Using GRASP and GA to design resilient and cost-effective IP/MPLS networks

Claudio Risso

UNIVERSITY OF THE REPUBLIC
ENGINEERING FACULTY
COMPUTER SCIENCE INSTITUTE
AND PEDECIBA INFORMATICS

P H D T H E S I S

to obtain the title of

PhD of Science

of the University of the Republic
Specialty : Computer Science

Defended by

Claudio Enrique Risso Montaldo

Using GRASP and GA to design resilient and cost-effective IP/MPLS networks

Thesis Advisor: Franco Robledo

Thesis co-advisor: Gerardo Rubino

prepared at UdelaR Montevideo, InCo/LPE Teams

and at INRIA Rennes, Dionysos Team

defended on May 5, 2014

Jury:

Reviewers :	Dr. Mauricio Resende	-	AT&T Labs (USA)
	Dr. Eng. Martín Varela Rico	-	VTT (Finland)
President :	Dr. Antonio Mauttone	-	UdelaR (InCo)
Examinators :	Dr. Francisco Barahona	-	IBM Research (USA)
	Dr. Eng. Eduardo Canale	-	UdelaR (IMERL)
	Dr. Eng. Gregory Randall	-	UdelaR (CSIC)

To Gaby, Migue, Male and Lu.

Acknowledgments

I owe most thanks to my academic advisors: Prof. Franco Robledo and Prof. Gerardo Rubino, for they encouraged me to venture into this challenge.

I would also like to thank to all those institutions that supported this work either logistically or financially. They are: ANII (Agencia Nacional de Investigación e Innovación, Uruguay), PEDECIBA (Programa de Desarrollo de las Ciencias Básicas, Uruguay), ANTEL (Administración Nacional de Telcomunicaciones, Uruguay), the joint Uruguay-France Stic-Amsud project "AMMA" (2013–2014), the "Ambassade de France à Montevideo" and the DIONYSOS team at INRIA-Rennes (France). These acknowledgments are extended to the persons Christophe Dessaux (Conseiller de Coopération et d'Action Culturelle at Ministère Français des Affaires Étrangères), Frédérique Ameglio (Attachée Culturelle Ambassade de France à Montevideo) and Gonzalo Perera (former ANTEL's vice-president).

Regarding the application cases, where some of the most important results of this work lie, there are also many fundamental contributors. I remark the participation of Engineers Diego Valle Lisboa and Laura Saldanha (ANTEL), for helping me to define the models as well as to gather the information necessary to feed and benchmark our algorithms. Complementarily, I also acknowledge the contributions of SeCIU (Servicio Central de Informática, Universidad de la República, Uruguay), especially of the Engineers Ida Holz, Luis Castillo, Sergio Ramírez and Mónica Soliño.

I would like to extend my deepest thanks to Dr. Mauricio Resende, Dr. Eng. Martín Varela Rico, Dr. Antonio Mauttone, Dr. Francisco Barahona, Dr. Eng. Eduardo Canale, and Dr. Eng. Gregory Randall, for honoring me by evaluating this work.

I cannot forget upon this occasion to mention to Raúl (El Güicha), former teacher and everlasting friend. He taught me maths, but far beyond of that, he awakened in me the interest by science and the passion for research.

And last, but not least, I thank to my family for supporting me all along this way. For in their love I always find the strength to stand and the will to move forward. This work is dedicated to them.

Using GRASP and GA to design resilient and cost-effective IP/MPLS networks

Abstract: The main objective of this thesis is to find good quality solutions for representative instances of the problem of designing a resilient and low cost IP/MPLS network, to be deployed over an existing optical transport network. This research is motivated by two complementary real-world application cases, which comprise the most important commercial and academic networks of Uruguay.

To achieve this goal, we performed an exhaustive analysis of existing models and technologies. From all of them we took elements that were contrasted with the particular requirements of our counterparts. We highlight among these requirements, the need of getting solutions transparently implementable over a heterogeneous network environment, which limit us to use widely standardized features of related technologies. We decided to create new models more suitable to fit these needs.

These models are intrinsically hard to solve (NP-Hard). Thus we developed metaheuristics to find solutions to these real-world instances. Evolutionary Algorithms and Greedy Randomized Adaptive Search Procedures obtained the best results.

As it usually happens, prospective real-world problems are surrounded by uncertainty. Therefore, we have worked closely with our counterparts to reduce the fuzziness upon data to a set of representative cases. They were combined with different strategies of design to get to scenarios, which were translated into representative instances of these problems.

Finally, the algorithms were fed with this information, and from their outcome we derived our results and conclusions.

Keywords: Multilayer networks, design of resilient networks, combinatorial optimization, metaheuristics, graph theory, optical transport networks, IP/MPLS

Contents

Chapter 1

Introduction

Contents

Some decades ago, the increasing importance of the telephony service pushed most TELecommunications COmpanies (TELCOs) to deploy optical fiber networks. Since the traffic volume requirements of the telephony service were low, the design process was guided by two main concerns: the cost and availability of the service.

This breakthrough took place in parallel with the digitalization of telephony service. Hence, the optical fiber network was used as the physical support to connect the infrastructure responsible of transporting the voice of customers. The two most common technologies used to transport the streams of bytes of phone calls were: Synchronous Optical NETworking (SONET) and Synchronous Digital Hierarchy (SDH). Both provide protection mechanisms against failures in the optical fibers.

At its early stages, the Internet service was implemented making use of the existing transport infrastructure. That is: connections between nodes of any IP network were implemented through circuits of existing SDH/SONET networks. As a consequence of the protection provided by the SDH/SONET network, the IP layer rarely suffered of unplanned topology changes.

Some years afterwards, the exponential growth of the Internet traffic volume demanded for networks of higher capacity. This demand caused the development

of Dense Wavelength Division Multiplexing (DWDM) technology. Instead of using several optical fibers to connect SDH/SONET nodes, DWDM allows multiplexing many connections over one single cable of optical fiber using different wavelengths. DWDM rapidly became very popular with telecommunications companies because it allowed them to expand the capacity of their networks without laying more fiber. Today, DWDM has turned out to be the dominant network technology in high-capacity optical backbone networks.

The main drawback of this array of technologies is the existence of multiple layers (overlays). Many layers imply the existence of many networks to maintain and operate, causing significant costs.

As a response to the previously described issues, the industry added several features to the traditional IP routers. The most relevant are: multiprotocol label switching (MPLS), traffic engineering extensions for dynamic routing protocols (e.g. OSPF-TE, ISIS-TE) and fast reroute algorithms (FRR). This new *technology bundle* is known as IP/MPLS. Since IP/MPLS allows recovering from a failure in about 50ms -a benchmark comparable to SDH/SONET-, opens a competitive alternative against traditional protection mechanisms, allowing savings coming from the elimination of the intermediate transport layer.

Since IP/MPLS allows the elimination of an intermediate layer, manages Internet traffic natively, and makes possible a much easier and cheaper operation for Virtual Private Network (VPN) services, it is gaining relative importance every day.

In this work we address the problem of finding the optimal -minimum cost- configuration for an IP/MPLS network to be directly deployed over an existing DWDM infrastructure. There are other works that analyze the design of networks with multiple overlays. Many of them focus on some variant of deploying an SDH/SONET network over DWDM, while others explore the integration of some IP/MPLS features. None integrates full capabilities nonetheless.

To fairly evaluate the suitability of the new interaction of technologies (IP/MPLS over DWDM), we developed two models consistent with it. Both problems are NP-hard and finding solutions was not affordable on instances with more than 15 nodes. Since practical applications are far beyond this limit and due to their intrinsic complexity, we developed metaheuristics to find good quality solutions to real-world size instances of both models.

Complementarily, we analyze the performance of metaheuristics using real-world scenarios provided by two different organizations. We remark this fact because due to the evolution of optical networks, they're prone to have structural issues against which newer technologies are better provided. The first application is over the new RAU (Uruguayan Academic Network); the network that supports connectivity among points of our own University. The second one is based on the planning of the IP/MPLS network of the national telecommunications company of Uruguay: ANTEL (Administración Nacional de TELecomunicaciones).

The main contributions of this work are:

i Models consistent with the characteristics of an updated interaction of technologies (IP/MPLS deployed over an optical/DWDM network). We analyzed two extreme strategies for IP/MPLS routing and developed correspondent models;

ii The design of metaheuristics to find good quality solutions for these models. The metaheuristics implemented are: Genetic Algorithms (GA) and Greedy Randomized Adaptive Search Procedure (GRASP);

iii The experimental evaluation based onto real-world network scenarios;

iv The assessment up from these scenarios, of the quality achievable over actual network implementations, using standard features of native protocols.

1.1 Evolution of telecommunications technologies

Internet is perhaps *"an exception"*, an atypical/unplanned event that has changed the world forever, not only within telecommunications industry but also on most human activities. It emerged from the academic community with the purpose to connect just a handful of points and nowadays spans globally; composing the greatest piece of technology ever deployed.

But Internet is not only a technological breakthrough, it also has its own global organization, which coexists with traditional: national, international, financial and corporative hierarchies. Despite former organizations, Internet is not intended to rule its members. It is mostly a plain, distributed organization that promotes and facilitates a basic set of resources (standard protocols, IP addresses, etc.) to connect to each other.

Some properties helped to hold previously unknown growth rates; they are: the scalability of the underlying network technology, the flexibility to take advantage of the existing telephone infrastructure and capacity to integrate new members and users easily. Along these 20-25 years since Internet's breakout from the academic world to the general public, traffic volume has been doubling itself year after year. Only between 2000 and 2009, the number of global Internet users rose from 394 million to 1.858 billion. By 2010, 22% of the world's population had access to computers with 1 billion Google searches every day, 300 million Internet users reading blogs, and 2 billion videos viewed daily on YouTube.

The current world has nothing to do with that: TELCOs, Computers Manufacturers and Software Providers envisaged for us some decades ago. Internet is a pretty good example of how a decentralized enterprise can achieve outstanding success. Within this work we shall only deal with some technological issues of this explosive phenomena, but before that, we brief some historical milestones, fundamental to understand the main contributions of this work.

1.1.1 Setting the board

Unlike telegraph communications, where characters are encoded by a discrete set of symbols (dots and dashes), the telephony service was born as an analog service. The principle is simple: human voice is transformed into a continuous electrical signal, which is base-band transferred in the access link (cooper-line/local-loop between the telephone and the central office), and sometimes modulated and multiplexed -mostly over microwave or satellite links- to connect distant points.

Along its almost 140 years of life the telephone service added several improvements to the original Bell's invention (1877). Just to mention some of them:

1940s - The "automatic telephone exchange" appears and electro-mechanical switchboards progressively substitute the switchboards operated by telephonists. As a consequence, people only needed dialing instead of talking to set-up a telephone call, and the time wasted to establish a connection decreased dramatically.

Complementarily, setting up a phone call not longer required human intervention. Any device capable of generating pulses (tones later) was able to connect to a peer through a phone line.

1950s - Long distance calls multiplexed over microwave links. Since microwaves tend to follow straight lines, the same radio spectrum can be shared among several links, as long as they do not overlap the line-of-sight with other nearby radio station. Microwave links are cheap and easy to deploy, so their irruption allowed communications between distant points became massive.

Microwave links at both sides of the ocean used Frequency Division Multiplexing (FDM) to separate channels, implementing over a point-to-point microwave connection the same principle used by AM radio stations (Amplitude Modulation) to coexist with other channels. Through the usage of microwave links, North America and Europe rapidly became highly connected.

1960s - International calls performed over satellite links. Even though microwave links were practical in many situations, relaying between too distant sites (transoceanic communications or even national calls on continental countries) were still hard to accomplish because of the curvature of the Earth's surface. Satellite links were the answer to this issue; through them, people in different continents were closer then than ever before.

1970s - Digital communications came to field. By the 60s, the main technical issue to be improved by Telephone Companies was the quality of the voice. The world was on its way to get globally connected but communications were still *noisy*, especially those with too many hops (i.e. long-distance calls).

Curiously, the solution came from the same approach used by the predecessor of the telephony service: the telegraph. Noise is an unavoidable guest in every

telecommunication instance. When an analog signal mixes with unwanted interference, some part of it always becomes part of the signal propagated to the next relay station. This fact is a fundamental component of the "Information Theory", developed by Claude Shannon in 1948.

When instead of a continuous signal a discrete signal is transmitted (one where only a prefixed set of values is allowed), there is room to correct tainted symbols (bauds). Moreover, if extra bits of redundant information are carefully added, it is possible to correct misinterpreted information. Exploiting this fact very low error rate links turn realizable.

Problem is that voice is analog by nature. Luckily, there is another theoretical result that closed the equation. The Sampling Theorem (Nyquist 1928 - Shannon 1949) guaranties that a continuous signal with a bounded bandwidth of B KHz, sampled at a rate of at least 2B samples per-second, can be reconstructed without any distortion.

Additionally, when these samples are discretized by an appropriate number of bits, the error incurred (digitization noise) can be plenty compensated by the lower error rates in digital links. Digitizing the voice is a very clever approach to deal with noise, in which controlled noise is intentionally introduced in order to keep unwanted noise under control.

The first AXE telephone exchange was presented in 1976. The analog voice coming from end-users (local-loop), passed through a Pulse Code Modulation (PCM) process, i.e. sampling and digitization, before being relayed to output interfaces, where different phone calls are to be arranged over Time Division Multiplexing (TDM) links.

1980s - Concurrently and complementarily to the evolution from electromechanical to digital exchange, microwave links were updated to digital ones. Since the early beginnings of analog multiplexing it was established that a bandwidth of 4KHz was enough for a standard-quality phone call. Therefore, 8000 samples by second (Sampling Theorem) were enough to code the voice. Using a logarithmic scale of 256 values (1Byte) the discretization noise was unperceivable, so, digital bandwidth of a TDM voice channel was fixed to $8\text{Ksamp/sec} \cdot 8\text{bits/samp} = 64\text{Kbps}$. In other words $1/8000\text{Hz} = 125\mu\text{secs}$ became the basic time-tick unit for TDM switches and 64kbps became the standard rate for a phone-call stream.

New digital microwave links were arranged over existing analog ones. Two digital TDM link standards were defined in order to re-use bandwidth of the electromagnetic spectrum: E-Carrier (Europe) and T-Carrier (North America), with basic capacities of: E1 - 32 voice channels (2.048 Mbps) and T1 - 24 voice channels (1.536 Mbps) respectively.

Both standard interfaces define the first level of its own Plesiochronous Digital Hierarchy (PDH), where several higher rate interfaces are also standardized ($\text{E2} = 4 \cdot \text{E1}$, $\text{E3} = 4 \cdot \text{E2}$, ..., $\text{T2} = 4 \cdot \text{T1}$, $\text{T3} = 7 \cdot \text{T2}$, ...).

A common problem to both PDH standards is that every source has its own "free running" clock. The rate is then allowed to vary by 2.048Mbit/s±50ppm. To compensate the eventual absence of a bit at the receiving multiplexer, stuffing bits are added to higher stream rates. This fact, together with the absence of references within super-frames pointing to lower rate streams, make too hard adding/dropping intermediate streams. PDH imposes a hierarchical network architecture and it hasn't native protection mechanism.

The 80s also witnessed the birth of commercial cellular telephony; perhaps the most exciting improvement for end-users since automatic telephone exchange.

1990s - By the 90s the Telephone Companies were at the summit of their business existence. It was the proper time for a large-scale deployment of optical fiber.

As physical media, optical fiber posses several convenient properties: broad base-band bandwidth, immunity to interference and low attenuation loss over long distances. In fact, SDH and SONET -the successors of PDH- were designed with aim to use optical fiber as physical support.

SDH/SONET also filled two important gaps of PDH standards: the possibility to easily add/drop a connection in any node of the network and protection mechanisms necessary to achieve the high availability pursued by TELCOs since many years before; the famous five-nines or 99.999% of availability for the telephone service.

Although different in fine detail, SONET (North American standard) and SDH (European standard) are essentially the same: a protocol-neutral transport technology capable of offering highly-available connections between points. The first cargo transported was: real-time, uncompressed, circuit-switched voice encoded in PCM; in other words: PDH streams.

Unlike PDH flavors, SDH/SONET nodes are tightly synchronized across the entire operator's network (usually inter-country networks). The sources of synchronism are atomic clocks (active and backup clocks) and synchronism is propagated through the very links of the network. Its protocol neutrality allowed SDH/SONET to evolve to next-generation SDH/SONET, a transport choice for later standards: ATM cells, IP packets, or Ethernet frames.

Late 80s consolidated the highest expression of the Public Switched Telephone Network (PSTN). The next planned step in PSTN's evolution was expanding digital connections to the customers' premises.

After too many years of coming back and forth, Integrated Services Digital Network (ISDN) came to life, but mostly upon TELCO's blueprints. ISDN came too late, its value proposition was too narrow and another disruptive competitor overtook its expansion.

1.1.2 Moving the pieces

Perhaps necessary at its time as the commercial framework to promote private initiative, Roosevelt's Communications Act of 1934 resulted in an aggregation of loosely-overlapped monopolies (broadcast TV and radio, cable TV, telephony).

After decades of constant growing, telecommunications services were at hand to most people. Nevertheless, diversity of services was stalled. Suffice to mention that during decades the three most significant changes in telecommunications value proposition were: color TV, cable TV and cellular telephony.

A mature technological environment, which included: accessible Personal Computers (PCs), widely deployed telecommunications infrastructure and a recently deregulated market (Clinton's Telecommunications Act of 1996), constituted the breeding ground for the impressive growth of Internet.

It's not easy to trace a timeline of Internet growing similar to that detailed in Section 1.1.1. Internet development is a decentralized process that comprises a huge number of threads at different levels of abstraction (networks, applications, protocols, standards). It was precisely this characteristic, which made possible such a remarkable high rate of growing.

We shall only mention here two historic milestones: the birth of Internet and the event that unleashed it from the academic community to the general public.

The first two nodes of what would become the ARPANET were interconnected between UCLA (University of California, Los Ángeles) and SRI International (an Independent nonprofit research institute), on 29 October 1969. The term *internet* was first used in 1974 as shorthand for *internetworking*.

With the support of the National Science Foundation (NSF) and other organizations, Internet progressively integrated more and more academic institutions. By the 80s it became the de-facto computer network for sharing scientific information among universities and related institutions, inheriting then the academic approach for doing things. Along two decades, Internet's technologies were maturing independently from the commercial issues.

The invention of the World-Wide-Web at CERN (Conseil Européen pour la Recherche Nucléaire) laboratories in 1990 was the unquestionable event that pushed Internet off the academy.

The WWW comprises the following elements:

1. A high level language convenient to specify documents in a rich format, which includes user-friendly objects such images or links to other pages.

   HyperText Markup Language (HTML) was the first and is still the most extended markup language.

2. Standards to bind, univocally identify and exchange objects within a distributed environment.

Hyper Text Transfer Protocol (HTTP) together with Uniform Resource Locator (URL) filled this need.

3. A set of applications (DNS, web browser, search engines), which implement these standards, isolating end-users from underlying complexity.

To work properly, the previously enumerated set of entities presumes the existence of always-available (on-line) connections among resources. Such a platform was already operational within the environment where the WWW was born; it was Internet in fact.

Unlike TV or radio, which are permanent and one-way communication media, telephony is on-demand and bidirectional. Telephonic infrastructure was regularly accessible but mostly unused; by the time when the WWW was released most people were off-line most of the time.

By fortune, the availability of good quality digital communications facilitated the technical deployment of Internet, providing permanent connections between Internet routers, and low-noise access links for end-users, which only needed a modem and an Internet Service Provider (ISP) to turn their phone-line into an Internet link.

After deregulation and as an effort to retain their customers, the Incumbent Local Exchange Carriers (ILECs) provided toll-free local telephone calls. This move kept Competitive Local Exchange Carriers (CLECs) under control, but opened the gates to ISPs, which rapidly became strong players.

Dial-up connections were cheap and accessible, but what was a convenient bandwidth for a voice stream (64Kbps), was far from being sufficient for Internet needs. Only five years later (by the early 2000s), many broadband access techniques such as: ADSL, Cable modems, fiber-to-the-building (FTTB) and fiber-to-the-home (FTTH) have become widely spread to small offices and homes.

By the mid 2000s, most telecommunications companies started the process to replace traditional telecommunication services by packet mode communication such as: IP telephony (VoIP) and IPTV.

It took almost 130 years to telephone companies to get to its maximum technological expression, which passed to the obsoleteness almost 15 years later.

1.2 Overlay networks

In the midway between legacy and next-generation networks, the multiplicity and interaction of technologies turned the operation and maintenance very complex. A typical context for such a reality is sketched in Figure 1.1, where networks are stratified according to their logical dependencies.

Physical level infrastructure comprises: optical fiber, coax distribution and local-loop networks. Local-loop (aka: cooper twisted pairs or outside plant) is the access

media for traditional telephone service. It connects customers' premises with telephone exchange (TDM switch of PSTN). In the beginning, residential users used modems to connect to Internet. These dial-up connections were established between customers' computers and a Remote Access Server (RAS) of an ISP, which answers the phone-call and negotiate a Point-to-Point connection to virtually link the computer with Internet.

ADSL access technology allowed permanent and higher bandwidth connections coexist with traditional phone service. Even though in this case the responsible of gathering customers' traffic towards the IP network (aggregation function) was an ATM network (instead of the PSTN), many characteristics of former dial-up service were inherited. One of them is the existence of an access server, updated to a Broadband Remote Access Server (BRAS) actually.

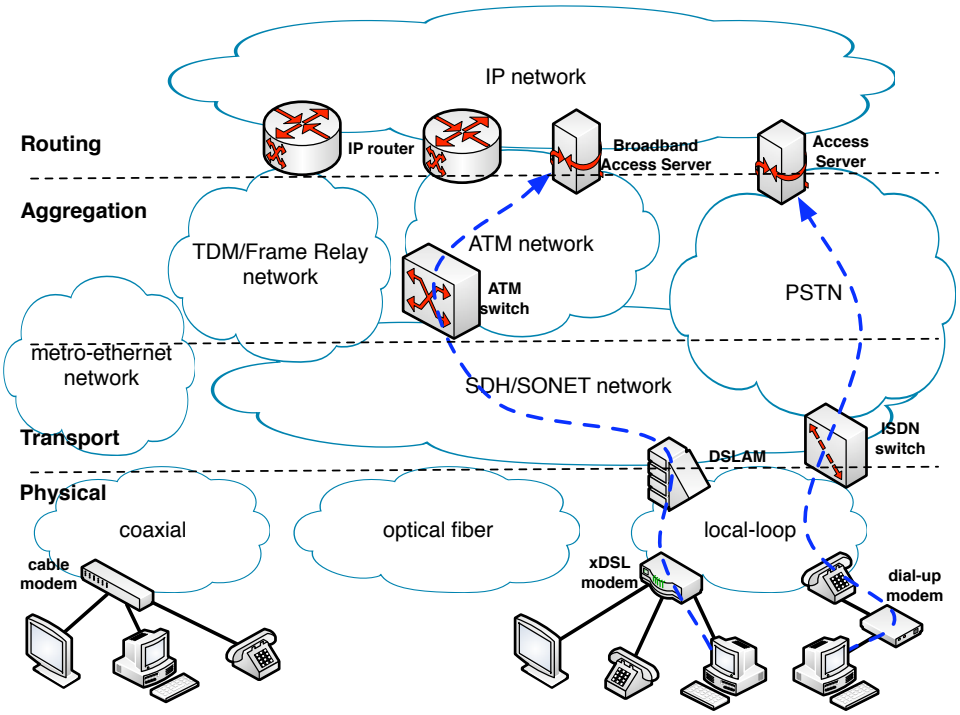


Figure 1.1: Technological context by early 2000s

Broadband connections made possible a richer offer of services. In order to obtain higher revenues from assets, ILECs evolved from classical phone services to triple-play services, a service-bundle that comprises: High Speed Internet (HSI), IP Television (IPTV, broadcast and Video-on-Demand or VoD) and a renewed phone service (voice, video and messaging), now supported over IP connections (Voice-over-IP or VoIP) instead of TDM circuits.

In parallel to the previously commented evolution (presented from the ILEC point-of-view), Cable TV operators had their own roadmap. These companies added HSI and VoIP services to their traditional broadcast TV service. Although access

technologies differ, the upper portions of both logical stacks are very much alike.

Internet routers were connected over leased-lines[1], ATM and/or Frame Relay networks -legacy infrastructure for permanent point-to-point data connections-, which were in turn connected by SDH/SONET circuits. Some TELCOs even deployed wide metro-ethernet networks for the aggregation function.

These trends led to networks composed by an increasing number of overlays. An *overlay network* is a network that is built on the top of another network. This happens when connections between nodes of a network are implemented as services of an existing one.

Legacy IP VPNs are a good example of overlay networks. The idea is simple, IP routers (not necessarily Internet routers) are physically connected to switches (TDM, Frame Relay or ATM switches) over which, IP packets are relayed among routers. For instance, Figure 1.2 presents a hypothetical network where routers: R1, R2 and R4 are linked to the switching network by Frame Relay links, whereas R3 is connected through an ATM one. Every time a router needs to send a packet to a peer, has to mark it with the appropriate tag: Data Link Connection Identifier (DLCI) for Frame Relay or with a combination of VPI/VCI (Virtual Path Identifier/Virtual Channel Identifier) in the ATM case.

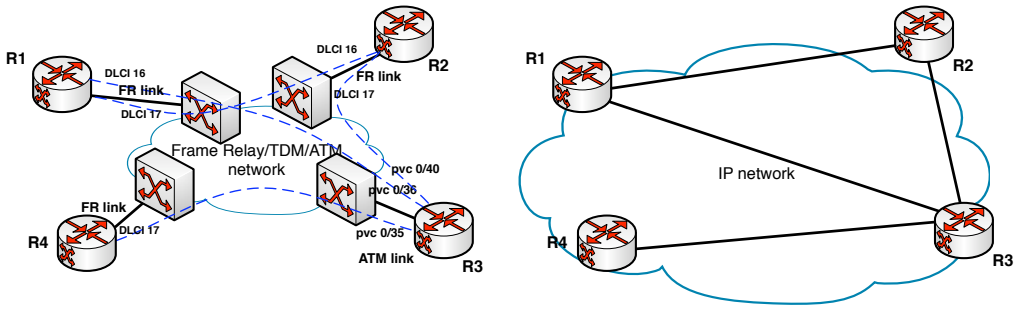


Figure 1.2: Technological context by early 2000s

When R2 (left half of Figure 1.2) tags a Frame Relay frame with DLCI=17, the switching network forwards and retags it until the frame's content is marked with VPI/VCI=0/40 and delivered to R3. From the routers point of view, they have different logical interfaces with the others (right half of Figure 1.2), although physically, many of these interfaces are supported over the same links.

This architecture allowed overlapping several logical networks without mixing traffic among them. Although Internet was the first and is still the most extended IP technology supported network, it is not the only one. Many organizations want to keep their internal communications apart from public access and nonetheless supported over IP technology. VPNs are an efficient solution to fulfill this need, since privacy is guaranteed whereas most infrastructures are shared among clients.

[1]Permanent TDM connections implemented over switches distinct from those responsible of temporary -phone call like- ones.

Frame Relay and ATM technologies were both developed with aim to become the standard to transport any payload (e.g. IP, IPX, AppleTalk, etc). Both were approaches to support any routing technology over a common forwarding plane, but none of them persisted over time. IP showed to be very scalable, it also was very widely-known -based in open standards-, so most organizations progressively went to IP networks, even for their private traffic, abandoning hence other technologies.

The standardization came from the top level, while the technology to interconnect IP nodes (TDM, Frame Relay, ATM, SDH, SONET, optical fiber, Ethernet) turned irrelevant. By early 2000s, TELCOs' roadmaps, carefully designed over decades, were being shattered by external forces. Against this unexpected lack of technological compass, most companies stacked technologies until they could be able to assess the future to come. In the long run, this strategy would be unsustainable.

1.2.1 Best of breed

For telecommunications companies, OPerational EXpenditures (OPEX) are much more important than CAPital EXpenditures (CAPEX). They usually represent 70% and 30% of the total costs, respectively. Therefore, keeping so many networks concurrently was too expensive. Since deregulation pushed companies to low down prices, important parts of this infrastructure were scheduled for deletion. Legacy switching infrastructure was the natural candidate to be substituted, and as the volume of Internet traffic became more and more important, Internet connections were gradually deployed over SDH/SONET.

Problem was that IP technology did not provide convenient mechanisms to isolate traffic amid clients. Reacting against this issue, CISCO Systems and other IP equipment providers spurred the development of Multi-protocol Label Switching (MPLS): a straightforward protocol for IP packets tagging, which allowed to treat them separately. Complementarily, traditional routing protocols were updated to support new features, and this technology was introduced as an update to the existing pure IP backbones, which evolved to what is now referred to as IP/MPLS.

By late 2000s other technologies -besides IP- were incorporated into what IP/MPLS could haul between routers (now renamed as IP/MPLS switches). Through pseudo-wires emulation, IP/MPLS was now capable of transporting: TDM, Frame Relay, Ethernet, ATM, SONET/SDH, PPP, HDLC, and other protocols. Hence, what was designed to be underneath the IP level, was being moved upwards in what constitutes a remarkable event, and a pretty good example that the level occupied by an overlay is determined by its function instead of its technology. Further details of IP/MPLS are analyzed in Section 2.2.

Along this process of rapid changes in the offer of services and their technological implementation, there was an ongoing challenge common to all TELCOs: Internet traffic volume was growing at exponential rates. Regardless of happenings in upper overlays, optical fiber networks, originally designed to last decades, were expending

resources swiftly.

During 2000s and unlike other technologies, optical network technology evolved regularly. Dense Wavelength-Division Multiplexing (DWDM), a technology that multiplexes several optical carrier signals onto one single optical fiber, was the solution to avoid laying more fiber.

Changes described within this section took place in a ten years period, where telecommunications technologies passed across a stressing best-of-breed process that converged to the current state of the art. Nowadays, a typical array of layers for a TELCO's backbone should roughly look like Figure 1.3.

This work aims to optimize the effectiveness of an IP/MPLS network deployed directly over an optical network; therefore in accordance with current tendencies the SONET/SDH layer will be omitted of our model. Moreover, since DWDM is basically a technology designed to multiply the number of optical fibers, we shall treat it as an indistinguishable part of the optical network. Throughout this work the optical network is referred to as the *physical layer* (see Figure 1.3).

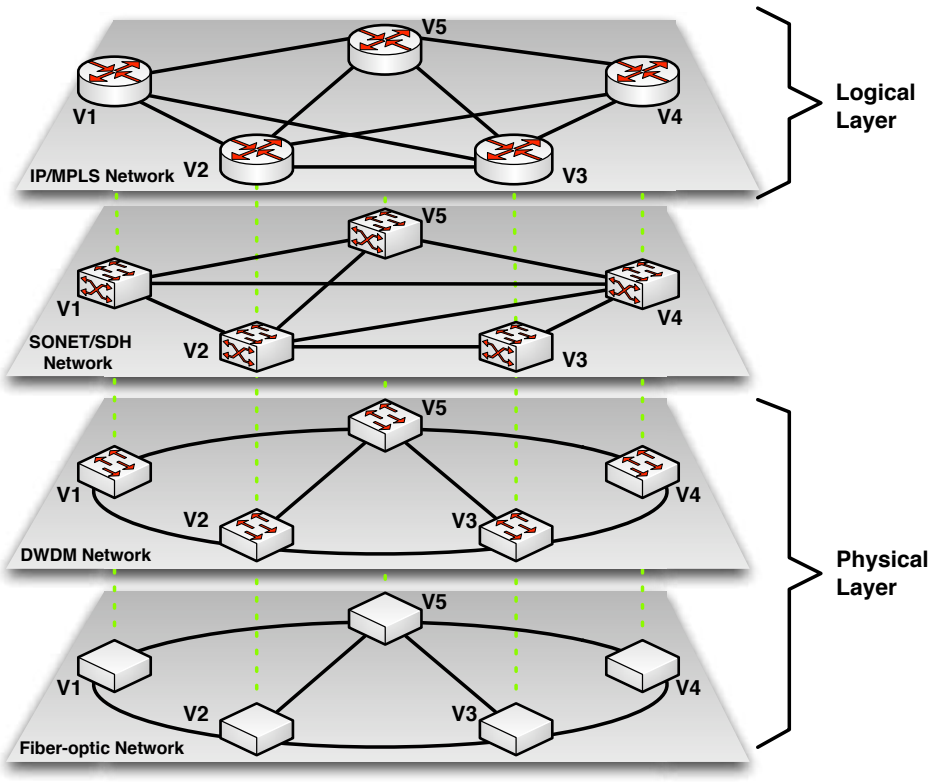


Figure 1.3: Stack of layers in the network of a typical telecommunications company

Within this model the nodes of the IP/MPLS layer are clients of the physical layer. There is no interaction between the control planes of both layers, so the IP/MPLS nodes are unaware of the existence of the physical layer. In fact, with

respect to the planes of control of the IP/MPLS nodes, they could be near to each other and connected directly. However, nodes are distant from the rest, and what really happens is that whenever a logical link is established, a *lightpath* must be configured in the physical layer to implement it.

A lightpath is determined by the sequence of links between DWDM nodes (a path in the optical network) and the wavelengths (colors) used at each step to implement the optical connection. Details about which color is used at each hop is out of the scope of this work.

These IP/MPLS nodes together with the lightpath-connections associated with the links between them form the so-called *logical layer* upon the physical one.

1.2.2 The physical layer

Optical fiber networks were originally deployed to support the telephony service and they constitute the foundation of the physical layers. In accordance with SDH/SONET service availability requirements, these networks were designed in such a way that several independent paths were available between each pair of nodes. To optimize these large capital investments, several models and algorithms were developed to achieve efficient network design.

Nevertheless, when optical networks were originally designed, requirements were quite different from current Internet ones; not only in its volume -which DWDM somehow compensates- but also in the shape/structure of the traffic matrix. Along all the process described in Section 1.1, traffic was mainly telephonic and most of it is local: terminates within the same city where the phone call is originated. Long distance phone calls were expensive and even much more expensive were international calls, which were mostly routed through satellite links. On the opposite direction, for any ISP most Internet traffic ends up outside of its network boundaries, turning upside down original design requirements.

DWDM is a technology to multiply the number of optical channels, not to modify the topology of the network. We developed this work without any intention to change the underlying optical fiber network[2]. Furthermore, important portions of the physical layer are rented to international carriers, turning unfeasible some changes. The topology of the physical network is an input to this problem.

We assume that the links of the physical network have no capacity-limit. If as an outcome of the design process we determine that the capacity in some portion of the physical network needs to be extended, this fact translates into installing more DWDM infrastructure to potentiate actual infrastructure. The consequences of these actions are considered in the cost function of the problem.

It is an accurate approximation to state that the cost of an optical fiber deployment is proportional to its length. Additionally, DWDM technology requires

[2]Which is by the way, the standard approach in exiting literature.

repeaters and amplifiers placed at regular intervals for compensating the loss in optical power while the signal travels along the fiber. As a consequence, the cost of a lightpath is proportional to its length over the physical layer. DWDM supports a set of standard high-capacity interfaces (e.g. 1, 2.5, 10 or 40 Gbps). The cost of a connection also depends of its bit-rate but not proportionally. For economies of scale reasons, the higher the bit-rate, the lower the per-bandwidth-cost.

Regarding this model, the information required from the physical layer reduces to: the topology (link lengths included) and the per-distance cost associated with standard capacities of the interfaces. We are also assuming that costs coming from the physical layer are the only costs to be considered in the problem.

Finally, we assume that once configured, the paths of the lightpaths are fixed. So, if any physical link fails, all logical links that make use of it do fail as well. The upper layer implements the mechanisms that sustain the availability of the services.

