteration. When a new node is candidate to be added to the partition, the algorithm checks the complementary part of the graph to be completely connected. The choice of the candidate that has to be considered at each iteration, is made following two different policies (each policy is performed in a dedicated stage):

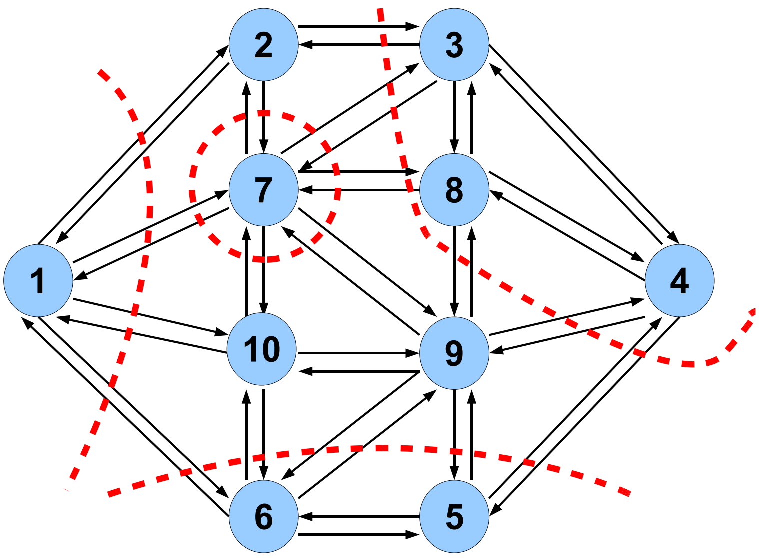


Figure 4.4: Examples of cuts in a graph. The different cuts are represented by the red dashed curves.

- The first build-partition stage consists into three nested cycles; the main outer loop selects, following an increasing order (from 1 to N), the node from which starting to build the current partition (node r). The middle loop (from 1 to N) picks out, always following an increasing order, the nodes added to the partition during the previous iteration (node k). Finally the inner loop (from $k+1$ to N) selects the the new node candidate to be added to the current partition (also here following an increasing order). The candidate node (node c) has to respect some characteristics: it has to be connected by a link to the current k node, its index has to be greater than current node k index and, obviously, it has not to be already a member of the partition built on the current node r.

- The second build-partition stage is very similar to the first, with only a small but significant modification: the inner loop is blocked when a new candidate is added to the partition. In this way the node k selected during the middle cycle, always corresponds to the node c just added.

The just described stages are shown in figure(4.5) and (4.6). When a candidate node c is added, the algorithm checks the full-connectivity of the complementary graph of the current partition. The operation is performed executing a Dijkstra algorithm with a random root node belonging to the complementary graph: the full-connectivity is confirmed if all the nodes of the complementary graph could be added to the SPT considered. If the full-connectivity is not confirmed, then the node c just added is excluded by the current partition (see figure(4.7)) .

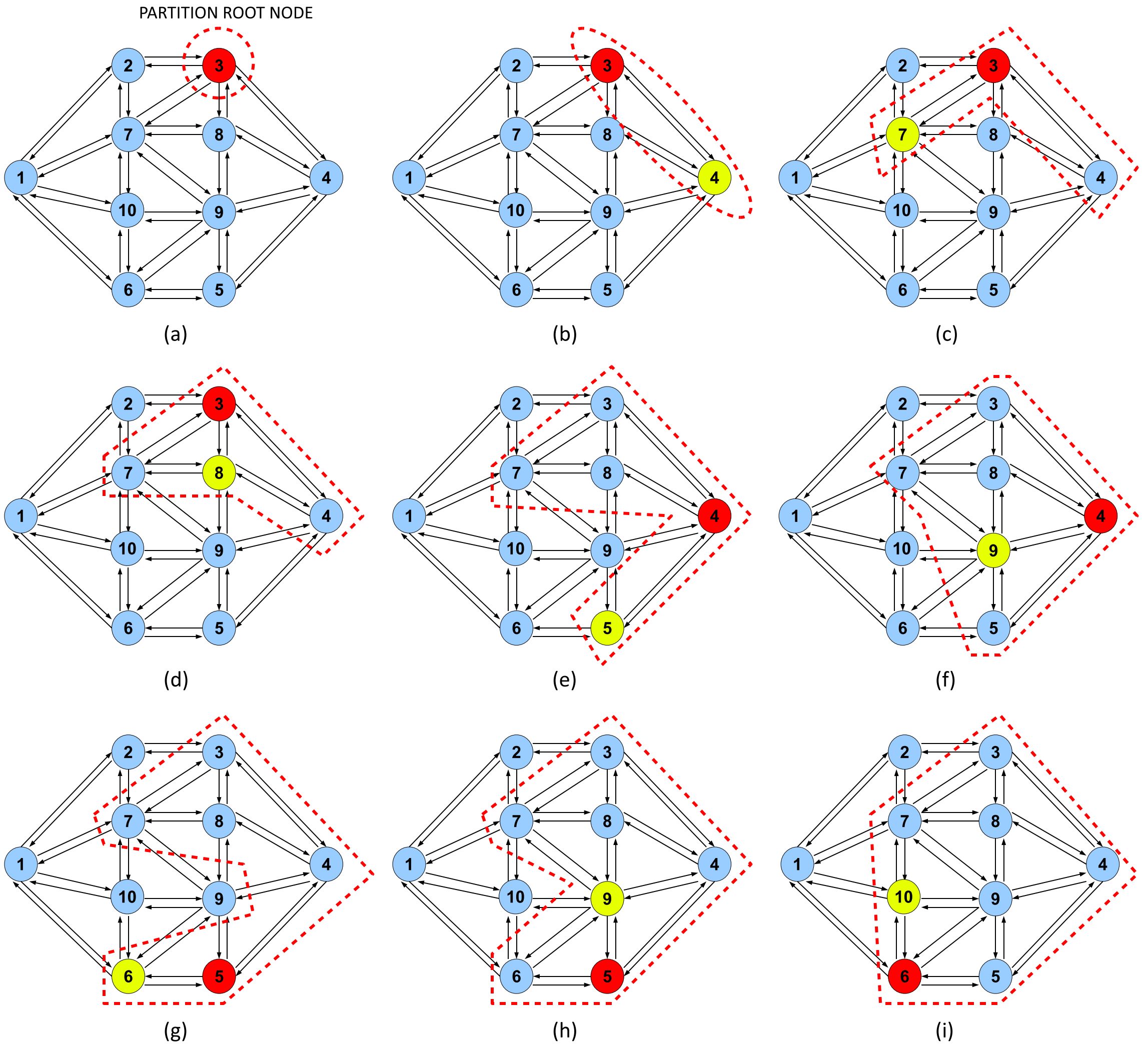


Figure 4.5: *Example of how the first building-partition stage operates. In this case the partition has node 3 as root. Red nodes represent the current node k and yellow nodes the current node c. Notice that the node k is switched, only when all the nodes with a greater index, and connected to it, have been considered as possible candidates. In this example we could not observe the case of a candidate node excluded because of the violation of the full-connectivity constraint.*

When a new cut has been found, we implement a procedure for the detection of the cover inequalities derivable from the cut itself. The main idea is to find the different sets of links that could be powered off, considering the total amount of flow that has to be routed through the cut.

The procedure follows the classic algorithm for the calculation of the cover inequalities: it consists in a nested cycle of knapsack and separation problems solved sequentially. The elements of the knapsack are represented by the links of the cut ($(i, j) \in T$). The cost of each element is unitary, while the weight is indicated by the link capacity c_{ij}; The capacity RC of the knapsack corresponds to the residual

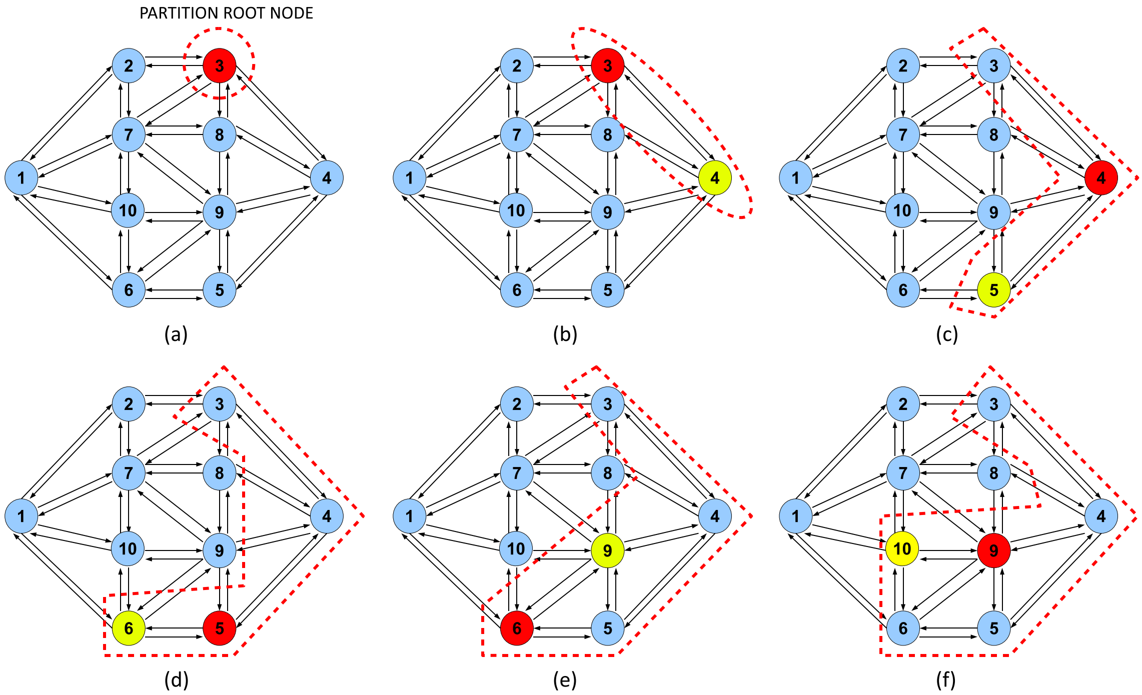


Figure 4.6: *Example of how the second building-partition stage operates. In this case the partition has node 3 as root. Red nodes represent the current node k and yellow nodes the current node c. Notice that the node k is switched when a new candidate node is added to the partition; this node becomes immediately the new node k. In this example we could not observe the case of a candidate node excluded because of the violation of the full-connectivity constraint.*

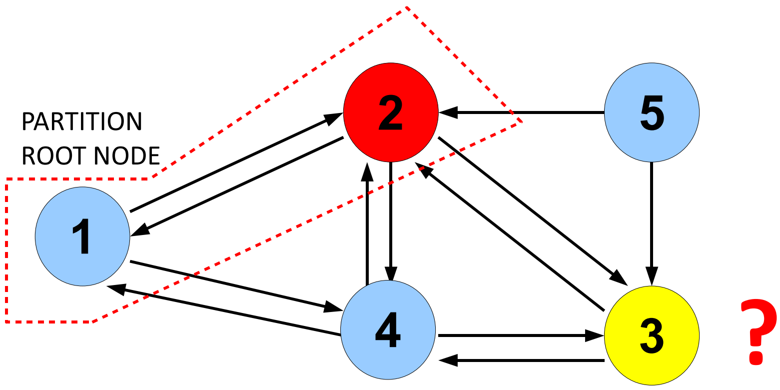


Figure 4.7: *In this example the second building-partition stage is operating; node 1 is the partition root node, node 2 is the current node k and node 3 is the current node c. Node c cant'be added to the partition because of the lack of connectivity in the complementary graph composed by nodes 4 and 5.*

capacity available through the cut T considered. RC is computed as the difference between the total capacity of the links that cross the cut, and the sum off all the traffic demands having source and destination in the two opposite sides of the cut:

$$RC_T = \sum_{(i,j)\in T} c_{ij} - \sum_{s\in P,\, d\in P^c} d_{sd} \tag{4.9}$$

In eq. 4.9, T, P, and P^C represent the current cut, the current partition and the current complementary partition, respectively.

The knapsack and separation problem formulations are the following:

Knapsack:

$$max \sum_{(i,j)\in T} b_{ij} \tag{4.10}$$

Subject to:

$$\sum_{(i,j)\in T} b_{ij} * c_{ij} \leq RC_T \tag{4.11}$$

$$b_{ij} \in [0,1] \quad \forall\,(i,j) \in T \tag{4.12}$$

Separation:

$$min \sum_{(i,j)\in T} (1 - b^*_{ij}) * u_{ij} \tag{4.13}$$

Subject to:

$$\sum_{(i,j)\in T} c_{ij} * u_{ij} \geq RC_T + 1 \tag{4.14}$$

$$u_{ij} \in \{0,1\} \quad \forall\,(i,j) \in T \tag{4.15}$$

(4.10) and (4.13) are respectively the objective function of the knapsack and of the separation problem. (4.11) is the knapsack capacity constraint, that does not allow to switch off links with an equivalent total capacity greater than the cut residual capacity RC_T. Eq. (4.14) simply represents the separation constraint as in its definition [20]. b^*_{ij} represents the solution variables calculated by the knapsack problem: these variables are then used as parameters in the separation model. A new constraint is added when (4.13) $<$ 0; if $u_{ij} = 1$, then the link $(i,j) \in T$ is a component of a minimal cover.

Table 4.2: *Traffic demands of figure 4.8.*

Source / Destination	**1**	**2**	**3**	**4**
1	/	20	20	20
2	20	0	20	20
3	20	20	/	20
4	20	20	20	/

In general a set $C \subseteq N$ is a cover if $\sum_{j\in C} a_j > b$. A cover C is minimal if $C \setminus \{j\}$ is not a cover for any $j \in C$. More details on covers and cover inequalities can be found in [20].