1.2.3 The logical layer

The logical layer is responsible of moving the traffic of the customers among different nodes. Internet traffic is usually the most important in volume, but there are many classes of traffic traversing the network: VoIP traffic, IPTV traffic and VPNs' traffic, among others. These traffic requirements are summarized into a matrix of demands between nodes, whose computation is out of the scope of this study.

The purpose of this work is designing an efficient logical network to be deployed over a fixed transport infrastructure. Several aspects should be considered to fulfill the design. First of all, we must determine which logical links will be connecting the IP/MPLS nodes and what capacity (bit-rate) will be assigned to each of them. Some links are not desirable in the solution and they should be excluded of the analysis. This usually happens due to engineering or architectural definitions.

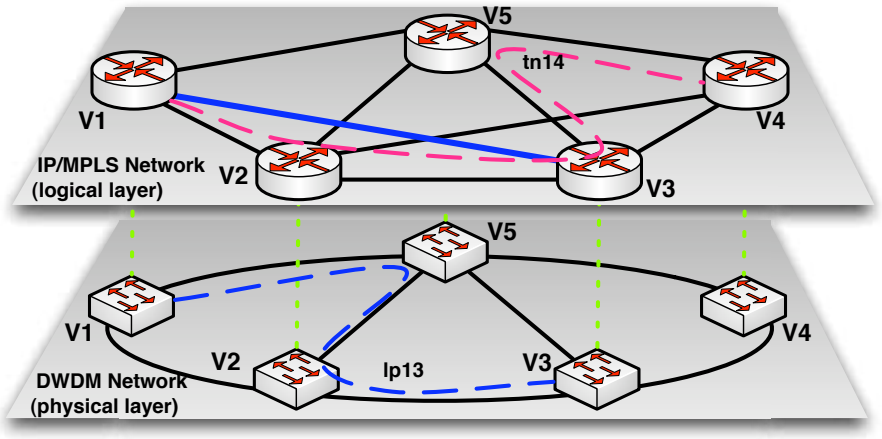


Figure 1.4: Possible implementations of the logical link (13) and tunnel (14)

On the other hand and since we are searching for minimum cost solutions, the

design process should avoid the usage of unnecessary links. These decisions rely on the model itself and the outcome defines the topology of the logical network in its nominal/fully-operating state.

The next step is determining the path followed by the lightpath necessary to implement each desirable logical link. This step is critical because forms the cost, and also because determines how faults on links of the physical layer affect logical links.

The paths chosen for the lightpaths determine the multiple topologies of the logical layer after failures in physical links.

Figure 1.4 sketches two hypothetical network layers with five nodes each. Let us take by example the logical link (13) -highlighted with a blue-bold line-. Its lightpath implementation is represented over the physical layer with a blue dashed-thin curve. For this configuration any fault on the physical links: (15), (25) or (23) will turn down the operational state of the logical link (13).

The logical network has many operational topologies. Eventually, it will have one for each link failure in the physical layer. Moreover, the increasing number of logical connections per physical link -intrinsic to DWDM- may cause multiple logical link failures from a single physical link failure (e.g., fiber cut).

As we shall comment in Section 1.3.1, the most extended native protection mechanism for SDH/SONET is the "1+1 protection", i.e., for every demand two independent lightpaths must be routed such that in case of any single physical link failure, at least one of them survive. Instead, our work replaces SDH/SONET by IP/MPLS as the technology of the logical layer. Therefore, models must be reformulated to adapt to particular characteristics of this technology.

Setting aside technical details, the IP/MPLS technology does not fit well with three natural features of the SDH/SONET technology.

1. The first one is the need of SDH/SONET to keep different demands between the same pair of nodes. In IP/MPLS networks, all the traffic from one node to another follows the same path on this network referred to as *IP/MPLS tunnel*. It is also outlined in Figure 1.4 -using a purple dashed-thin curve over the logical layer- a hypothetical path for the tunnel across which flows traffic between nodes v_1 and v_4.

2. The second difference is an improvement of IP/MPLS respect to SDH/SONET. In an IP/MPLS network, the path followed by a tunnel can automatically be reconfigured when a fault arises.

   In fact it might exist a different configuration for each tunnel on every logical topology. Moreover, although possible, there is no need to pre-establish all of these paths explicitly. If the appropriate information is fed to the routing protocols and the network is designed with care, the dynamic routing algorithms usually construct solutions of very good quality.

It is expected that the logical layer can be capable of accommodating the tunnels in such a way that, traffic demands associated with them can fit into the capacity of the logical links. This must be so, not only in the nominal topology, but also in all the topologies that may result from single faults on links of the physical layer. This model does not cover multiple and simultaneous physical faults. Through this constraint we aim to increase the quality of the designed solution, extending the strategy used by existing multiple-layers optimization models (see Section 1.3.2).

3. Finally, the third remarkable difference between models comes from how these technologies handle the existence of parallel links in the logical layer. In SDH the existence of parallel links is typical but in IP/MPLS it might conflict with some applications so we shall avoid it.

Because of the changes in the technology this model is significantly different from existing ones, and so are the algorithms needed to find solutions.

1.2.4 Synthesis of the problem

Before going into further details of the problem, we summarize the key points previously described.

The input data set is constituted by: the physical layer topology -DWDM network-; the client nodes of the logical layer -IP/MPLS nodes- and the potential links between them; the traffic demand to satisfy between each pair of nodes and the per-distance-cost in the physical network associated with each bit-rate available for the lightpaths.

The decision variables are: the logical links to be implemented, the bit-rate assigned to each one of them and the path for their lightpaths in the physical layer.

A feasible solution must be capable of routing every traffic demand over the remaining active links of the logical layer for every single physical link failure scenario.

The goal of this problem is to find the feasible solution with the lowest cost.

1.3 Design, dimensioning and capacity planning

Optical fiber laying requires the existence of conduits between the points to connect, whereas conduits installation usually requires digging trenches. After laying many fibers over a conduit, digging costs prorated over them, raise fiber costs in a range usually situated between 10% and 20%; so in the long run they are not a big issue. Moreover, when we consider that DWDM allows using a single fiber for several connections, this percentage is even lower.

Conduits constitute an important asset for TELCOs; the problem is that as a sole investment they are very expensive, whereas digging is a hard and time-

consuming activity. During many years deployment of optical networks were guided by telephony service, whose requirements do not span to traffic matters.

Basic carriers for SDH and SONET are referred to as STM-1 and OC-3 respectively. They share a line rate of 155.520Mbps and are capable of carrying 2,350 simultaneous phone calls. The immediate level of both hierarchies (STM-4/OC-12) raises this value to 9,400 (622.080Mbps).

For telephone calls the measure of offered load is quantified in Erlangs: a dimensionless unit that represent the average number of simultaneous phone calls. On normal situations telephone load varies along the day, taking its maximum/peak value sometime around midday (the *rush hour*). Minimum load occurs somewhere between 4AM and 5AM and its value only represents 2.5% of the peak one.

At the rush hour a city of one million inhabitants usually generates 10,000.00 Erlangs[3], which almost fit within one single STM-4/OC-12 link. Telephone traffic from such a city to the rest of the world is usually below 10% of the previous value. Finally, when we consider that any PSTN is composed of several exchange points (TDM nodes), connected by several links, we must conclude that basic capacities of SDH/SONET carriers are more than enough for telephonic traffic needs. Therefore, by the time when mayor portions of the optical network were designed, dimensioning and capacity planning was not a concern. The goal was to plan an optical network at lowest possible cost, minimizing the odds of its nodes become disconnected.

Nowadays, a 10Gbps link is capable of moving 15 times the telephone traffic of a city of one million inhabitants, while more and more single clients (e.g. Data Centers) demand several of these links for its own Internet connectivity. Internet traffic changed the paradigm under which optical networks were designed, forcing to integrate dimensioning and capacity planning into the models. We shall analyze both approaches in the following subsections.

1.3.1 Optimal design of a single layer network

The simplest class of mechanism for network survivability in the event of failure on a network element or link of a TDM transport network is Automatic Protection Switching (APS). APS schemes involve reserving a protection channel (dedicated or shared) with the same capacity as the channel to be protected.

There are different kinds of APS protection. All of them share one principle; the Network Element (NE) that detects the fault condition also initiates the protection switching action and is referred to as the *tail-end* node. The node at the other end of the protected circuit is referred to as the *head-end* node.

Rings are the basic construction units for TDM transport networks. A ring is a directed cycle-graph that spans several nodes, whose optical fibers do not repeat physical conduits. Since rings are the building blocks of TDM transport networks,

[3] In average, two of any hundred people are keeping a conversation over a telephone call.

APS schemes are specified over them.

In SDH/SONET, each optical fiber is used to transmit from one node to another (unidirectional transmission); so full-duplex transmission is achieved using the complementary portion of the ring. Figure 1.5 shows an example ring with four nodes (S_1, S_2, S_3 and S_4). The most basic connection for these nodes could be that marked as "ring 1", that is: links between (S_1, S_2), (S_2, S_3), (S_3, S_4) and (S_4, S_1).

When client nodes V_A and V_B demand a virtual connection between them, a circuit is established between S_1 and S_3 where the stream from V_A to V_B follows the path represented by a blue curve, while the stream from V_B to V_A follows the complementary one (red curve). Although feasible, this implementation is not reliable because any failure tears down the connection.

More fibers are needed to protect the previous circuit. APS uses another ring with links between the same nodes, whose transmission directions are reverted ("ring 2" of Figure 1.5). Normally, fibers connecting the same nodes are laid over common conduits.

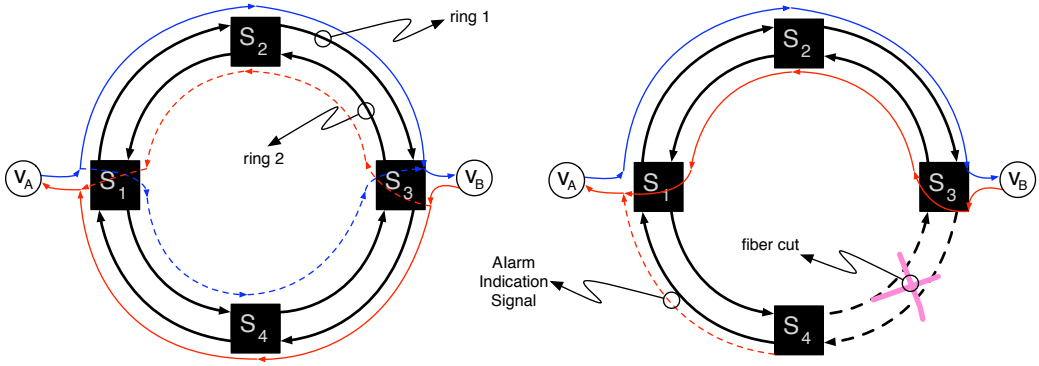


Figure 1.5: Transmission over rings and SNCP/UPSR (1+1) implementation

The simplest protection scheme in SDH is called: SubNetwork Connection Protection (SNCP). Its homologous in SONET is Unidirectional Path-Switched Rings (UPSRs). We shall refer to both flavors of APS as "1+1 protection". Despite details, both schemes send copies of the circuit's flows over both rings.

During normal operation (non-faulty state) head-end nodes receive both copies and select one of them to be delivered. Backup copies are represented by dashed curves upon Figure 1.5. Whether a fault on links between S_3 and S_4 disrupts active stream from V_B to V_A, node S_4 propagates an Alarm Indicator Signal (AIS) that is received by S_1, triggering a switching of streams that keeps operational the circuit (right half on Figure 1.5).

This mechanism is very straightforward, rapid (protections are always active/pre-established) and simple to maintain, even across multiple administrative domains (i.e. a circuit that spans several rings of different providers). Its main

disadvantage is its efficiency; because the same data is sent around the rings in both directions, the effective circuits' capacity is always the half of the physical one.

There is another protection scheme that may improve the efficiency under certain conditions. It's called Multiplex Section-Shared Protection RING (MS-SPRING) in SDH and Bidirectional Line-Switched Ring (BLSR) in SONET. Unlike SNCP/UPSR, MS/BLSR does not send redundant copies from ingress to egress. Rather, spare capacity is reserved along rings -in both directions-, and whenever nodes adjacent to the failure detect it, they reroute the traffic "the long way" around the ring making use of this spare capacity.

Left half of Figure 1.6 sketches how spare capacity (yellow dashed circle) is used in MS/BLSR to protect the circuit between V_A and V_B when conduit from S_1 to S_4) suffers a cut. This implementation is known as "1:1 protection". In a hubbing scenario (circuits terminating into a common node) the efficiency of MS/BLSR and SNCP/UPSR are identical.

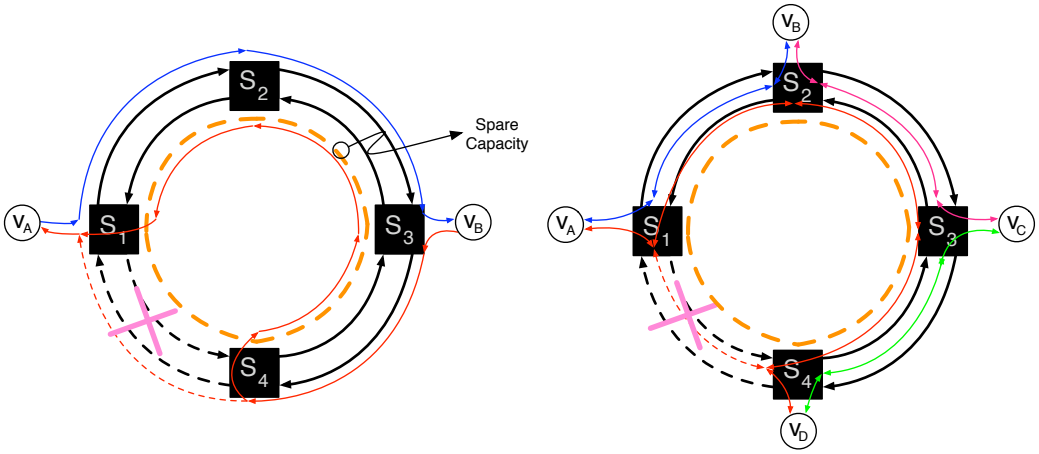


Figure 1.6: Protection based on MS/BLSR (1:1 and 1:N)

Efficiency improvements become possible when circuits are demanded between adjacent nodes. In this case, flows are one hop away from destination so they can use the direct link. The right half of Figure 1.6 corresponds to this situation, where circuits are indicated by bidirectional curves of different colors: blue for V_A to V_B, purple for V_B to V_C, green for V_C to V_D, and red for V_D to V_A.

The spare capacity in this case can be used to protect all circuits at once. For instance, after a physical failure between S_1 and S_4 (right on Figure 1.6), the circuit from V_D to V_A can be reestablished upon the spare capacity of remaining active links. When such a circuit protection is achieved is called "1:N protection". To protect circuits along several rings, a protection scheme must be provisioned at each one. Under some circumstances and whether the transport network is designed properly, MS/BLSR could help to increase the number of protected circuits.

Although from some point of view this scheme might be seen as a huge improve-

ment, links' efficiency is also bounded to 50% (spare capacity necessary), and this is in fact the theoretical limit due to the cycle topology, rather than the technological implementation for the protection.

In order to push the overall efficiency beyond, a protection mechanism capable of dealing with more complex topologies is necessary but this implies leaving behind APS (ring-local) protection schemes. In SDH/SONET domains this kind of protection only exists whether is implemented through an external Network Management System (NMS), but in IP/MPLS (see Section 2.2) this can be implemented through native/distributed protection mechanisms.

MS/BLSR is hard to design and coordinate among rings; it is also hard to troubleshoot (especially among administrative zones), a characteristic shared with relying protection on an NMS, which adds performance issues to the equation. Indeed, NMS protection is slower than APS and becomes a critical bottleneck when too many circuits fall at once.

APS is the most widely used protection family for SDH/SONET, particularly: SNCP/UPSRs (1+1 protection). Bandwidth efficiency wasn't a concern because -as we mentioned- traffic requirements were low. The relative abundance of bandwidth, aided to separate the tasks for network deployment into two complementary and loosely coupled groups: optical fiber and TDM transmission groups.

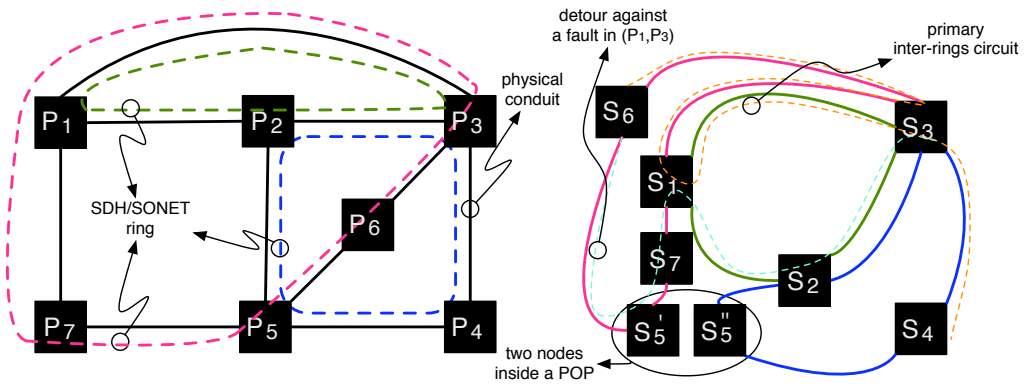


Figure 1.7: Example deployment of SDH/SONET rings

Figure 1.7 sketches a hypothetical situation. On left half, seven Points-of-Presence (POPs) are connected with conduits represented by black lines. Three SDH/SONET rings are implemented using fibers of these conduits. In this figure and from now on, we assume that all rings come in pairs (clockwise and anti-clockwise) to implement protection. Undirected lines will be used to represent both directions simultaneously. We also assume that opposite fibers of any link go back and forth over the same conduits. The paths followed by rings of this example are represented with dashed lines of different colors (green, blue and purple).

From the logical point-of-view the resulting network (TDM switches and rings) might look like the graph in the right half of the same figure. This network has one node-per-site, except inside P_5, where S'_5 and S''_5 are installed for example purposes. The green ring is a core ring, since circuits must traverse it to go from blue to purple ring. To exploit efficiency from this fact and since into this ring all nodes are adjacent, APS scheme in green ring could be 1:N while 1+1 might be used upon the others.

Whatever the scheme is, the solution is resilient to physical failures, even although several optical fibers use the same conduit. An accident that affects a conduit is typically a violent event, which cuts simultaneously several fibers in the conduit, perhaps all of them. Nevertheless, APS mechanisms guaranty these circuits keep working on.

For instance, if a backhoe cuts all fibers of the conduit between P_1 and P_3, then green and purple rings should be affected (both links between S_1 and S_3). A circuit between S_4 and S_6 whose operational path follows the yellow-dashed curve in Figure 1.7 would suffer of two simultaneous cuts. In spite of that and since independent APS processes, trigger restoration on both rings; point-to-point circuits only suffer a brief interruption ($\leq$ 50ms), almost imperceptible for active phone-calls.

The previous degree of resiliency, promoted the existence of loosely coordinated optical and transport engineering teams. Probably team members were unaware that this way of organizing tasks had theoretical support. Section 2.1 summarizes these results, while here we shall only mention that: *whenever nodes of a graph can be connected by two node-independent paths, this graph can be decomposed into cycles, i.e., any combination of two nodes or edges belongs to a common cycle.* Moreover: *if a set of nodes can be connected by k node-independent paths, then any subset of them with k nodes can be spanned by a cycle.*

These are exceptionally useful results. If a network (or a portion of it) is designed to be 2-connected, then, there are plenty of alternatives to group elements over SDH/SONET rings and therefore to protect point-to-point circuits between them.

The Minimum Weight Two Connected Spanning Network (MW2CSN, [Monma 1990], [Bienstock 1990]) and the Steiner Two-Node-Survivable Network Problem (STNSNP, [Baïou 1996]) are typical examples of this kind of physical network design problems. The first problem consists in finding the minimum-cost subset of links for a given graph, such that any pair of nodes can be connected by 2 node-independent paths. STNSNP is similar, although in this case some nodes are optional (Steiner nodes), provided just in case they can help to improve the quality of the solution, and allowed to be omitted from the result. Node-independent versions and edge-independent versions exist for both problems. Figure 1.8 shows a small example instance of STNSNP and its solution. Black-solid nodes are permanent nodes while white ones are optional/Steiner nodes. The cost of each edge is also indicated. Optimal cost for this instance is 14 and the solution is represented on the right half of the same figure.

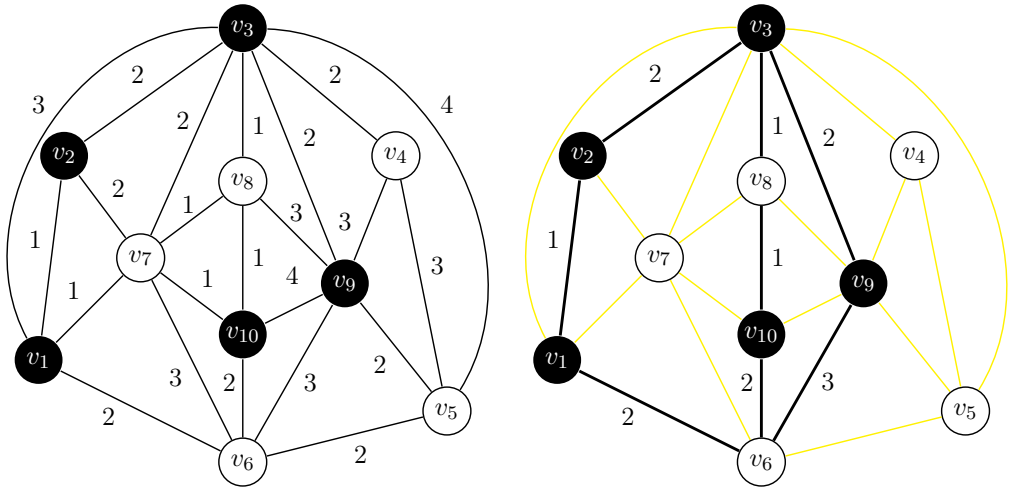


Figure 1.8: Example instance of STNSNP and its solution

Two-connectivity is a powerful tool to design optical networks, when they intended to bring support to an SDH/SONET overlay. Furthermore, whether is desirable that a higher number of nodes can be part of the same ring, it is sufficient to increase the connectivity between them. This could be the case for a core network within a bigger backbone.

Going upwards from the customers' premises, the outside plant ends up into a Central Office (CO) where telephone exchange is located. All COs are POPs. Some COs are of high priority because of the number or importance of their customers. For others, a temporary disconnection is a fair trade-off. Other POPs are usually added to this set to integrate functions (e.g. satellite links facilities).

So, to design an optical network the only input required from the rest of the corporation is the degree of importance of each POP, in terms of simple questions that might look like this: For which of these sites is acceptable a disconnection due to a failure in a single conduit? What nodes are candidates to conform the core of the backbone?

Answers to these questions can be translated into a vector of connectivity demands that determines constraints to feed an optimization problem. The k-Node-CONnectivity problem (kNCON, [Stoer 1993], [Kerivin 2005]) is an extension of STNSNP where different levels of connectivity are allowed. For instance, Figure 1.9 presents a graph with eight nodes and nine edges, for which the degree of connectivity is detailed in vector r.

Nodes whose indices in vector r correspond to value 3 (v_1, v_6 and v_8), count 3 node-independent paths between them. These nodes could constitute the POPs of the core ring to which all others are connected. Nodes whose indices correspond to 2 (v_2, v_5 and v_7), count 2 node-independent paths between them, and with those

previously connected. So, the backbone is composed of: (v_1, v_2, v_5, v_6, v_7 and v_8), i.e., those nodes with connectivity greater or equal to 2. There is only one path that connects v_3 to this backbone so it is likely to turn transitorily disconnected. Node v_4 is disconnected from the rest.

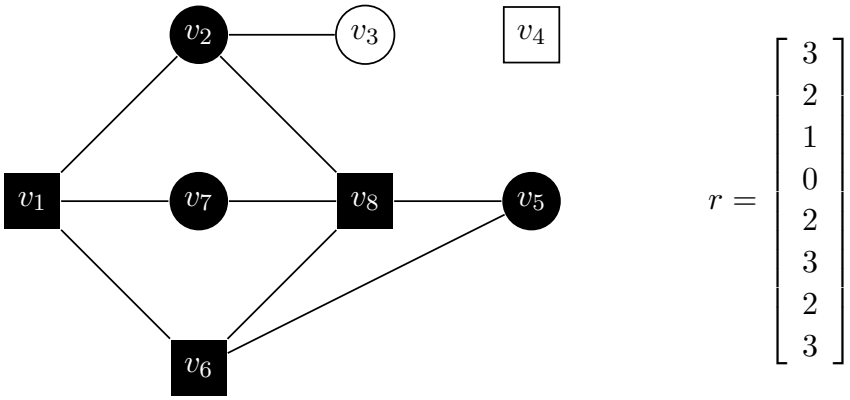


Figure 1.9: Connectivity levels between nodes of a graph

In synthesis, given a vector of integer numbers summarizing connectivity requirements and the costs necessary to implement edges between nodes, the output of kNCON represents a minimal cost network that spans all nodes with positive connectivity[4], whereas fulfills connectivity constraints. Not by chance, on late 80s, Bellcore played an essential supporting role to formalize kNCON, as well as to develop algorithms to solve it[5]. A widely known generalization of the kNCON that allows to set specific levels of connectivity between pairs of nodes, is known as Generalized Steiner Problem (GSP, [Agrawal 1995], [Canale 2009], [Winter 1986], [Winter 1987]).

Although computing the connectivity of any given graph is efficient in terms of computational performance, designing a low cost network with connectivity constraints is computationally hard to achieve (NP-Hard). In general, the optimal design of a single layer network is a challenging task that has been considered by many research groups, see: [Okamura 1981], [Stoer 1993] and [Kerivin 2005].

The design of physical networks is out of the scope of this work. We are commenting it here: because of its historical relevance, to clarify the improvements that newer technologies can provide, and also, because many of the paradigms these models aimed to solve need to be reviewed. Just to mention some of them:

Traffic - When important portions of optical fiber networks were designed, traffic volume between points was not an issue. Nowadays, sole customers demand several times the telephone bandwidth of an entire city. Besides, most of the traffic was metropolitan whereas long distance traffic flowed through satellite links, whose facilities were usually located near to cities themselves.

[4] A connectivity requirement of 0 means that the associated node is optional.
[5] Fiber Options: Software for designing survivable optimal fiber networks.

DWDM helps to increase the optical network capacity but doesn't change its topology. Many structural problems coming from outdated traffic requirements were inherited by actual optical networks.

Backbone/Core - Several backbone nodes were defined by the number or importance of its clients, so probably many of them are still important nowadays.

More critical is the fact that many times, cores within these backbones were arbitrarily defined. In other words those responsible for the transport network, defined where they wanted the core, instead of finding out where they needed it. The core of the network is responsible of moving most of the traffic. It counts with interfaces of the highest speed and should be placed to maximize profits from such interfaces.

Since traffic wasn't an issue cores are likely misplaced.

Topology - Physical networks were designed to deploy overlapping optical rings over them. As we shall see in Section 2.2, actual technologies allow much richer topologies, as well as mechanisms much more efficient than those of APS's flavors. Problem is that inherited physical topologies do not have to serve to current capabilities of technology.

Two decades ago, requirements were very different to current ones. Since important portions of the physical and transport networks were designed to fulfill former requirements, we are confident that through an upgrade of the design models there must be room to improve quality with current technologies.

1.3.2 Multi-layer aware models

Although capacity and demand are an intrinsic component of Okamura and Seymour's work ([Okamura 1981]), strictly speaking, this model refers to a single layer problem. Due to the growing of Internet demand, traffic and capacity are now a recurrent elements on most Network Design Problems (NDPs), particularly on multi-layer ones. However, what defines an NDP as of "multi-layer type" is the explicit existence of an overlay network, into which demands are to be routed. Within this section we analyze the evolution and the "state of the art" on related multiple layer networks works. We comment representative works, some of which have been grouped by their research team.

1.3.2.1 ZIB group contributions

The "Konrad-Zuse-Zentrums für Informationstechnik Berlin" (ZIB), from Berlin - Germany, is one of the groups that developed models related to ours. An early representative model for overlay networks dates from 1996: "A Network Dimensioning Tool" ([Alevras 1996]). This work effectively states the existence of two network levels, one responsible of connecting points physically, the other responsible of providing services to customers.

Instead of a physical network as was described in: Section 1.2.2 or Section 1.3.1, whose links support many logical connections simultaneously, in this work a supply network was introduced, whose edges (eventually parallel edges) represent different types of link technologies to connect nodes, such as: microwave, leased lines (point-to-point and permanent TDM connections), etc. Each technology has capacity bounds (minimum and maximum) with associated costs.

Failure scenarios are determined by failures on single edges or nodes of the supply network. This way of modeling failures implicitly demands the existence of an implementation in which links do not share a common physical component.

Given a set of demands among nodes is necessary a routing configuration to deliver them. Demands are distributed onto flows over multiple paths. Furthermore, a minimum level of diversity between points is a requirement. That is, the percentage of demand between any two points routed over a single path cannot exceed certain values. The goal is dimensioning supply edges, such that there exist a routing configuration that satisfies capacities for any failure-scenario and such that the sum of all capacity installation costs is as small as possible.

This model responds to a typical application case of a cellular operator (e-plus Mobilfunk GmbH), which owns: radio-bases, controllers and its backbone TDM switching, whereas leases connectivity among points to other companies. Authors detail a MIP (Mixed Integer Programming) problem formulation, and make use of Linear-relaxation to find solutions. Test instances are based on networks with less than 15 nodes.

Although revolutionary at its time and convenient for its application, this model is no appropriate for modeling current high-speed Internet networks. Nowadays, the number of logical links is much higher than the number of conduits, so a physical failure tears down several logical links simultaneously. Besides, IP/MPLS traffic must follow a single path between points, instead of diverse and concurrent paths.

Finally, traffic is not dynamically protected in this model. A failure in a supply element would disconnect active calls flowing across it, and clients should have to call again to re-establish them. Certainly this is the reason to impose traffic diversity. Actual applications demand better protection mechanisms.

Following the evolution of technologies this group worked on more updated models, issuing on 2006 an article called: "Two-layer Network Design by Branch-and-Cut featuring MIP-based Heuristics" ([Orlowski 2006]), and extending it on 2007 to another one called: "Single-layer Cuts for Multi-layer Network Design Problems" ([Koster 2008]). Both works are approaches to the problem: designing a transport SDH/SONET network to be deployed over a DWDM optical one. Here, we only comment: [Koster 2008], the most evolved of them.

This work presents a two-layers model with physical and logical levels. Multiple parallel logical links are introduced on the logical layer, to allow physically independent paths between points (the standard protection in TDM networks). These paths

are externally determined (prefixed) in order to guarantee effective protection.

The model integrates constraints to guarantee that certain demands can be protected. Protection is introduced by forcing copies of these demands to use different logical links[6]. Besides determining which logical links are necessary and what capacity should be assigned to each of them, the model considers the number of optical fibers necessary to allocate this capacity -over predetermined physical paths-, and the switching capacity of each node. Different models of logical nodes, with different switching capacities are allowed.

The objective aims at minimizing the total installation cost, that is, the sum of: optical fibers cost, nodes costs (different models have different cost), and costs coming from capacities assigned to logical links[7].

These works state the problem from a transport network designer point-of-view. Physical network is fixed and known, and a previous criterion to determine potential logical links as well as their lightpaths is assumed. Authors detail a MIP (Mixed Integer Programming) problem formulation, and use branch-and-cut techniques to find solutions. Test instances are of up to 17 nodes.

This model integrates a high degree of detail into the design (node models, link capacities, optical fibers used) and fits SDH/SONET technology very well. In our opinion it could be improved by integrating the "ring entity", because the existence of two disjoint paths for a demand does not guarantee that them can be implemented under an APS protection scheme, unless both match a sequence of logical rings all along their way.

Another practical drawback of this model is the need to predetermine physical paths for logical links (predetermine lightpaths). Finding an optimal configuration of disjoint paths over a network is a task hard to accomplish. In fact, there is another German university (University of Technology Berlin) that worked on such a problem ([Oellrich 2008]) and proved that it is NP-Hard.

There are several reasons why these works are inappropriate for our needs:

Instance size - This work found solutions for up to 17 nodes. One of the instances solved in Chapter 5 has 70 logical nodes and more than 200 physical ones.

Instead of using classic MIP heuristics, we decided to try metaheuristics to find good quality solutions for larger instances.

Technology - IP/MPLS traffic cannot be split into flows, it must be moved along a single path between each pair of nodes.

We agreed with our counterparts (ANTEL and RAU) to avoid parallel logical links between nodes. ZIB's models make use of them.

[6] In other words, flows are forced to recreate the scheme of a standard "1+1" SDH/SONET protection.

[7] Boards to install in nodes to implement links of different rates.

Since IP/MPLS and SDH/SONET have a different set of technological constraints, we need different models to recreate applications.

Costs structure - In both of our applications, the cost of logical nodes -even considering adaptation modules and backplane capacity- was far below the cost necessary to connect them.

Hence, we decided to dispense with details such as node or boards costs. Instead, we are providing plenty of freedom to choose the paths followed by lightpaths, since they are which form the cost. Further details are mentioned in Chapter 5.

We aren't stating here that referenced models are inappropriate in general. We are confident of their suitability to reflect costs into a metropolitan network, where nodes are close to each other; difference is that in our application cases, networks spanned nodes over an entire country. One of them even counted nodes at different hemispheres.

1.3.2.2 ZTiT group contributions

There are other groups that have worked with multi-layer models. As a representative of IP/MPLS overlay networks optimization, we selected another European group: "Zakład Teleinformatyki i Telekomutacji" (ZTiT), from the "Politechnika Warszawska"[8], Warsaw - Poland.

We selected two articles of this group to analyze here, they are: "An IP/MPLS over WDM network design problem" ([Kubilinskas 2005a]) and "Two design problems for the IP/MPLS over WDM networks" ([Kubilinskas 2005b]). Although evident, it is worth mentioning that both articles refer to an IP/MPLS overlay to be deployed over a (D)WDM physical network. So, both match our particular combination of technologies.

In spite of technological details, base modeling in these works is very similar to that used by ZIB group. Candidate paths for lightpaths are pre-computed and fed to the problem. Furthermore, candidate paths are also pre-established for realizing demands. From the technical point-of-view, ZTiT's models limit simultaneous path diversity, so at any time there must a single path realizing each flow demand; reflecting then the fact that traffic between points of an IP/MPLS network must flow across a tunnel.

Protection is achieved by selecting more than one path to realize demands, where each path is active on some physical failure scenarios. As we shall see in Section 2.2, IP/MPLS provides "hot active/standby protection" where a set of logical paths can be defined as candidates for tunnel implementations. Unlike APS, this protection mechanism is point-to-point and doesn't rely upon an underlying structure of rings, so unlike ZIB models, solutions to instances of ZTiT's models are always realizable.

[8] Which translated corresponds to: Department of Computer Networks and Switching - Warsaw University of Technology.

A remarkable difference with our approach is that these models set a budget for the installation, and attempt to satisfy the maximum number of demands within it. In our work, every demand *must be satisfied* and then the required budget is a result instead of an external constraint.

Test instances were computed with up to 60 nodes. Hence, problem size is convenient for real-world scenarios. Nevertheless, for instances with more than 40 nodes, the number of candidate paths to implement tunnels was limited to 3, constraining the search space too much (see Section 5.2). As we shall see in Section 3.1, our simplest model uses a pretty similar approach, i.e., imposes the existence of two physically disjoint paths to implement each tunnel. However, the model that allowed mayor improvements was the other. Main differences between works are:

- We used an objective function based on capacity and length required to implement lightpaths. Our work doesn't consider costs like: modules or boars necessary to implement the solution, because in our instances they represent less than 10% of the budget of the result.