The added constraints have the following form:

$$\sum_{(i,j)\in C} y_{ij} \leq |C| - 1 \tag{4.16}$$

where C is the minimal cover considered and $|C|$ is the cover cardinality. In figure (4.8) the output of the procedure is reported for a trivial case.

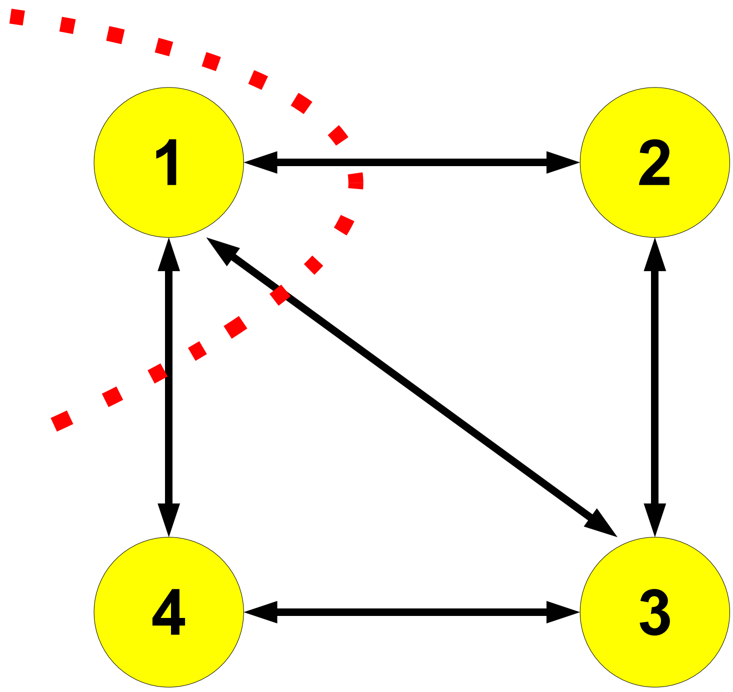


Figure 4.8: *Example of how the preprocessing procedure works when a cut is found. In this case each links is assumed to be bidirectional with a capacity in each direction of 100 units. In according with the traffic matrix of table 4.2 and considering the cut indicated with the red dashed line, it is possible to derive the two following additional constraints:*
$s.t.(1) \quad y_{12} + y_{13} + y_{14} \leq 2$
$s.t.(2) \quad y_{21} + y_{31} + y_{41} \leq 2$

The new added constraints suggest to the solver which combinations of switched off links are potentially valid. In this way the branch and bound tree is considerably reduced; there are therefore more possibilities for the solver to find more rapidly a good branch of the tree. During tests we have been able to add up more than 3000 constraints with network of about 100 nodes.

In addition to the cover inequalities constraints, we have also inserted a new constraint, deduced by some considerations about the minimum amount of traffic that the network has to be able to route. At first we have calculated the minimum hop path between every couple of nodes; then the hop number of each path has been multiplied with the traffic demand having as source and destinations the nodes connected by the path itself. All the values obtained are finally added together: this new parameter is called D and it represents a lower bound of the minimum global capacity that has to be guarantee to satisfy all the demands. The new constraint has the following structure:
Subject to:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(1-y_{ij})c_{ij} \geq D \tag{4.17}$$

Chapter 5

Optimized Shortest Path for Energy Saving: feasible routing stage

The feasible routing stage receives as input the reduced topologies calculated at the switching-off stage, and it has the aim to find a set of OSPF links weights that would be able to define the feasible shortest path routing (without exceeding links maximum utilization) of all the traffic demands through the reduced network.

The OSPF links weights configuration of the second stage, is implemented by the IGP-WO (Interior Gateway Protocol-Weight Optimization) algorithm presented in [2] by Bernard Fortz and Mikkel Thorup. The IGP-WO algorithm has the aim to minimize a cost function Φ based on links utilization. The cost function Φ is calculated as

$$\Phi = \sum_{(i,j):e_{ij}=1} \Phi_{(i,j)}(l_{ij}) \tag{5.1}$$

where l_{ij} is the total flow routed through link (i,j) and

$$\phi'_{(i,j)}(l) = \begin{cases} 1\,for & 0 \leq x/c_{ij} < 1/3 \\ 3\,for & 1/3 \leq x/c_{ij} < 2/3 \\ 10\,for & 2/3 \leq x/c_{ij} < 9/10 \\ 70\,for & 9/10 \leq x/c_{ij} < 1 \\ 500\,for & 1 \leq x/c_{ij} < 11/10 \\ 5000\,for & 11/10 \leq x/c_{ij} < \infty \end{cases} \tag{5.2}$$

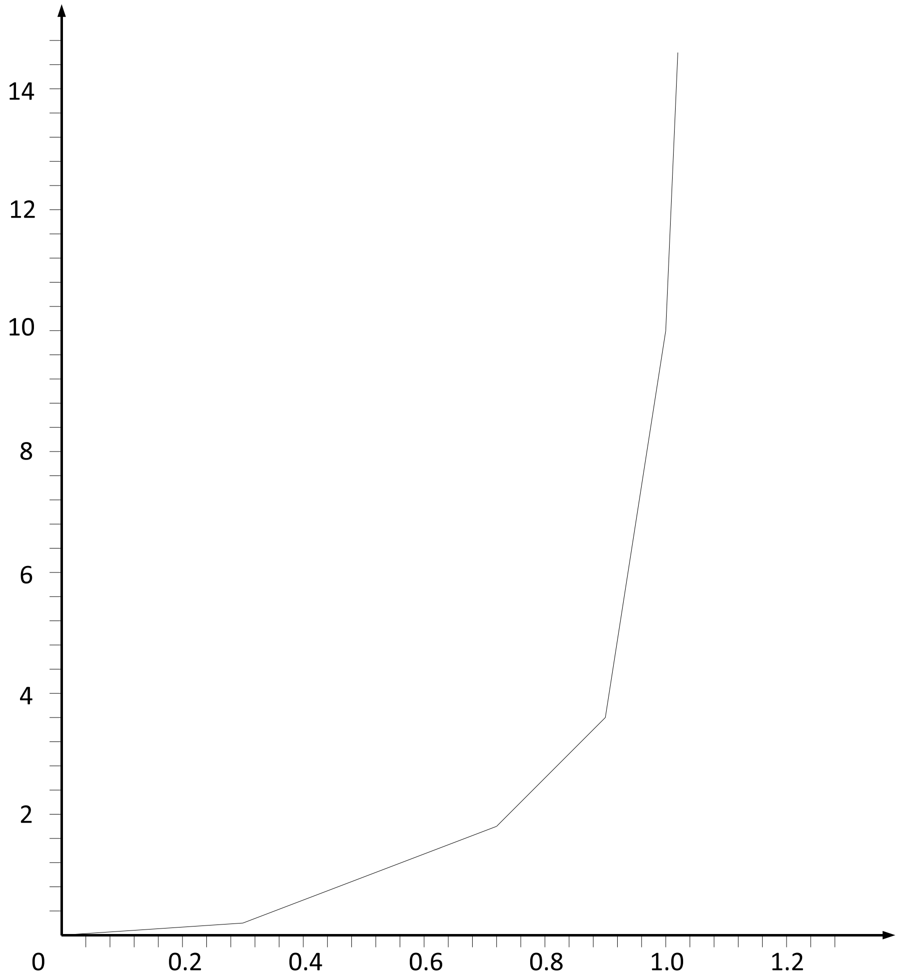


Figure 5.1: *Link cost $\phi_{(i,j)}(l_{ij})$ as a function of load l_{ij} for link capacity $c_{ij} = 1$. Source [2].*

The cost function $\Phi_{(i,j)}$ is illustrated in figure (5.1); The main idea is to encourage to route traffic demands through links with low utilization. The exact definition of $\Phi_{(i,j)}$ is not really vital: it only necessarily has to be piece-wise linear increasing and convex. These characteristics let to solve, at the optimum and in polynomial time, the general routing problem. The optimum of the general routing problem is very useful in order to compare the performances of the IGP-WO algorithm. A criticism that could be done to the structure of the cost function, is that, having the target to avoid saturated links, it would be probably more convenient to minimize a maximum than to minimize a sum. However, because of the very high penalty for the overloaded links, also overloaded solutions are greatly disadvantaged; moreover the general advantage of working with a sum rather than a maximum, is that the possible bottlenecks do not impede the objective function to minimize the utilization levels in the rest of the network.

Unfortunately, given s network topology and a traffic matrix, there is no guarantee concerning the existence of a set of OSPF links weights that could perform the feasible routing of the demands. In our case this situation principally occurs when the feasible routing stage operates on a reduced network topology, considering the traffic levels utilized to calculate the same reduced network at the first stage. The reduced network, has been in fact calculated considering a completely free routing (splittable multicommodity flow), that could not be exactly reproduced by the shortest path routing defined with the OSPF weights. In these cases, the unique way to get a feasible solution with no saturated links, consists into switching on some network elements (routers and links).

We have decided to handle this issue, exploiting the information (reduced traffic matrices and reduced network topologies) already at our disposition, and avoiding in this way to develop an apposite switching-on procedure: we have in fact noted that it could be possible to switch on network elements, simply using, as input of the second stage, the reduced network topology obtained considering higher traffic levels at the first stage. The computational results showed in chapter 7 demonstrate the validity of this approach: a gap of 10% between the traffic levels considered at the two stages, is in fact sufficient to find feasible solutions.

5.1 Optimal routing model

Making reference to the notation presented in the section 3.1 is possible to express the general routing problem as the following linear program:

$$min\,\Phi = \sum_{(i,j)\in E} \Phi_{(i,j)} \tag{5.3}$$

$$\sum_{j=1}^{N} f_{ij}^{sd} - \sum_{j=1}^{N} f_{ji}^{sd} = \begin{cases} t^{sd}, & \forall s,d,i = s \\ -t^{sd}, & \forall s,d,i = d \\ 0, & \forall s,d,i \neq s,d \end{cases} \tag{5.4}$$

$$l_{ij} = \sum_{s=1}^{N}\sum_{d=1}^{N} f_{ji}^{sd} \;\; \forall i,j \tag{5.5}$$

$$\Phi_{(i,j)} \geq l_{ij} \;\; \forall (i,j) \in E \tag{5.6}$$

$$\Phi_{(i,j)} \geq 3l_{ij} - \frac{2}{3}c_{ij} \ \ \forall (i,j) \in E \tag{5.7}$$

$$\Phi_{(i,j)} \geq 10l_{ij} - \frac{16}{3}c_{ij} \ \ \forall (i,j) \in E \tag{5.8}$$

$$\Phi_{(i,j)} \geq 70l_{ij} - \frac{178}{3}c_{ij} \ \ \forall (i,j) \in E \tag{5.9}$$

$$\Phi_{(i,j)} \geq 500l_{ij} - \frac{1468}{3}c_{ij} \ \ \forall (i,j) \in E \tag{5.10}$$

$$\Phi_{(i,j)} \geq 5000l_{ij} - \frac{19468}{3}c_{ij} \ \ \forall (i,j) \in E \tag{5.11}$$

$$f_{ij}^{sd} \geq 0 \ \ \forall (i,j) \in E; \forall s,t \in N \tag{5.12}$$

Eq. 5.4 represents the classical flow conservation constraints, eq. 5.5 defines the total amount of traffic routed through each link, and eq. 5.6-5.11 define the cost on each link.

5.2 The IGP-WO Algorithm

In OSPF routing each link (i,j) has a weight w_{ij}; these weights uniquely determine the shortest paths, the load on the links, and, finally, the cost function Φ. The problem of finding the optimal OSPF weights set is NP-hard, and it is therefore impossible to use exact methods.

The IGP-WO algorithm consists into a local search heuristic, that has the goal to determine a weights vector $(w_{ij})_{(i,j)\in E}$ that minimize the cost function Φ and therefore optimizes the load balancing in the network domain analysed. The main inputs of the algorithm are the network topology (nodes and link), one or more traffic demand matrices, the number of iterations to be executed during the heuristic search, and, finally, the maximum weight value (W_{max}) that can be assigned to the links. The load balancing is achieved by splitting the traffic demands equally among all the shortest paths (multiple path routing): more are the shortest paths between each couple of nodes, a better load balancing is obtained.

The local search heuristic run by IGP-WO, if compared with classic local search heuristics, is characterized by some differences concerning the neighbourhood definition, the way it is explored, and the choice of the next solution. Classic descent methods simply choose an improving neighbour, stopping the execution when a local minimum is found. Instead IGP-WO, as meta-heuristic like tabu search or simulated annealing, allows non-improving moves too and implements policies to avoid cycling.

5.2.1 Neighbourhood Structure

A solution of the weights setting problem is fully characterized by its weights vector w. A neighbour $w^{'} \in N(w)$ of w is obtained by executing the two following operations:

- **Single weight change:** this operation is very simple and consists into changing the weight of a single link.
- **Evenly balancing flows:** this operation consists in modifying the weights of the outgoing links connected to a certain node x, in order to obtain more shortest paths in the direction of a certain destination d; in this way load balancing is obtained, because traffic demands destined to node t and routed through node x will be split among the different shortest paths (see figure(5.2)).

The neighbourhood structure is very complex and the number of neighbours of a given solution is very high. Therefore a complete exploration of the neighbourhood leads to consume too time. This drawback is overcome by evaluating at each iteration, only a randomly selected set of neighbours. The algorithm starts by evaluating 20% of the neighbourhood: each time the current solution is improved, the sampling rate is divided by 3, while in the opposite case it multiplied by 10. The algorithm is forced to sample at least 1% of the neighbourhood.

When a local minimum is reached, there are many neighbours with the same cost; this situation could cause a long series of iterations with the same cost value. In order to avoid this situations, when 300 iterations with no improvements have occurred, the algorithm tries to move to possible more attractive regions, by randomly perturbing the current solution. The perturbation consists into adding a randomly defined value, uniformly chosen between -2 and +2, to 10% of the weights.

5.2.2 Hashing tables

As just written, the IGP-WO local search heuristic allows non-improving moves along the neighbourhood. This approach is however accompanied by the undesired possibility that non-improving moves could lead to cycling. An efficient approach

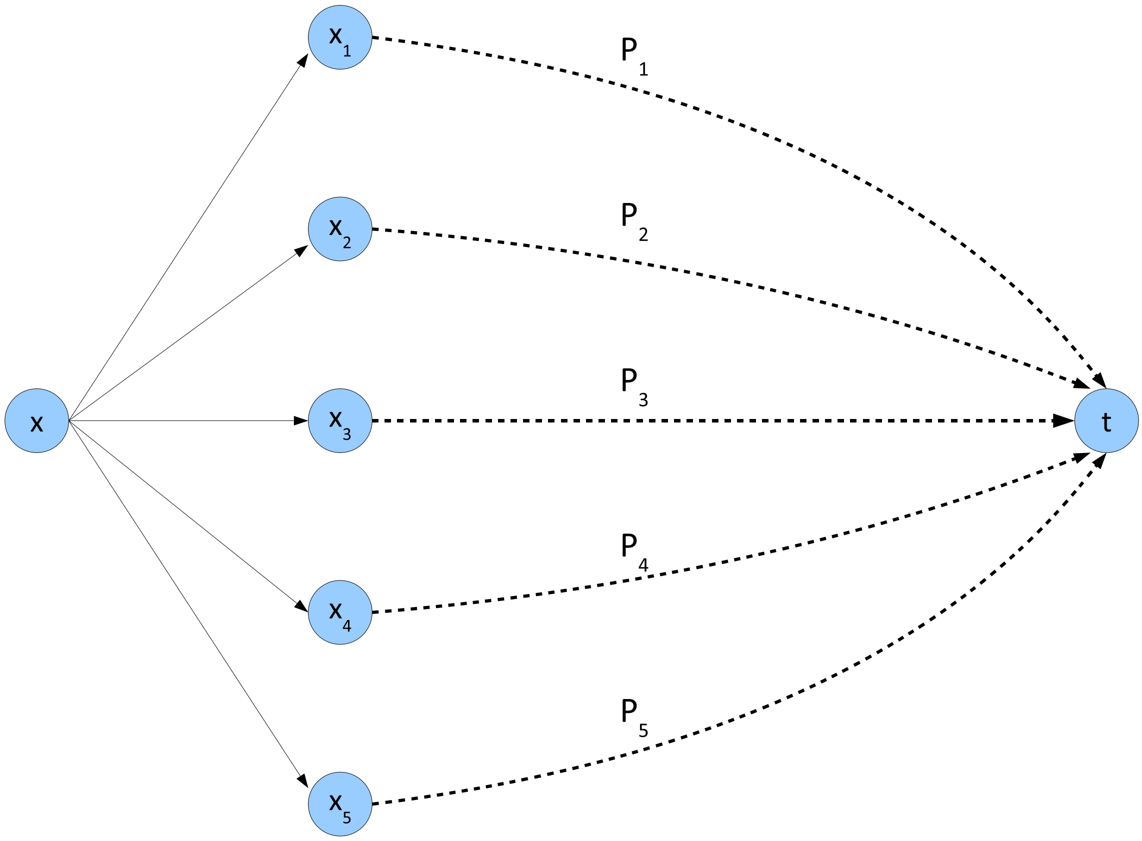


Figure 5.2: *The second type of move tries to make all paths from x to t of equal length. Source [2].*

against cycles could be represented by the utilization of hashing tables. Fortz and Thorup have in fact followed this idea and they have therefore developed a simple strategy based on hashing tables, that assures the absence of cycles. Their approach consists into exploit an especially chosen hashing function h, in order to map each weights vector to integers. Each integer is represented with $l = 16$ bits, therefore an hashing table T with 2^l entries is needed. The hashing table utilized is boolean and at the beginning of the algorithm all the entries of T are set to false. When a new iteration gives as results a new weights vector w, then the entry correspondent to $T(h(w))$ is set to true. The neighbourhood exploring stage reject any solution w' such that $T(h(w'))$ is true. This procedure completely avoids cycling, but causes the rejection of potentially excellent solutions having the same hashing value of a solution met before. It is therefore important to choose h carefully, in order to obtain a negligible probability of collision.

A secondary smaller hashing table is also used in order to avoid the evaluation of neighbours with the same hash value, and in order to escape from the regions where the local minimum has been already detected.

5.3 Computational results

In [2] the IGP-WO algorithm has been tested on a proposed AT&T WordlNet backbone with 90 nodes and 274 bidirectional links. The computational results are shown in table 5.1. The table highlights the comparison between the routings obtained following different configurations of the OSPF weights:

- **HeurOSPF:** configuration found by the IGP-WO algorithm.
- **InvCapOSPF:** the links weights are set as inversely proportional to the link capacity.
- **UnitOSPF:** all links weights are set to 1.
- **L2OSPF:** the links weights are proportional to the physical length of the links.
- **RandomOSPF:** the weighs set is chosen randomly.
- **OPT:** it does not represent a weights configuration, but it represents the optimum solutions found with the linear programming formulation (5.3-5.11).

Each entry of the table shows the normalized cost value and, in parenthesis, the link max utilization. Observe how the results obtained by the IGP-WO algorithm (HeurOSPF) are the unique comparable with the optimum values (OPT).

5.4 The Totem Toolbox

The code of IGP-WO algorithm is not directly available because it belongs to AT&T. Fortunately the researchers of the Totem Project [22] have developed an interesting toolbox for traffic engineering methods that implements the IGP-WO algorithm.

Table 5.1: *Comparison between the cost functions obtained with different configurations of OSPF weights. The link max utilization is reported between parenthesis. Source [2].*

Demand	InvCapOSPF	UnitOSPF	L2OSPF	RandomOSPF	HeurOSPF	OPT
3709	1.01 (0.15)	1.00 (0.15)	1.13 (0.23)	1.12 (0.35)	1.00 (0.17)	1.00 (0.10)
7417	1.01 (0.30)	1.00 (0.31)	1.15 (0.46)	1.91 (1.05)	1.00 (0.30)	1.00 (0.19)
11126	1.05 (0.45)	1.03 (0.46)	1.21 (0.70)	1.36 (0.66)	1.01 (0.34)	1.01 (0.29)
14835	1.15 (0.60)	1.13 (0.62)	1.42 (0.93)	12.76 (1.15)	1.05 (0.47)	1.04 (0.39)
18465	1.33 (0.75)	1.31 (0.77)	5.47 (1.16)	59.48 (1.32)	1.16 (0.59)	1.14 (0.48)
22252	1.62 (0.90)	1.59 (0.92)	44.90 (1.39)	86.54 (1.72)	1.32 (0.67)	1.30 (0.58)
25961	2.70 (1.05)	3.09 (1.08)	82.93 (1.62)	178.26 (1.86)	1.49 (0.78)	1.46 (0.68)
29670	17.61 (1.20)	21.78 (1.23)	113.22 (1.86)	207.86 (4.36)	1.67 (0.89)	1.63 (0.77)
33378	55.27 (1.35)	51.15 (1.39)	149.62 (2.09)	406.29 (1.93)	1.98 (1.00)	1.89 (0.87)
37087	106.93 (1.51)	93.85 (1.54)	222.56 (2.32)	476.57 (2.65)	2.44 (1.00)	2.33 (0.97)
40796	175.44 (1.66)	157.00 (1.69)	294.52 (2.55)	658.68 (3.09)	4.08 (1.10)	3.64 (1.06)
44504	246.22 (1.81)	228.30 (1.85)	370.79 (2.78)	715.52 (3.37)	15.86 (1.34)	13.06 (1.16)

Chapter 6

Generation of Data Input

6.1 Network topologies

The knowledge of realistic Internet topologies, is really important for green researchers. The topology is in fact one of the elements that mostly influences the performances of the various energy saving approaches. Unfortunately real network topologies are not commonly available, because the ISPs consider the information about them strictly confidential. In [21] the reaserchers have presented new measurement techniques able to infer high quality ISP maps. They have been able to map ten diverse ISPs, Abovenet, AT&T, Ebone, Exodus, Level3, Sprint, Telstra, Tiscali (Europe), Verio, and VSNL (India), by using over 750 publicly available traceroute sources as measurement vantage points. In table 6.1 we report the characteristics of the backbone networks of six of the previous ISPs. The files concerning the various maps are available at [19]. These maps include both access and backbone nodes. Three of the ten ISPs just listed, have compared the obtained maps with the real topologies, confirming their extreme accuracy.

The tests have been carried out on three of the six real network backbone topologies reported in table 6.1. In particular we have worked with Telstra, Exodus and Ebone (Exodus and Ebone topologies are illustrated in figure (6.1) and (6.2)) . Unfortunately, although the network topologies are known, no accurate information about network links capacity are available. We therefore have assumed all the network links with the same arbitrary capacity. We don't have any information about the OSPF weights too, but we don't need them as data input.

Finally, running the SNL formulation, we have had the necessity to select the set of edge and core nodes. Edge nodes could be both source and destination for the traffic demands, while core nodes play only the role of transit routers; therefore core nodes are the only that could be powered off. In the used topology, we had no

Network	AS number	Routers	Links	Cities	Inter-city Links
Telstra (Australia)	1221	104	302	53	55
Sprintlink (US)	1239	315	1944	44	83
Ebone (Europe)	1755	87	322	23	38
Tiscali (Europe)	3257	161	656	50	88
Exodus (Europe)	3967	79	294	22	37
Abovenet (US)	6461	141	748	22	42

Table 6.1: *Backbone topologies from Rocketfuel (with AS number and name). The table shows the number of links and routers of each topology. It reports also the information about the number of cities in order to give an idea of the real dimension of the topologies considered.*

Network	Total Routers	Edge Routers	Core Routers	% Edge Routers
Telstra (Australia)	104	65	39	62.5
Ebone (Europe)	87	31	56	35.6
Exodus (Europe)	79	38	41	48.1

Table 6.2: *Randomly division between edge and core routers considered for the tests. The percentage of edge routers is very high in Telstra topology because of the presence of 51 leaf nodes*

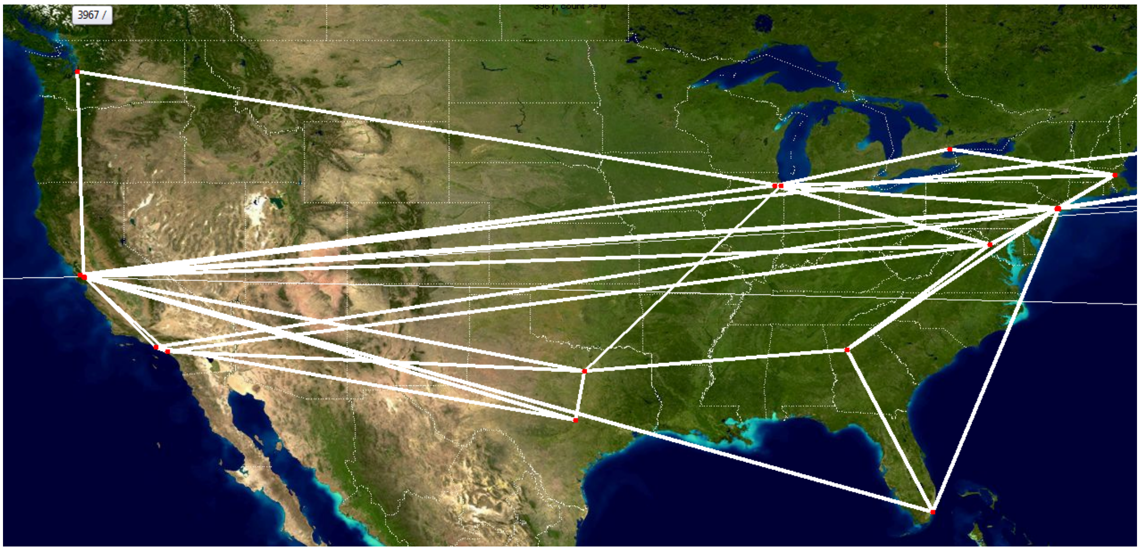


Figure 6.1: *Schematic representation of Exodus topology. Background images from Visible Earth at NASA.*