- On the other hand and due to the economic importance of lightpaths' length, we rely on the model for the construction of paths for lightpaths and tunnels, increasing then the complexity of the problem.

  The example instance of Figure 3.9 shows that paths followed by lightpaths are not always intuitive, so we preferred to integrate them into the problem to solve instead of using a fixed set of candidates computed in advance.

- The second model developed in this work (Section 3.2), corresponds to a much more complex routing scheme -also realizable in IP/MPLS-, where eventually there are as many paths for each tunnel as physical links (failure scenarios).

  It is worth mentioning that many remarkable improvements in our test cases can only be achieved with this technological variant, which is not covered by referenced works.

1.3.2.3 Other works

Two interesting articles: "Survivable IP/MPLS-over-WSON Multilayer Network Optimization" ([Ruiz 2011]) and "Towards the maximum resource sharing degree for survivable IP/MPLS over WDM mesh networks" ([Zhang 2013]), share many characteristics with works of ZIB and ZTiT groups. Concretely, they pre-compute routes and assume the existence of an active/standby protection scheme on the logical network and/or the capability to dynamically change the route followed by the lightpaths over the physical layer (second approach modeled in [Kubilinskas 2005b]).

The simplest approach for protecting logical links upon the physical layer is using APS mechanisms[9], but they share the same lack of efficiency. To dynamically

[9] DWDM and SDH/SONET implement equivalent ring-based protection schemes.

reconfigure point-to-point lightpaths on the DWDM network in a mechanism coordinated with the IP/MPLS network, it is necessary the integration of control planes of logical and physical layers.

For our application cases this is not possible because important portions of the physical layer are rented to third-party companies (carriers). Carriers are reluctant to share control planes of their networks with customers and they have good reasons for it. Actually, a key factor in the success of Internet's routing scalability is the existence of a specific routing protocol to share information between ISPs: Border Gateway Protocol (BGP), which neither allow foreign ISPs get to know internal details of others, nor to control traffic into a peer network.

In addition to scalability issues, coordinating control planes of both layers increase the complexity of the Operation and Maintenance (O&M) tasks. Since OPEX is much more important than CAPEX in terms of costs, it wouldn't be wise jeopardizing O&M by changing its environment.

Perhaps the closest work to ours is "Lightpath Routing and Capacity Assignment for Survivable IP-over-WDWM Networks" ([Kan 2009]). This work considers the deployment of an IP network over a fixed/non-dynamical configuration of physical lightpaths (as is our case), and finds paths for lightpaths by its own means (they are not given in advance). Another coincidence with our work is that the logical layer is the sole responsible of providing protection.

Nevertheless, the goal of this model isn't finding the lowest possible cost for a network with certain constraints. Rather, it aims to minimize the *spare capacity* necessary to protect traffic against physical failures. Another remarkable difference is that IP traffic is modeled as a fluid, which can be split over multiple paths between source and destination nodes. Authors even use some theoretical results of "maximum flow problems"[10] during the construction of models and algorithms to find routes for the lightpaths.

Problem is stated as of two stages. "Lightpath Routing" where routing strategies are developed to construct good quality routes (of expected low spare capacity) for logical links over the physical layer. At this stage solutions are found through a MIP formulation. On a second stage ("Joint Lightpath and Traffic Routing") and up from pre-computed lightpaths (first stage), demands are routed over the logical layer for each single physical link failure scenario, based on a standard Linear Programming (LP) formulation. As an outcome of this stage, capacity is assigned to logical links. Test instances counted up to 12 nodes in each layer. Unfortunately, IP routing is not that flexible. It is very difficult to configure and maintain a network where traffic behaves like this. Moreover, the native object to route traffic in an IP/MPLS network is the tunnel (aka LSP), which is per-se a single path. Although some Network Equipment Providers (NEPs) integrate tools for splitting traffic between nodes, these are proprietary mechanisms and keep them working on a multi-vendor

[10] Concretely, the equivalence between maximum-flow and the minimum-capacity cut-set.

network (as is our case) may turn tasks of operation into a nightmare. Finally, the sizes of instances are not appropriate for our requirements.

Other multi-overlay models worth to be referenced are: [Balakrishnan 1994a], [Balakrishnan 1994b], [Balakrishnan 1998], [Cruz 2003], [hark Chung 1992] and [Fouilhoux 2011].

1.4 Structure of the thesis

This work is organized as follows. Chapter 1 summarizes evolution of services and technologies, remarking elements that determined the state of the art we're attempting to improve. There are several approaches for improving network efficiency, based on the application of methods from such diverse areas as: Statistics, Economics, Electrical Engineering and Operations Research (OR). This work contributes from the OR area. Related OR works are also analyzed in this chapter and their main results were commented from our applications' point-of-view. As far as our knowledge goes this work constitutes an innovative contribution since: i) states a model with immediate/practical applications for an updated combination of technologies; ii) making use exclusively of standard and widely-used mechanisms, which do not require mayor changes in O&M practices; iii) using real-world test instances, which embed structural issues proper of a representative historical evolution of optical networks and iv) developing metaheuristics to find good quality solutions for them.

Chapter 2 summarizes fundamental theoretical and technological details, necessary to understand models and contributions. Later on, in Chapter 3, two mixed-integer programming (MIP) formulations are presented as approaches to model extreme practical applications over the same technology. We also show some exact solutions found with CPLEX for small/simple but illustrative cases; some of whom arise uncertainty over some assumptions often integrated to former models. Additionally, we analyze the intrinsic complexity of both problems proving that mere sub-components of them determine NP-Hard problems. Due to the intrinsic complexity of both models and the reduced number of nodes for test instances reported in existing bibliography, we decided to use metaheuristics to construct solutions. Most precedent works base their algorithms upon heuristics derived from exact-methods. Chapter 4 details the two implementations used in this work: Evolutionary Algorithms (EAs) and Greedy Randomized Adaptive Search Procedure (GRASP).

Chapter 5 depicts characteristics and results for two real-world application cases, the IP/MPLS networks of ANTEL and RAU respectively, the most important commercial and academic networks of Uruguay. In accordance with counterparts of both organizations, several scenarios are defined to reflect realistic prospective conditions. Scenarios are also determined in order to analyze potential benefits from changes in fine-detail implementation strategies. Both metaheuristics were able to obtain promising results. For some scenarios, outstanding improvements came from the widespread usage of IP/MPLS technology, that is, by integrating facilities not

covered by referenced models. Finally, Chapter 6 briefly summarizes most relevant results of this work, and details main-lines for related future research.

1.5 Published papers

Many works derived from this thesis have been presented in: congresses, workshops, symposiums, journals or books. Some of them correspond to branches of this research project, carried out by complementary developments. Those which follow, correspond strictly to presentations issued by this author.

2009 Networking School, LANOMS (Latin American Network Operations and Management Symposium), Punta del Este, Uruguay. Tutorial Speaker.

2010 Applied Mathematics and Engineering, Conference CIMPA School, Balneario Solís, Uruguay. Conference Presenter.

2010 Joint International Meeting, ALIO - INFORMS, (Association of Latin-Iberoamerican Operational Research Societies), Buenos Aires, Argentina. Conference Presenter.

2010 International Conference OR2010, ("Mastering Complexity"), Universität der Bundeswehr München, Munich, Germany. Conference Presenter.

2010 Tercer Seminario de la Red Latinoamericana "Optimización Discreta y Grafos: Teoría, Algoritmos y Aplicaciones", Instituto Científico Milenio, "Sistemas Complejos de Ingeniería", Departamento de Ingeniería Industrial, Universidad de Chile, Santiago, Chile. Technical Session Speaker.

2011 Third Workshop "Proyecto Anillo" ACT-88, Universidad Técnica Federico Santa María, Valparaíso, Chile. Technical Session Speaker.

2012 Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC-2012, University of Victoria, Canada. "A parallel evolutionary algorithm for multilayered robust network design". Conference Presenter.

2012 XVI Congreso Latino-Iberoamericano de Investigación Operativa, CLAIO/SBPO, Río de Janeiro, Brazil. Special Session Speaker (Optimization problems in real-life settings).

2013 Fifth International Workshop on Reliable Networks Design and Modeling, RNDM-2013, Almaty, Kazakhstan. "Using metaheuristics for planning resilient and cost-effective multilayer networks". Technical Session Speaker.

The following is the list of publications issued from this thesis work (with the same reference number as they appear in the Reference list at the end):

[Risso 2012]: Claudio Risso, Sergio Nesmachnow and Franco Robledo. *A Parallel Evolutionary Algorithm for Multilayered Robust Network Design.* In 3PGCIC'12 - 7th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pages 291-296, Victoria, BC, Canada, November 2012.

[Risso 2013c]: Claudio Risso, Franco Robledo and Pablo Sartor. *Optimal design of a multi-layer network. An IP/MPLS over DWDM application case.* Current Developments in Optical Fiber Technology, pages 3-20, June 2013.

[Risso 2013a]: Claudio Risso, Eduardo Canale, Franco Robledo and Gerardo Rubino. *Using metaheuristics for planning resilient and cost-effective multilayer networks.* In RNDM'13 - 5th International Workshop on Reliable Networks Design and Modeling (RNDM'13), pages 90-96, Almaty, Kazakhstan, September 2013.

[Risso 2013b]: Claudio Risso and Franco Robledo. *Using GRASP for designing a layered network: A real IP/MPLS over DWDM application case.* International Journal of Metaheuristics, vol. 2, no. 4, pages 392–414, December 2013.

[Risso 2014]: Claudio Risso, Eduardo Canale and Franco Robledo. *Optimal design of an IP/MPLS over DWDM network.* To appear: Pesquisa Operacional - Special Issue from CLAIO/SBPO 2012, 2014.

Chapter 2

Fundamental Knowledge

Contents

This chapter summarizes components of knowledge used during the rest of this document. The first part covers theoretical elements, mainly upon: *Graph Theory*, *Computational Complexity* and *Metaheuristics*. Most of them are widely known, and are just included for disambiguation purposes.

The second part of this chapter focuses on technological aspects. There are plenty of free access technical documents describing the IP technology. However, regarding IP/MPLS the situation is quite different. This technology was spurred by companies, network equipment providers mostly, and the access to free documents of good quality is much more limited.

Furthermore, when available, this documents are usually focused on particular details, i.e., the pieces of knowledge are segmented/circumscribed to particular application cases. Hence, getting the big-picture of such a huge and flexible family of technologies interacting with each other, was a very hard and time consuming task. From our point of view, it represents a complementary state of the art analysis, technological rather than academic.

2.1 Theoretical background

The following is our summary of existing theoretical elements, necessary to understand the remaining of this document.

2.1.1 Fundamentals of Graph Theory

Definition 1. *A graph is an ordered pair $G = (V, E)$ comprising a finite set V of vertices (or nodes) together with a set E of edges (or lines), which are pairs of vertices of V. When E is a multiset of pairs of vertices (not necessarily distinct), such a graph is called:* multigraph *or* pseudograph*, otherwise is called* simple*. Simple also requires graphs with no loops (edges connected at both ends to the same vertex). When pairs of nodes in E are* unsorted pairs*, the graph is referred to as* undirected*. To avoid ambiguity, if the type of the graph is not specified the graph is assumed to be:* undirected *and* simple*.*

For instance, Figure 2.1 sketches three different classes of graphs. That on the left side corresponds to an example of a directed non-multigraph graph. It is worth mentioning that though there are two edges between v_2 and v_3, they are pointing in opposite directions so are different. There is also a loop in v_3.

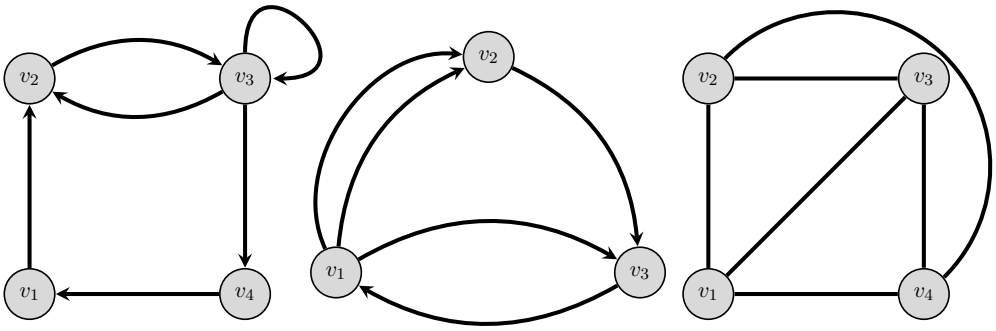


Figure 2.1: Representation for different types of graphs

The graph in the middle of Figure 2.1 corresponds to a multigraph because repeats a link between v_1 and v_2. As it was the case in the first graph, this is directed too. Finally, the graph placed at the right of the image is simple and undirected.

Graphs are basically an abstraction to represent a problem. Circumscribing the analysis to network applications, edges correspond to links. Edges on directed graphs are always represented by arrows, which indicate direction, i.e., the possibility to go from one node to another through it. If the intention is going back and forth between nodes in a directed graph, an edge must be included in each direction. On undirected graphs, edges always allow going in both directions.

Definition 2. *A* planar graph *is a graph that can be embedded into the plane, that is: it can be hand-drawn in such a way that its edges intersect only at their endpoints. In other words, it can be drawn in such a way that no edges cross each other.*

The left part of Figure 2.2 presents an example planar graph. The middle sketches a possible representation (notice that edges do not cross any other). However, when

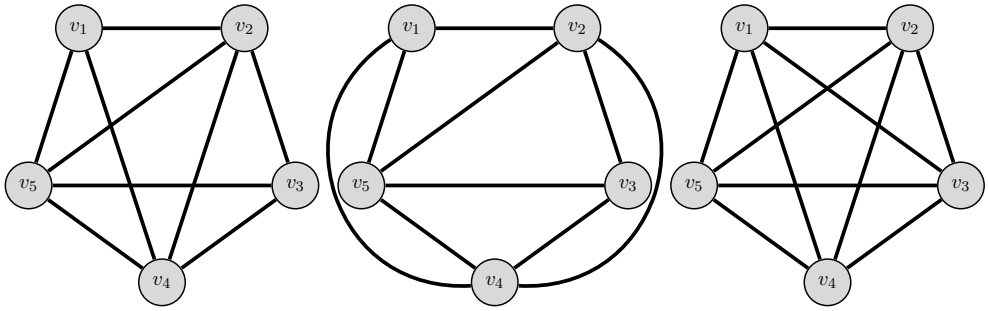


Figure 2.2: Planar graph, its planar representation and a non-planar example

an extra edge is added between v_1 and v_3, the result looses its planarity. Graph on the right of the figure cannot be drawn without crossing edges.

Definition 3. *Given a graph and two vertices, if these vertices are connected by an edge they are called* adjacent*, whereas the edge is called* incident *to the vertices. A* circuit *or* closed walk *consists of a sequence of vertices starting and ending at the same vertex, where each two consecutive vertices in the sequence are adjacent to each other. A* cycle *is a circuit with no repetitions of vertices or edges allowed, other than starting and ending vertices.*

When a planar graph is drawn without any crossing, any cycle that surrounds a region without any edges reaching from the cycle into the region, forms a face*. This definition also comprises the* external *or* unbounded face*.*

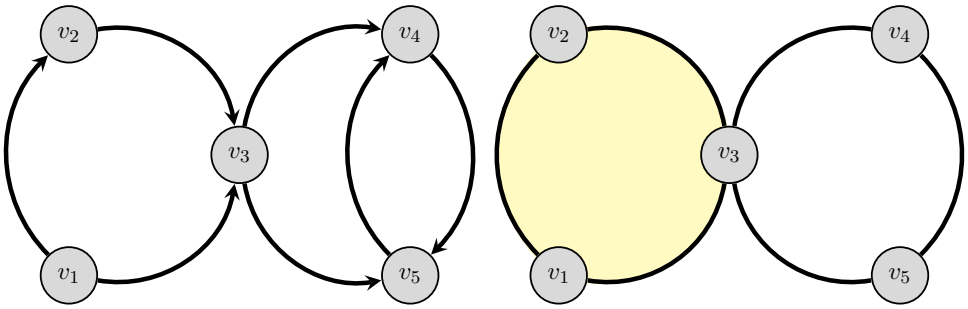


Figure 2.3: Circuits and cycles over graphs

Figure 2.3 presents two example graphs. The left one is directed and its only circuit is that that goes from v_4 to v_5 and back again. Since nodes are not repeated, it is also a cycle.

Disregarding directions and starting vertices, the example on the right half embeds three circuits, which span: $\{v_1, v_2, v_3\}$, $\{v_3, v_4, v_5\}$ and $\{v_1, v_2, v_3, v_4, v_5\}$. First and second circuits are also cycles. One of the three faces defined by this planar representation is colored with light-yellow. The remaining two faces are determined by the other cycle and by the outside of the graph.

Definition 4. *Given a planar graph G and a planar representation of it, let G', the* dual graph *of G, be a graph (usually a multigraph) that has a vertex corresponding to each face of G, and an edge joining two neighboring faces for each edge in G.*

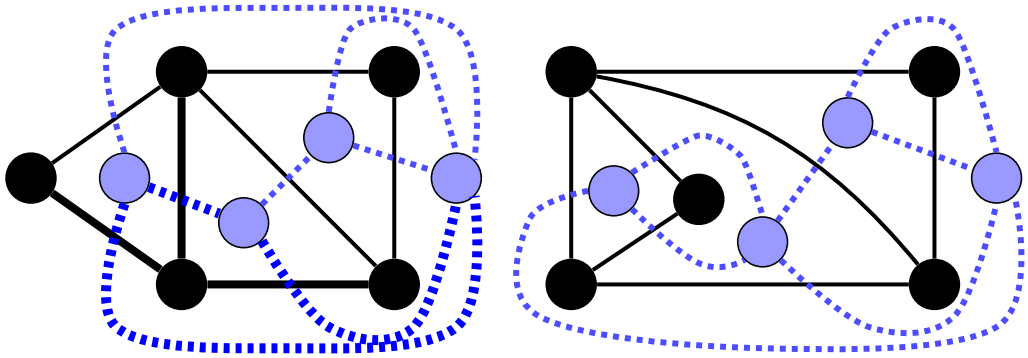


Figure 2.4: Planar representations and corresponding dual graphs

For instance, in Figure 2.4 are sketched two planar representations for the same graph. Primal graphs are marked with black nodes and lines; while in each case the corresponding *dual graph* is represented with blue nodes and dashed-blue curves.

Definition 5. *Let G be a graph of any type. Given two vertices u and v, a* path *from u to v consists of a sequence of distinct vertices starting at u and ending at v, such that each two consecutive vertices in the sequence are adjacent to each other. To avoid ambiguity in multigraphs case a sequence of distinct edges must be specified.*

For instance on left graph of Figure 2.3 there are two paths between v_1 and v_4, they are: (v_1, v_3, v_4) and (v_1, v_3, v_5, v_4). However, there is no path from v_4 to v_1. In the graph on the right half and since all nodes are spanned by a circuit, there exists a path between each pair of nodes.

Definition 6. *Let G be an undirected graph. Two vertices u and v of G are called* connected *if G contains a path from u to v. Otherwise, they are called disconnected. A graph is said to be* connected *if every pair of vertices in the graph is connected.*

For instance, the graph on the right of Figure 2.3 is connected.

Definition 7. *Let $G = (V, E)$ be an arbitrary undirected graph.*

- *If $G' = (V, E \backslash X)$ is connected for all $X \subseteq E$ where $|X| < k$, then G is* k-edge-connected.
- *If $G' = (V \backslash Y, E)$ is connected for all $Y \subseteq V$ where $|Y| < k$, then G is* k-vertex-connected *(or simply* k-connected*).*

Actually, the graph on the right half of Figure 2.3 is not just connected but 2-edge-connected. To disconnect the graph is necessary to remove a subset of 2 edges, for instance: (v_3, v_4) and (v_3, v_5). However, it is worth pointing out that the removal of any subset of two edges doesn't guarantee the graph disconnection; e.g., after removing (v_1, v_3) and (v_3, v_4) the graph remains connected. Finally, this graph isn't 2-node-connected. The removal of v_3 splits the graph into two.

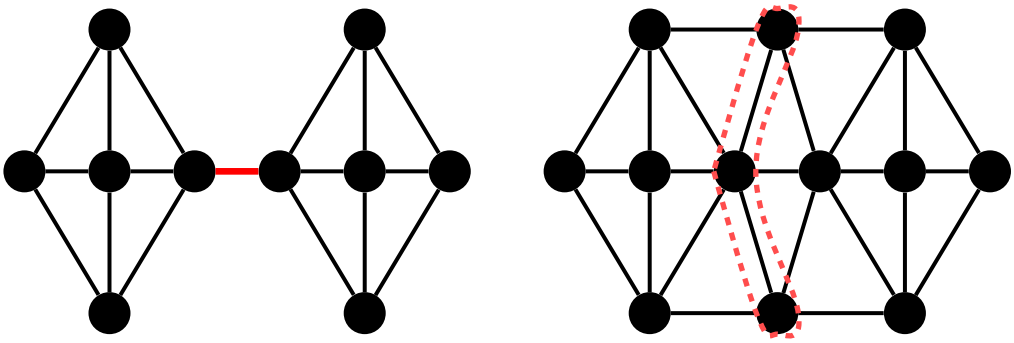


Figure 2.5: Examples of node and edge connectivity

Figure 2.5 presents other two examples. The left one is connected but no more than that, because after removing the edge highlighted with a red line, the graph is split into two connected sub-components. When such an edge exists is referred to as a *bridge*. Complementarily, an *articulation point* or *cut-vertex* is any vertex whose removal disconnects the graph. Node v_3 in Figure 2.3 or endpoints of the bridge in Figure 2.5 are good examples of it. The right half of Figure 2.5 sketches a 3-node-connected graph. It is pretty clear that is necessary a group of at least three nodes (like that remarked into the figure) to disconnect the graph.

Definition 8. *A maximal connected subgraph without a cut-vertex is called a* block. *Thus every block of a graph G is either a maximal 2-connected subgraph, or a bridge edge, or an isolated vertex. By their maximality different blocks of G overlap in at most one vertex, which is then a cut-vertex of G. Let $\mathcal{A}$ be the set of cut-vertices of G and $\mathcal{B}$ the set of its blocks. The* block graph *of G is a graph whose vertices are $\mathcal{A} \cup \mathcal{B}$, while the edges have the form (ab) where a is a cut-vertex of block b.*

Definition 9. *Let $e = xy$ be an edge of the graph $G = (V, E)$. By G/e we denote the graph obtained from G by* contracting *the edge e into a new vertex v_e, which becomes adjacent to all the former neighbors of x and y. When this transformation relaxes the premiss regarding the existence of an edge between nodes is called an* identification.

The left half of Figure 2.6 presents an example graph where *blocks* are shadowed with different colors. The example comprises seven blocks which overlap in two vertices (a_1 and a_2). The right part of the figure shows an example contraction between node x and y and its result.

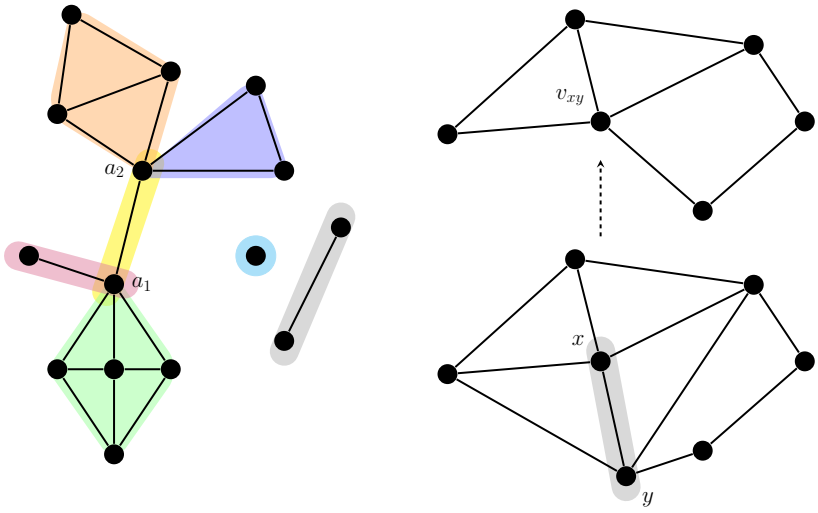


Figure 2.6: Examples of blocks in a graph and an edge contraction

Definition 10. *Let $G = (V, E)$ be an arbitrary undirected graph*

- *Given any node $v \in V$ we denote as $N(v)$ to the set of nodes adjacent to v.*
- *Given any node $v \in V$ we denote as $d_G(v)$ to the number of nodes adjacent to it (i.e. $|N(v)|$). This number is called* degree of v.
- $\delta(G) = \min\{d_G(v) | v \in V\}$, *is called the* minimum node degree *of G.*
- *Let k be the greatest integer for which G is k-node-connected. We call $k(G)$ to such number, which is referred to as* node connectivity *of G.*
- *Let m be the greatest integer for which G is m-edge-connected. We call $\lambda(G)$ to such number, which is referred to as* edge connectivity *of G.*

Theorem 1. *Given any graph $G = (V, E)$, it must stand that: $k(G) \leq \lambda(G) \leq \delta(G)$.*

Previous results show that node-connectivity is stronger than edge-connectivity. Up from this result we can be sure that graph on the right of Figure 2.5 is at least 3-edge-connected.

Theorem 2. *(Menger 1927)*
This theorem has two versions:

- *Edge-connectivity: Let G be any arbitrary, undirected graph. Given any two distinct vertices u and v of G, the minimum edge cut for u and v (the minimum number of edges whose removal disconnects u and v) is equal to the maximum number of pairwise edge-disjoint paths from u to v.*
- *Vertex-connectivity: Let G be any arbitrary, undirected graph. Given any two distinct, non-adjacent vertices u and v of G, the minimum vertex cut for u and v (the minimum number of vertices whose removal disconnects u and v)*

is equal to the maximum number of pairwise vertex-independent paths from u to v.

Previous theorem is of fundamental importance. It univocally binds physical resiliency/survivability of a network with the existence of independent paths among nodes.

Lemma 1. *(Berge 1973)*
Let $G = (V, E)$ be an arbitrary, undirected graph. The following are equivalent:

- *G is 2-connected.*
- *Any two nodes of G share a common cycle.*
- *Any two edges of G share a common cycle.*
- *Any node and edge of G share a common cycle.*

The previous result guarantees that counting with two independent paths between nodes is a sufficient condition to span with a cycle any arbitrarily two elements subset of a network. A typical design issue of a TDM transport network.

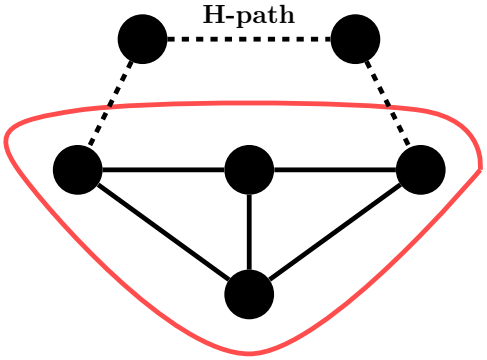


Figure 2.7: An H-path of a given graph

Definition 11. *Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be two arbitrary undirected graphs such that: $H \subset G$ (i.e. $V_H \subset V_G$ and $E_H \subset E_G$). An* H-path *of H is a nontrivial path G, whose vertices only intersect V at its endpoints.*

Figure 2.7 show an example H-path of the graph enclosed by a red curve. Dashed lines mark the H-path.

Theorem 3. *Let $G = (V, E)$ be any arbitrary, undirected graph. G is 2-connected if and only if it can be built up from a cycle, integrating successive* H-paths *to* H-graphs *previously built.*

Figure 2.8 shows through an example instance, how a 2-connected graph (G) can be built up from a cycle (H_0), appending H-paths iteratively (H_1,H_2,H_3,H_4).

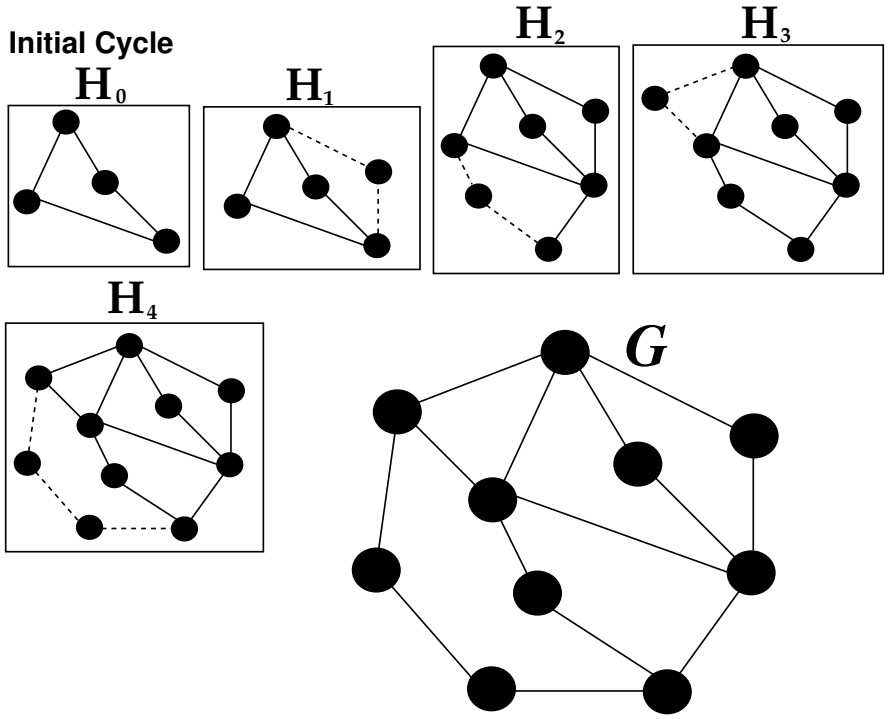


Figure 2.8: Construction of a G by appending a sequence of H-paths

The easiest way to deploy terrestrial conduits of optical networks is over earth's surface -digging the soil superficially-. Hence, physical networks are mostly planar graphs and the previous result guarantees that they can be incrementally deployed as a sequence of 2-connected graphs, which each in turn can be decomposed into a set of adjacent faces (rings).

Theorem 4. *Let $G = (V, E)$ be any arbitrary, undirected graph. If G is k-connected ($k \geq 2$), then, vertices of any set $U \subseteq V$ such that $|U| \leq k$ are spanned by a cycle.*

This result guarantees that to extend the number of nodes capable of begin spanned by a cycle, it is only necessary to increase the connectivity degree among them. We summarized theoretical results which fit like a hand in a glove with design requirements of single layer networks.

Definition 12. *Given a graph $G = (V, E)$, a* cut *is a partition $S \subseteq V$ of the set of nodes into parts: S and $V \backslash S$. Associated with each cut is a set of edges $CS \subseteq E$, where each link has one node in S and the other in $V \backslash S$. We refer to CS as the* cut-set *associated with S or simply $CS(S)$.*

A bond *is a minimal cut-set, i.e., it is a minimal (but not necessarily minimum), proper, not empty set of edges whose removal disconnects the graph. Thus, a bond splits a connected graph into two connected components, and from this point of view is an extension of the* bridge edge *to a set-bridge of edges.*

Figure 2.9 shows examples of different types of *cut-sets*. The left half indicates with dashed blue curves minimum and minimal cut-sets. Since is not possible disconnect a 2-connected graph by only removing an edge, the curve cutting red edges -(v_1, v_2) and (v_1, v_5)- corresponds to a minimum cut-set.

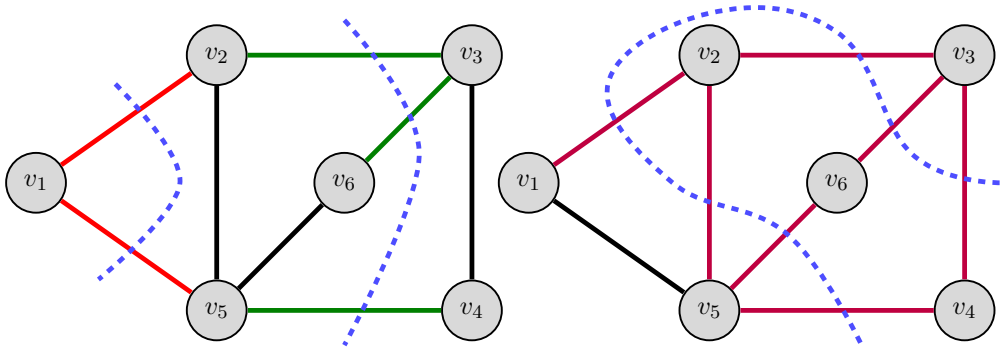


Figure 2.9: Minimum, minimal and maximal cut-sets of a graph

The second cut-set is that which crosses edges colored with green -(v_2, v_3), (v_3, v_6) and (v_4, v_5)-. It is minimal because whether any edge is removed from it, the remaining set of edges isn't capable of disconnecting the graph any longer, i.e., it is not a cut-set anymore. However, this is not a minimum cut-set, because we already found another with only two edges. Both cut-sets on left half of Figure 2.9 are *bonds* of this graph. On the opposite extreme, the right of Figure 2.9 highlights the maximal cut-set for the same graph, which comprises all edges of the graph but one (i.e. v_1v_5).

As we shall see later on this work, bonds are an important element in the structural analysis of an overlay network. Lemma 2 binds bonds with cycles. This relationship is highlighted with bolder lines and curves on left half of Figure 2.4.

Lemma 2. *Bonds of a planar graph are those whose edges determine a cycle on the associated edges of the dual graph.*

Until now, all properties described within this section referred to *unweighted graphs*, i.e., graphs where nodes and edges do not have: capacity, cost, length or other quantitative attribute associated.

On Section 1.3.1 we described three problems (MW2CSN, STNSNP and GSP) consisting of finding a subgraph of a given weighted graph (edges have associated cost). Before proceeding to Computational Complexity, we describe other two weighted graph problems.

Definition 13. *Given a weighted, directed graph $G = (V, E)$, two vertices s (*source*) and t (*sink*) and a capacity function $c : E \to \mathbb{R}^+$, a* flow function *or* s-t-flow *is a mapping $f : E \to \mathbb{R}^+$ such that:*

1. *For each $e \in E$ it fulfills $f(e) \leq c(e)$. That is: the flow of an edge cannot exceed its capacity. This is known as* capacity constraint.

2. *For each* $v \in V$ *other than* s *and* t*, it must hold that* $\sum_{(wv)\in E} f(wv) = \sum_{(vw)\in E} f(vw)$*. That is: the sum of the flows entering a node must equal the sum of the flows exiting a node, except for the source and the sink nodes. This is known as* conservation of flows.

The value of flow *is defined by* $|f| = \sum_{(sv)\in E} f(sv)$*. It represents the amount of flow passing from the source to the sink. The* maximum flow problem *consist in finding a flow* f *with the maximum possible value.*

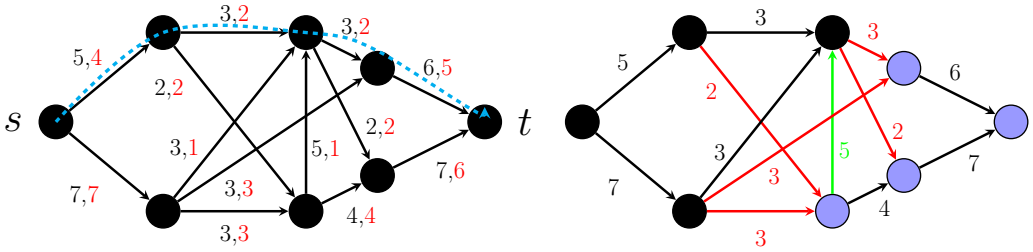


Figure 2.10: Capacitated graph with flow assignments and cut-set capacity

For instance, Figure 2.10 shows a hypothetical undirected graph whose capacities are indicated in black labels on the left figure. This figure also details a possible flow assignment, marked using red numbers. This flow isn't maximum since there is at least one path along which, the flow value can be incremented (dashed cyan curve). The right part sketches a possible cut of nodes. Those nodes in black contain the source node, whereas the complement (blue nodes) contains the sink. Since graph is directed, the cut-set only includes red edges (green edge is not into the cut-set), i.e., those from black to blue nodes.