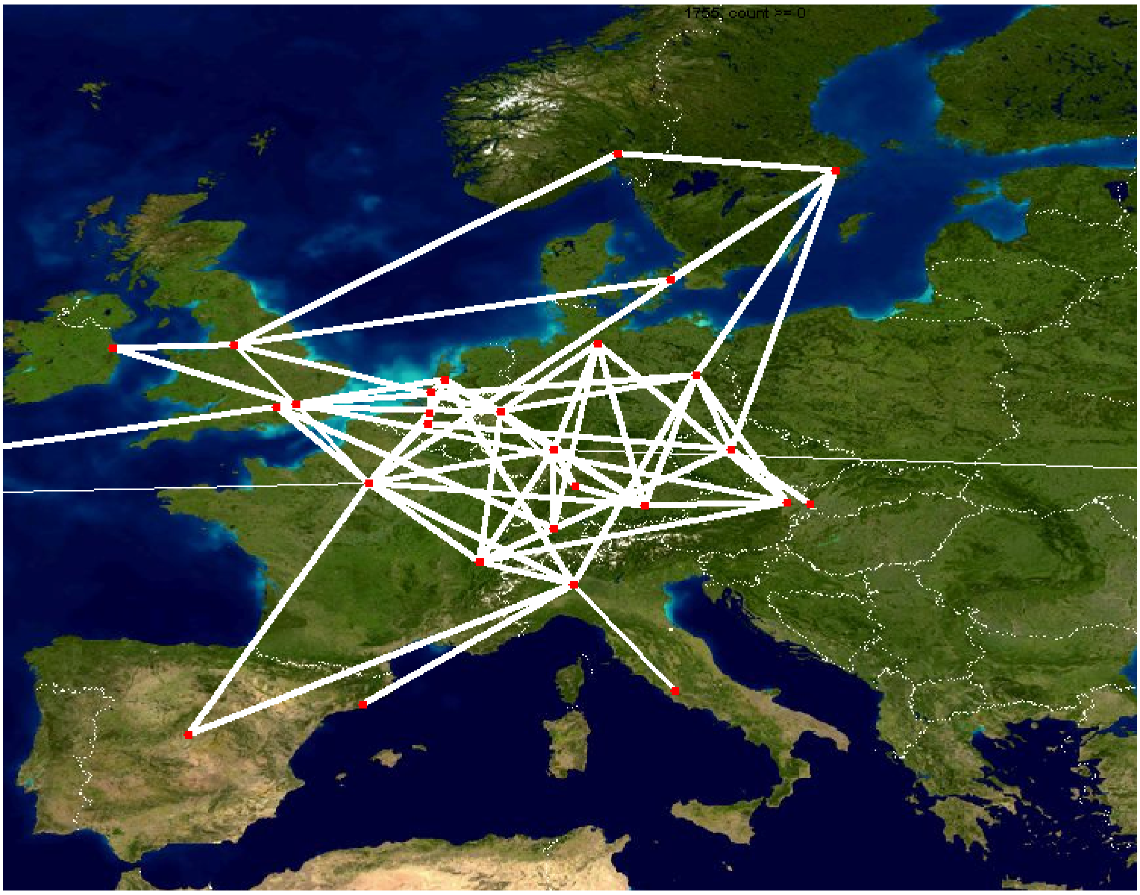


Figure 6.2: *Schematic representation of Ebone topology. Background images from Visible Earth at NASA.*

information about the node status (core/edge): we hence have randomly selected a set of edge routers for each of the three topologies considered (see table 6.2). Obviously we have considered all the leaf nodes as edge nodes. In this way we avoid the trivial cases where all the core leaf nodes could be easily powered off: a similar situation would not be realistic. We have also decided to select at least one edge node for each single city.

6.2 Traffic matrices

Like in the case of real network topologies, also information about real traffic matrices are not easily available. Because of this absence of data about peak workload traffic matrices, we have decided to use a linear programming formulation, in order to obtain, for each of the three topologies, a maximum sustainable traffic matrix; "maximum sustainable traffic matrix", means that it is possible to satisfy all the traffic demands only switching on all the network links. We have used two LP formulations: the first is the SLOM (Switch Links Only Matrix) formulation and the second is the SNLM (Switch Nodes and Links Matrix) formulation. The main difference between them is that SNLM has to distinguish edge nodes from core nodes.

6.2.1 SLOM formulation

Objective Function:

$$max \quad \sum_{i=1}^{N}\sum_{j=1}^{N} M * t_{ij} + a_{ij} \tag{6.1}$$

Subject to:

$$\sum_{s=1}^{N} f_{ijs} \leqslant c_{ij} * e_{ij} \quad \forall i, j \in V \tag{6.2}$$

$$\sum_{j=1}^{N} f_{iji} = \sum_{d=1}^{N} t_{id} + a_{id} \quad \forall i \in V \tag{6.3}$$

$$\sum_{j=1}^{N} f_{jis} - \sum_{j=1}^{N} f_{ijs} = t_{si} + a_{si} \quad \forall i, s \in V : s \neq i \tag{6.4}$$

$$t_{ii} + a_{ii} = 0 \quad \forall i \in V \tag{6.5}$$

$$t_{ij} \leq \sum_{h=1}^{N} \frac{e_{ih} * c_{ih}}{N - \alpha} \quad \forall i, j \in V \tag{6.6}$$

$$t_{ij} \geq \beta \quad \forall i, j \in V : j \neq i \tag{6.7}$$

$$t_{ij}, a_{ij} \geq 0 \ \forall i, j \in V, \quad f_{ijs} \geq 0 \ \forall i, j, s \in V \tag{6.8}$$

Eq. (6.8) expresses the variables; let t_{ij}, a_{ij} be positive real variables that represent the traffic demands between node i and node j ($t_{ij} + a_{ij}$ indicates the total traffic demands between i and j), and let f_{ijs} be the flow variables that show the amount of traffic routed through the link (i, j) and originated by node s. We use two variables (the main t_{ij} and the auxiliary a_{ij}) to represent the traffic demands because of the necessity to avoid the trivial solutions where there are only traffic demands equal to the links capacities between the couples of nodes connected by a link (see figure (6.3)). These situations are avoided by the introduction of constraints (6.6) and (6.7), that put some limitations on the feasible values of t_{ij}; in particular eq. (6.6) and eq. (6.7) define respectively the maximum and minimum tolerated value for each traffic demand. Parameters α and β can be varied in order to obtain more

S/D	1	2	3	4	5
1	/	0	0	0	12
2	0	/	1	0	0
3	0	0	/	0	0
4	0	0	5	/	0
5	0	3	21	10	/

Table 6.3: *Traffic demands of figure 6.2.*

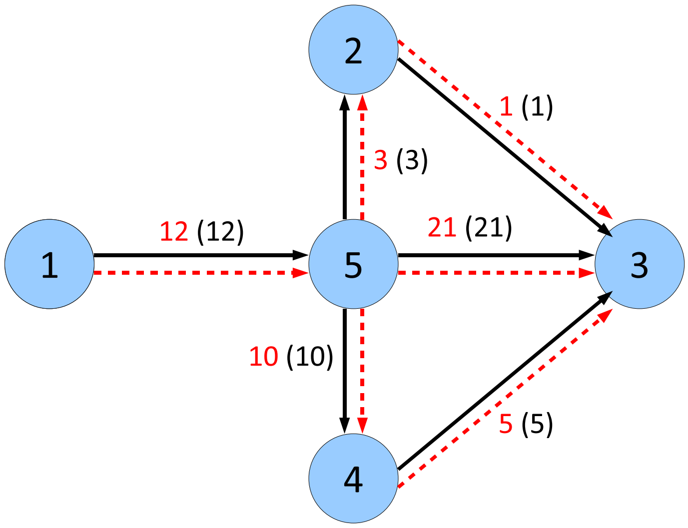


Figure 6.3: *Observe how the traffic matrix reported in table 6.3 could maximally load the illustrated topology. For each link (i, j), a traffic demands t_{ij} equal to the link capacity c_{ij} exists. Red values represent the traffic flows on the links, and black in parenthesis values indicates the links capacity.*

or less balanced matrices. In the objective function (6.1), t_{ij} is multiplied for M (parameter with a very large value) and has therefore a bigger weight than a_{ij}; In this way, the formulation tries at first to maximize, following constraints (6.6) and (6.7), the values of t_{ij}. The unconstrained variables a_{ij} are considered only in a second moment, in order to complete the traffic matrix obtained with the values of t_{ij}. The necessity of using an auxiliary variable a_{ij}, comes up because constraints (6.6) and (6.7), make almost impossible, for the majority of the topologies, to get a full traffic matrix with the only feasible values of t_{ij}. Eq. (6.2) is the link capacity constraint, while (6.3) and (6.4) are the usual flow conservation constraints. Finally eq. (6.5) allows only traffic demands between distinct nodes. Obviously it is necessary to add together t_{ij} and a_{ij} to get the final traffic matrix.

The matrices obtained with the approach just described, could be routed without exceeding the links capacity, only if a completely free routing is adopted (splittable multicommodity flow). The OSPF protocol would not be able to perform the same routing. The full traffic matrices represent therefore an upper bound of the maximum traffic load that could be really managed by the network.

6.2.2 SNLM formulation

The SNLM formulation has to guarantee the absence of traffic demands between core nodes. This goal is reached adding a single new equation to the SNOM formulation just presented:

$$t_{ij} + a_{ij} \leq M * G_i G_j \quad \forall i, j \in V \tag{6.9}$$

Eq. 6.9 reachs its goal by exploiting the big-M technique. The binary parameter G_i is equal to 1 when node i is an edge node.

The maximum traffic matrices obtained with SNLM formulation are not able to completely load the network considered. This fact does not depend by the formulation, but by two other elements: the structure of the topology, and the impossibility of generating traffic demands in each node.

6.2.3 Low workload traffic matrices

The full matrices obtained by SLOM and SNLM are exploited to derive the other matrices representative of the various traffic scenarios. We assume all these matrices to be proportional to the full matrix; we simply multiply the original matrix coefficients with a parameter comprised between 0 and 1.

Chapter 7

Computational Results

We have tested the OSPES procedure on three real backbone network topologies, in order to evaluate the algorithm performances under realistic conditions. The procedure is composed of two different stages, that we prefer to evaluate separately.

7.1 Switching-off Stage Computational Results

The problem of maximizing the total amount of energy saved, guaranteeing the feasible routing through the network of all the traffic demands (4.1-4.8), falls in the class of the NP-hard problems. We have however chosen to handle the problem using two MIP formulations (SLO and SNL), because of our belief concerning the possibility of reaching satisfactory solutions by stopping the solver (CPLEX 10.0) after some hours of elaboration.

7.1.1 SLO formulation

Simple SLO formulation

Tables 7.1, 7.2, 7.3, report the results obtained running the SLO formulation (4.1)-4.5) without any form of data pre-processing (constraints (4.16) and (4.17) are not included in the formulation). The gap from the bound calculated by the solver, except in one case (Exodus 60%), is always lower than 30%. η_{l-tot} represents, the total percentage of switched off links

$$\frac{\sum_{i=1}^{N}\sum_{j=1}^{N} y_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{N} e_{ij}}$$

and η_l the percentage of switched off links, in respect to the maximum number of links $L_{INF-CAP}$ that is possible to switch off considering to have links with infinite capacity

Exodus 294 links 79 nodes - $L_{INF-CAP} = 193$				
Load	**SLO**	Gap_{SLO}	η_{l-tot}	η_l
10%	170 (204)	20,00%	57,82%	88,08%
20%	/	/	/	/
30%	147 (178)	21,09%	50,00%	76,17%
40%	/	/	/	/
50%	/	/	/	/
60%	85 (112)	31,76%	28,91%	44,04%
70%	/	/	/	/
80%	/	/	/	/
90%	29 (36)	24,14%	9,86%	15,03%

Table 7.1: *Exodus: SLO formulation computational results.*

Ebone 322 links 87 nodes - $L_{INF-CAP} = 211$				
Load	**SLO**	Gap_{SLO}	η_{l-tot}	η_l
10%	182 (217)	19,23%	56,52%	86,25%
20%	/	/	/	/
30%	133 (172)	29,32%	41,30%	63,03%
40%	/	/	/	/
50%	/	/	/	/
60%	67 (85)	26,87%	20,81%	31,75%
70%	/	/	/	/
80%	/	/	/	/
90%	4 (4)	0,00%	1,25%	1,90%

Table 7.2: *Ebone: SLO formulation computational results.*

Telstra 302 links 104 nodes - $L_{INF-CAP} = 125$				
Load	**SLO**	Gap_{SLO}	η_{l-tot}	η_l
10%	119 (124)	5,04%	39,40%	95,20%
20%	/	/	/	/
30%	84 (101)	20,23%	27,81%	67,20%
40%	/	/	/	/
50%	/	/	/	/
60%	32 (32)	0,00%	10,60%	25,60%
70%	/	/	/	/
80%	/	/	/	/
90%	2 (2)	0,00%	0,66%	1,60%

Table 7.3: *Telstra: SLO formulation computational results.*

Exodus - $L_{INF-CAP} = 193$				
Load	SLO_{PRE}	$Gap_{SLO_{PRE}}$	η_{l-tot}	η_l
10%	178 (201)	12,92%	60,54%	92,23%
20%	153 (191)	24,84%	52,04%	79,27%
30%	139 (178)	28,05%	47,28%	72,92%
40%	119 (153)	28,57%	40,48%	61,66%
50%	107 (138)	28,97%	36,39%	55,44%
60%	89 (112)	25,84%	30,27%	46,11%
70%	70 (85)	21,43%	23,81%	36,27%
80%	57 (70)	22,81%	19,39%	29,53%
90%	30 (35)	16,67%	10,20%	15,54%

Table 7.4: *Exodus: SLO_{PRE} formulation computational results.*

Ebone - $L_{INF-CAP} = 211$				
Load	SLO_{PRE}	$Gap_{SLO_{PRE}}$	η_{l-tot}	η_l
10%	170 (216)	27,06%	52,80%	80,57%
20%	155 (193)	24,52%	48,14%	73,46%
30%	130 (171)	31,54%	40,37%	61,61%
40%	108 (137)	26,85%	33,54%	51,18%
50%	88 (114)	29,54%	27,33%	41,71%
60%	67 (85)	26,87%	20,81%	31,75%
70%	40 (52)	30,00%	12,42%	18,96%
80%	28 (35)	25,00%	8,7%	13,27%
90%	4 (4)	0,00%	1,25%	1,90%

Table 7.5: *Ebone: SLO_{PRE} formulation computational results.*

Telstra - $L_{INF-CAP} = 125$				
Load	SLO_{PRE}	$Gap_{SLO_{PRE}}$	η_{l-tot}	η_l
10%	119 (124)	5,04%	39,40%	95,20%
20%	99 (117)	18,18%	32,78%	79,20%
30%	83 (99)	19,28%	27,48%	66,40%
40%	64 (81)	26,56%	21,19%	51,20%
50%	41 (43)	4,88%	13,58%	32,80%
60%	32 (32)	0,00%	10,59%	25,60%
70%	23 (23)	0,00%	7,62%	18,40%
80%	12 (12)	0,00%	3,97%	9,60%
90%	2 (2)	0,00%	0,66%	1,60%