Definition 14. *Given a maximum flow problem defined by a weighted, directed graph* $G = (V, E)$ *and* $S \subset V$*, an s-t-cut of* V *is a cut that contains the source but doesn't contain the sink, the* capacity *of such s-t-cut:* $c(S, V\backslash S)$ *or simply* $c(S)$*, is defined by* $c(S) = \sum_{(uv)\in E, u\in S, v\in V\backslash S} c(uv)$*.*

For instance, the capacity of the cut upon the right part of Figure 2.10 is 13. There is a fundamental theorem that binds flow and capacity.

Theorem 5. *(Ford-Fulkerson 1962)*
The maximum value of an s-t-flow is equal to the minimum capacity over all s-t-cuts.

Actually, Theorem 5 is a generalization of Theorem 2. Assigning capacity 1 to each edge of a graph and finding the maximum flow between any two nodes, we also find the connectivity degree between them. Besides this famous theorem, Ford and Fulkerson developed an algorithm capable of finding the maximum flow for a graph in a computationally efficient way. As an immediate consequence, determining the connectivity of a given graph can be also efficiently computed. Moreover, determining the minimum cut set (see Figure 2.9) can be efficiently computed too.

Definition 15. *Given a weighted, undirected graph $G = (V, E)$, and a length function $l : E \to \mathbb{R}^+$, the* shortest path problem *consists in finding a path between two vertices, such that the sum of the lengths of its constituent edges is minimized.*

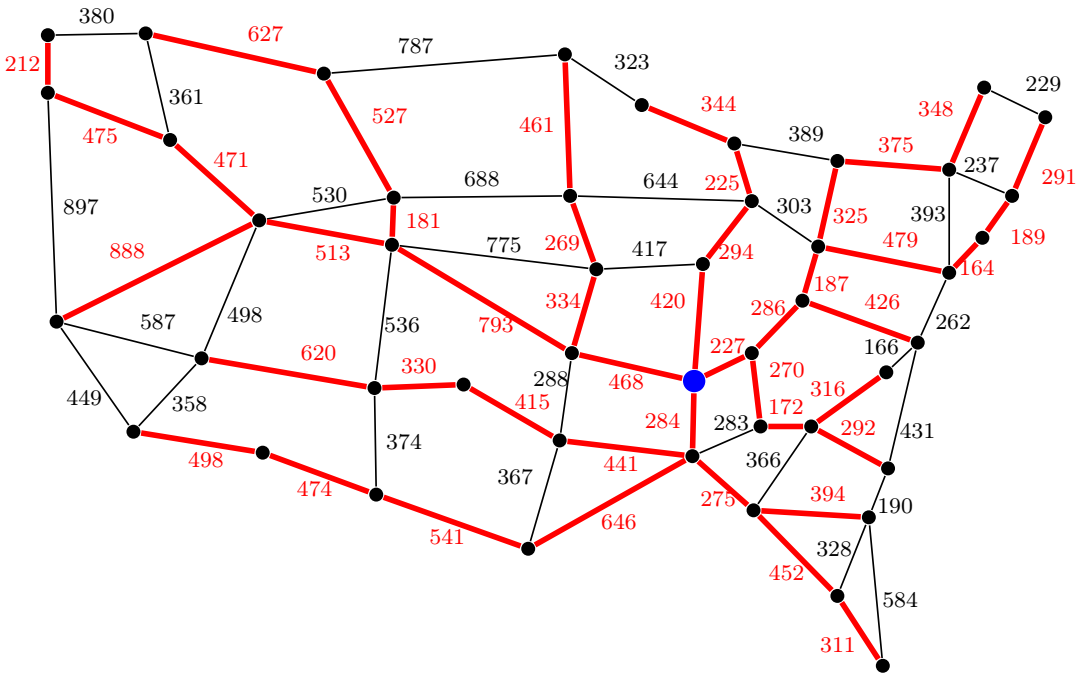


Figure 2.11: Graph corresponding to the USNet's network (USA)

Finding minimal paths between nodes can be very difficult to the bare eye. See for instance Figure 2.11, where optimal paths from the blue node to all the rest are highlighted with red. Just as it was the case with maximum flow problems, there are computationally efficient algorithms to find solutions to instances of this problem.

The most famous algorithm to solve this problem is known as "Dijkstra's algorithm" (published in 1959). This algorithm requires positive lengths on edges, which is appropriate for network application cases.

Algorithm 1 Pseudo-code for Dijkstra's algorithm

Procedure Dijkstra(V,E,$l : E \to \mathbb{R}^+$):

1: $S \leftarrow \emptyset$; $dist(1) \leftarrow 0$; $dist(i) \leftarrow \infty$ for $i \neq 1$;
2: **while** $S \neq V$ **do**
3: Take $j \notin S$ such that $dist(j) = \min_{i \notin S}\{dist(i)\}$;
4: $S \leftarrow S \cup \{j\}$;
5: **for all** $i \in N(j), i \notin S$ **do**
6: **if** $dist(i) > dist(j) + l(ji)$ **then**
7: $dist(i) \leftarrow dist(j) + l(ji)$
8: **return** $dist$;

A former algorithm called Bellman-Ford's algorithm (published in 1956), can be used upon graphs with negative edge weights, as long as the graph contains no negative cycle reachable from the source vertex. Despite that, Dijkstra's algorithm is the most widely used because is capable of finding solutions with less computations, i.e., in shorter time. Actually Dijkstra's algorithm is usually the working principle behind link-state routing protocols, OSPF and ISIS being the most common ones (see Section 2.2).

The references [West 1995] and [Diestel 2012] extensively cover these and many other graph theory properties.

2.1.2 Fundamentals of Computational Complexity

Previously, we commented the existence of *computationally efficient algorithms* to find solutions to instances of the *maximum flow* and of the *minimal path* problems. Because of the equivalence of this final problem with the *minimum cut set* (Theorem 5), and of it with computing *connectivity degree* between nodes, we must conclude that all problems mentioned into this paragraph are *computationally efficient to be solved*.

The immediate questions are: When is an algorithm efficient? When is a problem easy to be solved? The classical definition for the efficiency of an algorithm relies upon time-complexity and is: "the amount of time taken by an algorithm to run, as a function of the length of the string representing the input".

As measuring translates into comparing, at this point we introduce some useful concept to compare algorithms. The following definition allow us to compare the complexity of two algorithms.

Definition 16. *Let $f(x)$ and $g(x)$ be two real functions defined on some input string set, whose outputs (positive reals) respectively represent the time expended by two algorithms -running upon the some computer- to find a solution to an instance associated with each input string.*

We say that g is of higher complexity order *than f (denoted as $f(x) = O(g(x))$, if and only if, there is a positive constant M such that for all sufficiently large strings x, $f(x)$ is at most M multiplied by $g(x)$. That is, exists x_0 and M such that: $f(x) \leq Mg(x)$, for all $|x| > x_0$.*

This definition focuses on the tendency of the algorithm as the input increases in size, so, coefficients and lower order terms are usually excluded. For example, if the time required by an algorithm on all inputs of size n is at most $5n^3 + 3n$, the *asymptotic time complexity* is $O(n^3)$.

Definition 17. *An algorithm is referred to as* efficient *or* tractable*, if and only if, its* asymptotic time complexity *is a monomial. In other words, for any input instance x of size n, the algorithm can find a solution in* polynomial time.

Formally, there is a $p \in \mathbb{N}$ *such that the time required to the algorithm to find a solution* ($f(x)$) *is of* lower complexity order *than* n^p *(i.e.* $f(x) = O(n^p)$*).*

Let us suppose that computing power (in terms of computations per second) increases by a factor of 1000 at decade, and we have to choose between four different algorithms to find solutions for a critical application, which demands an answer within a day. As a baseline, performance for these algorithms was computed on 2010 and the best all of them could do, was finding solutions within a day for instances of size 100. Besides, it is known that complexity orders for these algorithms are: n^2, n^{10}, 10^n and $n!$ respectively.

Table 2.1 shows along decades -as computer performance evolves-, the expected size for instances that each algorithm can solve within a day. First of all we must observe that the first algorithm is the most promising one. Although worse in performance, the second algorithm shares a characteristic with the first one, decade after decade the instance size increases by a common factor: $\sqrt{1000} \approxeq 31.62$ in the first case, $\sqrt[10]{1000} \approxeq 1.995$ in the second.

year	$O(n^2)$	$O(n^{10})$	$O(10^n)$	$O(n!)$
2010	100	100	100	100
2020	3,162	200	103	102
2030	100,000	398	106	104
2040	3,162,278	794	109	105
2050	100,000,000	1,585	112	106

Table 2.1: Affordable size of instances for algorithms of different complexity orders

Unlike polynomial time algorithms, third and fourth are respectively of exponential and over-exponential complexity. When this happens the size of instances behaves as immutable to evolution of computer's performance. This is the practical consequence of intractability, it means that we cannot rely on hardware efficiency improvements to find solutions for big instances.

A polynomial time algorithm is usually a reasonable option. Problem is that there are many problems for which a polynomial time algorithm has never been found. Computational complexity theory as a branch of the "theory of computation" in "theoretical computer science and mathematics" is an abstract area. Outstanding contributors to this area were: Alan Turing, Stephen Cook and Richard Karp. A detailed analysis on it is out of the scope of this document, nevertheless we give here a descriptive/extendable set of concepts for *decision problems*, i.e., problems where the output limits to two values: Yes or No.

Definition 18. *Given a decision problem* π*, we call an* instance *of it to a concrete set of parameters that can be used to feed univocally an algorithm in order to find an answer (Yes or No). Any decision problem* π *has an associated domain-set of instances* D_π*, where the problem makes sense. Let* $Y_\pi \subseteq D_\pi$ *be that subset of instances for which the answer is* Yes*.*

Definition 19. *We say a decision problem π is of class* **P**, *if and only if, there is an algorithm capable of finding answers (solutions) in polynomial time as a function of the instance size.*

We say a decision problem π is of class **NP**, *if and only if, there is an algorithm capable of checking a solutions (given by an external oracle) in polynomial time as a function of the instance size.*

It is pretty clear that whether a problem can be *solved* in polynomial time it also can be *checked* in polynomial time, so $P \subseteq NP$. Although the opposite inclusion looks unlikely, until today no one has ever found a formal proof of it. In fact, the conjecture $P \neq NP$ is probably the most important open problem in computer science and we are not intending to search for an answer here. Instead, we take the usual approach to establish the intrinsic complexity of our problems, that is, we compare them to other well known complex problems.

Definition 20. *Given any two decision problems π and π', and being D_π and $D_{\pi'}$ their respective sets of instances, we call* polynomial reduction *of π' to π ($\pi' \preccurlyeq \pi$) to any function $f : D_{\pi'} \to D_\pi$ of polynomial complexity, such that for all $d \in D_{\pi'}$, it fulfills that $d \in Y_{\pi'}$ if and only if $f(d) \in Y_\pi$.*

The existence of a polynomial reduction from π' to π ($\pi' \preccurlyeq \pi$) means that if anyone develops an efficient algorithm to find solutions to any instance of π, through the reduction process it can be used as the kernel to construct an efficient algorithm to find solutions to any instance of π'.

Definition 21. *Given a problem π we say that it is* NP-Hard *if and only if for all $\pi' \in NP$ it holds $\pi' \preccurlyeq \pi$. When besides this it holds that $\pi \in NP$, we say π is* NP-Complete *or* NP-C.

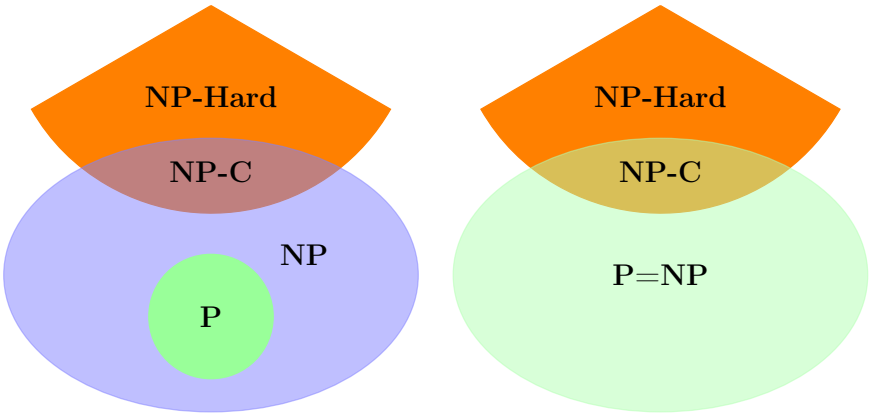


Figure 2.12: Inclusion relationship amid complexity types

The inclusion relationship amid different complexity types of problems is represented in Figure 2.12. The representation on the left corresponds to the case where $P \neq NP$, the right one to $P = NP$.

On 1971 Stephen Cook proved (Cook's theorem) that the boolean SATisfiability (SAT) problem is *NP-Complete*, that is: the SAT is as hard as any other problem of *NP*. An yet on 1972 Richard Karp started the construction of a list of polynomial reductions of SAT to other *NP* problems, thereby showing that all of them are *NP-complete* (Karp's 21 NP-complete problems).

Today, the standard procedure to prove that a problem $\pi \in NP$ is *NP-Complete*, consists in finding a well known *NP-complete* problem (π') and a polynomial reductions from π' to π (i.e. prove that $\pi' \preccurlyeq \pi$). Hence, the transitivity of "$\preccurlyeq$" guarantees that: $SAT \preccurlyeq \pi$. Complementarily and since SAT is the hardest NP problem (Cook's theorem), both complexities are equivalent and π is *NP-Complete* too.

It is worth pointing out that the previous procedure guarantees π is *NP-Hard*, even if we cannot prove $\pi \in NP$.

As we mentioned at the begging of this subsection, many important problems are known to be in *P*. However there many other important problems that are *NP-Complete* or even *NP-Hard*. Just to mention some them:

i) Determining if there is a cut-set of certain size for a graph $G = (V, E)$ is *NP-Complete*. So it is finding a "Maximum Cut Set", because this can be recursively achieved starting by $|E|$ and decreasing size by one at each step.

ii) Counting how many "Minimal Cut Sets" of certain size are embedded into a graph is *NP-Hard*. Counting how many cycles are embedded in it, is in general *NP-Hard* too, because of the relationship between bonds and cycles for planar graphs (see Lemma 2).

iii) The "Number Partitioning Problem" is *NP-Complete*. That is: given a set of nonnegative integers, determine whether there is division of it into two subsets such that the sums of numbers in each one match.

iv) Given a weighted graph, finding a subgraph of it of a given cost with connectivity constraints to fulfill between nodes is *NP-Complete*, so is finding the minimal cost subgraph with integer costs. This covers: MW2CSN, STNSNP and GSP problems. When costs are real numbers, finding the minimum cost subgraph turns *NP-Hard* for all these problems.

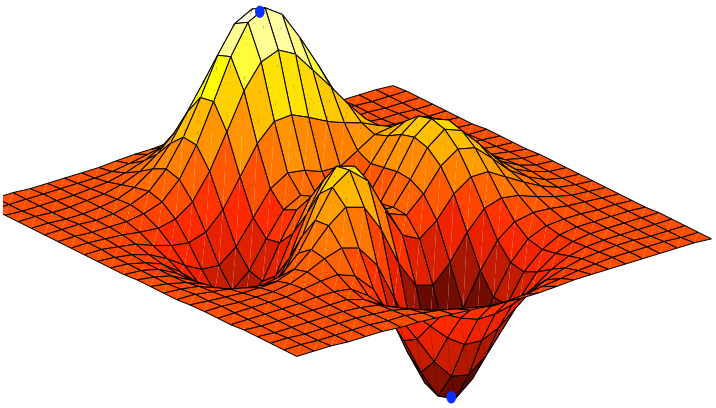
v) The "Travelling Salesman Problem" (TSP) is *NP-Hard* in general. That is: given a list of cities and distances between each pair of them, what is the cycle of lowest length that spans all cities. If distances are integer numbers TSP turns *NP-Complete*.

vi) A *vertex cover* of a graph is a subset of vertices such that each edge of the graph is incident to at least one vertex of the subset. The problem of finding a minimum vertex cover (with the minimum number of vertices) is a classical *NP-hard* problem, while its decision version: "finding whether there is a vertex cover of certain size", is one of the Karp's 21 *NP-complete* problems.

Definition 22. *Given a domain X for a set of n variables (e.g. $X = \mathbb{R}^n$, $X = \mathbb{N}^n$ or $X = \{0,1\}^n$), an objective function $f : X \to \mathbb{R}$ and a set of m constraints to be fulfilled $g : X \to \mathbb{R}^m$, an optimization problem consist in finding $\bar{x} \in X$, such that $f(\bar{x})$ is the minimal (or maximal) value of f while $g(\bar{x}) \leq 0$.*

For example, the surface in Figure 2.13 represents a hypothetical instance for a two variables problem of a generic (P) optimization problem. The goal is finding the points marked with blue dots in the figure.

On 1984 Narendra Karmarkar proved that when $X = \mathbb{R}^n$, g is a linear function and f is also linear (or even quadratic), the problem (P) can be solved in polynomial time. So Linear Programming (LP) problems are computationally easy to solve (i.e. they are in P class).



$$(P) \begin{cases} \min f(x) \\ g(x) \leq 0 \\ x \in X \end{cases}$$

Figure 2.13: A generic optimization problem representation

However, when $X \subseteq \mathbb{Z}^n$ the problem (P) is hard in general, even when f and g are linear. We call an optimization problem as a *combinatorial optimization problem* when all variables are of integer type.

This extends to cases where only some of the variables are of integer type whereas the others are real numbers. These problems are called Mixed Integer Programming (MIP) problems. In general MIP problems are hard to solve. To justify complexity it suffices to point out that most problems listed previously (i.e. GSP, TSP et al) can be written like a MIP problem.

For further information about these matters we recommend [Garey 1979].

2.1.3 Fundamentals of Metaheuristics

As we have seen in Section 1.3.2, most multi-layer works make use of classical optimization tools and iterative methods to solve LP or MIP problems (simplex, linear relaxation, branch-and-cut, etc). Iterative methods were designed with aim to find an optimal/exact solution, but as we saw in Section 2.1.2, for some sort of

problems (*NP-Hard* problems) this strategy is not efficient to find solutions for large instances (see Figure 2.1).

As we'll see later on (Section 3.1, Section 3.2) both of our models lie upon the category of *NP-Hard* problems. So, unlike former works, we decided to apply metaheuristics as the strategy to find good quality solutions.

Compared to classical optimization methods, metaheuristics do not guarantee that a globally optimal solution can be found. Instead, they attempt to construct good quality solutions in low computation time.

Most literature on metaheuristics is experimental in nature, describing empirical results based on computer experiments with the algorithms. Along this work and in accordance with usual practices, several heuristics and metaheuristics were used to solve instances, e.g.: Genetic Algorithms, Tabu Search, Variable Neighborhood Search, Simulated Annealing and Greedy Randomized Adaptive Search Procedure (GRASP) among others. Into this document we only elaborate on those that achieved best results: Genetic Algorithms with Tabu Search and GRASP.

2.1.3.1 Evolutionary Algorithms

Evolutionary Algorithms (EAs) are non-deterministic methods that emulate the evolutionary process of species in nature, in order to solve optimization, search, and other related problems [Bäck 1997, Davis 1991]. In the last twenty five years, EAs have been successfully applied for solving optimization problems underlying many real applications of high complexity.

Algorithm 2 Schema of an evolutionary algorithm

```
1: initialize(P(0))
2: generation ← 0
3: while not stopcriteria do
4:    evaluate(P(generation))
5:    parents ← selection(P(generation))
6:    offspring ← variation operators(parents)
7:    newpop ← replacement(offspring, P(generation))
8:    generation ++
9:    P(generation) ← newpop
10: return best solution ever found
```

The generic schema of an EA is shown in Algorithm 2. An EA is an iterative technique (each iteration is called a *generation*) that applies stochastic operators on a pool of individuals (the population P) in order to improve their *fitness*, a measure related to the objective function.

Every individual in the population is the encoded version of a tentative solution of the problem. The initial population is generated by a random method or by using a specific heuristic for the problem. An evaluation function associates a fitness

value to every individual, indicating its suitability to the problem. Iteratively, the probabilistic application of *variation operators* like the *recombination* of parts of two individuals or random changes (*mutations*) in their contents are guided by a selection-of-the-best technique to tentative solutions of higher quality.

The stopping criterion usually involves a fixed number of generations or execution time, a quality threshold on the best fitness value, or the detection of a stagnation situation.

Specific policies are used to select the groups of individuals to recombine (the *selection* method) and to determine which new individuals are inserted in the population in each new generation (the *replacement* criterion). The EA returns the best solution ever found in the iterative process, taking into account the fitness function considered.

Parallel implementations are popular options to improve the efficiency of EAs. By using several computing elements, Parallel Evolutionary Algorithms (PEAs) allow reaching high quality results in a reasonable execution time even for hard-to-solve optimization problems.

Three main paradigms have been proposed in the related literature to design parallel EAs, regarding the criterion used for the organization of the population [Alba 2002, Alba 2013]:

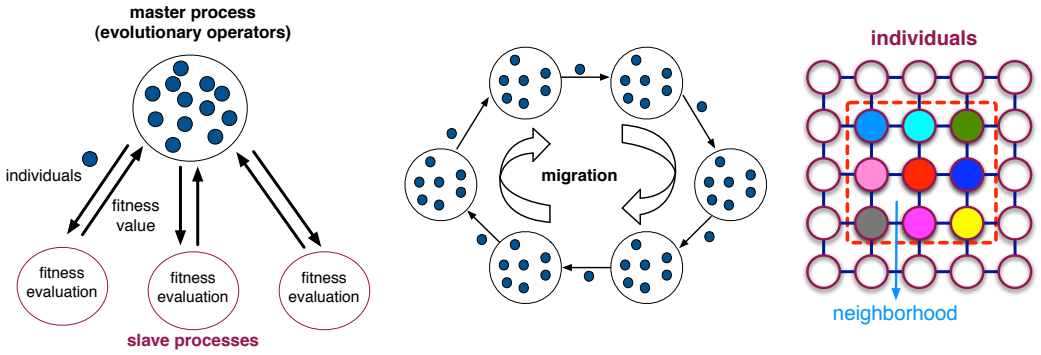


Figure 2.14: Parallel models for EAs

- The *master-slave* model follows a functional decomposition of the EA. The evaluation of the fitness function is the main candidate to perform in parallel when solving optimization problems, since it usually requires larger computing time than the application of the variation operators.

  Thus, a master-slave parallel EA is organized in a hierarchic structure: a master process performs the evolutionary search, while it controls a group of slave processes that evaluate the fitness function (see left of Figure 2.14).

- The *distributed subpopulations* model splits the global population in several subpopulations (also called *demes*), separated geographically from each other.

Each deme runs a sequential EA, and the individuals are able to interact only with other individuals in the deme.

An additional *migration* operator is defined: occasionally some selected individuals are exchanged among subpopulations, introducing a new source of diversity in the EA (see middle of Figure 2.14).

- The *cellular* model considers an underlying spatial structure for the individuals in the population, most usually a two-dimensional toroidal grid. The interactions in the evolutionary search are restricted only to neighboring solutions. The propagation of good characteristics in the solutions follows the *diffusion* model, gradually spreading through the grid.

  The limited interaction between individuals is useful to provide diversity in the population, often improving the efficacy of the evolutionary search (see right of Figure 2.14).

The evolutionary algorithm used in this work to solve the the problem presented in Section 3.1 follows the *distributed subpopulation model*.

2.1.3.2 Greedy Randomized Adaptive Search Procedure

The Greedy Randomized Adaptive Search Procedure (GRASP) is a well known metaheuristic, which has been applied for solving many hard combinatorial optimization problems with very good results [de Aragão 2001, Festa 2004, Mavridou 1998, Martins 2000, Pardalos 1999, Rosseti 2001, Resende 1998a, Resende 1998b, Ribeiro 2002]. Extensive surveys of the associated literature are given in [Feo 1995, Resende 2003, Festa 2004].

Before describing the main ideas of GRASP, we formulate a generic combinatorial optimization problem based on the description introduced in [Resende 2003]. Let us consider:

i) $N = \{n_1, \ldots, n_m\}$ is the finite basic set containing the potential elements which will be able to integrate a feasible solution.

ii) F denotes the set of feasible solutions satisfying: $F \subseteq 2^N$.

iii) $f : 2^N \rightarrow \mathbb{R}$ is the objective function. Without losing generality, we assume the minimization version, i.e. the aim is to find a global optimal solution $S^* \in F$ such that $f(S^*) \leq f(S), \forall S \in F$.

These points will be determined, when specifying the optimization problem to be studied. For example, in the case of the Minimum Vertex Covering Problem:

- $N = \{v_1, \ldots, v_n\}$ is the set of nodes to be considered,
- E is the set of edges connecting the nodes of N,

- F is composed of all the subsets of N such that when $S \in F$ any edge in E has at least one endpoint in S,

- $f(S)$ is the number of nodes belonging to S.

A GRASP is an iterative process, where each iteration consists of two phases: *construction* and *local search*. The *construction phase* builds a feasible solution, whose neighborhood (in some sense to be defined when adapting the method to each specific problem) is explored during the second phase, looking for an improvement. The best solution over all GRASP iterations is returned as the result.

Algorithm 3 GRASP pseudo-code

Procedure GRASP($ListSize$,$MaxIter$,$Seed$**):**

```
1: Read_Input_Instance();
2: for k = 1 to MaxIter do
3:    InitialSolution ← Construct_GRSol(ListSize, Seed);
4:    LocalSearchSolution ← Local_Search(InitialSolution);
5:    if cost(LocalSearchSolution) < cost(BestSolutionFound) then
6:       Update_Solution(BestSolutionFound, LocalSearchSolution);
7: return BestSolutionFound;
```

We describe now a generic GRASP implementation, whose pseudo-code can be seen in Algorithm 3. This generic implementation serves as a template to be mapped into the problems introduced in Section 4.2, where specific GRASP methods customized to our problems are proposed.

The GRASP heuristic has three main parameters: the number of iterations $MaxIter$, the candidate list size $ListSize$, and a third implicit parameter, the initial seed $Seed$ for the pseudo-random number generator. The first parameter corresponds to the number of iterations in the outer loop of the algorithm. The second parameter will be seen in more detail when explaining the construction phase, but roughly speaking, it is a measure of how many alternatives will be taken into account at each constructive step.

In some GRASP versions the size of the restricted candidate list is recomputed dynamically (i.e. the value of $ListSize$ is not fixed), being used in this case a threshold parameter denoted by α. Later on, we explain in detail both variants.

Looking again at the pseudo-code, it can be seen that GRASP iterations are carried out between lines 2 and 6. Each GRASP iteration consists of the construction phase (line 3), the local search phase (line 4) and, if necessary, the solution update (lines 5-6).

In the construction phase, a feasible solution is built. Algorithm 4 shows a generic pseudo-code for the construction phase. The solution is usually represented as a set of elements (the precise meaning of these elements depends on the specific

problem); the construction phase starts from an empty set and iteratively adds an element until the set corresponds to a feasible solution.

At each step of the construction phase, a restricted candidate list (denoted by RCL) is determined by sorting all non already selected elements with respect to a greedy function that measures the (myopic) benefit of including them in the partial solution. In general, this function evaluates the incremental increase in the cost function $f(\cdot)$ when incorporating each new element into the solution under construction. Specifically, by applying this function, we build the RCL containing those elements whose addition to the current partial solution induce the smallest incremental costs (this is the greedy component of GRASP).

Algorithm 4 Pseudo-code for procedure Construct_GRSol (Construct Greedy Randomized Solution)

Procedure Construct_GRSol($ListSize$,$Seed$**):**

1: $Solution \leftarrow \emptyset$;
2: Incremental costs evaluation for the candidate elements;
3: **while** not_feasible($Solution$) **do**
4: $RCL \leftarrow$ the restricted candidate list;
5: $s \leftarrow$ select randomly an element from the RCL;
6: $Solution \leftarrow Solution \cup \{s\}$;
7: Incremental costs revaluation;
8: **return** $Solution$;

The next element to be included into the partial solution is randomly chosen (uniformly or in some biased form) from the RCL (this is the probabilistic component of GRASP). In this way, GRASP allows for different solutions to be obtained at each GRASP iteration. When the chosen element is added to the partial solution, the benefits associated with every element not yet added to the partial solution are updated in order to reflect the change induced by the insertion of the new element. Thus, the heuristic recomputes the RCL and reevaluates the incremental costs (this is the adaptive component of GRASP). Once the construction phase is finished, the solution built is returned.

The solutions generated by the construction phase are not guaranteed to be locally optimal with respect to simple neighborhood definitions. Hence, it can be beneficial to apply a local search to attempt to improve each constructed solution. A local search algorithm works in an iterative fashion by successively replacing the current solution by a better one taken from its neighborhood. It finalizes once there is no better solution found in the neighborhood.

Algorithm 5 shows a generic pseudo-code for the local search phase. It has as input a feasible solution $Solution$ and searches for a better solution within a neighborhood $N(Solution)$ previously defined. In most of the cases, the local search phase takes as entry the feasible solution $Solution$ delivered by the construction

phase, but for certain applications, we could have several local search phases working in a combined form by exploring different neighborhoods, implying thus that their entries are not necessarily the solutions given by the construction phase.

Algorithm 5 Local_Search pseudo-code

Procedure Local_Search(*Solution*):

1: **while** not_locally_optimal(*Solution*) **do**
2: Find $Neigh\_Sol \in N(Solution)$ satisfying $f(Neigh\_Sol) < f(Solution)$;
3: $Solution \leftarrow Neigh\_Sol$;
4: **return** *Solution*;

The success when applying the local search phase is strongly related with the following points:

- the suitable choice of a neighborhood structure,
- efficient neighborhood search techniques,
- easy evaluation of the cost function when exploring the neighborhood,
- the quality of the starting solution.

The construction phase plays an important role with respect to this last point, since it must produce good starting solutions for this local search sub-procedure. Depending on the problem, the used neighborhoods are generally not complex. There exist two basic different strategies to explore a neighborhood, which are:

best-improvement: all neighbors are investigated and the current solution is (possibly) replaced by the best neighbor.

first-improvement: when finding the first better neighbor solution (i.e. whose cost value is smaller than that of the current solution), the current solution is replaced by this one.

In [Resende 2003], the authors mention that empirically (when applying both strategies on many applications), in most of the cases, both strategies reach the same final solution, but in general the *first-improvement* takes a smaller computational time. Besides, they observe that is more frequent the premature convergence to a non-global local optimum by using *best-improvement* than *first-improvement*.

One important characteristic of GRASP is its low parametrization; few parameters need to be set and tuned. This implies that the main effort can be focused on implementing efficient data structures to obtain fast iterations. Let us analyze the influence of the GRASP parameters and the RCL construction.

A GRASP algorithm finalizes once performed $MaxIter$ iterations. Clearly, the probability of finding a new solution improving the currently best one decreases with the number of iterations already computed, the quality of the best solution found

may only improve with the latter. In general, the computation times from iteration to iteration are relatively similar, therefore the total computation time depends linearly on $MaxIter$. Thus, when increasing $MaxIter$, the global computation time will be increased as well as the probability of finding better solutions.

At any GRASP iteration, let us denote by $c(e)$ the incremental cost associated with the insertion of element $e \in E$ into the solution under construction and by c_{min} and c_{max} the smallest and the largest incremental costs respectively. There are two main variants to compute the RCL used in the construction phase. Next, we shall describe both approaches.

i) Given a positive integer $ListSize$, the RCL is composed of the $ListSize$ elements of E with the best (i.e. smallest) incremental costs. In this case, we say that the RCL is cardinality-based. The size of the RCL can be smaller than $ListSize$ since, depending on the instance, we could not get to compute exactly the $ListSize$ best elements.

ii) The second variant uses a threshold parameter denoted by $\alpha \in [0, 1]$. In this case the size of the RCL is dynamically adapted according to the quality of the elements to be added (we say that the RCL is value-based). Fixed α, the RCL is formed by all "feasible" elements $e \in E$ which can be inserted into the partial solution under construction without losing feasibility and whose quality is superior to the threshold value; that is to say:

$$e \in \text{RCL} \Leftrightarrow c(e) \in [c_{min}, c_{min} + \alpha(c_{max} - c_{min})].$$

If we set $\alpha = 0$ the resulting algorithm is purely greedy, and with $\alpha = 1$ we obtain a random construction. Hence, we can infer that α regulates the amounts of greediness and randomness in the construction phase.

For further details of GRASP the reader may consult the references [de Aragão 2001, Feo 1989, Feo 1995, Martins 2000, Resende 2003, Ribeiro 2002], which provide an extensive analysis of the GRASP metaheuristic based on many applications. Topics discussed include: successful implementation techniques, parameter tuning strategies, alternative solution construction mechanisms, techniques to speed up the local search, reactive GRASP, cost perturbations, bias functions, memory and learning, local search on partially constructed solutions, hashing, filtering, implementation strategies of memory-based intensification and post-optimization techniques using path-relinking, hybridizations with other metaheuristics and parallelization strategies.

2.2 Technical background

At this point we summarize several technical concepts and preconditions that were surveyed during first steps of the analysis and constitute the foundations of models described in Section 3.1 and Section 3.2.

2.2.1 Network components

Although from the clients point of view the network looks like a whole, engineers and network designers disaggregate networks into components. This decomposition relies on the function and/or the technology of different components.

The most basic categorization of network elements is into: *access* and *backbone*.

Access - The access infrastructure is responsible of hauling the traffic of clients from their premises (fixed services) or dynamic positions (mobile services), to a POP of the service provider network.

Backbone - The backbone comprises the infrastructure responsible of moving traffic among POPs as well as delivering it to other networks.

Those backbone nodes that are only connected to other backbone nodes compose the *core* of the backbone. The complement, that is: the set of nodes also connected to access nodes, is referred to as *backbone's edge*, because they constitute the frontier before entering the backbone.

Historically from the telephony service point-of-view, the *access network* comprised the local-loop, while transport/transmission and telephone exchange compounded the backbone[1] (see Figure 2.15). An analogous scheme might be detailed for cable operators.

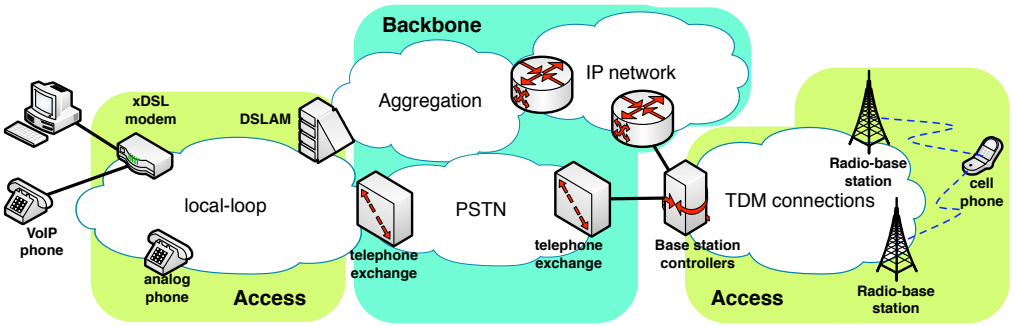


Figure 2.15: Access and backbone portions of a network

This architecture was inherited by cellular telephony, although in this case the access network added some active components (radio-bases, controllers) to handle the handover process behind a user moving from one radio-base to another.

When broadband Internet services were launched, additional elements were integrated to access (e.g. Digital Subscriber Line Access Multiplexers) and backbone (aggregation and the Internet network itself), but except for the appearance of overlays (Section 1.2), the underlying abstract structure remained unchanged.

Nowadays many companies are replacing cooper twisted-pairs of their local-loops by optical fibers (FTTH). Despite that the previous specialization persists. We focus

[1] Although they weren't yet referred to as so.

this work on backbone design. It is a fact the within a backbone, Internet traffic coming from broadband fixed services is by far the most important in volume[2].

Models developed during this work are inspired in real-world application cases. They are analyzed into detail in Chapter 5 while now we summarize overall percentages and characteristics that fundament definitions and complement the analysis.

Within ledgers of our reference TELCO the access network is the most important piece of capital. Over scenarios analyzed for ANTEL the access portion of the network comprises assets always placed over 66% of the network's total[3]. Only the remaining corresponds to backbone and yet we agreed to optimize this one.

Several reasons sustain this decision for ANTEL:

i) Major portions of access networks were deployed long ago and are now repaid. Just as was described for optical fibers and related investments on conduits, local-loop deployment was a huge budget and time consuming process, carried out on times when ILECs constituted monopolies.

   Later on, technologic evolution (xDSL, DWDM) allowed ILECs to get additional revenues from already installed assets, but companies are reluctant to further changes on their physical infrastructure;

ii) Physical access networks are stiff, with little margin to change. For instance, after simulating different strategies to connect DSLAMs to backbone[4] total access costs only decreased about 3%. Over the same scenarios, extreme costs for the backbone spread a range of 147%. So backbone optimization is much more cost-effective, in relative and absolute terms;

iii) The cost of the access network is important for ANTEL, because dedicated resources are assigned to each user for massive scale services. When access costs apply to nonresidential applications, backbone costs become relatively more important. Regarding this final point and to reinforce this premiss, we remark that in our second real-world application (RAU), access costs only represent from 5% to 13% of the total. So basic hypothesis are consistent in both cases. It is worth pointing out that this is sustained by additional efficiency in the access connections of RAU and not by issues coming from the design of ANTEL's access network.

   Physical access resources are provisioned to fulfill peak traffic committed with customers[5], which in ANTEL's case are residential subscribers, i.e., hundreds of thousands of dedicated access connections dimensioned on a per-customer

[2]The sum of traffic generated by phone calls (fixed lines and cellular phones) and mobile Internet devices is under the 10% of the total.

[3]Although these values were after the optimization, so the actual backbone's percentage is probably higher.

[4]The only alternative with some degree of freedom/configurability.

[5]If access connections were dimensioned under this value, customers would never reach what is committed on service contracts.

peak basis. Backbone resources on both cases and access connections in RAU's project (many centers count hundreds of hosts) summarize a large number of users and are designed to comply with certain *statistical performance*. Capacities are provisioned assuming that only a fraction of the potential users would be using the service at any moment. This *overbooking factor* is applied by all ISPs all over the world and its value usually ranges between 20 and 200.

In other terms, if instead of selling hundreds of times the capacity installed, the backbone of ISPs were designed to fulfill with that state where all customers are transmitting simultaneously, the cost of the access network would be far below the 10% of the total.

Besides access and backbone division, the backbone itself can consist of several components with specialized functions. For instance, the network of RAU holds a backbone of a single piece in which traffic from users can flow between any pair of nodes if necessary. ANTEL's network on the other hand, holds two network portions: *aggregation* and *public IP* networks. Access nodes are connected to aggregation backbone and over it are conducted to the public IP backbone (see Figure 2.16). It is only within this last network where traffic among customers is routed.

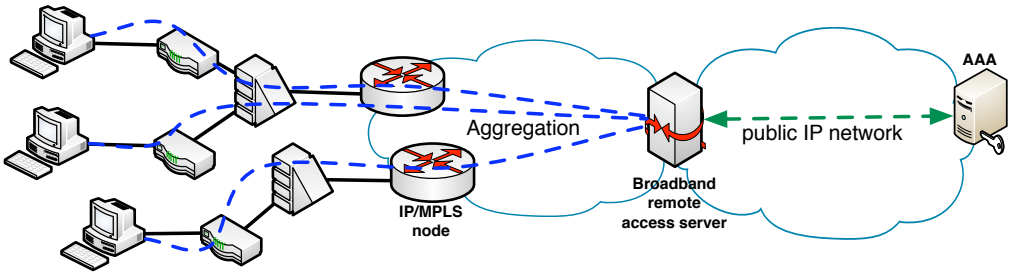


Figure 2.16: Aggregation and public portions of a backbone

The juncture function between both networks is carried out by Broadband Remote Access Servers (BRASes). This architecture may look unnatural and inefficient but it is yet widely used because mixes convenient operative improvements. BRASes combined with Authentication Authorization and Accounting (AAA) servers allow the record of: the IP assigned to each customer at any moment[6], differentiated/selective accounting and/or service profiles for individual customers, as well as centralized inspection of the traffic coursed by users.

2.2.2 IP/MPLS technology

As we commented on Chapter 1, DWDM implements local protection mechanisms (circumscribed to each ring) equivalent to those of SDH/SONET. Thereby suffers from the same lack of efficiency, i.e., to protect all connections, 50% of the capacity of each link must be reserved as *spare capacity*; instead, protection schemes covered by models of this work are sustained on mechanisms of the IP layer.

[6]Fundamental to track computer security incidents.

2.2.2.1 Pure IP networks

In IP technology, data is sent in *packets* rather than over a continuous stream of bytes as in TDM. The basic working principle behind IP technology is the *best-effort delivery*, where network services neither guarantee that packets are delivered nor that they are treated specially in terms of performance/quality-of-service (e.g. delay, jitter or packet loss). Despite this apparent lack of commitment, the network is actually entrusted to do as best as it can with the means at its reach. These capabilities have been evolving over decades and are now a competitive alternative for all applications.

Whenever a router receives an input packet onto one of its interfaces performs a *lookup* to find out the best-known destination for it, and to determine the corresponding output interface. On this basis the packet is then delivered to the next router, hoping that eventually it will reach the destination. The process by which packets are moved between ports of a node is called *forwarding*, whose implementation constitutes what is called the *data plane* of a router.

Naturally, forwarding decisions are based on rules and algorithms running as processes, which are in turn fed by information of the actual network. The process by which routers share information among them and apply rules to determine their forwarding entries is called *routing*. The set of processes performing routing functions from which routing tables are fed is called *control plane*.

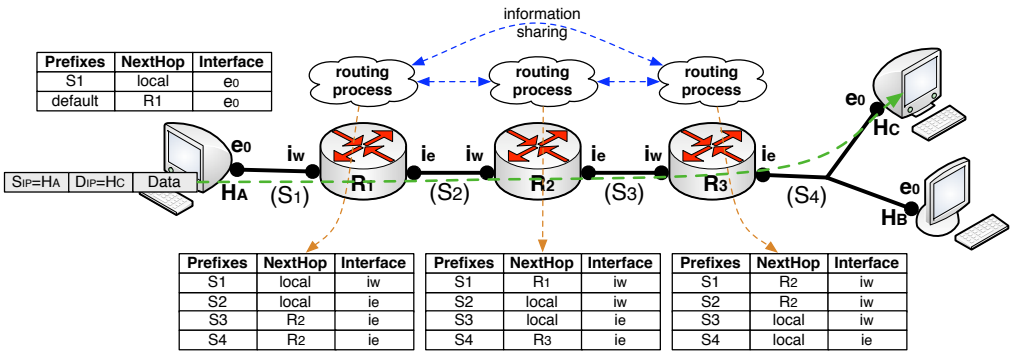


Figure 2.17: Forwarding and routing processes

A key difference between TDM and IP forwarding paradigm is the existence of *distributed* and *dynamic* "routing protocols". Flows onto SDH/SONET as well as lightpaths on DWDM are administratively pre-provisioned. On IP paradigm, all a sender needs to successfully transmit a packet is: a valid source IP assigned over an operative connection, a destination IP to where the information is meant to be sent, and a network working properly.

Figure 2.17 shows a very simple IP network. The network consists of four segments (groups of IP addresses): S_1, S_2, S_3 and S_4, and three routers: R_1, R_2 and R_3. The routing processes running at each router, share information about the segments

connected to their hosts, and up from it they construct the forwarding table of its respective router. Simple hosts -like computers- usually limit to a simple forwarding table consisting of addresses directly accessible across the interface (e_0 in this case) and a *default gateway*, which is supposed to known what to do with the rest of the addresses.

Let us suppose that an application running at the host of IP = H_A resolves to send a packet to the host of IP = H_C. Since destination IP hasn't a matching entry in its routing table, the host delivers the packet towards its default (R_1) through its unique interface (e_0). Router R_1 matches H_C within network addresses covered by segment S_4 and from it determines the best path to get to it, is across R_2. This process continues until the packet is delivered by R_3 towards its destination.

At each hop the packet moves without changing any of its attributes[7] or data into the payload. As we shall see this a vital difference between routing and switching technology: forwarding on a routed network always relies on global addresses, whereas tags/labels in a switched network only have local significance.

Routing protocols play an essential role within this strategy. A router can get to know by its own the information of its environment (e.g. interfaces, IP segments assigned to them, neighbor routers), but in order to construct a global information database, a collaborative approach is needed. This is the spirit of routing protocols: each router publishes what it is certain of, and redistributes what it has learned from the others.

As there are segmentations for network portions, there are also segmentations for routing protocols. A first categorization comes from the kind of information that routers interchange. *Distance vector* protocols are the simplest ones. As a basis they periodically interchange their routing tables with neighbors and reconstruct/update these tables accordingly. For instance, R_3 in Figure 2.17 knows is connected to segment S_3 so it tells that to R_2 which adds this entry into its routing table. In turn R_2 forwards this information to R_1. An equivalent process takes place from R_1 to R_3 to learn that S_1 is at its west. After several iterations, routing tables of all routing processes converge and the network is fully operational.

Although simple, distance vector protocols suffer from many issues from which we highlight: stability, convergence and performance. Over a more complex topology than that depicted in Figure 2.17, these algorithms can keep stalled on inconsistencies from which it would take too many iterations (too much time) to recover up. Furthermore, when there is more than one alternative to reach a destination, the suitability of one route against another -known as the *metric measure* for this entry- merely limits to count the number of hops, what tells very little about the quality of the route chosen. Routing Information Protocol (RIP) is the most popular distance vector routing protocol.

[7] Except for its Time-To-Live (TTL) field, which is decreased before forwarding, to prevent the existence of routing loops.

A more complex but clever approach is that implemented by *link state* routing protocols. Instead of their routing tables, they share information of links and nodes connected to each one. After gathering all the information of the network these protocols construct a *topology database*. In other words, for *link state* routing protocols, each process running in the network keeps a representation of a graph with: nodes, links and their costs. For instance, each router of Figure 2.11 is aware of the entire picture. Whenever a change happens in this topology, routers next to this event immediately propagate a *link state advertisement* to all routing processes of the network. So, the topology table is rapidly updated.

The mechanism to construct the routing tables is pretty direct: in order to find the minimal path toward each possible segment, processes run Dijkstra's algorithm over the instance determined by their *shared topology database*. These routing tables remain static until a topology change occurs. Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (ISIS) are the most widely used link state routing protocols. Unlike distance vector protocols, link state ones run the same algorithm over a common instance (the topology database), so it is expected that each router constructs a routing table consistent with those of the rest.

All protocols mentioned up to this point have scalability issues. It is not conceivable that a flapping interface of a router placed at any point in Internet, triggers events that impact to the rest of the network and force the re-computation of the entire routing tables over the whole planet. This concern extends to issues coming from mistakes caused by either accidental or malicious human intervention. Certain level of isolation among administrative domains is necessary in order to assure a minimal stability to a global infrastructure like Internet.

Global Internet is an aggregation of *network isles*. Each ISP is responsible of what happens within its network, which is assigned with a unique number that identifies this Autonomous System (AS) globally. Some routers from different ASes are connected with peers of other ASes to share routing information. These routers are known as *border routers*. To share routing information between them, they make use of a: specialized, highly-scalable routing protocol known as Border Gateway Protocol (BGP).

The left part of Figure 2.18 sketches the interconnection scheme for a few South-American ISPs. In the routing system of the Internet there are over 42.000 ASes coexisting, which summarize over 500.000 BGP network entries. The right half of Figure 2.18 shows a visualization of routing paths over the actual Internet, where each point corresponds to an ISP, and the lines to peering instances.

Border routers are a good example of routers that need to run more than one routing protocol. In order to be useful, information fed from BGP peers needs to be redistributed to routing protocols running inside the AS network. According on their efficiency, reliability and function, protocols have different priorities. When an entry to the same group of addresses appears in more than one routing table, only that entry coming from the highest priority protocol enters to the forwarding table.

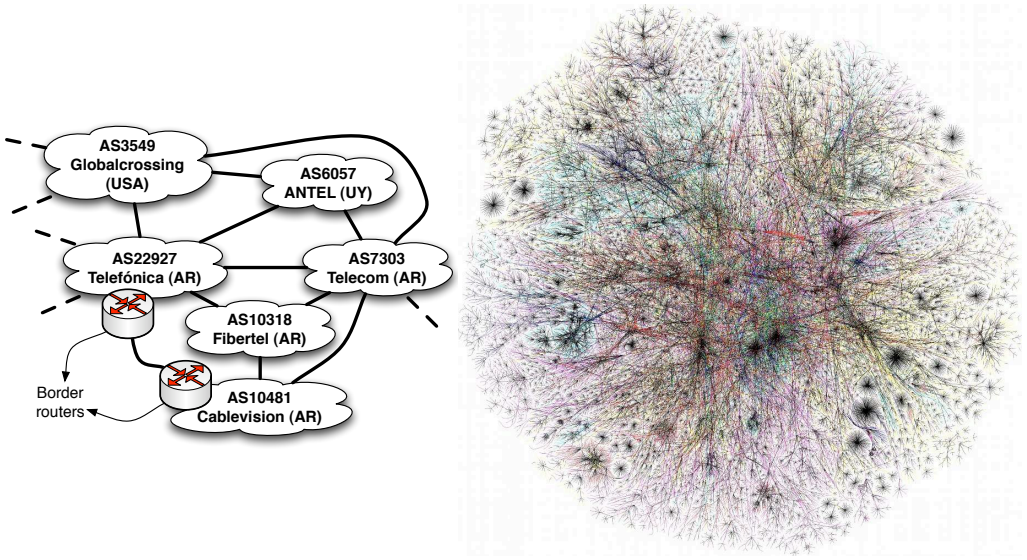


Figure 2.18: Autonomous Systems of Internet

The truth is that although flexible and scalable, BGP has a long way to go before bringing support to optimization matters. Nowadays, optimization is sustained in internal network capabilities and upon a proper network design. Internal routing relies on Interior Gateway Protocols (IGPs): RIP, OSPF and ISIS are typical examples of IGPs. This categorization is complementary to that of *distance vector* or *link state* protocols. BGP is a highly configurable and scalable distance vector protocol, but it converges too slowly and it is not equipped with fine mechanisms to control traffic flow over networks.

So, IGP protocols used on important networks are often of the link state family. This work focuses on the optimization of the internal structure of the network of an ISP. Thereby, we only go further into details of link state IGPs.

2.2.2.2 Extensions to IP technology

On early stages of Internet expansion the previously described state of technologies was more than enough to keep the network up and running. Although IP architecture was indeed scalable, it lacked from the finesse to optimally utilize the resources of networks, particularly in the ISP's backbones. This was an important issue because traffic volume was growing exponentially. Another pressing issue was the need to reduce the number of networks, i.e., to converge into a multi-service network capable of serving all kinds of traffic: Internet, VPNs and also multimedia. This transmutation required improvements on the best-effort paradigm to turn it into a Quality-of-Service (QoS) aware network.

First of all, traffic of different classes should be treated differently. When a file is being downloaded from Internet the overall performance is not significantly affected by an extra delay in the delivery of some packets. Furthermore, certain rates of

packet loss are well tolerated by applications and transport protocols. On the other extreme: delay, jitter and packet loss must be kept under strict boundaries to sustain a good quality for voice on a phone-call.

The *differentiated services* (or diff-serv) architecture is the most successful mechanism for classifying and managing network traffic. It consists in marking packets according on applications' needs and configure "diff-serv aware routers" to implement *per-hop behaviors* (PHBs), which define the packet-forwarding properties associated with each class of traffic at each hop. There are bits within the IP header for this purpose, they are: Differentiated Services Code Point (DSCP) and Explicit Congestion Notification (ECN). Different PHBs are defined to offer: low-loss, low-latency, high-priority or other treatments.

This architecture performs correctly when marking of packets is placed next to points where packets are originated, and when PHBs are coordinated all along the way packets flow towards their destinations. The coordination between *priorities* over different *queues* and the *scheduling* for dequeuing packets into output interfaces are classical applications of *queueing theory*, out of the scope of this work.

Complementarily to the design of optimal diff-serv policies, a network requires certain amount of provisioned resources to work properly. However, capacity -by itself- is not sufficient to avoid congestion and guarantee quality of service; *traffic engineering* is necessary in conjunction with capacity planning. Traffic engineering consists in establishing paths to be followed by packets over a network.

It is immediate that to implement traffic engineering (or traff-eng) is necessary to know in advance the topology of the network, so traff-eng requires a level of information detail comparable to that used by link state protocols to tackle down Shortest Path First (SPF) problem (solved by Dijkstra's algorithm). Nonetheless the approach followed by OSPF and ISIS is not sufficient to avoid congestion.

"The Fish Problem in Routing" (depicted in Figure 2.19) is a classical problem of Shortest Path First (SPF) routing. Its name derives from the shape of network, which resembles a fish with R_1 being the head and R_7 and R_8, the tail of the fish. Let us guess that costs of links are those represented in the left figure, and a packet is sent from R_7 towards R_1. According on these costs, packets will follow the path highlighted with a dashed-blue curve. Problem is that this path will be used by any packet arriving to -or originated at- R_6 with R_1 as its destination. This is because routers make a local decision, based on what each thinks is the optimal path from its perspective. Since all routers run the same SPF algorithm, with the same link state database, they all come up with the same shortest path, which very soon turns congested, while the non-shortest path remains idle/unused.

The right scheme of Figure 2.19 shows that changing the costs doesn't amend the problem but only inverts the idle/congested condition. The truth is that whether the routing decision is based solely on the destination address of the IP packets, this problem cannot be avoided in general. *Policy routing* is a mechanism that

integrates rules other than the destination IP to determine the next hop for packets. There are examples where policy routing helps to improve network quality, but they are based on small instances. Over a network of important size, coordinating the configurations of routing processes for policy routing to work fine, is a challenging task that may turn operations into a nightmare and shatter down to pieces the actual IP scalability.

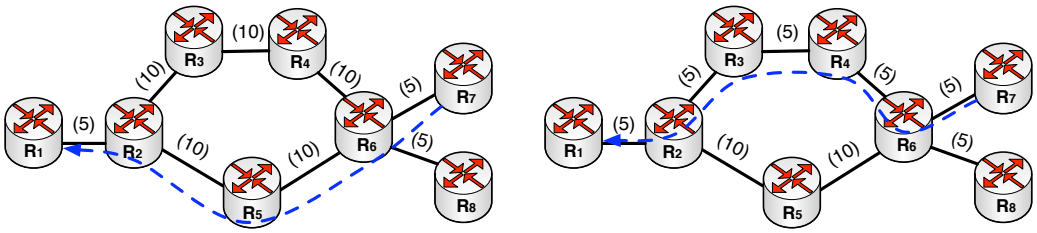


Figure 2.19: Example of the fish problem

The de-facto solution to control paths of IP traffic within a network is MultiProtocol Label Switching (MPLS). The working principle of MPLS is that forwarding is based on local labels/tags rather than on global IP addresses. Once a node pushes a packet as the payload of an MPLS frame with certain label, its path is determined regardless of the addresses in the packet. Moreover, intermediate forwarding decisions are realized without considering IP addresses at all. In this sense, MPLS resembles ATM, Frame Relay and other legacy switching protocols.

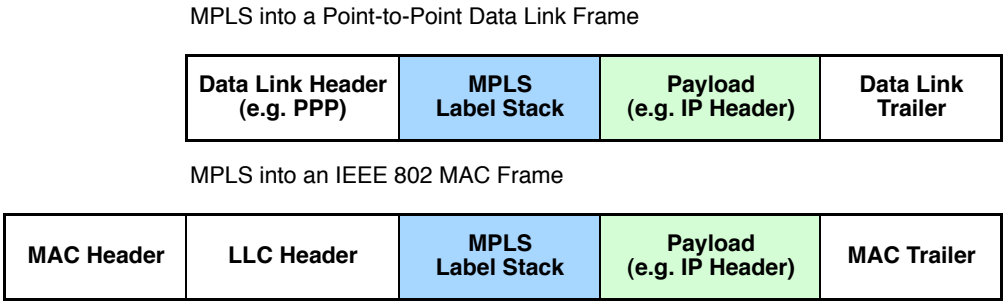


Figure 2.20: MPLS frames over different Layer 2 technologies

Furthermore, MPLS is standardized as cargo for most known connection protocols. Figure 2.20 show how MPLS can be transported over PPP in a point-to-point connections between two MPLS switches, which in turn might be implemented over an SDH/SONET stream connection. The picture also shows how is intended to be transported over Ethernet links. In fact, there are standard mechanism to transport MPLS over a wide spread of technologies (Frame Relay, ATM among others). So, an MPLS overlay can be deployed over a heterogeneous environment of underlying networks. That is a key for its success though is not the only one.

Another key factor for MPLS success is that it is an evolution of traditional IP

technology, i.e., MPLS switches are IP routers whose control and data planes were enriched with a set of complementary functionality. Actually, a decade ago many IP routers already operative could be upgraded to MPLS switches with minimal expenditures. Since they are capable of handling both: IP packets and MPLS frames, the technology is usually referred to as IP/MPLS. Regarding the classification of Section 2.2.1, nodes of this technology are designed to constitute backbones. According on its function IP/MPLS nodes are classified into: Label Edge Routers (LERs) and Label Switch Routers (LSRs). LERs are placed at the edge of the network and they receive/deliver packets from/towards clients. LSRs constitute the core of IP/MPLS backbones; they only switch frames coming from LERs or other LSRs.

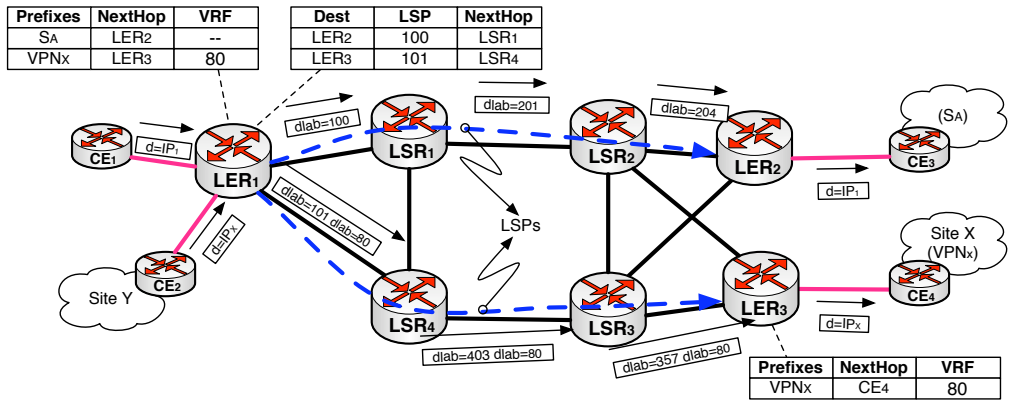


Figure 2.21: Unicast IP and MPLS VPNs

MPLS frames travel along the network into *IP/MPLS tunnels* also known as Label Switched Paths (LSPs). Tunnels are defined by a sequence of labels that are swapped at each hop. For instance, when the "pure access IP router" CE_1 (Customer Equipment) of Figure 2.21 forwards an Internet packet to LER_1 with destination in IP_1, LER_1 performs a lookup into its routing table and discovers that this IP belongs to segment S_A, which is announced by IGP as adjacent to LER_2. Subsequently, the router looks into its Label Forwarding Table (LFT) an entry to get to LER_2 and finds that the corresponding label is 100 with next hop in LSR_1. So, LER_1 prepends this label to the IP packet and transmits the resulting frame towards LSR_1. After receiving this frame and finding the entry in its LSPs table, LSR_1 switches label 100 by 201 and sends the frame towards LSR_2 without inspecting frame's payload (the IP packet). Analogously, the frame is retagged by LSR_2 and sent towards LER_2, which removes the label, analyzes the destination IP, and sends the IP packet towards CE_3. Sometimes LSR_2 directly sends the untagged IP packet to LER_2, popping rather than switching labels, which minimize offloading operations of LER_2, avoiding the double lookup (labels and routing tables); this functionality is called penultimate hop popping (PHP).

Besides the label value itself (20 bits), the MPLS label contains: three EXP bits, one Bottom-of-Stack bit and eight Time-To-Live (TTL) bits. The placing of each

bit within an MPLS frame is sketched in Figure 2.22. The meaning is the following: TTL is decreased at each hop to prevent loops (as in IP); the three EXP bits are used to mark up to eight QoS classes, hence IP/MPLS switches can identify different classes of service without inspecting frame's contain. The association between the six DSCP bits and the two ECN bits of the IP header and EXP bits of IP/MPLS frame is performed by LERs, using rules administratively set by network operators, whom determine how clients' packet-marking is translated into the backbone. The Bottom-of-Stack bit indicates this label is the last on the MPLS header. An MPLS header is indeed a stack of labels manifesting nested levels of tunnels. The Bottom-of-Stack bit optimizes hardware implementation for certain operations, such as penultimate hop popping. The most common usage of nested labels is for VPNs transporting.

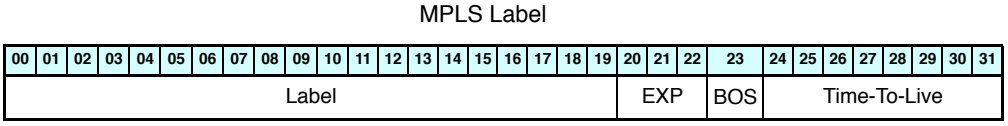


Figure 2.22: Bits usage within an MPLS label

The forwarding sequence between CE_1 and CE_3 corresponds to a shared IP addressing environment, where addresses are unique within the network, so routing tables can identify addresses segments univocally (e.g. Internet addressing). In spite of some -not yet seen- traffic engineering features, traditional IP networks are perfectly capable of accomplishing this delivery function. When instead of a hop-by-hop routing architecture, LERs make use of MPLS tunnels to send IP packets towards a destination determined up from the main/default routing table (e.g. fed from IGP information), the scheme is called "MPLS Unicast IP Forwarding" (aka unicast-ip). The two most important contributions of unicast-ip over pure-ip architectures are: i) the possibility to perform traffic-engineering on tunnels; ii) the isolation of core routers from lookups in massive IP tables. A full-routing table of Internet can contain 90,000 entries, however through unicast-ip, LSRs (core routers) only switch labels, hence lookups are onto much smaller tables.

Before going into traffic engineering, we analyze how IP-VPNs are implemented over IP/MPLS. Routers are capable of handling more than one independent routing table. One of these tables is the main/default, whose prefixes are used whenever a not more specific directive is indicated. Other tables are used for sustaining VPNs' operations. For instance, a customer with two sites: X and Y (see Figure 2.21), who transmits a packet from site Y towards X sends this packet to its default router (CE_2) as the first step. When this packet reaches LER_1 it does it into a well defined interface, which is attached to private routing/forwarding table, other than the default. Hence, LER_1 lookups into this specific table (let us guess is the 80th table) to find out that LER_3 is the router who knows how to get to VPN_X: the segment that contains IP_X. Before pushing the label to inject the packet into the appropriate LSP (101), LER_1 prepends another label of value 80 to indicate destination router to use this specific private routing space in order to determine the packet destination.

Penultimate hop popping doesn't affect this mechanism because only the top label (357) would be removed.

Virtual Routing and Forwarding (VRF) is the technology that allows multiple and independent instances of routing tables coexist within the same router at the same time. For this gets to work, it is necessary to attach routing processes with independent forwarding tables or Forwarding Information Bases (FIBs), each one applying to packets/frames entering into specific physical or logical interfaces. Through VRFs and FIBs, the same set or overlapping pools of IP addresses can be used without conflicting each other.

Sharing forwarding information among LERs is by itself a challenging issue. When a client needs to propagate routing information among its sites automatically (i.e. using a routing protocol, e.g. CE_2 to LER_1), LERs instantiate processes of traditional routing protocols (e.g. RIP) attached to client's interfaces. However, this procedure[8] doesn't scale well for LERs to share routes among them. The standard mechanism is Multiprotocol BGP (MBGP). Whereas standard BGP only supports IPv4 unicast addresses, MBGP supports IPv4 and IPv6 addresses, with unicast and multicast variants of each; it also supports the marking of routes with VPN tags, so LERs avoid mixing routes of different VRFs. Hence, besides its classical application, i.e., *sharing information among different ASes* (known as eBGP for external-BGP), BGP also can be used to share routing information within an AS (known as iBGP for internal-BGP) by selectively peering LERs, avoiding then this distribution to be carried out by IGPs, and isolating LSRs from learning unnecessary information.

An additional technical detail about technologies previously described is far beyond the purpose of this document. The only aspect where we need to go further in, is traffic engineering.

2.2.2.3 IP/MPLS traffic engineering

As we mentioned before *traffic engineering* consists in determining and setting paths to be followed by traffic over a network. Since over an IP/MPLS network most traffic (all but control data) flows across tunnels, traffic engineering reduces to mechanisms to set paths for them. Until now we haven't seen how IP/MPLS networks construct their tunnels. The most simple and yet widely used mechanism for this purpose is Label Distribution Protocol (LDP).

LDP embeds the spirit of distance vector routing protocols. It is implemented through processes running into routers, which rather than IP tables share label tables (LFTs), also known as LIBs (Label Information Bases). An LDP process feeds from IGP information and from updates coming from other LDP processes. Once launched, an LDP process imports IGP's routing table and assigns an arbitrarily label to each entry. If this entry corresponds to a local route (i.e. IGP hasn't learned it from a peer process), prefix and label constitute an entry that is published to LDP

[8] That is, running an instance of a routing process by each combination of VRFs.

processes running at neighbor routers.

When an entry for a yet unclosed label association is received from a peer process, and this process is hosted into the neighbor which is next-hop for the best known path to that prefix: the LDP process creates a new label association (completes the entry in its LIB), updates the LFIB (Label FIB) accordingly and triggers LDP updates to its peers. Eventually LDP tables converge and IP/MPLS switches fill entries to all destinations. Thereby once booted up, an IP/MPLS switch starts as an IP router and from that state evolves until it turns into a fully operational IP/MPLS switch.

For instance, let us suppose that an IP/MPLS switch is running OSPF (the IGP) and LDP on its control plane, as it is represented in Figure 2.23, and an OSPF update informs about the existence of a new prefix 10.0.0.0/8[9]. Afterwards, OSPF process runs SPF and determines from the shortest path to this prefix that the next-hop should be the neighbor of address 1.2.3.4; subsequently the entry is installed into the OSPF routing table. Immediately a contest starts among OSPF process and other routing processes to determine the most trustable entry[10], which is then installed into the Routing Table (RT) of this router and from there into the FIB. From this moment on any IP packet arriving with a destination IP belonging to the set 10.0.0.0/8 will be directed towards 1.2.3.4.

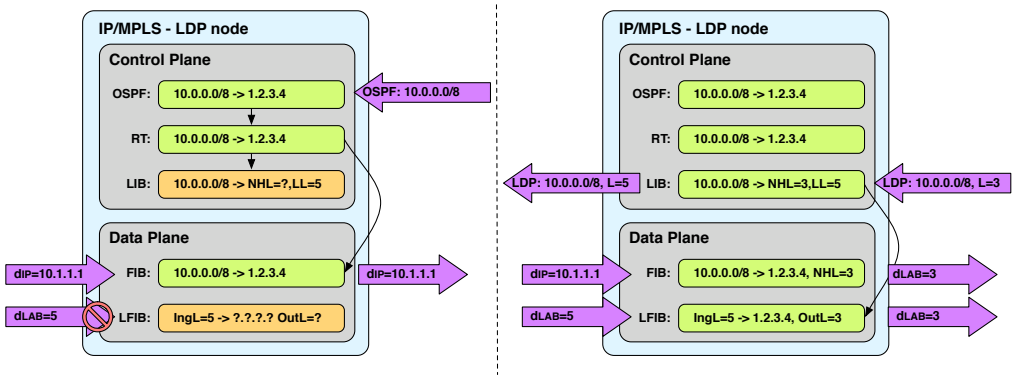


Figure 2.23: LDP incremental assignment of labels

After the new entry is installed in RT, LDP adds an entry in its LIB and assigns a Local Label (LL) of value 5 to it. However, this entry isn't valid yet because there is no known Next Hop Label (NHL) for it. So up to this point, MPLS frames coming into the router whose outer/top-of-stack label value is 5 would be discarded. This is illustrated on the left half of Figure 2.23. Later on (see right half of figure) an LDP update coming from 1.2.3.4 informs that the label to reach 10.0.0.0/8 through it has a value of 3. This closes NHL value and turns valid the entry in the LIB, which is now propagated to LFIB and sent to neighbors over LDP updates. From

[9]In IP terminology this means that only the first eight bits of the address are fixed. This spans a range of 16,777,216 addresses, from 10.0.0.0 to 10.255.255.255.

[10]That coming from the highest priority protocol.

this point on, an IP/MPLS frame with label=5 would be switched to label=3 and sent towards 1.2.3.4. Besides, whether the router is indeed a LER and receives an IP packet like before, the router pushes the packet into this tunnel by adding a label of value 3 prior to send it to 1.2.3.4.

Behind the algorithm used by LDP to construct paths for tunnels there is a premiss: "do it as the IGP does". LDP is very simple and easy to main, but replicates paths computed by IGP, so it is as exposed to traffic issues as IGPs are (fish problem). Efficient traffic engineering demands abandoning hop-by-hop paradigm.

IP/MPLS tunnels are unidirectional, so frames between a pair of nodes might go back and forth following different paths. This is unlikely for tunnels constructed by LDP (it uses IGP paths) but not for tunnels built by other mechanisms. Moreover, LDP paths determined by SPF algorithm replicate a tree-like topology from each node towards the rest (see Figure 2.11). Resource Reservation Protocol - Traffic Engineering (RSVP-TE) is a protocol for reservation of resources across an IP/MPLS network. Nodes of an IP/MPLS network can use RSVP-TE to indicate to other nodes the path and attributes (bandwidth, jitter, maximum burst, and so forth) for each IP/MPLS tunnel that departed from there.

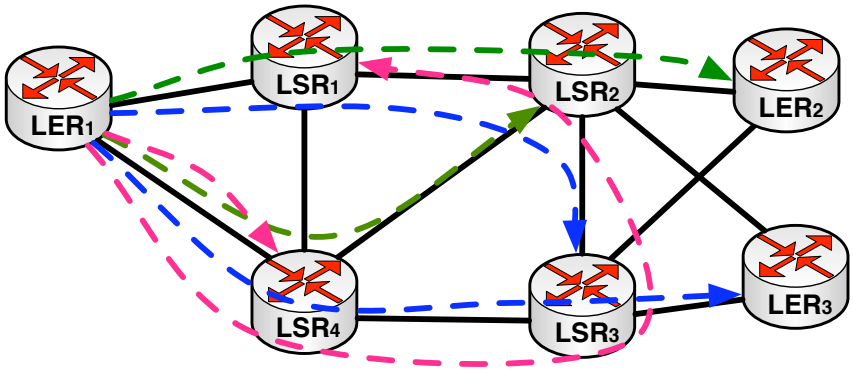


Figure 2.24: IP/MPLS tunnels with traffic engineering

With RSVP-TE any node of the backbone can determine the path to be followed by packets sent from it to the rest. Figure 2.24 sketches an arbitrarily configuration for paths of LSPs. Different colors were used just for clarity. It is worth pointing out that such a configuration for paths could never have been accomplished with LDP. For instance, if the shortest path between LER_1 and LER_2 is that marked with the green curve, then and because of the "principle of optimality", the optimal path to LSR_1 and LSR_2 would also be placed upon the same path all along, whereas in the example all paths differ.