Table 7.6: *Telstra: SLO_{PRE} formulation computational results.*

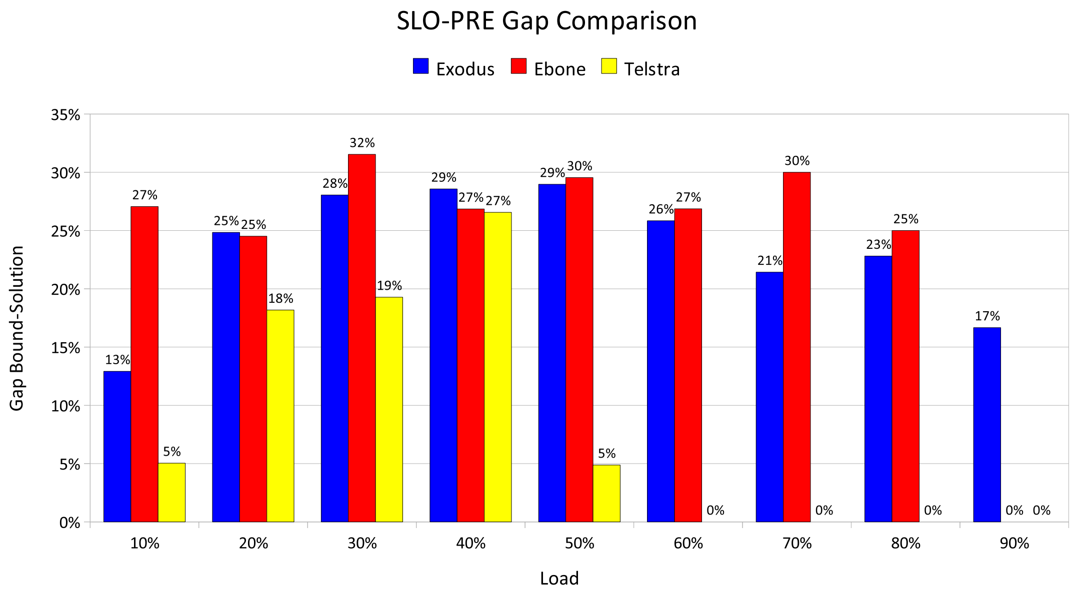


Figure 7.1: *SLO_{PRE}: comparison between gap levels of the three networks considered.*

$$\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}}{L_{INF-CAP}}$$

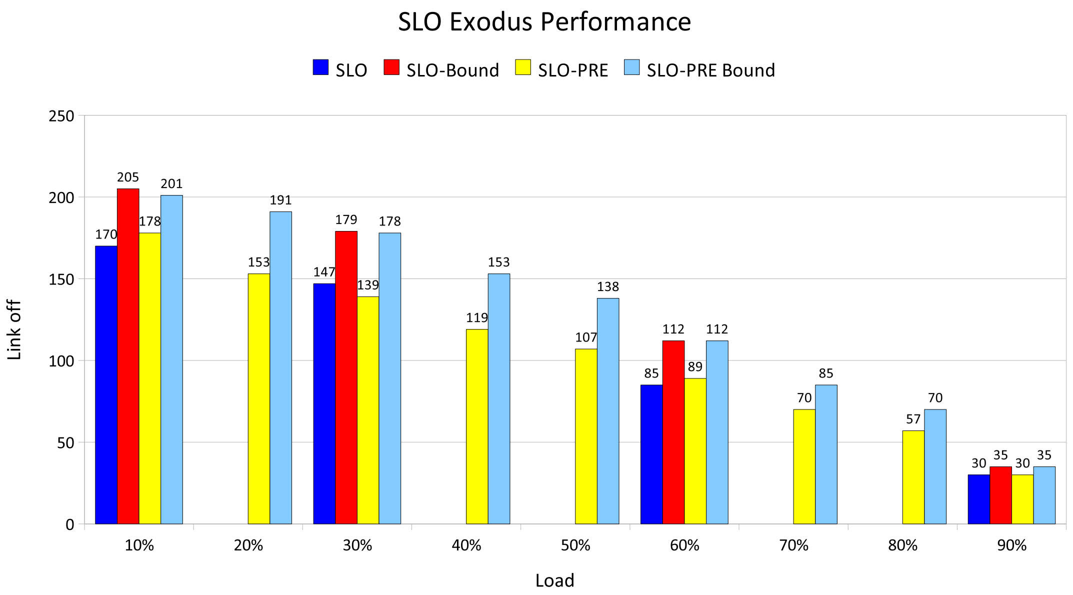
$L_{INF-CAP}$ is calculated by running the SLO formulation on the modified versions (extremely large links capacity) of the three network topologies. The values of η_{l-tot} and η_l are really satisfactory; their observation suggests that the bounds found by SLO could be considerably reduced, improving in this way the quality of the solutions found. The simple SLO formulation has been tested only with the 90-60-30-10% versions of the traffic matrix.

SLO$_{PRE}$ formulation

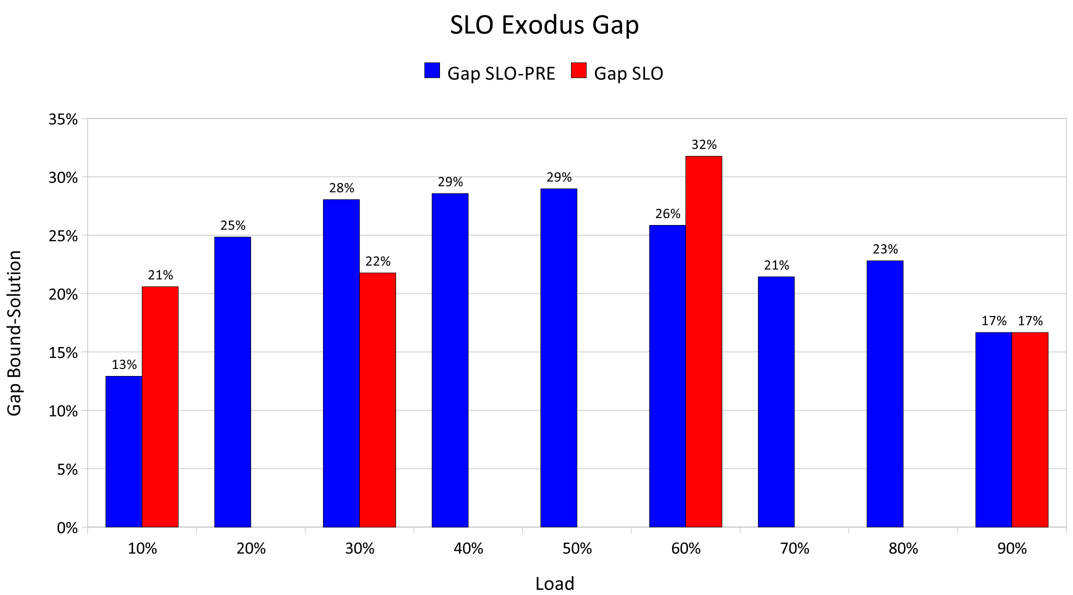
Tables 7.4, 7.5, 7.6 show instead the results achieved with the pre-processed version (SLO_{PRE}) of the SLO formulation. All the reduced versions of the traffic matrix have been used for the tests. The values of gap, η_{l-tot} and η_l reported in the tables, confirm the goodness of the solutions obtained. Figure (7.1) shows the trend followed by the gap of the SLO_{PRE} formulation, when the traffic level on the network is increased. The solutions obtained for Exodus and Telstra networks highlight a typical gaussian shape; this behaviour could be explained by the fact that the structure of the solution space is simpler and easier to be explored when the network is full or, in reverse, empty. Ebone network does not point out the same trend.

SLO-SLO$_{PRE}$ comparison

It is interesting to compare the performances of the simple SLO formulation with its pre-processed version SLO$_{PRE}$.

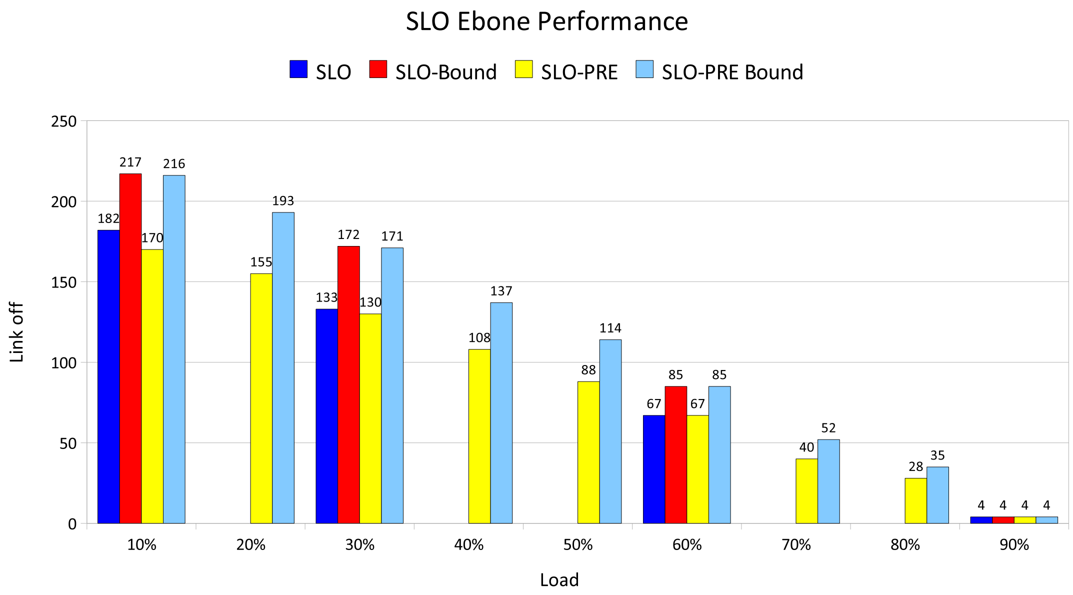


(a)

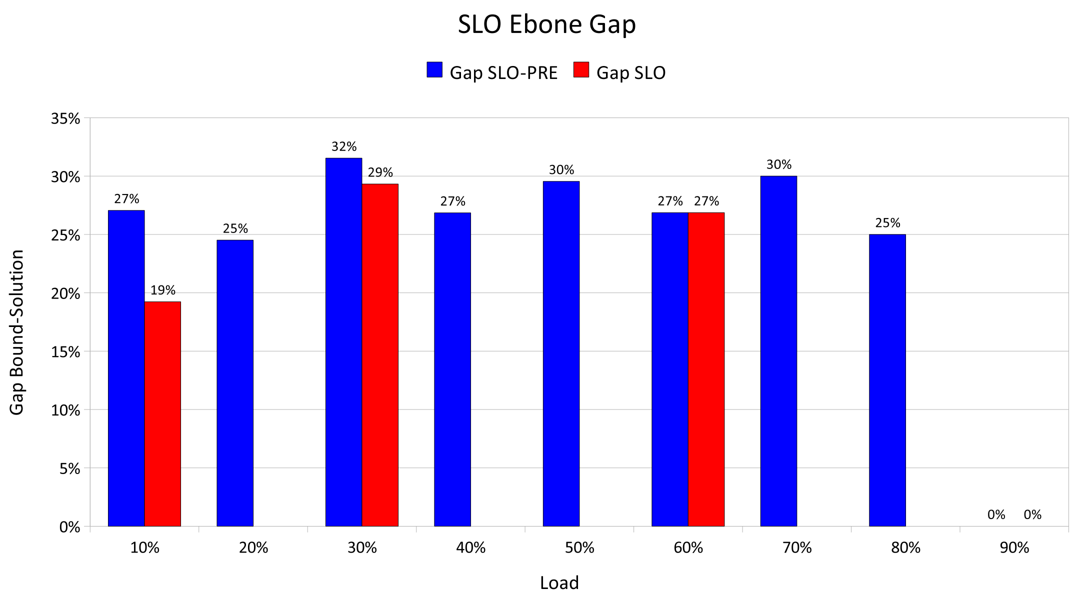


(b)

Figure 7.2: *Exodus: SLO and SLO$_{PRE}$ formulations computational results. a) Number of links switched off compared with the bounds calculated. b) Gap between solutions and bounds calculated.*

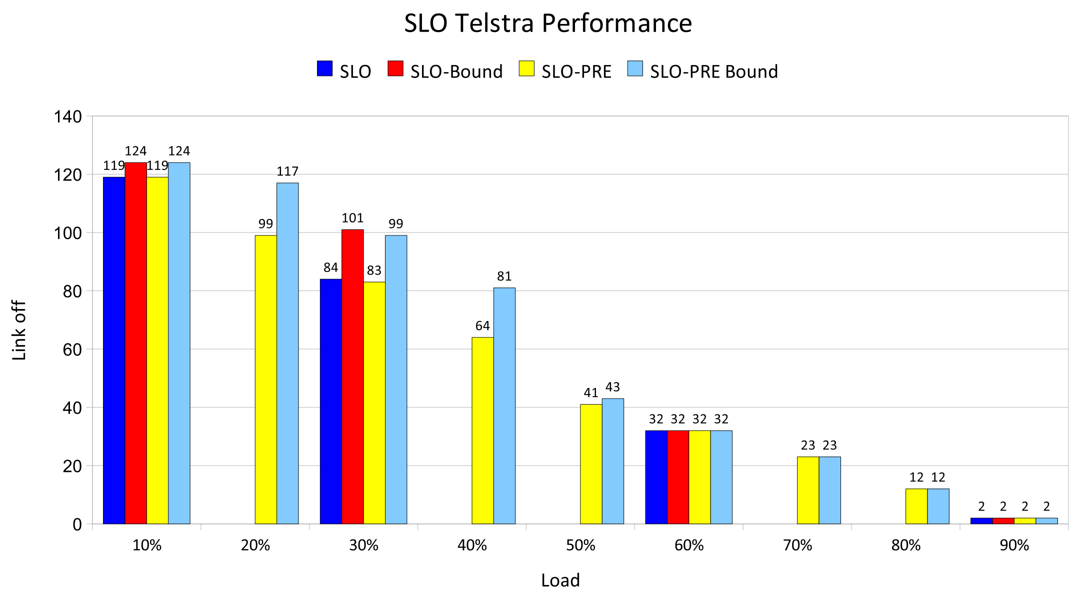


(a)

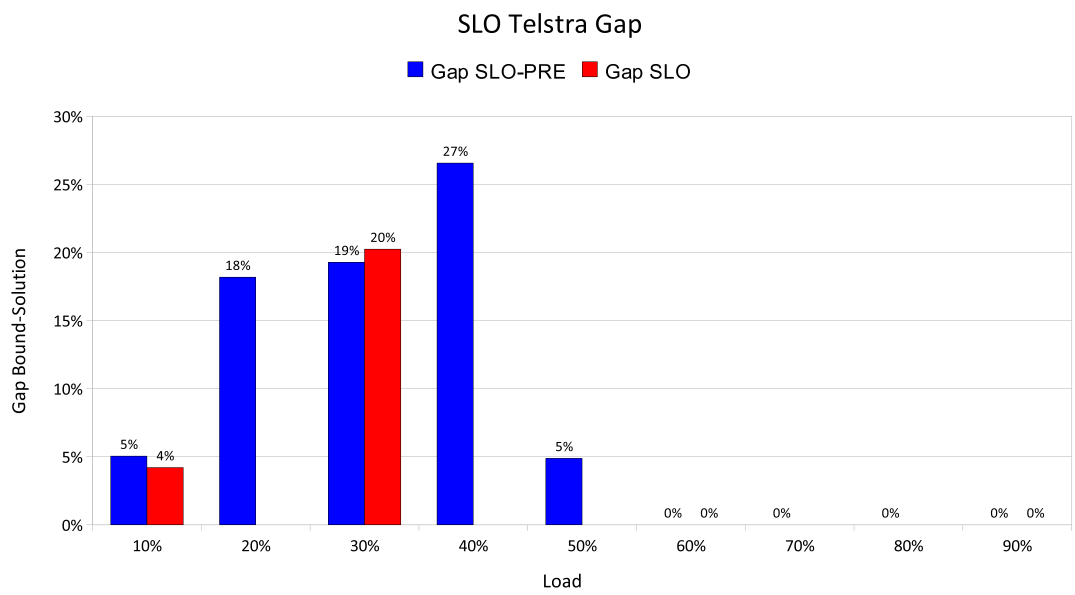


(b)

Figure 7.3: *Ebone: SLO and SLO_{PRE} formulations computational results. a) Number of links switched off compared with the bounds calculated. b) Gap between solutions and bounds calculated.*



(a)



(b)

Figure 7.4: *Telstra: SLO and SLO_{PRE} formulations computational results. a) Number of links switched off compared with the bounds calculated. b) Gap between solutions and bounds calculated.*

Figure (7.2), (7.3) and (7.4) show the comparison between SLO and SLO_{PRE} performances, considering in succession the three network topologies used. Figure (7.2) highlights a general better performance of SLO_{PRE}; SLO formulation obtains the best solution (both objective function and bounds) only one time (Exodus 30%) against the tree of SLO_{PRE}. Figure (7.3) and (7.4) denote instead a general equivalence between the two formulations. The reasons of this behaviour could be principally two:

- The branch and cut technique implemented by the solver is an heuristic procedure, where the capacity of finding the most convenient branch of the tree depends from many factors, comprise the casualness; the pre-processing algorithm has the target to reduce the dimensions of the tree (solution space), increasing in this way the probabilities of finding the good branch. However, an increase of the probabilities does not mean the certainty to find a better solution.

- The topology structure could influence positively or negatively the performances of the pre-processing procedure. For example a topology like Telstra, with a lot of leafs and a low average node degree, causes the generation of a lot of less significant constraints.

In the end we can affirm that the simple SLO formulation, is already able to find satisfactory solutions. In some case its performances could be improved by the utilization of the pre-processing algorithm.

7.1.2 SNL formulation

Table 7.7 and figure (7.5) report the computational results obtained by the SNL formulation with the Exodus topology. The total saving values are calculated considering, as already written in section 4.1, a unitary power consumption for each links, and a node power consumption $PN_i = \lceil \frac{3g_i}{2} \rceil$. The parameter η_{n-tot} represents the efficiency of the procedure, and it is calculated as the rate between the number of core nodes switched off and the total number of core nodes (maximum number of nodes that could be switched off). The bounds are not reported because they never exceed 1%. The results obtained are therefore really excellent, and, in reverse of the SNO formulation, it has not been necessary to execute any operation of data pre-processing.

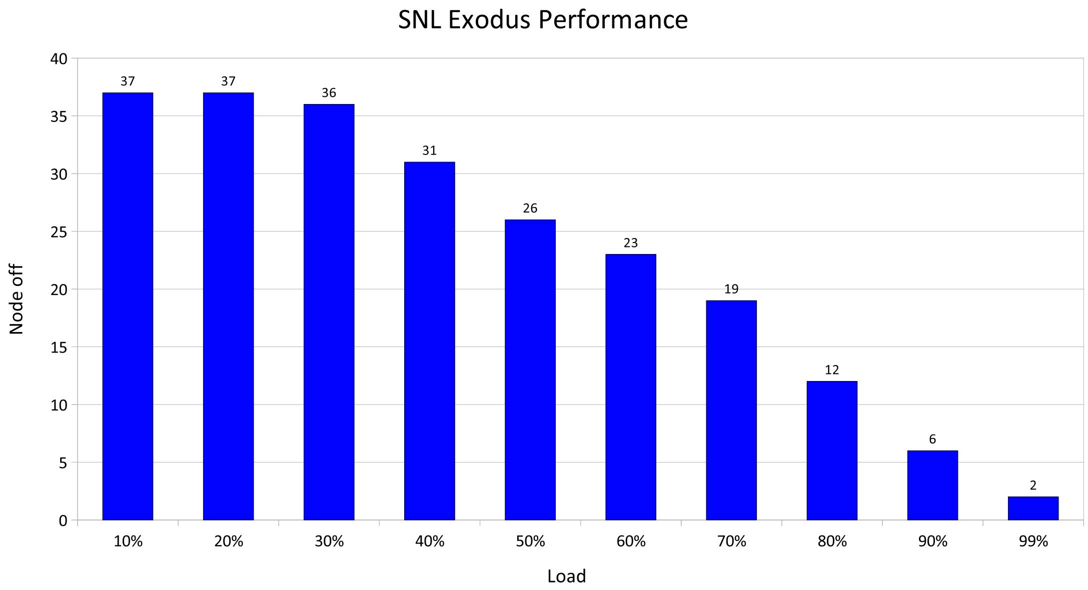
Exodus 294 links 79 nodes - 38 edge 41 core				
Load	**Links off**	**Nodes off**	**Total Saving**	η_{n-tot}
10%	217	37	571	90,24%
20%	204	37	543	90,24%
30%	191	36	512	87,80%
40%	169	31	436	75,61%
50%	148	26	370	63,41%
60%	126	23	309	56,10%
70%	103	19	250	46,34%
80%	77	12	167	29,27%
90%	50	6	86	14,63%
99%	28	2	40	4,88%

Table 7.7: *Exodus: SNL formulation computational results.*

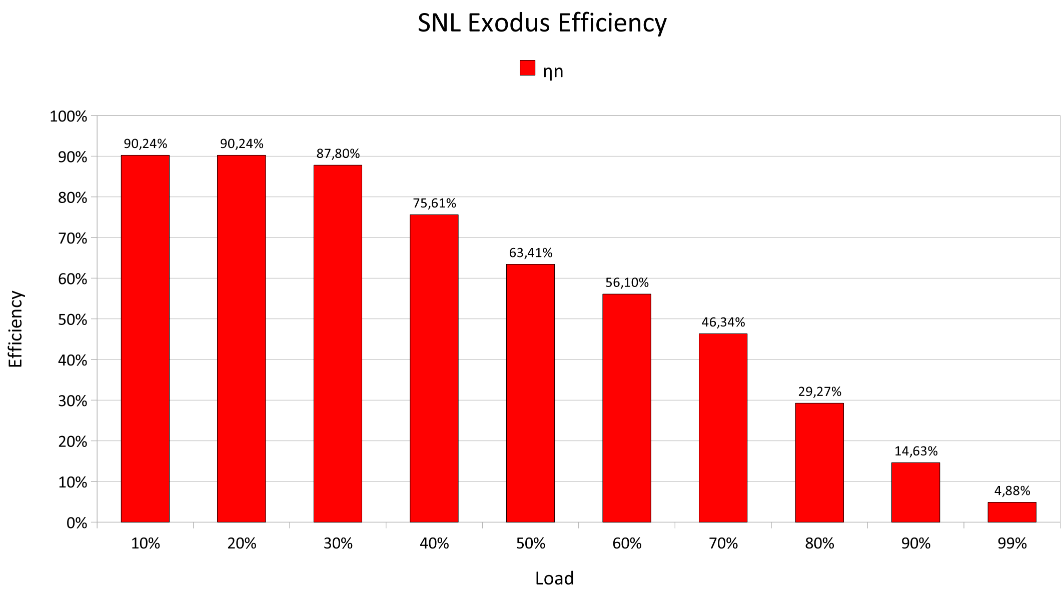
The reasons of these surprising performances are the following:

- **Smaller number of commodities:** the traffic matrices generated for the SNL formulation, contain traffic demands only for the couples of edge nodes (core nodes cannot be source or destination of any traffic demands). In the case of the Exodus topology, the traffic demands managed by the SNL formulation are $38 \times 38 = 1444$, while those managed by the SLO formulation are $79 \times 79 = 6241$: the routing sub-problem is therefore considerably less complex in the case of the SNL formulation.

- **Greater weight of nodes:** when the SLO formulation is processed, the problem consists into choosing the set of links that the lets to maximize the power saving. We have assumed that all the links have the same power consumption and therefore the same weight inside the objective function: for this reason it is more difficult for the solver to decide which links should be selected. For example, considering the Exodus topology the solver has to choose between 294 links with the same weight. Instead, in the case of the SNL formulation, there are some elements, the core nodes, that have a greater weight if compared with links weights; in fact the power consumption of a node is considered as 50% greater than the total power consumption attributable to its connected links. Additionally, switching off a nodes, it is then necessary to switch off also all the connected links. The problem presented by the SNL formulation, consists so principally into selecting the core nodes to switch off: if the Exodus topology is considered, the solver has to choose inside a set of only 41 core nodes. The possible combinations of 41 elements (nodes of SNL formulation) are obviously fewer of those possible with 294 elements (links of SLO formulation).



(a)



(b)

Figure 7.5: *Exodus: SNL formulations computational results. a) Number of nodes switched off. b) Efficiency* η_{n-tot} *of the solutions obtained.*

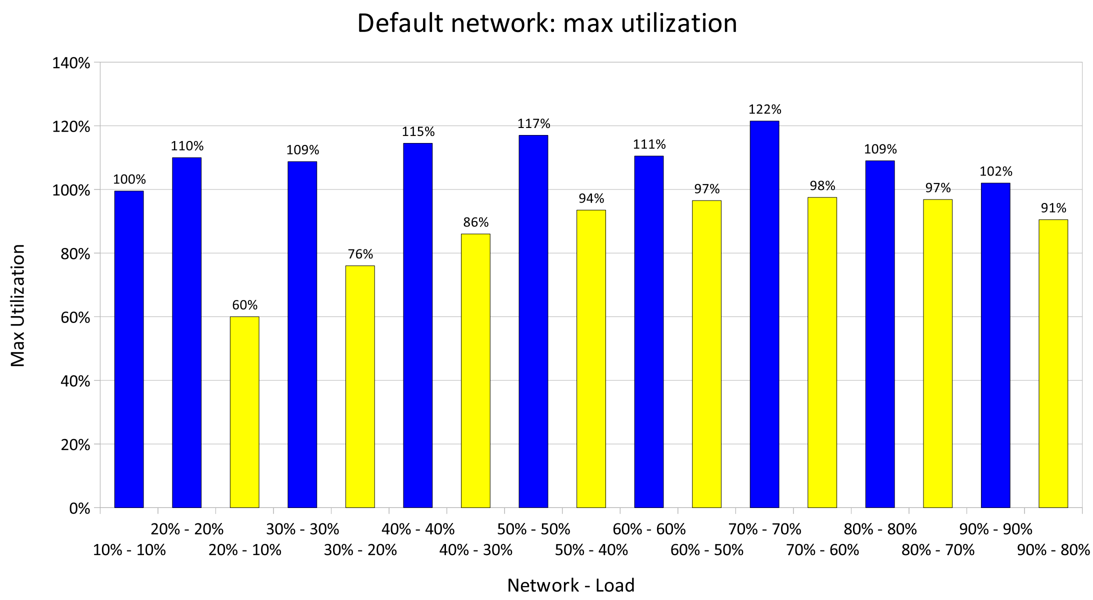
7.2 Feasible Routing Stage Computational Results

The reduced networks obtained in the switching-off stage, are then elaborated by the feasible routing stage. The aim is to find a set of OSPF weights that lets to route all the traffic demands without exceeding the links capacity. The main problem consists into the fact that, given a reduced topology as input, there is not the guarantee to find a feasible set of weights: this happens because, for reasons of complexity, the reduced network obtained by the MIP formulation, is calculated considering, in reverse of OSPF routing, a completely free routing of the traffic demands. Table 7.8 and figure (7.6) report the results obtained with the processing of a small default topology (see figure (7.7)): the observation of these data is very significant in order to better understand how we solve the problem.