To avoid labels inconsistencies RSVP-TE negotiates the values of each label at each hop (with each node) of the path. Besides this, RSVP-TE informs additional attributes such as the traffic volume this LSP is expected to inject. This is fundamental to coordinate capacity resources because RSVP-TE processes run

independently over nodes. In other words, processes hosted in the originating node determine paths from this node to the rest. An additional mechanism is necessary to dynamically distribute information, in order for processes running onto different nodes can coordinate their actions. The answer lies upon extensions to IGPs.

Originally, link state protocols were designed to construct a topology database where the only attribute of links to be distributed to the rest was "the cost". Recent version of OSPF and ISIS allow extending links' information with: capacity, usage, point-to-point delay and names for it (aka colors). Both flavors were renamed as: OSPF-TE and ISIS-TE, appending the Traffic Engineering (TE) suffix. With these extensions IGPs become a real-time mean for sharing traffic-engineering information. Immediately after an RSVP-TE process reserves certain capacity on a link, all the remaining routers get aware of it through an update into the topology database.

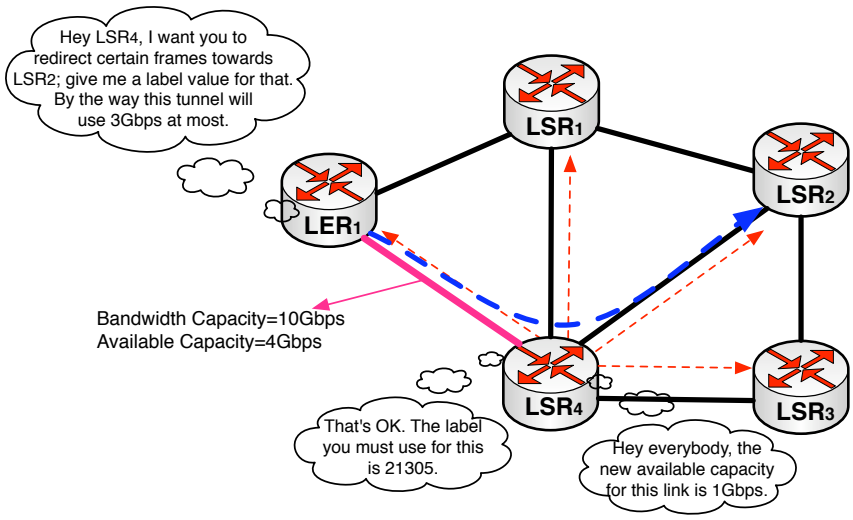


Figure 2.25: RSVP-TE and IGP-TE interaction

For instance, if this interaction would be held in terms of natural human language would look like is sketched in Figure 2.25. LER_1 has determined (later on we shall see how) that the best path for its tunnel to LSR_2 is that marked with a blue-dashed curve. It has also checked that links of this path have enough available capacity to allocate the tunnel, so negotiates labels hop-by-hop. Once labels are established intermediate nodes inform to the rest about changes through the IGP.

The algorithm used to determine these paths is a simple variant of former SPF known as Constrained Shortest Path First (CSPF). The path computed using CSPF is a shortest path fulfilling a set of constraints. The implementation simply runs Dijkstra's algorithm after pruning all those links that violate a given set of constraints. Hence SPF and CSPF have the same computational complexity. The most intuitive constraint would be *available capacity*, but other attributes are frequently used. For instance, if certain links were forbidden for a critical application, an automatic routing implementation would consist in marking these links with label "red" (coloring)

and instruct the CSPF to avoid the usage of "red links" for this path. Actually a rich combination of constraints is supported by CSPF, which is impressive since the core algorithm for computing routes is the same old SPF.

There are two technical issues still pending of further detail before closing this section. They are: how paths are bonded with LSPs and how can LSPs be protected. Both are related actually. Whenever an LSP is administratively configured, a set of one or more paths for it must be established. Almost any imaginable configuration for a path is realizable by one way or another. Some examples are:

a) A path could be picked up from LDP tables. This path would be dynamic, because it is recalculated after topology changes, to follow IGP changes. Since these computations are performed hop-by-hop, traffic engineering is not supported.

b) A path could be explicitly detailed by a sequence of hops (nodes) to be followed (e.g. n_1, n_3, n_8 using only direct links between them).

c) A path could be explicitly detailed by sequence of nodes and links. For instance: n_1, n_3, n_8 using only direct links of label "SomeLabel". Of course the label must be set as an attribute of involved links, which is operationally expensive.

d) A sequence of intermediate nodes. For instance: the path should go from here to n_1, and from there to n_3, and from there to its destination; complying with this, use any intermediate path between hops.

e) A minimal cost path constructed up from constraints, e.g., this tunnel requires 1Gbps of available capacity at each intermediate link, that's it.

The Operating System (OS) of each IP/MPLS switch tries to set-up paths for LSPs respecting the order in which they are attached. If for some reason the first path isn't available, then the OS switches to the second, eventually to the third, and so on. By adding dynamic entries in the list (computed by LDP or CSPF), it is not necessary to set-up a large number of paths for having a high level of resiliency.

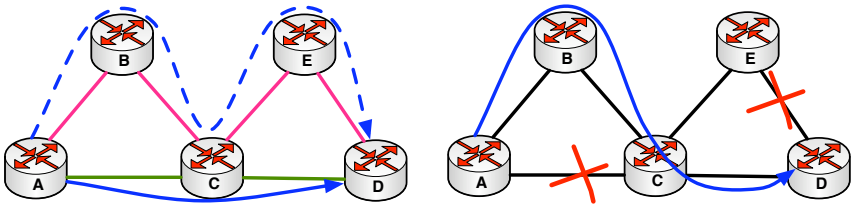


Figure 2.26: Active/standby protection and loose extensions

See for instance the left picture upon Figure 2.26. Coloring links with "green" and "purple" it would be possible to force two strict link-independent paths for the LSP that goes from A to D (blue curves). The same effect could have been achieved specifying the nodes sequence explicitly. Let us guess that A-C-D appears first in the list, A-B-C-E-D in the second place and there is no third alternative. Then

these paths would conform the primary/secondary aka active/standby paths for the LSP. This configuration is resilient to single link failures, however provisioning a *loose* (non-strict) third path alternative, where the design of an automatic route is transferred to a dynamic mechanism (e.g. CSPF/RSVP-TE) would allow recovering even from those double failures that aren't a cut-set (right half of figure).

The degree of protection and the performance achievable by combining these mechanisms is huge. IP/MPLS capabilities are far-beyond those of transport networks. This work develops and finds solutions to instances of two extreme models:

i) The first (Section 3.1) corresponds to a protection strictly based on active/standby paths, where paths are independent not only regarding logical but physical links;

ii) Complementarily the other model (Section 3.2) analyzes that case where all sufficiently capacitated routes are allowed/considered as paths for LSPs.

Although strictly speaking it is not necessary to know further technological details to understand models, before closing this "MPLS tutorial" we describe another IP/MPLS protocol, which resembles local protection mechanism of SDH/SONET.

Even though protection mechanisms previously described are extremely flexible, on some circumstances they may have lacks of performance. The switching from one path to another is commanded by the head-end node, which gets aware of the failure by two possible means. The easiest one is checking changes on the topology database of the link state IGP. The other is polling the status of each LSP end-to-end, by keeping special packets circulating over logical rings. This mechanism would be Bidirectional Forwarding Detection (BFD) for MPLS LSPs.

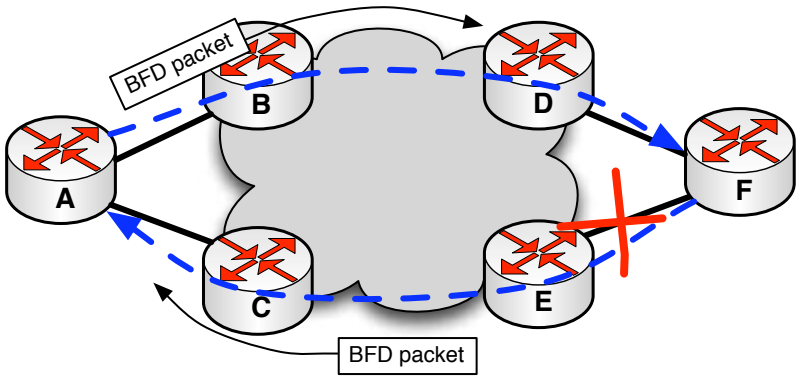


Figure 2.27: End-to-end failure detection mechanisms

For instance (Figure 2.27), to detect failures on LSPs between A and F, BFD packets (marked as of high-priority) are periodically sent by end-points. After a certain timeout without receiving any BFD packet, A and F become aware that a fault has arisen and switch to the second path, as it was established into the LSP configuration. This is usually faster than waiting the sequence of updates in

the IGP, necessary for them to get to know the topology change. In other words, BFDs are handled by the data plane rather than by the control one. However, when endpoints are far apart both mechanisms might react slowly[11].

The complementary mechanism used to prevent major consequences against previous situations is Fast ReRoute (FRR). FRR replicates local protection mechanisms of SDH/SONET (ring protection) over a logical IP/MPLS neighborhood.

Whether FRR is activated to protect intermediate physical links (or even nodes) of an LSP, each intermediate node computes and pre-provisions an independent/backup path to the next-hop called *detour*. When a failure happens on a link incident to an intermediate node, this detects it immediately, prepends the pre-established label and sends frames through the detour until either link gets repaired or head-end switches to an alternate path for the LSP.

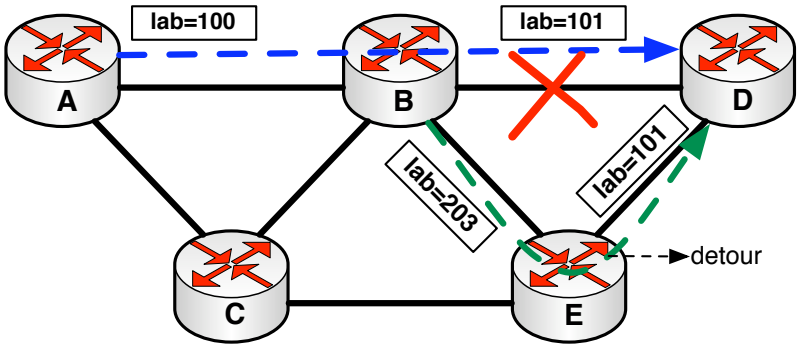


Figure 2.28: Fast-reroute as a local protection approach over MPLS

For example, let us suppose that certain node (B in Figure 2.28) has an entry in its LIB (and LFIB), guaranteeing that MPLS frames arriving to it -on any interface- with label 100, will be forwarded to node D after switching label value to 101. Node A uses this entry to implement the LSP that moves frames towards D. In order to protect this connection, the node B has also negotiated a *detour* (of label 203) across node E to get to node D. When link BD fails, node B starts prepending label 203 to frames formerly sent to D, and rather send them to E, which performs PHP (pops label 203) and sends the originally intended MPLS frame towards node D -over another interface- restoring rapidly the LSP from the fault.

FRR relies on PHP and RSVP-TE (necessary to establish the detour); it also increases the stack of labels by one while frames travel over the detour. It is a very fast mechanism, which allows restoration within 50ms as it was in TDM networks. When besides the path, FRR reserves capacity for detours, the overall efficiency of bandwidth usage falls down to levels of TDM networks. The truth is that in a data-network there is no need to keep a spare capacity of 50%. The sub-50ms limit is only important for a few applications, whose packets are marked as of high-priority (e.g.

[11]This process completes within 1s. Moreover, over non-international networks it usually does it under 300msec. Pretty good times indeed, although still far from the traditional 50msec target.

voice packets). Since volume of this traffic is far below Internet's traffic, the eventual transitory congestion over links used by a detour (e.g. B-E and D-E in Figure 2.28) would briefly affect packets coming from applications resilient to congestion, while head-ends facilities establish a traffic-engineered end-to-end backup path.

2.3 Summary

The beginning of this chapter summarizes some theoretical elements used either upon the next chapter (e.g. graph theory and computational complexity) or in that after it (e.g. metaheuristics). Afterwards, the fundamental characteristics of IP/MPLS technology are described, putting emphasis upon high level features and the overall interaction of protocols.

On IP/MPLS networks, nodes can behave as both: pure IP routers and/or MPLS switches. IP/MPLS is an extension of former IP technology that combines artifacts from legacy ATM networks. It requires several technologies and protocols working simultaneously. Most were mentioned during this section: RIP, OSPF-TE, ISIS-TE, MBGP (external and internal), LDP, RSVP-TE, BFD and FRR. Additionally, T-LDP is used to share information of non-IP VPNs (pseudo-wire emulation).

Further technical details on technologies are far beyond the purpose of this document. Actually, figuring out alternative usages of this large combination of protocols requires high technical skills. This is perhaps the main reason why existing literature falls back onto variants of previous (and much simpler) TDM technology. Regardless of that, within the vast exiting literature we recommend reading: [Hucaby 2004] for comprehensive application cases about building medium scale networks by combining IP routing and Ethernet switching: [Gough 2004] for extensive detail about IP routing protocols (both IGP and BGP); [Pepelnjak 2001] and [Pepelnjak 2003] for technical detail about how merging MBGP and MPLS to construct IP-VPN and large-scale networks and [Osborne 2002] for information about how design, configure, and manage MPLS-TE to optimize network performance.

IP/MPLS provides a wide set of mechanism to support protection. Any protection the physical layer can provide, may be logically implemented over IP/MPLS. Furthermore, the IP/MPLS layer can implement protections much richer than those of TDM transport networks, and can do this over a heterogeneous environment of physical portions, and different administrative divisions. Finally, all this can be accomplished within the same layer where traffic from Internet and VPNs is naturally supported, providing QoS mechanism that make possible most services -even multimedia- coexist within the same backbone.

We are confident that optimal resiliency relies upon logical layer protection mechanisms. Models covered into this work were developed accordingly. Results found for real application cases (Chapter 5) sustain these assertions.

Chapter 3

Design of Communications Networks

Contents

This chapter presents two formal models that correspond to an abstract approach to the problem of designing an optimal IP/MPLS deployment over an optical network. Each model derives from a technological alternative to implement protection upon an IP/MPLS network.

They will be used later on, to improve the quality of real applications. Besides presenting a MIP formulation and finding exact solutions for small size instances, the complexity of each problem is formally analyzed. Other theoretical results, useful to estimate bounds and finding exact solutions are enunciated and proved.

3.1 The simplest protection scheme

As it was detailed on Section 2.2.2, traffic demand between each pair of IP/MPLS nodes flows into an LSP or tunnel. At any moment this tunnel follows a path over the IP/MPLS (logical) network. It is possible to assign more than one path to each LSP; these paths are established as a list that is part of the LSP's configuration. When traff-eng is active, head-end logical nodes check the status for paths defined at each list (a per-tunnel list) and they use the first operative path in the list as the active path for the LSP.

It is worth mentioning that any protection scheme the DWDM layer can provide is realizable by IP/MPLS technology. For instance, let us suppose a physical network

of ten nodes and four internal faces as it is represented in Figure 3.1. For example purposes, let us also suppose that logical DWDM rings (colors blue, purple, green and orange) are deployed copying this structure. A circuit from A to F could be physically protected for instance over: blue, orange and green rings.

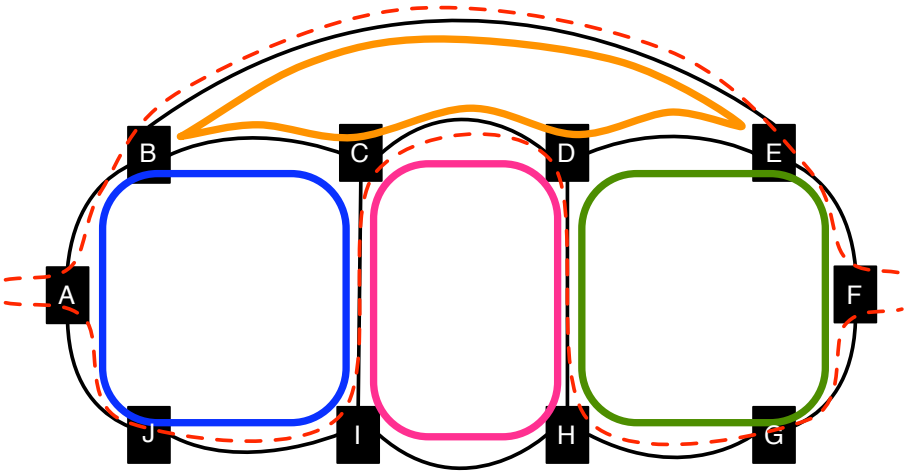


Figure 3.1: Overlapping logical rings

The immediate way to replicate this protection over IP/MPLS would be:

i) Copying the DWDM topology, with as many links as it. That is: one link between A and B, two links between B and C, and so on.

ii) Coloring links belonging to different rings with different colors. For instance: blue, purple, green and orange as in the figure.

iii) Picking up a sequence of nodes starting at A and ending at F, whose intermediate nodes have presence into more than one ring and which order is determined by the sequence of rings the circuit follows towards its destination. For instance: A - B - D - F.

iv) Configure a unique traffic engineering path for the LSP from A to F, where intermediate nodes are those previously define (e.g. B and D), and where the only constraint at each hop is using links of a specific color.

    For instance: blue links to go from A towards B, orange to go from B towards D and green for the final leap.

Node protection (besides link protection) can also be accomplished although is slightly harder. Furthermore, whether demands can be accommodated into links' capacities, it is possible to implement the protection without repeating links, for instance, assigning more than one color (e.g. blue and orange to the link between B and C). Since optical fibers of the same conduit usually fall down simultaneously, this is probably as effective as the former configuration, so IP/MPLS is at least as efficient as DWDM to protect traffic.

In fact, we already know that DWDM efficiency is bound to 50% at each component ring, what we need to find out are referential values for IP/MPLS. Thereby the only protections considered in our models are those the logical network can provide.

This section formalizes one of the protection schemes realizable by IP/MPLS network capabilities. The basis of this model is that each LSP has two static/pre-provisioned physically independent paths, recreating then a protection scheme as is sketched in Figure 2.26, i.e., an active/standby protection scheme where the sets of physical links utilized by each path are disjoint.

It is worth pointing out that this is the simplest protection scheme possible, because the mapping of both paths over the physical resources constitutes a circuit and the circuit is the simplest 2-edge-connected structure. Moreover, this protection is closely related to traditional ring protection, since the circuit can be determined by *xoring* rings over conduits. The active/standby paths associated with the protected circuit of the example sketched in Figure 3.1 are marked with dashed-red curves, and are the outcome of eliminating from the tandem of rings those conduits with repeated logical links.

Besides independency of physical links, paths will be constructed to fulfill with capacity constraints over logical paths. In other words, the model assumes that summarized demand between each pair of nodes is known in advance. To avoid congestion, the sum of these demands over each logical link should not exceed the link's capacity.

For simplicity we assume that demand values are symmetrical and that complementary tunnels (tunnels between the same nodes going in opposite directions) use the same paths. Unlike existing literature, this model also determines the path over the optical (physical) network used to implement each logical link (i.e. the lightpath).

Finally, we assume that parallel logical links are not allowed. This guarantees that the resulting paths for tunnels can be implemented in a strict hop-by-hop (see item b, page 71) scheme, without the need of coloring links to seek out (or avoid) their usage, which would simplify tasks of operation and maintenance. So, this is a practical rather than technological definition.

3.1.1 Active/standby MIP formulation

We introduce now the basic mixed-integer programming model that corresponds to the detailed application of IP/MPLS technology.

Parameters The physical network is represented by an undirected graph (V, P), and the logical network is represented by another undirected graph (V, L). Both layers share the same set of nodes. The links of the logical layer are potential -admissible logical links- while the links of the physical layer are definite. In both graphs the edges are simple since multigraphs are not allowed.

For every different pair of nodes p, q in V, it is known the traffic volume d_{pq} to fulfill along the unique path (tunnel) this traffic follows throughout a logical layer configuration. These paths are unique at every moment, but in case of link failures they may change to follow an alternate route. For simplicity we assume that the traffic volume is symmetric (i.e. $d_{pq} = d_{qp}$).

Let $\hat{B} = \{b_1, \ldots, b_{\bar{B}}\}$ be the set of possible bit-rate capacities for the lightpaths on the physical layer and therefore for the links of the logical one. Every capacity $b \in \hat{B}$ has a known per-distance cost c_b. For economies of scale reasons it holds that if $b' < b''$ then $(c_{b'}/b') > (c_{b''}/b'')$.

Since both graphs of this model are simple and undirected, we express links as pairs of nodes. The length l_{ij} of every physical link (ij) is known in advance.

Variables This model comprises three classes of variables. The first class is composed of the logical link capacity variables. We use boolean variables τ^b_{pq} to indicate whether or not the logical link (pq) has been assigned with the capacity b in $\hat{B}$. As a consequence the capacity of this link could be computed as: $\sum_{b\in\hat{B}} b \cdot \tau^b_{pq}$.

The second class of variables determines how are going to be routed the logical links over the physical network. If $\sum_{b\in\hat{B}} \tau^b_{pq} = 1$ then the logical link (pq) was assigned with a capacity, it is going to be used in the logical network and requires a lightpath in the physical one. The boolean variable y^{ij}_{pq} indicates either the physical link (ij) is being used to implement the lightpath of (pq) or not.

Since lightpaths cannot automatically recover from a link failure, whenever a physical link (ij) fails all the logical links (pq) such that $y^{ij}_{pq} = 1$ do fail as well. The only protection allowed into this model is that provided by the logical layer. The approach used here to protect demands against single physical link failures, consists in setting up two physical-link independent paths for each tunnel. The third and final class of variables determines both paths for each IP/MPLS tunnel. The boolean variable ${}^{rs}x^h_{pq}$ indicates that the logical link (pq) is going to be used either for the active $(h = 1)$ or the standby $(h = 2)$ path, of the tunnel that routes traffic demand $d_{rs} > 0$.

NOTE: To keep the nomenclature of the variables as easy as possible we always placed: logical links subindices at bottom right position, physical links subindices at top right position and demands subindices at top left position.

Constraints This problem comprises three groups of constraints. The first group establishes the rules that the routes of the lightpaths must follow to be feasible.

The meaning of constraints in group (3.1) is the following: (i) establishes that the number of capacities assigned to any logical link is at most 1[1]; (ii) and (iii) guarantee that if any particular link $(pq) \in L$ was assigned with a capacity (i.e. $\sum_{b\in\hat{B}} \tau^b_{pq} = 1$) then there must exist one and only one outgoing -or incoming- physical link used for its lightpath.

[1]It could be 0 if the link is not going to be used

$$
\begin{cases}
\sum_{b\in\hat{B}} \tau^b_{pq} \leq 1 & \forall (pq)\in L. \quad (i)\\
\sum_{j/(pj)\in P} y^{pj}_{pq} = \sum_{b\in\hat{B}} \tau^b_{pq} & \forall (pq)\in L. \quad (ii)\\
\sum_{i/(iq)\in P} y^{iq}_{pq} = \sum_{b\in\hat{B}} \tau^b_{pq} & \forall (pq)\in L. \quad (iii)\\
\sum_{j/(ij)\in P} y^{ij}_{pq} = 2\theta^i_{pq} & \forall (pq)\in L, \forall i\in V, i\neq p, i\neq q. \quad (iv)\\
y^{ij}_{pq} - y^{ji}_{pq} = 0 & \forall (pq)\in L, \forall (ij)\in P. \quad (v)\\
\tau^b_{pq},\ y^{ij}_{pq}, \theta^i_{pq} \in \{0,1\} & \forall (pq)\in L, \forall (ij)\in P, \forall b\in\hat{B}, \forall i\in V. \quad (vi)
\end{cases}
\tag{3.1}
$$

Before going any further we introduce a set of auxiliary variables: θ^i_{pq}. These variables are defined for every combination of logical links (pq) and physical nodes i in V. Hence, (iv) guarantees flow balance while routing the lightpaths through the remaining -non terminal- nodes. That is, any physical node other than p and q is either intermediate to the lightpath of (pq) (has two incident links used to implement it) or has nothing to do with it.

Finally (v) guarantees that the lightpaths go back and forth through the same path, while (vi) stands the integrity of the variables.

$$
\begin{cases}
\sum_{rs:d_{rs}>0} d_{rs}\cdot ({}^{rs}\dot{z}^{ij}_{pq} + {}^{rs}\ddot{z}^{ij}_{pq}) \leq \sum_{b\in\hat{B}} b\cdot\tau^b_{pq} & \forall (pq)\in L, \forall (ij)\in P. \quad (i)\\
\sum_{q/(rq)\in L} {}^{rs}x^h_{rq} = 1 & \forall d_{rs}>0, h\in\{1,2\}. \quad (ii)\\
\sum_{p/(ps)\in L} {}^{rs}x^h_{ps} = 1 & \forall d_{rs}>0, h\in\{1,2\}. \quad (iii)\\
\sum_{q/(pq)\in L} {}^{rs}x^h_{pq} = 2\cdot {}^{rs}\mu^h_p & \forall d_{rs}>0, h\in\{1,2\}, \forall p\in V, p\neq r, p\neq s. \quad (iv)\\
{}^{rs}x^h_{pq} - {}^{rs}x^h_{qp} = 0 & \forall d_{rs}>0, \forall (pq)\in L, h\in\{1,2\}. \quad (v)\\
{}^{rs}x^h_{pq},\ {}^{rs}\mu^h_p,\ {}^{rs}\dot{z}^{ij}_{pq},\ {}^{rs}\ddot{z}^{ij}_{pq} \in \{0,1\} & \forall d_{rs}>0, \forall (pq)\in L, h\in\{1,2\}, \forall p\in V. \quad (vi)
\end{cases}
\tag{3.2}
$$

The second group of constraints establishes the rules that the paths of the IP/MPLS tunnels, must follow over the logical layer. The meaning of the constraints in (3.2) is similar to those of (3.1) except for (i), where the inequalities were added to guarantee that primary and secondary paths assignment cannot overload logical links' capacities. There is more than one realizable alternative for this constraint. The simplest one consists in reserving capacity for both: the primary and the secondary paths, reproducing hence the classical physical protection scheme

SNCP/UPSRs but over the logical layer. Upon this case, for every logical link (pq) the aggregated demand over it reduces to: $\sum_{rs:d_{rs}>0} d_{rs} \cdot ({}^{rs}x^1_{pq} + {}^{rs}x^2_{pq})$, regardless of the physical fault (ij), that is, ${}^{rs}\dot{z}^{ij}_{pq} = {}^{rs}x^1_{pq}$ and ${}^{rs}\ddot{z}^{ij}_{pq} = {}^{rs}x^2_{pq}$, for all physical failure (ij). Keeping these values below each logical link's capacity ($\sum_{b\in\hat{B}} b \cdot \tau^b_{pq}$), guarantees no link is congested. However this model uses another alternative, which will be detailed later on.

Constraints (ii) and (iii) from (3.1) and (3.2) are equivalent, except for the fact that in the latter the existence of a tunnel relies on the existence of demand and this is known in advance. Variables ${}^{rs}\mu^1_i$ and ${}^{rs}\mu^2_i$ are homologous to θ^i_{pq}; so are constraints from (iv) to (vi).

Before proceeding any further we must notice that (3.1) and (3.2) are not independent. After a physical link failure, many logical links may be unavailable. Which logical links are in this condition, relies on how the lightpaths are routed in the physical layer.

$$\begin{cases} {}^{rs}x^1_{pq} + y^{ij}_{pq} + {}^{rs}\lambda^{ij}_{pq} \leq 2 & \forall d_{rs}>0, \forall (pq)\in L, \forall (ij)\in P. \quad (i) \\ {}^{rs}x^2_{\bar{p}\bar{q}} + y^{ij}_{\bar{p}\bar{q}} - {}^{rs}\lambda^{ij}_{pq} \leq 1 & \forall d_{rs}>0, \forall (pq)\in L, \forall (ij)\in P, \\ & \forall (\bar{p}\bar{q})\in L. \quad (ii) \\ {}^{rs}x^h_{pq}, y^{ij}_{pq}, {}^{rs}\lambda^{ij}_{pq} \in \{0,1\} & \forall d_{rs}>0, \forall (pq)\in L, \forall (ij)\in P, \\ & h\in\{1,2\}. \quad (iii) \end{cases} \tag{3.3}$$

The group of constraints (3.3) guarantees that the sets of physical links over which are supported the logical links used to establish either the primary or secondary paths of any LSP, are independent. The reason is the following: if link (pq) is part of the primary path of the tunnel that supports demand d_{rs} (i.e. ${}^{rs}x^1_{pq} = 1$) and the physical link (ij) is being used to support the lightpath of (pq) (i.e. $y^{ij}_{pq} = 1$) then it must stand that ${}^{rs}\lambda^{ij}_{pq} = 0$ to satisfy (i). As a consequence, any logical link $(\bar{p}\bar{q})$ used to support the secondary path for this demand (i.e. ${}^{rs}x^2_{\bar{p}\bar{q}} = 1$) must avoid using physical link (ij) for its lightpath (i.e. $y^{ij}_{\bar{p}\bar{q}} = 0$), otherwise members (i) and (ii) of (3.3) wouldn't be satisfied simultaneously. Hence, the value of the variable ${}^{rs}\lambda^{ij}_{pq}$ is 0 when the primary path of the tunnel corresponding to $d_{rs} > 0$, uses the logical link (pq), and this in turn uses the physical link (ij) as part of its lightpath. The value of ${}^{rs}\lambda^{ij}_{pq}$ is 1 otherwise.

Instead of reserving capacity for backup paths, this model uses the logical layer equivalent to MS-SPRING/BLSR approach, that is, backup paths only consume spare capacity when they are working. Thus the value of ${}^{rs}\dot{z}^{ij}_{pq}$ here is 1, only when (pq) is used by the primary path of $d_{rs} > 0$, and this logical link is not affected by a failure on physical link (ij). Conversely, ${}^{rs}\ddot{z}^{ij}_{pq}$ equals 1, if and only if, (pq) is used by the secondary path of $d_{rs} > 0$, and a fault on (ij) is affecting the primary path.

In terms of equations, this translates into the group of constraints (3.4). If ${}^{rs}x^1_{pq}$ equals 1 for some combination of: $d_{rs} > 0$, $(pq) \in L$ and $(ij) \in P$, the constraint (i) forces the value of ${}^{rs}\dot{z}^{ij}_{pq}$ to 1, unless some ${}^{rs}\lambda^{ij}_{\bar{p}\bar{q}}$ equals 0, which would mean that

the fault (ij) is affecting some logical link $(\bar{p}\bar{q})$ of the primary path of d_{rs}.

$$\begin{cases} {}^{rs}x^1_{pq} + \sum\limits_{\bar{p}\bar{q}\in L} {}^{rs}\lambda^{ij}_{\bar{p}\bar{q}} - |L| \leq {}^{rs}\dot{z}^{ij}_{pq} & \forall d_{rs}{>}0, \forall (pq)\in L, \forall (ij)\in P. \quad (i) \\ {}^{rs}x^2_{pq} - {}^{rs}\lambda^{ij}_{\bar{p}\bar{q}} \leq {}^{rs}\ddot{z}^{ij}_{pq} & \forall d_{rs}{>}0, \forall (pq)\in L, \forall (ij)\in P, \\ & \forall (\bar{p}\bar{q})\in L. \quad (ii) \\ {}^{rs}x^h_{pq}, {}^{rs}\lambda^{ij}_{pq}, {}^{rs}\dot{z}^{ij}_{pq}, {}^{rs}\ddot{z}^{ij}_{pq} \in \{0,1\} & \forall d_{rs}{>}0, \forall (pq)\in L, \forall (ij)\in P, \\ & h\in\{1,2\}. \quad (iii) \end{cases} \tag{3.4}$$

Conversely, if ${}^{rs}x^2_{pq}$ equals 1 and the primary path is being affected by a fault on some physical links (ij), there is at least one logical link $(\bar{p}\bar{q})$ for which ${}^{rs}\lambda^{ij}_{\bar{p}\bar{q}}$ equals 0, and one of the constraints (ii) in (3.4) would be forcing ${}^{rs}\ddot{z}^{ij}_{pq}$ to take the value 1. Behind some of these assertions, relies the fact that the values of ${}^{rs}\dot{z}^{ij}_{pq}$ and ${}^{rs}\ddot{z}^{ij}_{pq}$, are pushed downwards because of constraints (i) of (3.2), and because higher values of τ^b_{pq} increase the objective function, as we shall see now.

Objective The function to minimize is the sum of the cost to implement every logical link. According on which capacity was assigned to a logical link, there is an associated per-distance-cost (c_b), and according on how the corresponding lightpath was routed over the physical layer, there is an associated length $(\sum_{(ij)\in P} l_{ij} y^{ij}_{pq})$.

The product of both terms is the cost of a particular logical link and the sum of these products for all of the logical links is the total cost of the solution. The direct arithmetic expression for the previous statement would be: $\sum_{(pq)\in L}(\sum_{b\in\hat{B}} c_b\tau^b_{pq})(\sum_{(ij)\in P} l_{ij}y^{ij}_{pq}) = \sum_{(pq)\in L,(ij)\in P,b\in\hat{B}} c_b l_{ij}\cdot\tau^b_{pq}y^{ij}_{pq}$.

Although straightforward, this approximation is inappropriate because is non-linear; instead the subproblem (3.5) expresses the objective value with an equivalent linear expression. To do so, we use the real variable ${}^b\eta^{ij}_{pq}$ rather than $\tau^b_{pq}\cdot y^{ij}_{pq}$ and add some extra constraints to guarantee the equivalence between the values of both. Since ${}^b\eta^{ij}_{pq}$ is being multiplied by a positive constant in a minimization problem it would take its lowest value at the optimal point. This value would be at least zero because of constraints (iii) of (3.5).

$$\begin{cases} \min \sum\limits_{\substack{(pq)\in L \\ (ij)\in P \\ b\in\hat{B}}} c_b l_{ij}\cdot {}^b\eta^{ij}_{pq} & (i) \\ {}^b\eta^{ij}_{pq} \geq \tau^b_{pq} + y^{ij}_{pq} - 1 & \forall (pq)\in L, \forall (ij)\in P, \\ & \forall b\in\hat{B}. \quad (ii) \\ {}^b\eta^{ij}_{pq} \geq 0 & \forall (pq)\in L, \forall (ij)\in P, \\ & \forall b\in\hat{B}. \quad (iii) \end{cases} \tag{3.5}$$

The result of $\tau^b_{pq}\cdot y^{ij}_{pq}$ is also a boolean variable whose value is zero except when τ^b_{pq} and y^{ij}_{pq} values are both 1. However, in this case and because of constraint (ii), the

value of ${}^{b}\eta_{pq}^{ij}$ should be 1 as well. The complete MIP is the result of merging: (3.1), (3.2), (3.3), (3.4) and (3.5). Let us call ASP-MORNDP (Active/Standby Protection Multi-Overlay Resilient Network Design Problem) to the former problem.

3.1.2 ASP-MORNDP exact solutions

A particular instance would extend the understanding of the model previously described. Before going into it, we show a useful theoretical property. A bond as it was defined at Definition 12 is a minimal cut-set whose removal disconnects the graph, splitting it into two connected subcomponents. We extend the definition to our two-layers problem.