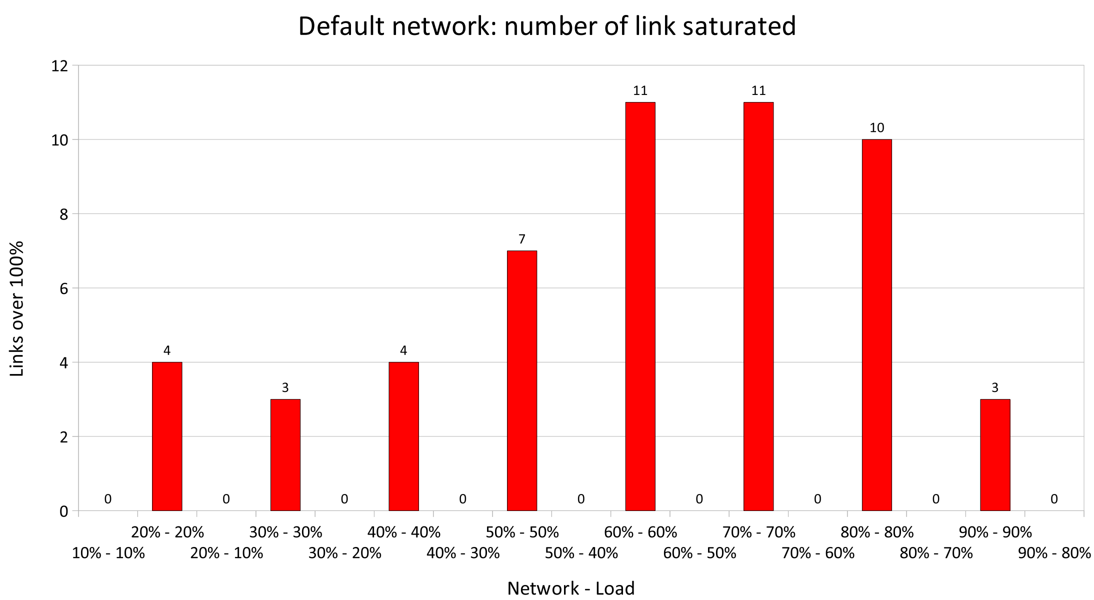
The column *Network* indicates the reduced network version considered; for example, a value of 10% represents the network calculated by the SLO formulation with the traffic matrix multiplied for 0.1. The column *Load* shows the reduced traffic matrix considered as input of the second stage. The columns U_{MAX} and $Links_{>100\%}$ report, respectively, the max links utilization and the number of saturated links, recorded performing the OSPF routing with the set of weights found. The data reported in the table show that it is really difficult (almost impossible), to find a

Default network - 10 nodes and 42 links			
Network	**Load**	U_{MAX}	$Links_{>100\%}$
20%	10%	60,00%	0
30%	20%	76,00%	0
40%	30%	86,00%	0
50%	40%	93,50%	0
60%	50%	96,50%	0
70%	60%	97,50%	0
80%	70%	96,83%	0
90%	80%	90,50%	0
10%	10%	99,50%	0
20%	20%	110,00%	4
30%	30%	108,75%	3
40%	40%	114,50%	4
50%	50%	117,00%	7
60%	60%	110,50%	11
70%	70%	121,50%	11
80%	80%	109,00%	10
90%	90%	102,00%	3

Table 7.8: *Default network: feasible routing stage SLO computational results.*



(a)



(b)

Figure 7.6: *Default network: SLO feasible routing stage performance: a)Max utilization. b)Number of saturated links.*

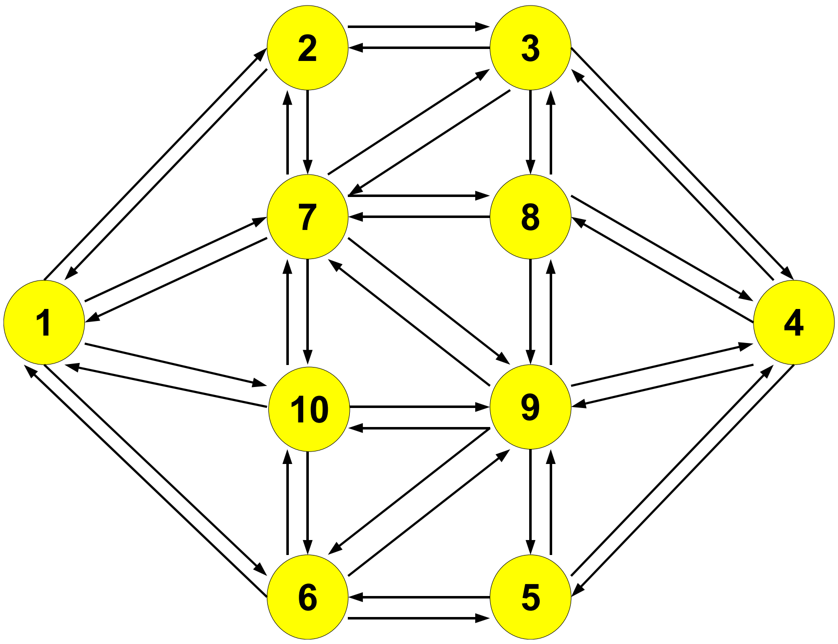
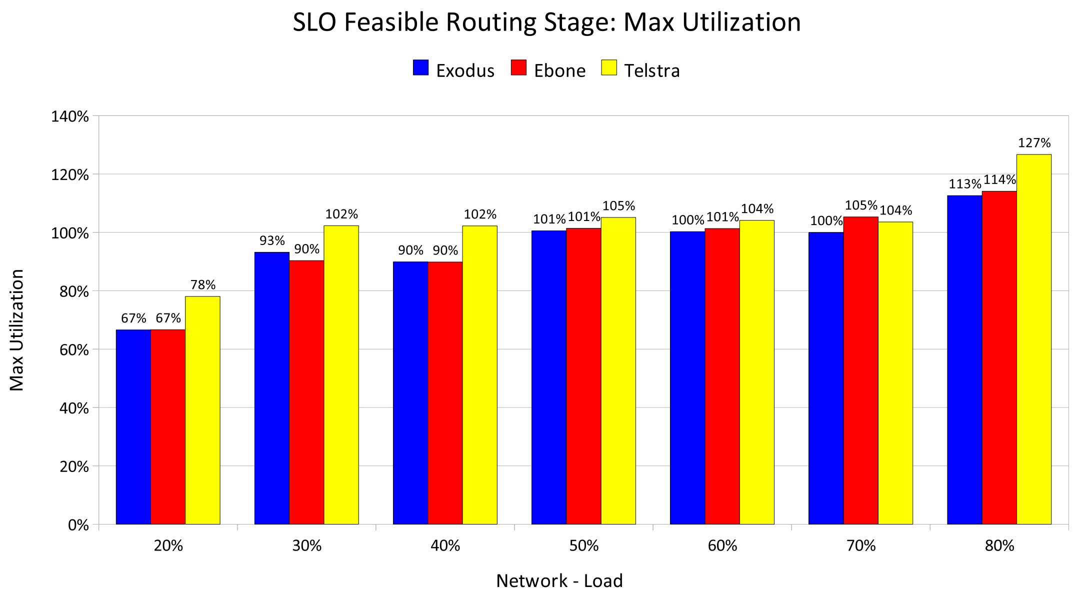


Figure 7.7: *Default network topology: 10 nodes and 42 links.*

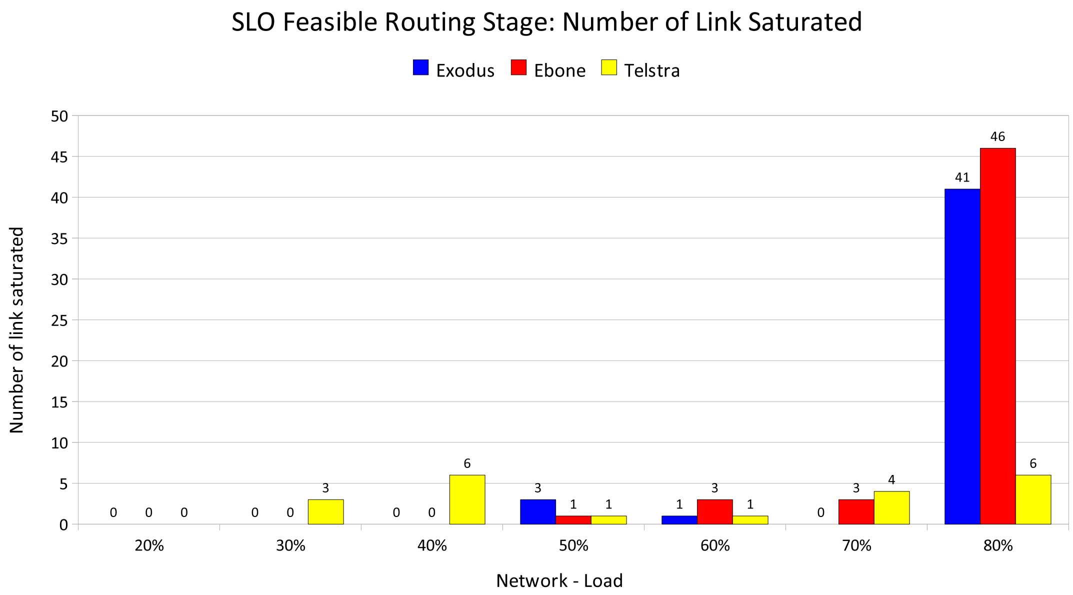
feasible set of weights when we use as input of the second stage, the same traffic matrix utilized in the first stage: the final solution is in fact feasible only in one case (10% - 10%). The infeasibility of the solution could be solved only switching on some links. Our idea for switching on more links is very simple and consists into use at the second stage a traffic matrix smaller than that used at the first stage. The table shows how a difference of 10% between the traffic matrices, involves the generation of feasible solutions.

7.2.1 SLO formulation data input

Table 7.9 and figure (7.8) report the computational results obtained running the feasible routing stage on the three real network topologies. For each couple network-load, we run the IGP-WO algorithm for 600 iterations, starting from a random set of weights (the set is generated taking as input a seed number). The results highlight good performances for the couples from 20%-10% to 70%-60%; only few links are sometimes saturated and the maximum utilization never exceeds 105%. We think that it would be possible to find completely feasible solutions performing more tests with different seed numbers, and exploring in this way new solution areas. For what concerns the couple 80%-70% and 90%-80%, the scarce quality of the results obtained could be easily explained: the full traffic matrix considered, represents an amount of traffic that could be routed through the network only using a completely free routing, that, obviously, could not be performed by the OSPF protocol. It's therefore reasonable to think that the 80% traffic matrices could be the biggest versions supportable by the networks that use the OSPF protocol. The impossibility of routing the 80% and 70% traffic matrices through a reduced version of the networks it is therefore explained. As expected, the IGP-WO algorithm finds less difficult when the traffic load is low: this behaviour is highlighted by the max utilization values that raise with the increase of the traffic load.



(a)



(b)

Figure 7.8: *Feasible routing stage stage global performances comparison: a) Max utilization. b) Number of links saturated.*

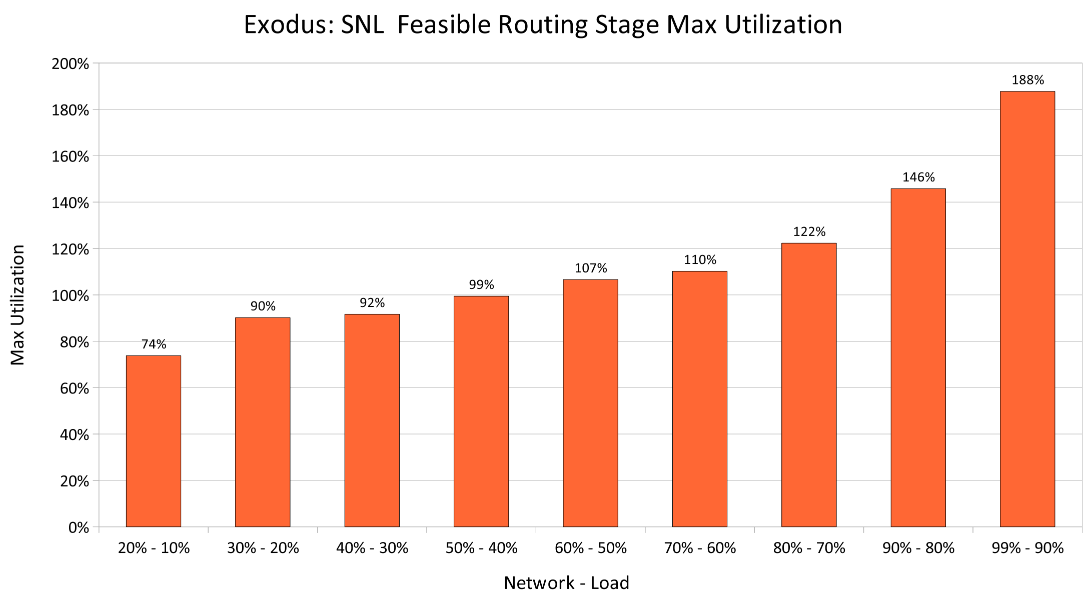
		Exodus		**Ebone**		**Telstra**	
Network	**Load**	U_{MAX}	$Links_{>100\%}$	U_{MAX}	$Links_{>100\%}$	U_{MAX}	$Links_{>100\%}$
20%	10%	66,60%	0	66,65%	0	78,05%	0
30%	20%	93,20%	0	90,33%	0	102,28%	3
40%	30%	89,90%	0	89,85%	0	102,22%	6
50%	40%	100,55%	3	101,40%	1	105,11%	1
60%	50%	100,25%	1	101,32%	3	104,11%	1
70%	60%	100,00%	0	105,30%	3	103,56%	4
80%	70%	112,60%	41	114,10%	46	126,69%	6
90%	80%	110,88%	26	/	/	/	/