Definition 23. *Let (V, P) and (V, L) be the graphs defined by the physical and logical networks. Any $bond_P$ of the physical layer will be also called a "multilayer bond" when the following holds: if (V', P') and (V'', P'') are the two connected components of $(V, P \backslash bond_P)$, then $(V', L \bigcap V'^2)$ and $(V'', L \bigcap V''^2)$ are both connected components too. For simplicity, we will refer to $L \bigcap V'^2$ and $L \bigcap V''^2$ as L' and L'' respectively. We also define $bond_{L,P} = \{e \in L \,/\, e = pq, p \in V', q \in V''\}$, which is actually a regular bond of (V, L) induced by $bond_P$.*

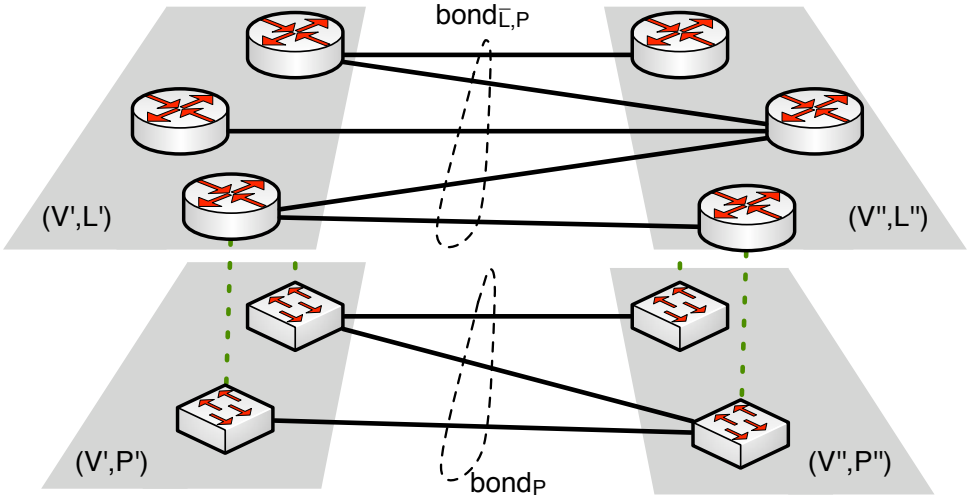


Figure 3.2: Extended version of Bonds to problem ASP-MORNDP

Figure 3.2 shows through an example, the relationship between bonds on both layers, while the following property binds the capacity of them with demands volume between components.

Lemma 3. *Given any solution to ASP-MORNDP, let $\bar{L} \subseteq L$ be the subset of arcs with positive capacities. In order for this solution to be feasible, it must hold that for every* multilayer bond *$bond_P$, the condition:*

$$\sum_{p \in V', q \in V''} d_{pq} \leq b_{\bar{B}} \left\lfloor \frac{|bond_{\bar{L},P}|(|bond_P| - 1)}{|bond_P|} \right\rfloor \tag{3.6}$$

must be satisfied, where $b_{\bar{B}}$ *is the maximum bit-rate available for dimensioning links, and* d_{pq} *is the traffic demand between nodes* p *and* q*.*

Instead of a complete proof, we infer this property up from other two. The first one is the equivalent property (Lemma 5), for the model to cover into the next Section 3.2 (FRP-MORNDP). The second one is the relaxation relationship between models (Theorem 8), i.e., FRP-MORNDP is a relaxation of ASP-MORNDP, thus a necessary condition for FRP-MORNDP must also be necessary for ASP-MORNDP.

Formalities aside, this property is intuitive. For instance, in Figure 3.2 and when $b_{\bar{B}} = 1\text{Gbps}$, the best bandwidth scenario to connect components is that where all links are assigned with 1Gbps (i.e.: $\bar{L} = L$). However, some physical link must be being used at least twice by lightpaths, and a fault on it, would reduce the available capacity between components to at most $1\text{Gbps}\lfloor 5 \cdot 2/3 \rfloor = 3\text{Gbps}$, which is then the limit of demands between components. The condition is necessary but not sufficient, as we shall see into Section 3.2, and into the proof of Theorem 7.

We show here an example instance for ASP-MORNDP. Let $V = \{v_a, v_b, v_c, v_d\}$, $P = L = \{v_a v_b, v_a v_c, v_b v_c, v_b v_d, v_c v_d\}$ (i.e. physical and logical graphs match), $d_{ab} = d_{bc} = 1$, $\hat{B} = \{1\}$, $l_{ij} = 1$ for all (ij) in P and $c_1 = 1$. The graph is represented in Figure 3.3.

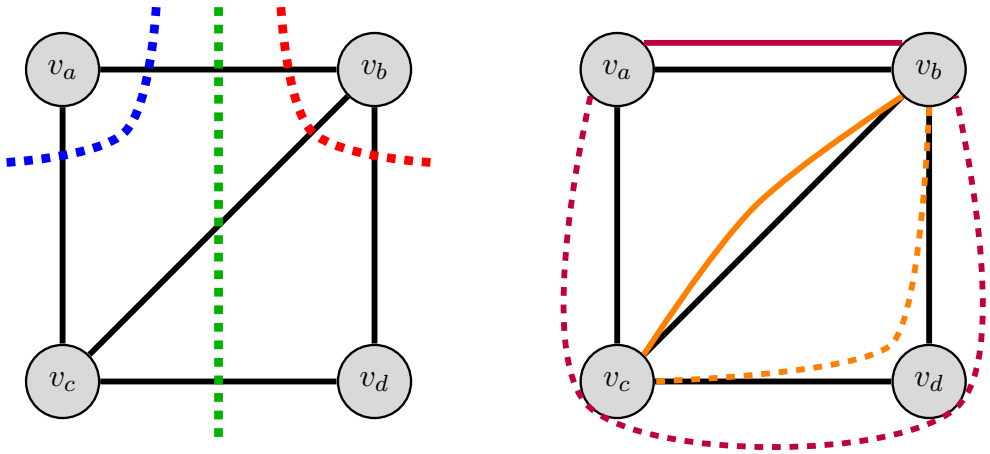


Figure 3.3: Example topology, bonds and paths used during analysis

Applying (3.6) to the bond marked by a blue-dashed curve onto the left half of Figure 3.3, results that: $1 \leq \lfloor |bond_{\bar{L},P}|/2 \rfloor$, from where both logical links from v_a must be used into the solution. Repeating the idea for the green-dashed bond, results that: $2 \leq \lfloor |bond_{\bar{L},P}| \cdot 2/3 \rfloor$, and now there must be 3 the number of logical links across the bond. Finally and for identical arguments, the red-dashed bond determines that all the potential logical links must be used in the construction.

Since all physical costs are identical, there is no reason to search an implementation for lightpaths, other than that where each one is implemented by the physical link homologous to each logical one.

In terms of the MIP formulation equivalent to ASP-MORNDP, this translates into the following: $\tau^1_{ab} = \tau^1_{ac} = \tau^1_{bc} = \tau^1_{bd} = \tau^1_{cd} = 1$, $y^{ab}_{ab} = y^{ac}_{ac} = y^{bc}_{bc} = y^{bd}_{bd} = y^{cd}_{cd} = 1$, whereas $y^{ij}_{pq} = 0$ otherwise. Besides, since no node is intermediate to any lightpath implementation, $\theta^i_{pq} = 0$ for all logical link (pq) and for all i in V. These values fulfill constraints contained into (3.1).

Regarding the paths of the tunnels, the right half of Figure 3.3 proposes a solution, where paths for tunnel corresponding to d_{ab} are represented by purple curves, whereas paths for d_{bc} are orange. In both cases, solid curves correspond to primary and dashed ones to standby paths.

Hence, the values for the remaining variables of the MIP formulation are the following: ${}^{ab}x^1_{ab} = {}^{ab}x^2_{ac} = {}^{ab}x^2_{cd} = {}^{ab}x^2_{bd} = 1$, ${}^{bc}x^1_{bc} = {}^{bc}x^2_{cd} = {}^{bc}x^2_{bd} = 1$ and ${}^{rs}x^h_{pq} = 0$ otherwise. Complementarily, the values of the auxiliary variables ${}^{rs}\mu^h_p$ are: ${}^{ab}\mu^2_c = {}^{ab}\mu^2_d = {}^{bc}\mu^2_d = 1$ and 0 otherwise, because v_c and v_d are the unique intermediate nodes, the first one for the secondary path of d_{ab}, the second one for the secondary path of both d_{ab} and d_{bc}. These values satisfy constraints groups (ii) to (vi) of (3.2).

It is worth pointing out that if we had chosen the approach where spare capacity is reserved, this solution wouldn't be feasible because logical links v_bv_d and v_cv_d are being used twice. To determine the values of ${}^{rs}\dot{z}^{ij}_{pq}$ and ${}^{rs}\ddot{z}^{ij}_{pq}$ in our case, we will see first those of ${}^{rs}\lambda^{ij}_{pq}$. The only λ's whose values are 0 are ${}^{ab}\lambda^{ab}_{ab}$ and ${}^{bc}\lambda^{bc}_{bc}$. Constraints into (3.3) are satisfiable, because active and standby paths of both tunnels are independent. In general ${}^{rs}\lambda^{ij}_{pq} = 1$ closes the equations, except for $({}^{ab}x^1_{ab}, y^{ab}_{ab})$ and for $({}^{bc}x^1_{bc}, y^{bc}_{bc})$, because their values are 1. Thus ${}^{ab}\lambda^{ab}_{ab} = {}^{bc}\lambda^{bc}_{bc} = 0$ are the exceptions.

To wrap up this example, let us notice that $\sum_{b\in\hat{B}} b \cdot \tau^b_{pq} = 1$ for every logical link (pq). Therefore, given (pq) and (ij) and in order to satisfy (i) into (3.2), there cannot be two values within the sets $\{{}^{ab}\dot{z}^{ij}_{pq}, {}^{ab}\ddot{z}^{ij}_{pq}\}$, $\{{}^{bc}\dot{z}^{ij}_{pq}, {}^{bc}\ddot{z}^{ij}_{pq}\}$ different than 0. In general we take ${}^{rs}\dot{z}^{ij}_{pq} = {}^{rs}\ddot{z}^{ij}_{pq} = 0$; the only exceptions arise from (3.4) and they are: ${}^{ab}\dot{z}^{ij}_{ab} = 1$ when $(ij) \neq (ab)$, ${}^{bc}\dot{z}^{ij}_{bc} = 1$ when $(ij) \neq (bc)$, ${}^{ab}\ddot{z}^{ab}_{pq} = 1$ when $(pq) \in \{(ac), (bd), (cd)\}$ and ${}^{bc}\ddot{z}^{bc}_{pq} = 1$ when $(pq) \in \{(bd), (cd)\}$.

Finally, ${}^1\eta^{ab}_{ab} = {}^1\eta^{ac}_{ac} = {}^1\eta^{bc}_{bc} = {}^1\eta^{bd}_{bd} = {}^1\eta^{cd}_{cd} = 1$ satisfies (3.5). Since all logical links must be used to comply with Lemma 3, and 1 is the minimum length to implement each one of them, it is immediate that 5 is the lowest possible cost.

The MIP formulation for ASP-MORNDP lacks of other value than the formalization of the problem. Its exact numerical solution is unaffordable, and before entering into the analysis of its intrinsic complexity, we present a simple theoretical result, which shows how to construct solutions for a particular family of instances.

Lemma 4. *Given $\mathcal{C}^n$ (an n nodes cycle) as the physical layer and some logical layer L such that: $\mathcal{C}^n \subseteq L$, if demands conform to $d_{pq} \leq D$ is always possible to find minimal feasible solutions when either: $b_{\bar{B}} = Dn^2/4$ and n is even, or when*

$b_{\bar{B}} = D(n^2-1)/4$ *and* n *is odd. Moreover, the optimal solution in both cases reduces to use* $\mathcal{C}^n$ *as the logical layer.*

Proof. First of all, let us observe that for having two physically independent output paths from any node, it is necessary to deploy at least one of them in each direction over the physical ring. Hence, that logical network that copies the physical topology and maps all lightpaths in just one hop to the neighbor, is both: minimal (simplest 2-edge-connected topology) and of minimum cost, because it only uses each physical link once, which is the minimum necessary to be able to keep physical independence between active and standby paths.

Suffices then to establish a links' capacity for guaranteeing feasibility. Optimality is inherited from the previous comment. To conclude this proof, let us observe that because of the simplicity of this topology, each link supports either the primary or the secondary path for each LSP, but not both (physical independence). Let us use as the primary path, that with the minimum number of hops towards its destination. This reduces the number of times a logical link is being used by active paths, to at most $\lceil n/2 \rceil$, and plenty fulfills the premisses of this property.

The most stressing case is when a physical link fails, because the logical layer reduces to a sequence of adjacent nodes, Without loss of generality the surviving arcs of the logical layer could be: $\{(v_1v_2), (v_2v_3), \ldots, (v_{n-1}v_n)\}$. Given any of these arcs (v_k, v_{k+1}), $1 \leq k < n$, the traffic across this link would be at most: $\sum_1^k \sum_{k+1}^n D = k(n-k)D$, when $d_{pq} = D$ for every $1 \leq p < q \leq n$.

If n is even this function takes its maximum at $n/2$ which image is $Dn^2/4$, precisely the value chosen to dimension links upon this case. If n is odd this function takes its maximum at $(n-1)/2$ and $(n+1)/2$ which image is $D(n^2-1)/4$, and also matches the capacity chose of the logical link. □

It is pointless to go further into the search of exact solutions for other cases. As we shall see into Section 4.1, the algorithms used to find solutions, dispense with these results during their constructions.

3.1.3 ASP-MORNDP complexity analysis

Until now we expressed the problem as of two layers. Although from a technological perspective this is an accurate approximation of what really happens, from a more abstract perspective we may think about this problem as of three layers.

Besides the formerly defined logical and physical layers, there is another abstract overlay on the top of them "the service perspective about the problem", i.e., how are paths assigned to point-to-point demands between POPs (Figure 3.4).

The analysis of ASP-MORNDP's complexity developed in this section, relies on this abstract structure of three layers. ASP-MORNDP is not only NP-Hard, it

embeds problems computationally hard to solve into the mapping from each layer over the underneath one.

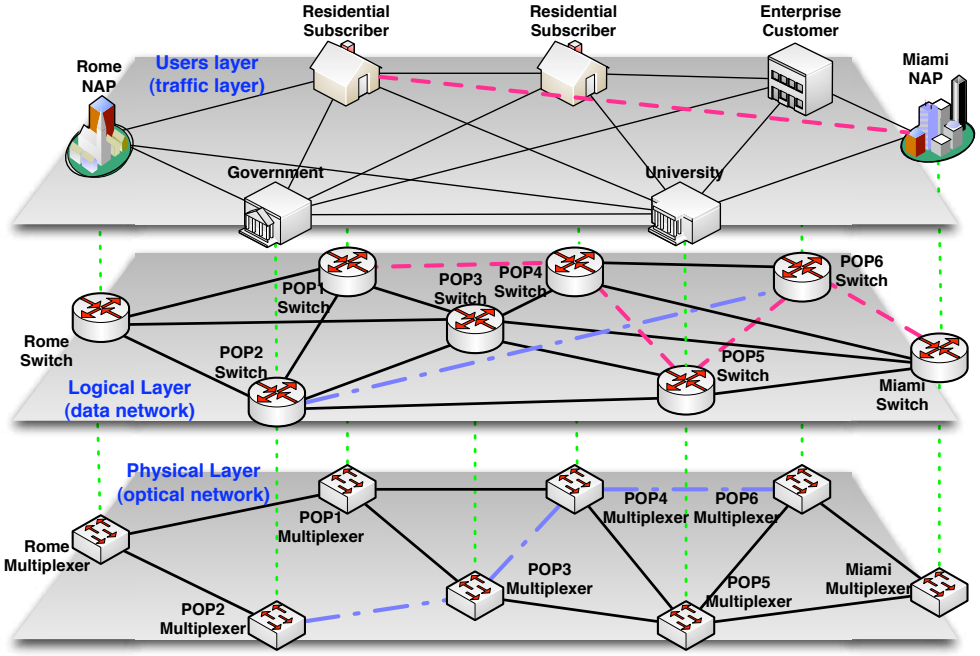


Figure 3.4: A three layers conceptual model.

Theorem 6 establishes the character of NP-Hard for the subproblem of optimally deploying a logical over a physical network. That is, when traffic volume is negligible compared with links' capacity, the subtask of determining lightpaths in order to achieve an optimal cost deployment for the logical layer constitutes by itself an NP-Hard problem.

Complementarily (Theorem 7) and since traffic between nodes can only be moved over two prefixed paths, the mere task of arranging tunnels over a predetermined logical layer -routing of the IP/MPLS tunnels over a data network- constitutes an NP-Hard problem too. Hence, ASP-MORNDP is indeed the composition of solely very hard to solve problems.

Theorem 6. *The subproblem of routing the logical layer over the physical one, turns NP-Hard the problem ASP-MORNDP.*

Proof. The proof lies under reduction of 2ECSS (Two-Edge-Connected Spanning Subgraph) to ASP-MORNDP. The 2ECSS problem consist in finding a minimum-weight 2-edge-connected subgraph of a weighted graph. In general 2ECSS is an NP-Hard problem and it is NP-Complete when weights are integer numbers (see [Eswaran 1976]). It is closely related to MW2CSN (item iv, page 47) though in this case nodes are 2-edge-connected rather than 2-node-connected.

We use decision versions of both problems to validate instances. Let π_k be the decision problem consisting in finding whether or not exists a feasible solution for

ASP-MORNDP of integer cost k, such that for any other integer k' where $k' < k$ the answer to $\pi_{k'}$ is negative. Analogously, let π'_k be the decision problem consisting in finding whether or not exists a feasible solution for 2ECSS of integer weight k, such that there is no other solution of lower weight. We propose a mapping from instances of π'_k to instances of π_k, and prove that it is a polynomial reduction.

($\Rightarrow$) Given an instance $G = (V, E, W)$ of 2ECSS -for simplicity we assume that all the weights are positive integers- we create an instance of ASP-MORNDP by taking: logical and physical graphs with the same topology of G (i.e. $L = P = (V, E)$), $|\hat{B}| = 1$ with $b_1 = N(N-1)/2$ where $N = |V|$, $l_{ij} = w_{ij}$ for all (ij) in P (i.e. weights of arcs in G are lengths of arcs of P) and $d_{ij} = 1, \forall 1 \leq i < j \leq N$. Finally $c_b = 1$ for the unique available capacity.

Before going any further please notice that because of how demands and capacities were set-up for this ASP-MORNDP instance, any logical link has enough capacity to support within it all active demands of the logical network.

This part of the proof consists in assuring the existence of a positive instance for π_k, whenever there is a positive instance for π'_k. If the instance G satisfies the decision problem π'_k, then exists $G' \subseteq G$ of weight k so that G' is two-edge-connected and there is no other solution of lower weight than k. The rules to construct a solution for π_k up from a solution $G' = (V, E')$ of π'_k are the following:

1. Set as definite those logical links homologous to E'. That is, take every edge of G' and dimension its homologous logical link with capacity b_1.
2. The remaining logical links will not be used.
3. Implement in one hop the lightpath for every effective logical link, making use of its associated physical link. This is always possible since P is equal to L.

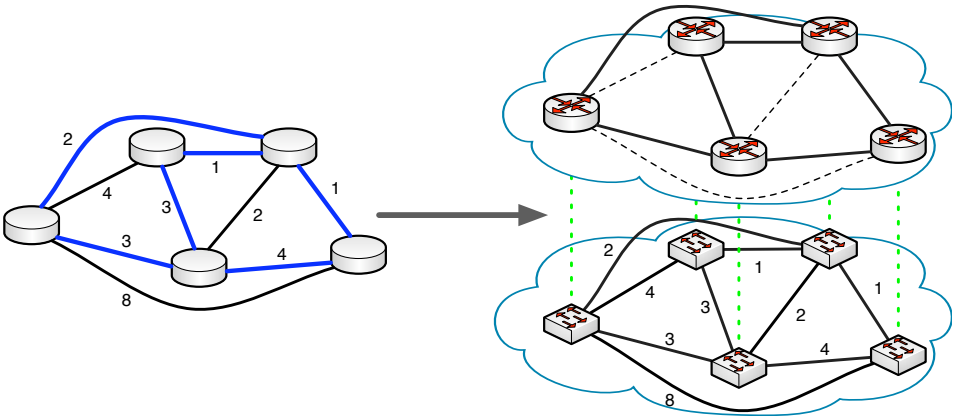


Figure 3.5: Reduction from 2ECSS to ASP-MORNDP

Figure 3.5 presents an example instance for 2ECSS (left side) and its corresponding image through this transformation (right side). Links highlighted with

bold-blue constitute a solution for this instance of π'_{14}. Their corresponding logical links in the image (marked as solid) are the only links that are used in the solution to ASP-MORNDP, and their lightpaths are the associated projections over the physical layer. Both constitute the solution to this instance of π_{14}.

It is immediate that the cost of the previous solution is always k. Complementarily, logical links are one-to-one with physical links, and therefore preserve independence. Besides, G' is 2-edge-connected, and because of Theorem 2 there must be two link independent paths between each pair of vertices. Since the logical layer copies the topology of G', and the construction preserves independence amid logical and physical links, the paths of Theorem 2 are physically independent and can be used as primary and secondary paths for each demand.

Finally, the capacity used on links guarantees that none can be saturated. The construction mechanism turns connectivity into feasibility, so the solution proposed for π_k is feasible from the network point-of-view. However, to close this part of the proof we must be sure that there is no other solution for π_k with lower cost.

This is sustained by the following facts. In the case that a solution of lower cost than k could be found, it would be possible to find an alternate construction of the same cost, whose lightpaths' mapping results from projecting effective logical links one-to-one over physical ones (see the reciprocal part of the proof). The key is that on such a case, the pre-image of this construction would also be a solution to π'_k of lower cost what constitutes an absurd.

($\Leftarrow$) First of all, we claim that *given any solution to a positive instance of π_k constructed by the previous mechanism from some π'_k, it holds that its lightpaths are edge-disjoint.* Indeed, suppose that (ij) and $(i'j')$ were two logical links whose lightpaths intersect, like in Figure 3.6. Then, another solution can be built up by exchanging (ij) and $(i'j')$ by the logical links associated to its implementation of lightpaths, like it is sketched in the second half of Figure 3.6.

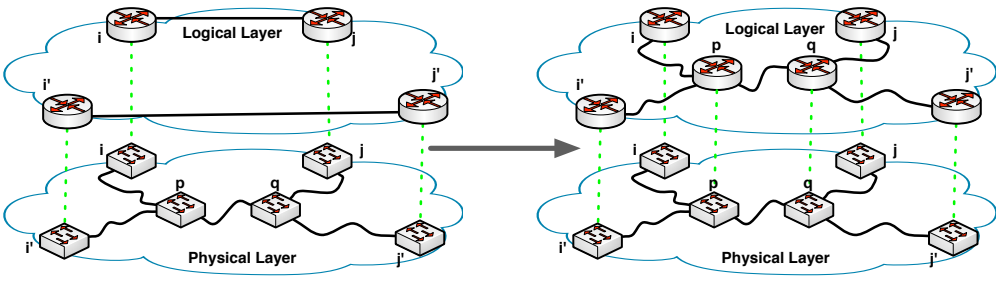


Figure 3.6: Non-disjoint lightpaths implementation and an alternate configuration.

If during this transformation we find out that one of the new logical links is already present on the remaining set of logical links, we just skip this particular appending. The transformation preserves -may even surpass- the logical connectivity in every physical failure scenario. It is also cheaper because the physical links of path between p and q are only used once in this solution. The last cannot hold

because it would exist a solution whose cost is lower than k, which is explicitly forbidden in this decision problem. This ends the proof of our claim.

As a corollary of the previous property, we can prove that *given any solution to a positive instance of π_k, it is always possible to build an equivalent one where topologies of logical and physical layers match.* The transformation process is based on the following recurrence:

1. If any logical link is not implemented using its corresponding physical link, we replace it with the logical links homologous to those links of the lightpath.
2. If during the previous step a logical link is repeated, the former is replaced by the logical links associated to the links of its lightpath. This can always be done because physical implementations must be disjoint.
3. Repeat the process until logical and physical networks match.

Since this transformation preserves the usage of links in the physical network, the cost of the solution and the failure scenarios are not affected. Remains to be seen that the topology found after the transformation (let us call it $G' = (V, E')$) is two-edge-connected, and so it is also a solution for π'_k.

Let us observe that the existence of a solution to π_k, implies the existence of two physically independent paths between any pair of nodes (i.e. primary and secondary paths). To determine the corresponding logical paths after the transformation, suffices to expand each original logical link to the corresponding newer sequence, according on the steps followed during the logical links transformation. Since physical independence is preserved, and new logical links are one-to-one with the originals, these new set of pairs of paths onto G', are under the premisses required by Theorem 2 to guarantee that G' is 2-edge-connected.

Finally, it cannot exist another feasible solution to π'_k of cost $k' < k$, because the outcome of the transformation (described in $\Rightarrow$) to this solution would be a feasible solution to π_k of cost lower than k. Since all the transformations are of polynomial complexity it stands that $\pi'_k \preccurlyeq \pi_k$, and due to the fact that 2ECSS is NP-Hard, ASP-MORNDP is NP-Hard too. □

Theorem 7. *The subproblem of routing traffic demands over the logical layer turns NP-Hard the problem ASP-MORNDP.*

Proof. The proof lies under reduction of NPP (Number Partitioning Problem) to ASP-MORNDP. NPP problem consist in finding two subsets with almost the same sum for a known multiset of numbers. Formally, given a list of positive integers: $a_1, a_2, \ldots, a_N$, a partition $\mathcal{A} \subseteq \{1, 2, \ldots, N\}$ must be found so that discrepancy:

$$E(\mathcal{A}) = |\sum_{i \in \mathcal{A}} a_i - \sum_{i \notin \mathcal{A}} a_i|,$$

finds its minimum value within the set $\{0, 1\}$. NPP is a very well known NP-Complete problem (see for instance [Hayes 2002] and item iii - page 47).

($\Rightarrow$) Given an instance of NPP (a list of positive integers) we create an instance of ASP-MORNDP by taking: logical (L) and physical (P) graphs with the same topology schematized in Figure 3.7, which will be called $G = (V, E)$, $|\hat{B}| = 1$, $M = (\sum_{1 \leq i \leq N} a_i)$ and $b_1 = \lceil M/2 \rceil$, the length of any (ij) in P is $l_{ij} = 1$ and $d_{iD} = a_i$ for all v_i such that $1 \leq i \leq N$.

Since logical and physical topologies are the same (as in the proof of Theorem 6), given any optimal solution, another one of the same cost can be constructed where lightpaths are implemented using homologous physical links (similar proof to internal corollary of Theorem 6).

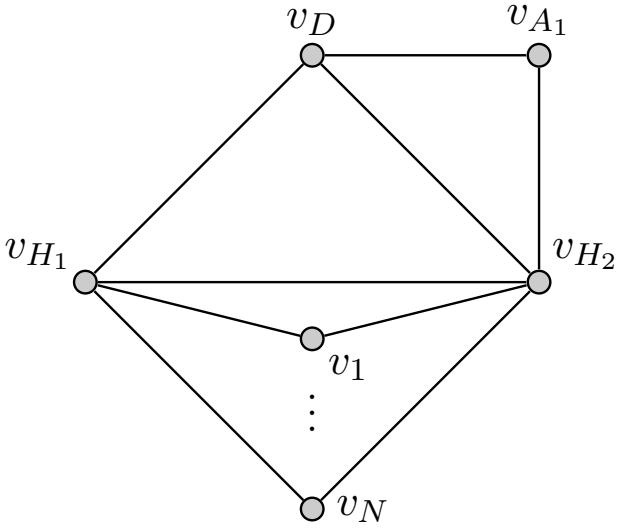


Figure 3.7: Graph used for NPP reduction to ASP-MORNDP

The problem is if all of these logical links are necessary as a part of an optimal solution. There are two possibilities here; the easiest one is when for every a_i it holds that $a_i \leq b_1/2$, in which case we can manage without the logical link (v_{H_1}, v_{H_2}).

First of all we must observe that all logical links from any v_i to either v_{H_1} or v_{H_2} are necessary to provide two physically independent paths from v_i for the tunnel determined by $d_{iD} > 0$. Besides, the three links with an endpoint in v_D are necessary as a minimal capacity between subcomponents determined by the bond between $\{v_D\}$ and other nodes of V ((3.6) of Lemma 3). This is because $|bond_P| = 3$ and $\sum_{1 \leq i \leq N} d_{iD} = M$, so

$$M \leq \lceil M/2 \rceil \left\lfloor \frac{2 \cdot |bond_{\bar{L},P}|}{3} \right\rfloor,$$

requires $|bond_{\bar{L},P}| = 3$. The same lemma applied form the bond $\{v_D, v_{A_1}\}$ to its complement, determines the usage of all logical links, other than (v_{H_1}, v_{H_2}).

Finally and since all distances are $l_{ij} = 1$, the cost equals the number of hops followed by lightpaths and it cannot be an optimal construction other than that

where all lightpaths are implemented by the homologous physical link (at one hop). Hence, the reduction determines that any solution must use the entire logical layer but (v_{H_1}, v_{H_2}), and that logical failures are one-to-one with physical ones.

The argumentation up to this point is also valid onto the complementary part of the proof. To conclude this part, let us suppose that the base NPP instance is positive, that is, exists $\mathcal{A}$ such that $|\sum_{i\in\mathcal{A}} a_i - \sum_{i\notin\mathcal{A}} a_i| \in \{0, 1\}$. For the values of the terms $\sum_{i\in\mathcal{A}} a_i$ and $\sum_{i\notin\mathcal{A}} a_i$, this implies either both values are b_1 or one of them is $b_1 - 1$, both not both. Without loss of generality we assume that $|\mathcal{A}| \geq \lceil N/2 \rceil$, i.e., $\mathcal{A}$ has more elements than its complement.

The general procedure to set-up a feasible configuration for paths of tunnels under these premisses is simple:

- For each i in $\mathcal{A}$ set the primary path for d_{iD} as: $(v_i v_{H_1})$, followed by $(v_{H_1} v_D)$. This doesn't overload $(v_{H_1} v_D)$ because $\sum_{i\in\mathcal{A}} d_{iD} \leq b_1$.
- For each i in $\mathcal{A}$ set the secondary path for d_{iD} as: $(v_i v_{H_2})$, followed by $(v_{H_2} v_{A_1})$, followed by $(v_{A_1} v_D)$.
- For each i not in $\mathcal{A}$ set the primary path for d_{iD} as: $(v_i v_{H_2})$, followed by $(v_{H_2} v_D)$. This doesn't overload $(v_{H_2} v_D)$ because $\sum_{i\notin\mathcal{A}} d_{iD} \leq b_1$.
- For each i not in $\mathcal{A}$ set the secondary path for d_{iD} as: $(v_i v_{H_1})$, followed by $(v_{H_1} v_j)$, followed by $(v_j v_{H_2})$, followed by $(v_{H_2} v_{A_1})$, followed by $(v_{A_1} v_D)$, for any arbitrary $j \in A$, not yet used during this step of the construction. This is always possible because $\mathcal{A}$ has more elements than its complement.

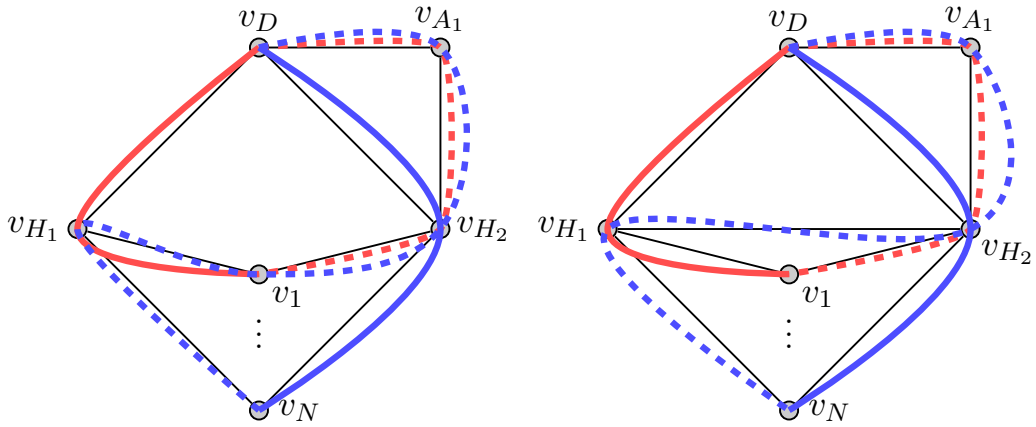


Figure 3.8: Primary and secondary paths construction

The left half of Figure 3.8 sketches this construction. Let us observe that primary and secondary paths on this construction are link-independent. Besides, secondary

paths of both families are not active simultaneously, so the aggregated load of active secondary paths, doesn't overload neither $(v_{H_2}v_{A_1})$, nor $(v_{A_1}v_D)$.

This construction relies on the fact that $a_i \leq b_1/2$, so there is always room to accommodate two demands within one link. When this premiss fails, the procedure might not be capable of finding a feasible construction. This is because the number of logical links is not sufficient.

For example, for the NPP instance $\{1, 2\}$ the previous procedure fails, because $b_1 = 2$, and there is not enough spare capacity to accommodate demands. This is indeed a pretty simple counterexample to the bonds condition Lemma 3.

However on these cases, the construction can be fixed by using all the logical links, i.e., by appending (v_{H_1}, v_{H_2}), and by using it to detour "blue secondary paths" as is indicated upon the right part of Figure 3.8. The resulting construction is minimal, and uses the minimum number of hops to implement lightpaths, so it is an optimal configuration.

This closes the consistency for the direct part of the proof.

($\Leftarrow$) The complementary part of the proof is easier. Let us guess that there is a positive instance of ASP-MORNDP constructed by the previous mechanism from some instance of NPP. The mapping between links of logical and physical layers for a solution to this instance is one-to-one (arguments included in $\Rightarrow$).

Since the instance is positive, there must be a configuration of paths, such that demand can be accommodated into (v_{H_1}, v_D) and (v_{H_2}, v_D) when (v_{A_1}, v_D) fails. Because of the limited set of capacities, both links must have been assigned with b_1, and such accommodation of demands and paths (either primary or secondary), can only be done when discrepancy is not greater than one. Hence, through this assignment we indirectly found a solution for the original NPP problem.

Since the transformation process is of polynomial complexity, it must stand that $\text{NPP} \preccurlyeq \text{ASP} - \text{MORNDP}$ and therefore ASP-MORNDP is NP-Hard too. □

3.2 A much more versatile scheme

The second model covered by this work is actually an extension of that presented at Section 3.1 (ASP-MORNDP). It shares all conceptual components and portions of the model but those regarding the path conformation for LSPs.

The former model establishes two physically independent paths for the LSPs as a minimum degree of resilient connectivity. This one in contrast allows the existence of as many path configurations for any LSP as the number of single physical link failure scenarios, covering then the opposite extreme of potential configurations allowed by the technology.

The ASP-MORNDP problem is inspired on former protection mechanisms of TDM and DWDM Optical Transport Networks (OTNs), although here active and

standby paths are not confined to a logical ring, and rather they are extreme-to-extreme. ASP-MORNDP can be easily configured on a node-by-node specification of primary and secondary paths for each LSP, which is widely supported by Operating Systems (OS) of Network Equipment Providers (NEP), and thus is realizable on actual networks. To avoid manual intervention on this process, a module on an external Network Management System (NMS), which implements the model, computes solutions and also sets up paths for each LSP, could support the administrative process automatically, whereas control planes of nodes do the actual work.

Conversely, the model covered in this section is inspired on a dynamic demand-constrained routing for paths. It doesn't preset paths and rather relies on capabilities of dynamic routing protocols to establish them when faults arise. Although strictly speaking the existence of a feasible paths configuration does not guarantee CSPF can find it (see *bumping* into Section 4.2.2), results computed for real instances (Chapter 5) are very promising with regard to the performance that standard traffic-engineering IGPs can achieve.