Table 7.9: *SNO Feasible routing stage: global comparison between the three topologies.*

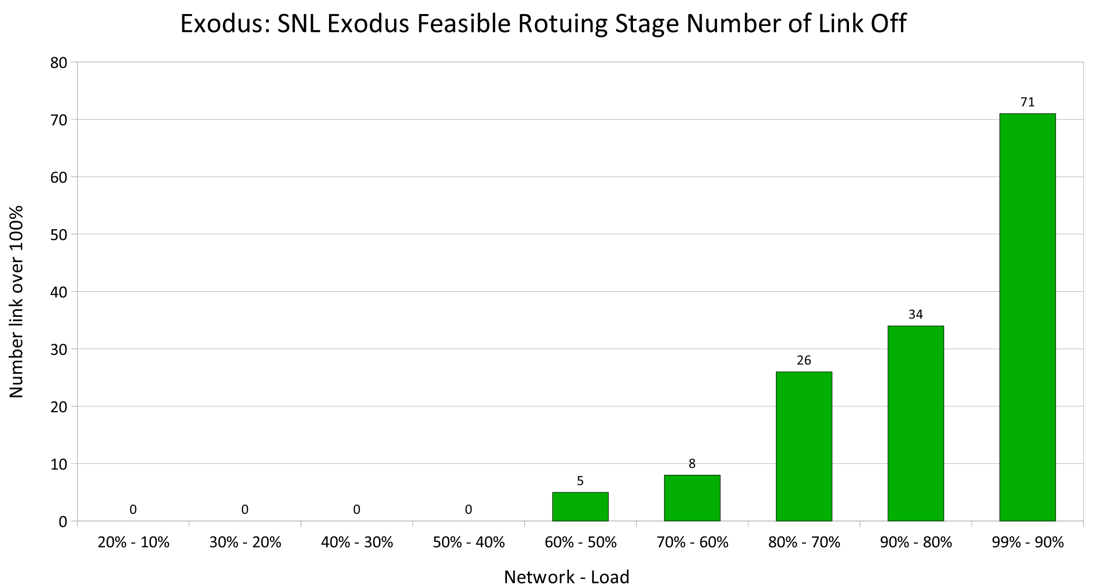
7.2.2 SNL formulation data input

Table 7.10 and figure (7.9) show the computational results obtained by the second stage elaborating the reduced Exodus topologies calculated by the SNL formulation.

The data reported in table 7.10 denotes the same behaviour already found in table 7.9: the algorithm performances are good till the network-load couple 80%-



(a)



(b)

Figure 7.9: *Exodus: SNL feasible routing stage performance: a) Max utilization. b) Number of links saturated.*

70%. The reasons of this behaviour (full traffic matrix too big) have already been explained in the previous section. The confirm to the validity of these reasons, is given by the results reported in table 7.10, where it is showed that the 90% traffic matrix could be not feasibly routed through the 99% Exodus version.

Analysing the results, we have moreover discovered that, after the feasible routing stage elaboration, it is sometimes still possible to switch off more links. Some links remain in fact completely empty, because off the impossibility to belong to any shortest path trees.

Exodus - 79 nodes - 38 edge 41 core			
Network	U_{MAX}	$Links_{>100\%}$	
20%	10%	73,80%	0
30%	20%	90,20%	0
40%	30%	91,65%	0
50%	40%	99,45%	0
60%	50%	106,60%	5
70%	60%	110,20%	8
80%	70%	122,30%	26
90%	80%	145,80%	34
99%	90%	187,75%	71

Table 7.10: *Exodus: SNL feasible routing stage computational results.*

Chapter 8

Conclusions

The scientific research in the green ICT area is still at the beginning. In chapter 2 we have analysed some recently published works concerning energy consumptions measurements and models [4] [5], sleeping strategies for network devices [7], power aware network design [10], adaptive link rate procedures [11] and optimization of energy costs [13].

In chapter 3 we have deeply examined the theme of the green IP routing, presenting some works about heuristic algorithms for the efficient management of network elements [14] [15] [16] [17]. A deep analysis has highlighted a scarce adaptability of these procedures to realistic scenarios.

For this reason we have developed a two-stages procedures called OSPES (Optimized Shortest Path Tree for Energy Saving), able to successfully operate with higher traffic loads and with more realistic network topologies, as those available at [19] and presented in chapter 6. OSPES procedure is presented in chapter 4 (switching-off stage and pre-processing data procedure) and 5 (feasible routing stage).

The procedure is based on the main idea of exploiting the configuration of the OSPF links weights, in order to perform a routing that could involve only a part of the network elements (links and routers): the first stage, considering the network topology and the traffic load, decides through a MIP formulation (4.1-4.8), which links and routers could be switched-off; the OSPF weights of these links are then set equal to a very big value, in order to exclude them from all the shortest path trees.

At this point the feasible routing stage, always considering the same traffic matrix, has to find a set of OSPF weights for the active links, with the aim of guaranteeing a feasible routing of all the traffic demands. When it is not possible to find a feasible weights configuration, it becomes necessary to switch on some links and routers and try again: in chapter 7, the computational results have shown how it could be possible to easily execute this operation, exploiting a gap of 10% between the traffic levels used in the two stages (higher level at the first stage).

The computational results have confirmed the validity of our new approach: the elaboration times are acceptable (some hours) and the quality of the feasible solutions found is completely satisfactory (high percentage of elements switched off).

However we think that the performances of OSPES could be still improved and there are many research lines that could be followed in the future:

- Development of a third elaboration stage based on the local search (like [14]), able to switch off other network elements without modifying any more the OSPF weights of the active links. In this way the final solution would be still improved.
- Execution of tests with realistic traffic matrices communicated directly from the ISPs, in order to better evaluate the realistic impact of OSPES.
- Execution of tests with bigger topologies.
- Execution of tests on different type of topologies: not only backbone, but also access topologies.
- Improvement of the pre-processing algorithm.
- Elaboration of new formulations able to model more complex topology structures (not only routers and links, but also chassis and line cards).

Ringraziamenti

Questa tesi rappresenta il punto di arrivo di un lungo percorso iniziato diciotto anni fa il primo giorno di scuola elementare. Tante persone in tutti questi anni mi sono state vicine, aiutandomi a raggiungere questo importante traguardo che verrá coronato con il conseguimento della Laurea in Ingegneria delle Telecomunicazioni.

Vorrei per prima cosa ringraziare i miei genitori, Marina e Andrea, che mi hanno affiancato durante tutto questo lungo percorso; li ringrazio per avermi sempre appoggiato con tutti i mezzi a loro disposizione, gratificandomi in occasione dei successi piú belli, e incoraggiandomi nei momenti piú difficili. Li ringrazio per aver sempre creduto in me.

Ringrazio i miei parenti, nonni, zii e cugini, riservando un pensiero speciale per mio nonno Bruno, che pur essendo mancato dieci anni fa, ho sempre sentito al mio fianco.

Ringrazio gli amici di una vita, Johnny, Stumatz, Rada, Mila, Birba, Gamba, Merons, Ragoz, Bertuz, Pitta, La Fede, Laura, Vale, Costy, Condo, coi quali ho condiviso gioie e delusioni, e sopratutto coi quali ho passato tantissimi momenti indimenticabili (vedi in ultimo il magnifico Triplete).

Ringrazio i miei allenatori Mauro, Alby, Lanfra, Yann, Papi, Riccardo e Christian, per avermi aiutato a diventare il tennista che sono oggi (forse dovrei fargli causa piú che ringraziarli!), senza ovviamente dimenticare tutti i miei compagni di squadra della Pro Patria coi quali ho condiviso ore di allenamento e domeniche di gara a squadre passate nei circoli di tutta la Lombardia, e, quest'ultimo anno, di tutta Italia.

Ringrazio i miei compagni di squadra del Politecnico, Sciuscio, Maurino, Pozz, Jose, Pedro, Marco, Paga, Francesca, Julia, Diletta, Paola, Edo, Andy, Andrea, coi quali anche quest'anno si cercherá di trionfare nella Coppa delle Universitá Milanesi.

Ringrazio i miei compagni di universitá Dany, Don, Gabry, Feo, Grillo, Ste, Teo, Galimba, Andrea e tutti gli altri che non posso nominare per motivi di spazio, grazie ai quali sono sopravvissuto durante questi cinque anni di Poli.

Ringrazio i miei amici del mare Marto, Spi, Tommy, Cippo, Cirlo, Ventu, Pio, Tax, Eli, Fra, Lero, Lollo, coi quali passo ogni anno delle mitiche vacanze.

Ringrazio molto sentitamente i miei due soci Andrea Martinetti e Daniele Tieghi, per avermi permesso di provare concretamente a inseguire il mio sogno di diventare imprenditore. Lavorando al loro fianco ho avuto la possibilitá di crescere tantissimo sotto il profilo sia personale che professionale.

Ringrazio anche alcune persone, che per differenti motivi, ritengo siano state fondamentali nella mia vita: Cesare Sciarra, Salvatore Scintu, Enrico Boselli, Stefano Gualandi, la famiglia Farronato.

Ringrazio infine con tutto il mio cuore una ragazza speciale che da piú di un anno é al mio fianco, e con la quale penso senza ombra di dubbio di aver trascorso i momenti piú belli della mia vita: Hélène, grazie mille!

Secondo facebook ho 264 amici, quindi deduco che avró sicuramente dimenticato di ringraziare qualcuno. Chiedendo umilmente perdono colgo quindi l'occasione di ringraziare anche tutti coloro che non ho nominato, ma che hanno comunque contribuito al raggiungimento di questo importantissimo traguardo.

Grazie a tutti!

Bibliografia

[1] J. G. Koomey, *Estimating total power consumption by servers in the U.S. and the world*, Technical Report Final Report, Lawrence Berkeley National Laboratory, February 2007.

[2] B. Fortz, M. Thorup *Internet Traffic Engineering by Optimizing OSPF Weights*, Proceedings of INFOCOM 2000, March 2000.

[3] M. Minami, H. Morikawa, *Some Open Challenges for Improving the Energy Efficiency of the Internet*, Proc. 3rd International Conference on Future Internet (CFI 2008), June 2008.

[4] M. Gupta and S. Singh, *Greening of the Internet*, in Proceedings of the ACM conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM 2003), (Karlsrhue, Germany), pp. 19 - 26, ACM, Aug. 2003.

[5] J. Baliga, K. Hinton, and R. S. Tucker, *Energy Consumption of the Internet*, in Proceedings of the Joint International Conference on Optical Internet and the 32nd Australian Conference on Optical Fibre Technology. (COIN-ACOFT 2007), (Melbourne, Australia), pp. 1-3, June 2007.

[6] M. Roughan, A. Reenberg, C. Kalmanek, M. Rumsewicz, J. Yates, Y. Zhang *Experience in Measuring Internet Backbone Traffic Variability: Models, Metric, Measurements and Meaning*, in proceeding of the 2nd ACM SIGCOMM Workshop on Internet Measurement, Marseille, France, 2002.

[7] P.C. Gunaratne, K.J. Christensen, *Predictive power management method for network devices*, IEE 2005 29 March 2005, Electronics Letters online no: 20051149.

[8] R. Jain, I. Chlamtac, *The P2 algorithm for dynamic calculation of quantiles and histograms without storing observations*, Commun. ACM, 1985, 28, (10), pp. 1076-1085.

[9] K. Christensen, P. Gunaratne, B. Nordman, A. George *The Next Frontier for Communications Networks: Power Management*, Computer Communications, Vol. 27, No. 18, pp. 1758-1770, Dec 2004.

[10] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, S. Wright, *Power Awareness in Network Design and Routing*, In Proceedings of IEEE INFOCOM 2008.

[11] P.C. Gunaratne, K.J. Christensen, B. Nordman, S. Suen *Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR)*, IEEE Transactions on Computers 57 (April 2008) 448-461.

[12] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, D. Wetherall, *Reducing Network Energy Consumption via Sleeping and Rate-Adaptation*, in Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NDSI'08) (USENIX Association, ed.), (San Francisco, California, USA), Apr. 2008.

[13] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, B. Maggs *Cutting the Electric Bill for Internet-Scale Systems*, In SIGCOMM 2009 Proc.

[14] L. Chiaraviglio, M. Melia, F. Neri, *Energy-Aware Networks: Reducing Power Consumption By Switching Off Network Elements*, In FEDERICA-Phosphorus tutorial and workshop (TNC2008), Brugges, BE, 2008.

[15] L. Chiaraviglio, M. Melia, F. Neri, *Energy-Aware Backbone Networks: a Case Study*, GreenComm'09 - First International Workshop on Green Communications, Dresden, Germany, June 2009.

[16] L. Chiaraviglio, M. Melia, F. Neri, *Reducing Power Consumption in Backbone Networks*, IEEE ICC 2009, Dresden, Germany.

[17] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, E. Vittorini *An Energy Saving Routing Algorithm for a Green OSPF Protocol*, IEEE INFOCOM 2010, San Diego (USA), 15 March - 19 March 2010

[18] T. G. Crainic, M. Gendreau, I. Ghamlouche, *Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design*, Technical report CRT-2001-01, Centre de recherche sur les transports, Universite de Montreal, 855.

[19] Rocketfuel Project [Online], available at www.cs.washington.edu/research/networking/rocketfuel.

[20] L.A. Wolsey *Integer Programming*, Wiley-Interscience, John Wiley & Sons, Inc. 1998.

[21] N. Spring, R. Mahajan, D. Wetherall *Measuring ISP topologies with rocketfuel*, In Proceedings of ACM SIGCOMM 2002, August 2002.

[22] TOTEM Project [Online], available at http://totem.run.montefiore.ulg.ac.be/.