3.2.1 Free routing MIP formulation

We introduce now the other mixed-integer programming model, which corresponds with this extended application of the IP/MPLS technology. Rather than doing it from scratch, here we're only detailing the differences with the former model.

First of all we must mention that parameters of this model are identical to those of ASP-MORNDP, so are variables τ^b_{pq} and y^{ij}_{pq}, as well as constraints (3.1) and (3.5), because the construction of lightpaths and its associated cost in this case follows the same set of rules that in the precedent.

$$
\begin{cases}
\displaystyle\sum_{rs:d_{rs}>0} d_{rs} \cdot {}^{rs}x^{ij}_{pq} \leq \sum_{b\in\hat{B}} b \cdot \tau^b_{pq} & \forall(pq)\in L, \forall(ij)\in P. \quad (i) \\
\displaystyle\sum_{q/(rq)\in L} {}^{rs}x^{ij}_{rq} = 1 & \forall d_{rs}>0, \forall(ij)\in P. \quad (ii) \\
\displaystyle\sum_{p/(ps)\in L} {}^{rs}x^{ij}_{ps} = 1 & \forall d_{rs}>0, \forall(ij)\in P. \quad (iii) \\
\displaystyle\sum_{q/(pq)\in L} {}^{rs}x^{ij}_{pq} = 2 \cdot {}^{rs}\mu^{ij}_{p} & \forall d_{rs}>0, \forall(ij)\in P, \forall p\in V, p\neq r, p\neq s. \quad (iv) \\
{}^{rs}x^{ij}_{pq} - {}^{rs}x^{ij}_{qp} = 0 & \forall d_{rs}>0, \forall(pq)\in L, \forall(ij)\in P. \quad (v) \\
{}^{rs}x^{ij}_{pq},\ {}^{rs}\mu^{ij}_{p} \in \{0,1\} & \forall d_{rs}>0, \forall(pq)\in L, \forall(ij)\in P, \forall p\in V. \quad (vi)
\end{cases}
\tag{3.7}
$$

Differences between models come from how both of them build their tunnels, which introduces changes in the variables ${}^{rs}x^h_{pq}$ that are now reformulated as ${}^{rs}x^{ij}_{pq}$, expressing the configuration of paths followed by tunnels to recover from a failure on physical link (ij). Hence ${}^{rs}x^{ij}_{pq}$ is a boolean variable that indicates whether or not

the logical link (pq) is going to be used to route traffic demand $d_{rs} > 0$, under a fault condition in the physical link (ij). We explicitly omitted variables corresponding to the nominal/non-faulty states, because any feasible configuration for tunnels when some lightpaths are under a fault condition is also feasible when none is.

The group of constraints that establishes the rules that the paths of the IP/MPLS tunnels must follow in the logical layer is (3.7). The meaning of constraints into this group is analogous to those of (3.2), except for (i). Constraints (i) in (3.2) guarantee that when active, primary and secondary paths of each tunnel, count with enough capacity all along its way. However, the computation of such physically independent paths is shared with (3.3) and (3.4), which results in a very complex set of rules.

In this model, constraints group (i) only guarantees that such spare capacity would be available for each potential physical fault, given more freedom to construct these paths. However, this freedom is not absolute, and as it happened in ASP-MORNDP, we must notice that (3.1) and (3.7) are not independent, and additional constraints must be included to guarantee that logical links used for paths on each scenario, are not being affected by the corresponding physical link failure.

$$ {}^{rs}x_{pq}^{ij} \leq 1 - y_{pq}^{ij} \qquad \forall rs{:}d_{rs}>0, \forall (pq)\in L, \forall (ij)\in P. \tag{3.8}$$

Given any failure scenario (ij), the group of constraints (3.8) simply prevents from using logical link (pq) to route any tunnel $({}^{rs}x_{pq}^{ij} = 0, \forall rs : d_{rs} > 0)$ on that scenario, when the failure affects this link (when $y_{pq}^{ij} = 1$).

As we mentioned, the objective function in this model as well as its linear equivalent, match exactly with those of former model. So now the complete MIP is the result of merging: (3.1), (3.5), (3.7) and (3.8). Let us call FRP-MORNDP (Free Routing Protection Multi-Overlay Resilient Network Design Problem) to the problem described within this section.

3.2.2 FRP-MORNDP exact solutions

We start this section by showing particular solutions for some simple but illustrative example cases. The first example has four nodes $V = \{v_1, v_2, v_3, v_4\}$, the physical layer is the cycle $(\mathcal{C}^4)$ while the logical layer is the complete-graph or clique $(\mathcal{K}^4)$. The remaining parameters are: $B = \{3\}$, $d_{pq} = 1$ for all p, q such that: $1 \leq p < q \leq 4$ and $l_{ij} = 1$ for every physical link (ij). The value of c_b is irrelevant in this case because there is only one bit-rate available.

The optimal solution found, uses all of the logical links. Figure 3.9 shows with dashed-blue lines the route followed by each lightpath over the physical cycle. This is an example where lightpaths' routes are not intuitive, even for a very simple input data set. Before proceeding with the next example, let us analyze whether this solution is feasible or not. Unlike ASP-MORNDP case -where values for variables were all detailed-, in this case we are giving a descriptive approach to check that all constraints are fulfilled.

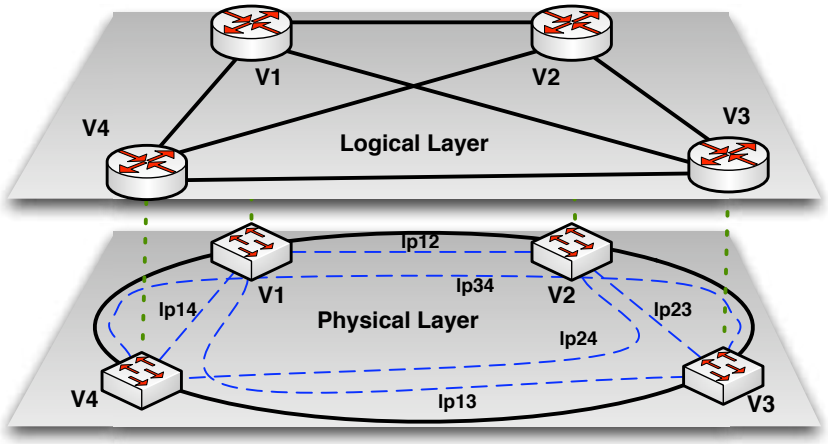


Figure 3.9: Optimal solution found for $\mathcal{K}^4$ over $\mathcal{C}^4$, with $d_{pq} = 1$ and $B = \{3\}$

Given the fact that all logical links are used in the construction (i.e. $\sum \tau^1_{pq} = 1$ for all $(pq) \in L$), is immediate that the paths outlined in Figure 3.9 to represent the route of the lightpaths (lp_{ij}), are consistent with constraints into group (3.1).

It is now necessary to check out that against every single failure in the physical network, the surviving logical network has capacity to route the tunnels, which would be equivalent to comply with (3.7) and (3.8). Because of symmetry matters there are only two representative failures for this example: $(3, 4)$ -which is equivalent with $(1, 2)$-, and $(1, 4)$ -which is equivalent with $(2, 3)$-.

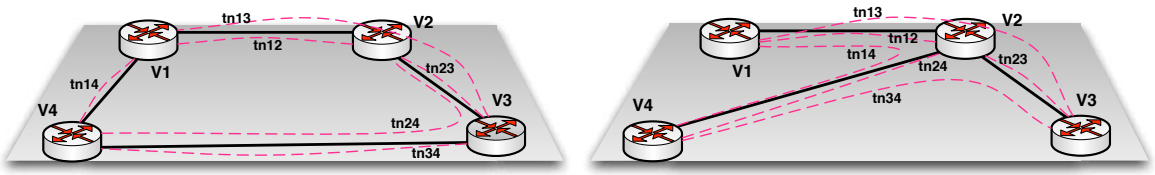


Figure 3.10: Paths followed by tunnels against failures in link $(3, 4)$ and $(1, 4)$

The surviving topology after a failure on physical link $(3, 4)$ is sketched on the left of Figure 3.10. The figure also shows a path, that could be followed by the tunnel (tn_{pq}) associated to each demand d_{pq}. Complementarily, the example corresponding to a failure on $(1, 4)$ is represented on the right of Figure 3.10. It is worth pointing out that the cases associated to a failure of the physical link (14) -or in (23)- are the most stressing ones, since they require full capacity of operational links.

The following example comprises seven nodes and explores again the clique-over-cycle case. The remaining parameters are analogous: $B = \{3\}$, $l_{ij} = 1$ for every physical link (ij), except for demands, that now are to/from one single node ($d_{1q} = 1$ for all q such that $1 < q \leq 7$).

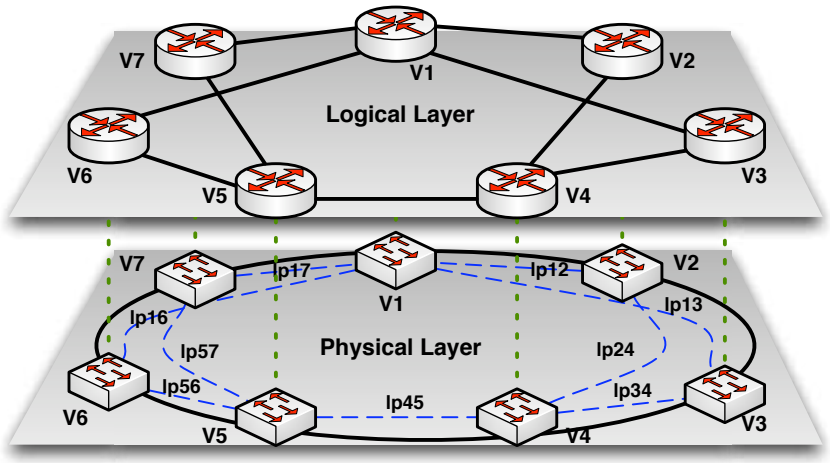


Figure 3.11: Optimal solution found for $\mathcal{K}^7$ over $\mathcal{C}^7$, with $d_{1q} = 1$ and $B = \{3\}$

Unlike the previous example, the optimal solution in this case (sketched in Figure 3.11) does not make use of the whole logical network. Although the route followed by each lightpath looks more natural in this example, it is not immediate why this set of logical links ought to be the appropriate to construct the optimal solution.

Through these two examples we attempted to show that solutions are not intuitive even for very simple cases. This was an initial concern when we analyzed existing models (Section 1.3.2.1 or Section 1.3.2.2). To find optimal solutions we used ILOG CPLEX v12.1. All computations were performed on a Linux machine with an INTEL CORE i3 Processor and 4GB of DDR3 RAM.

$\|V\|$	b_1 *range*	*number of variables*	*number of constraints*	*final time minus initial time (hh:mm:ss)*
5	2 - 6	1230	1640	00:00:00 - 000:00:11
6	3 - 9	3390	4035	00:00:02 - 000:19:31
7	2 - 12	7896	8652	00:00:05 - 087:19:05(*)
8	3 - 16	16296	16772	00:00:02 - 100:10:17(*)

Table 3.1: Overall results for some particular cases

(*)Note: The solver aborted for some intermediate cases.

Table 3.1 shows information for several test instances similar to those represented in Figure 3.9, that is: $\mathcal{K}^n$ over $\mathcal{C}^n$ with $d_{pq} = 1$ for all p, q such that $1 \leq p < q \leq n$, and where $l_{ij} = 1$ for every physical link (ij), for the unique capacity available b_1 ($|\hat{B}| = 1$). The lower and upper bounds used for b_1 in Table 3.1 were taken in accordance with Lemma 6 and Lemma 7. Before going into them, we present for FRP-MORNDP, the necessary condition already used during ASP-MORNDP analysis (Lemma 3). This result also makes use of the bonds extension for multilayer graphs established in Definition 23.

Lemma 5. *Given any solution to FRP-MORNDP, let $\bar{L} \subseteq L$ be the subset of arcs with positive capacities. In order for this solution to be feasible, it must hold that for every* multilayer bond $bond_P$*, the condition:*

$$\sum_{p \in V', q \in V''} d_{pq} \leq b_{\bar{B}} \left\lfloor \frac{|bond_{\bar{L},P}|(|bond_P| - 1)}{|bond_P|} \right\rfloor \tag{3.9}$$

must be satisfied, where $b_{\bar{B}}$ is the maximum bit-rate available for dimensioning links and d_{pq} is the traffic demand between nodes p and q.

Proof. The proof is based on a particular case of the "Pigeonhole Principle". First of all we observe that if there is any demand between components V' and V'' then $|bond_P|$ must be greater or equal to 2; otherwise the term $|bond_P| - 1$ would avoid (3.9) to be satisfied. So physical 2-edge-connectivity is implicit.

Regardless of how links in $bond_{\bar{L},P}$ are implemented over the physical layer, their lightpaths must use some link of $bond_P$ to connect (V', P') with (V'', P''), because $bond_P$ is a cut-set. Then, there must exist at least one physical edge $e \in bond_P$ used at least $\lceil |bond_{\bar{L},P}|/|bond_P| \rceil$ times by lightpaths of $bond_{\bar{L},P}$ (Pigeonhole Principle).

As a consequence, if the physical link e fails, the remaining capacity to route traffic from (V', L') to (V'', L''), falls down to at most:

$$b_{\bar{B}}(|bond_{\bar{L},P}| - \lceil |bond_{\bar{L},P}|/|bond_P| \rceil).$$

If this capacity is below the traffic demand between components $(\sum_{p \in V', q \in V''} d_{pq})$, and since this pool of logical links must be traversed to implement demands ($|bond_{\bar{L},P}|$ is also a cut-set), the logical network cannot satisfy demands.

The previous condition is numerically equivalent to state that if:

$$\sum_{p \in V', q \in V''} d_{pq} > b_{\bar{B}} \left\lfloor \frac{|bond_{\bar{L},P}|(|bond_P| - 1)}{|bond_P|} \right\rfloor,$$

there is always an edge $e \in bond_P$ used so many times by lightpaths, that its fault leaves the resultant operational data network without enough spare capacity in $bond_{\bar{L},P}$ to route traffic between nodes of V' and V''. □

It is worth mention that the previous property it is a generalization of [Okamura 1981] to our two layers model.

As it happens with Lemma 3, the condition specified on Lemma 5 is not sufficient to guarantee a solution. However, finding representative counterexamples is much harder on this case. It is worth pointing out that counterexamples can be easily designed, when they aim into demands' values, like: $d_1 = 1.1$ and $d_2 = 0.9$, which cannot fit into links of capacity 1, although its semi-sum does it perfectly. Conversely, finding a counterexample that fails, not because a mismatch between demands values and links' capacities, is nontrivial.

Consider the following problem: $V = \{v_1, v_2, v_3, v_4\}$, $P = L = \{(v_1v_2), (v_2v_3), (v_3v_4), (v_4v_1), (v_1v_3)\}$, $\hat{B} = \{1\}$ and $d_{13} = d_{24} = 1$. Physical and Logical topologies are represented upon the left side of Figure 3.12, while the second half sketches the traffic demands.

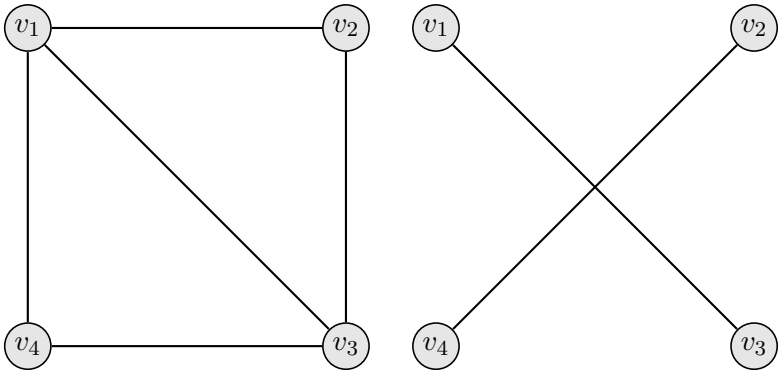


Figure 3.12: Counterexample to bonds conditions

Because of symmetry, $(\{v_1\}, \{v_2v_3v_4\})$, $(\{v_2\}, \{v_1v_3v_4\})$ and $(\{v_1v_2\}, \{v_3v_4\})$ comprise the representative failing scenarios for bonds. Let us start by checking out that (3.9) is satisfied for these scenarios:

$$1)1 \leq 1 \left\lfloor \frac{3(3-1)}{3} \right\rfloor = 2; \qquad 2)1 \leq 1 \left\lfloor \frac{2(2-1)}{2} \right\rfloor = 1, \qquad 3)2 \leq 1 \left\lfloor \frac{3(3-1)}{3} \right\rfloor = 2.$$

So the bond condition is always satisfied. Let us suppose that some candidate solution is given for this instance. Regardless of how lightpaths are implemented, there must be some physical link which fault affect logical link v_1v_3 lowering down the operational logical layer to at most $\mathcal{C}^4$. Since paths for demands from v_1 to v_3 and from v_2 to v_4, must cross into some surviving logical link and $\hat{B} = \{1\}$, this reduced logical network cannot satisfy both demands.

Lemma 6. *Given $\mathcal{C}^n$ as the physical layer and any logical layer L such that: $\mathcal{C}^n \subseteq L$, if demands conform to $d_{pq} \leq D$ is always possible to find minimal feasible solutions when either: $b_{\bar{B}} = Dn^2/4$ and n is even, or $b_{\bar{B}} = D(n^2-1)/4$ and n is odd. Moreover, the optimal solution in both cases reduces to use $\mathcal{C}^n$ as the logical layer.*

All the steps of this proof are analogous to those of Lemma 6. The lowest computation times in Table 3.1 were found for extreme cases covered by Lemma 6. Other instances with very low computation times are covered by the following result.

Lemma 7. *Given $\mathcal{K}^n$ and $\mathcal{C}^n$ respectively as logical and physical layers, and if demands conform to: $d_{pq} \leq D$ is always possible to find minimal feasible solutions when either: $b_{\bar{B}} = 2D$ and n is odd, or when $b_{\bar{B}} = 3D$ and n is even. Moreover, the solution when $d_{pq} = D$ and n is odd requires the usage of all the links of $\mathcal{K}^n$, whereas if n is even only diagonal links can be discarded, except for $n = 4$.*

Proof. This proof has several steps. First of all let us observe that $d_{pq} = D$ is the hardest demand case, so proving that if $d_{pq} = D$ and n is odd, the entire $\mathcal{K}^n$ with links dimensioned with a capacity $2D$ is an optimal solution for the logical layer, guarantee both: feasibility (when $d_{pq} < D$) and the optimality (when $d_{pq} = D$) for the odd case. Analogously, for n even we prove that: $\mathcal{K}^n$ minus diagonals links, dimensioned with a capacity $3D$ is optimal.

The next part of the proof consists in determining lower bounds for the capacity $b_{\bar{B}}$ and the number of logical links to use when $d_{pq} = D$. Afterwards, we show how to construct a feasible solution using that bounds. The minimal nature of that construction will close the optimality of the solution.

To determine lower bounds we apply Lemma 5 using the bond defined by any node (e.g.: v_1) to its complement. Since we are seeking for lowest values for $b_{\bar{B}}$ the maximum degree of logical connectivity will be allowed. Since v_1 has 2 neighbors in the physical layer and $(n-1)$ neighbors in the logical one, using Lemma 5 we have that: $D(n-1) \leq b_{\bar{B}} \lfloor (n-1)/2 \rfloor$. If n is odd this inequality turns out to be $D(n-1) \leq b_{\bar{B}}(n-1)/2$ and $b_{\bar{B}}$ must satisfy: $b_{\bar{B}} \geq 2D$ for the existence of solutions.

On the other hand, if n is even the inequality converts into $D(n-1) \leq b_{\bar{B}}(n-2)/2$ and $b_{\bar{B}}$ must satisfy: $b_{\bar{B}} \geq 2D(n-1)/(n-2)$. Since demand between each pair of nodes is D and our problem does not allow splitting traffic through more than one path, the capacity of any logical link has to be an integer multiple of D to be useful. In other words: no tunnel can fit within a fraction of D. The first integer multiple of D greater or equal to $2D(n-1)/(n-2)$ is $3D$, so practical solutions actually require $b_{\bar{B}} \geq 3D$.

It is now the time to prove that such values allow building feasible solutions. if $n = 3$ then $\mathcal{K}^n$ matches $\mathcal{C}^n$ (links of both layers are one-to-one), and lightpaths must be implemented by the homologous physical link. Hence, whenever a link fails the only way to deliver traffic is across the link with the remaining logical neighbor and capacity $2D$ is enough for that.

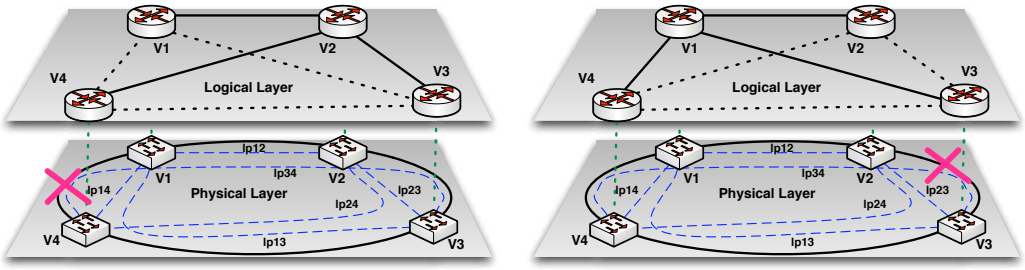


Figure 3.13: How failures on links $(1,4)$ or $(2,3)$ affect the solution for $\mathcal{K}^4$ over $\mathcal{C}^4$

The case $n = 4$ is out of the rule. Figure 3.9 shows the best choice for the path of each lightpath (found with CPLEX) under these premisses. The failure scenarios that affect most links are: (v_1v_4) and (v_2v_3), reducing the logical layer to those

sketched in Figure 3.13. Figure 3.10 shows paths for tunnels to sustain feasibility when $b_{\bar{B}} = 3D$.

For values of n greater than 4, a more general rule must be used. Let us guess that $n > 4$ and odd, so $n = 2k + 1$. We propose a logical network that uses all of the logical links ($\mathcal{K}^n$). Besides, the physical implementation uses the lowest number of physical hops to implement each lightpath. Since n is odd such physical mapping is unique.

To set-up the paths followed by tunnels in the non-faulty state, the direct route may be used -all logical nodes are neighbors-. For simplicity we assume that adjacent nodes are numbered consecutively, like: $(v_0, \ldots, v_{2k})$. Due to the fact that $\mathcal{C}^n$ and $\mathcal{K}^n$ have cyclic symmetry, we can analyze a particular physical link failure without loss of generality. Let (v_0, v_{2k}) be the physical link under a fault. Taking as a referential representation that of Figure 3.14, a fault into this physical link affects all logical links between the first and the second halves of $(v_0, \ldots, v_{2k})$, whose clockwise distance is grater or equal to $k+1$, because their counterclockwise paths are shorter and thereby used by lightpaths. That is, links of the form $(v_{i'}, v_{i''})$ where:

$$\left.\begin{array}{rcl} i' & = & 0, 1, \ldots, k-1 \\ i'' & \in & \{k + i' + 1, \ldots, 2k-1, 2k\} \end{array}\right\} \tag{3.10}$$

Upon the left half of Figure 3.14 we represented an example of the effects of this fault over the logical layer, for $n = 5$ ($k = 2$). Bold-blue lines highlight affected logical links.

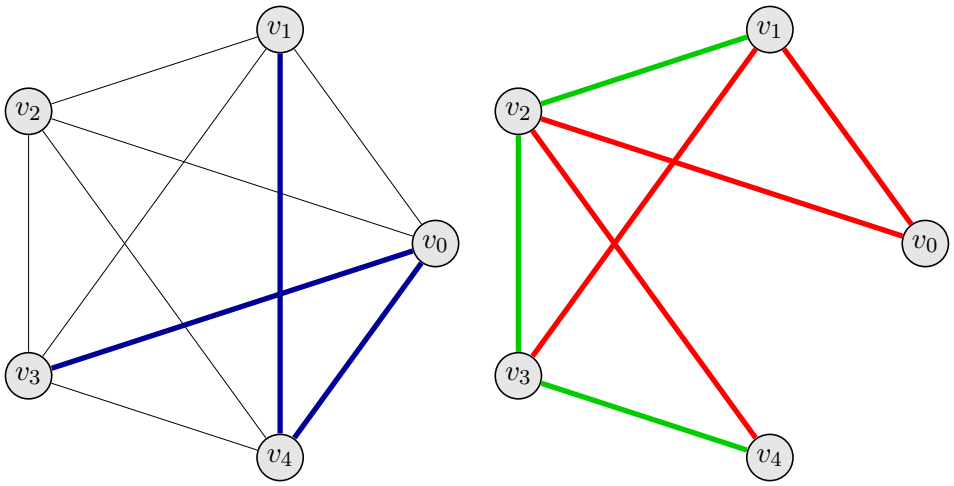


Figure 3.14: Example construction for $n = 5$.

For those tunnels which are not affected by the fault, the direct logical route is preserved. Then the surviving logical links still have a gap of capacity (spare capacity) of magnitude D, and we must find a way to deploy affected tunnels over them. The strategy is the following: take affected nodes in decreasing order of severity and for each one attempt to detour all its affected tunnels in the minimum

number of hops. This is a greedy strategy, so when viable is that which uses the minimum number of resources.

This can be accomplished in two hops from v_0 -the most severely affected- and in three hops for the remaining nodes. The alternative route for those tunnels that followed a path of the form $(v_0, v_{i''})$ now switches into: $\{(v_0, v_{i''-k}), (v_{i''-k}, v_{i''})\}$ where $i'' = k+1, \ldots, 2k-1, 2k$. All the remaining faulty tunnels would be detoured in three hops according to the following rule: a faulty tunnel $(v_{i'}, v_{i''})$ will follow the path $\{(v_{i'}, v_{i''-k}), (v_{i''-k}, v_{i''-i'}), (v_{i''-i'}, v_{i''})\}$, $i'' = k + i' + 1, \ldots, 2k-1, 2k$.

For instance, when $n = 5$ the new paths configuration is also sketched in Figure 3.14 (right half). Red lines highlight new paths for affected tunnels ending at v_0, whereas green lines correspond to paths with an endpoint on v_1.

To be sure this construction is feasible we must check that: logical links are operational in this state and none is used more than once. Both are straightforward:

- Since all logical links of this construction are clockwise with a number of hops less or equal than k, they cannot intersect links of set (3.10) and must remain in operational state.
- The first logical link used to detour $(v_0, v_{i''})$ always starts at v_0, while the second leap always maps onto k physical hops. Since no link within the three hops group has such length or origin, both sets cannot intersect.
- It's immediate that within each set, logical links cannot repeat by construction.

The construction is based on the limits determined by bonds condition (necessary condition), so bond's capacity is at bottom. Because of the capacity selected for links all of them are necessary to reach bond's capacity. Hence, the full logical network is minimal. Besides, the construction balances the usage of physical links, so any physical failure tears down the same number of logical links: $k(k+1)/2$. Until now we haven't considered physical costs.

Let us suppose that there is a mapping for lightpaths of lower cost. This mapping must raise the utilization of one physical link over the others, and against a failure on this link the surviving logical layer would not have enough capacity to accommodate demands. Regardless of physical costs, there is no margin to change lightpaths mapping, so the construction is optimal for $d_{pq} = D$.

The final step of this proof is for n greater than 4 and even. Let us say $n = 2k$. As it was seen, $b_{\bar{B}} = 3D$ is a lower bound for the capacity so we shall attempt to use this value as a starting one. Unlike the n odd case, in this, we opted by suppressing diagonal links of the logical layer. This decision keeps univocal the election of the lowest number of hops to implement lightpaths and besides eliminates those links potentially most expensive. The remaining aspects of the construction stay equal, except for some paths followed by the tunnels within the logical layer.

Since in this construction we cannot use diagonals, the traffic between v_i and v_{i+k} $(0 \leq i \leq k-1)$ will be routed through $\{(v_i, v_{i+k-1}), (v_{i+k-1}, v_{i+k})\}$, $i = 0, \ldots, k-1$. This leaves a spare capacity of $2D$ or $3D$ on all of the logical links.

Instead of going into details of the construction of tunnels for a representative fault on a physical link (v_0, v_{2k-1}), we present an illustrative workaround. Figure 3.15 outlines in its first half a diagram for the logical layer when $n = 6$, and remarks with blue lines the logical links affected by a failure of the physical link (v_0, v_5). The second half of the picture details the operational logical links in this faulty state. If instead of this figure we look back to the second picture of Figure 3.14, we might conclude that both graphs look very similar. In fact they only differ in v_5 and its operational links: (v_3, v_5) and (v_4, v_5).

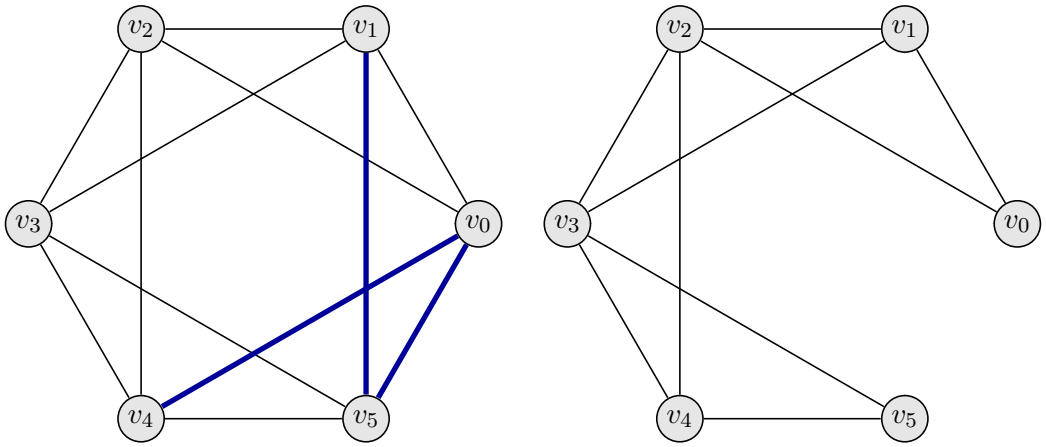


Figure 3.15: Example case for $n = 6$.

Moreover, in both cases demand between v_0 and v_4 must be reestablished, but instead of demands between v_0 and v_3, and between v_1 and v_4, in this case we have to figure out a solution to detour tunnels for pairs: (v_0, v_5) and (v_1, v_5). An easy workaround would be using the former solution for $n = 5$ and adapting it to the new case. This can be achieved by appending an extra hop (v_3, v_5) to the existing path between v_0 and v_3, and appending (v_4, v_5) to that between v_0 and v_4. Both are realizable because of extra spare capacity on this case. The previous ideas can easily generalized for any $k > 6$ and even. □

Corollary 1. *It is always possible to find minimal feasible solutions when:*

1) The logical layer is $\mathcal{K}^n$;

2) The physical layer is a 2-edge-connected graph G (on the n vertices of the logical layer), such that each one of its k blocks *is a cycle;*

3) The demands conform to: $d_{pq} \leq D$;

4) Capacities comply either $b_{\bar{B}} = 3D$ when n is even or $b_{\bar{B}} = 2D$ when n is odd.

Proof. First of all, let us notice that any graph G in the condition of 2), is made up by identifying $k-1$ vertices from a cycle of $n+k-1$ vertices. These vertex identifications originate the cut-vertices of G. An example instance of such G is represented on the right of Figure 3.16. A cycle, which can generate this instance after a selective identification process (marked with thin dashed lines), is sketched on the left half of the same figure. Within this proof, we refer to such a cycle with the list of identifications as an *expansion* of G.

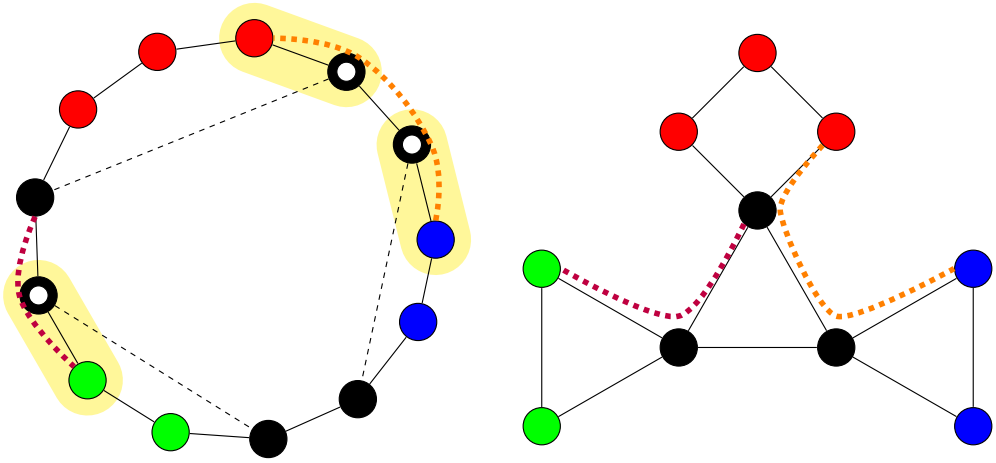


Figure 3.16: Example case for $n = 10$, $k = 4$.

It is tempting the idea of simply applying Theorem 7 over the expanded $\mathcal{C}^{n+k-1}$. However, this cannot be done directly because nodes upon both layers do not match after expansion. In other terms only one on each set of identified physical nodes can have an associated logical node. This can nonetheless be fixed easily by contracting $k-1$ nodes. The first step consists in assigning -arbitrarily- the logical node corresponding to each expanded set to one of the physical nodes. The remaining $k-1$ logically unpaired nodes are recursively contracted with neighbors in $\mathcal{C}^{n+k-1}$ until get to $\mathcal{C}^n$. For instance in Figure 3.16 those nodes without a logical peer are marked with a white dot into the middle. Besides, in this example, those nodes selected for being contracted are shadowed with pale yellow.

Once in this state we can apply Theorem 7 construction to determine capacities and routing configurations on both layers. To roll back this configuration over $\mathcal{C}^{n+k-1}$ we need to reinsert those edges contracted that would be over the path. This is represented in Figure 3.16 using bold-dashed purple and orange curves.

Since edges of $\mathcal{C}^{n+k-1}$ after identifying the corresponding vertices, are the same edges of G, a failure in any edge of G produce the same effect than in the corresponding edge of $\mathcal{C}^{n+k-1}$. Therefore replicating paths over the logical layer following the per-scenario path configuration of each LSP found for Lemma 7 works fine. □

3.2.3 FRP-MORNDP complexity analysis

Theorem 8. *FRP-MORNDP is a relaxation of the ASP-MORNDP problem.*

Proof. The proof consists in showing how a feasible solution for FRP-MORNDP can be made up from a feasible solution of ASP-MORNDP, conserving the cost of the former. First all, let us recall that ASP-MORNDP is the result of combining constraints groups: (3.1), (3.2), (3.3), (3.4) and (3.5), whereas FRP-MORNDP arises from (3.1), (3.5), (3.7) and (3.8). Equations into (3.1) and (3.5) are common to both problems, so are decision variables: τ_{pq}^{b}, y_{pq}^{ij} and ${}^{b}\eta_{pq}^{ij}$, which are the only set of variables that form part of the objective function. The remaining variables of ASP-MORNDP: ${}^{rs}x_{pq}^{h}$, ${}^{rs}\mu_{p}^{h}$, ${}^{rs}\lambda_{pq}^{ij}$, ${}^{rs}\dot{z}_{pq}^{ij}$ and ${}^{rs}\ddot{z}_{pq}^{ij}$, and of FRP-MORNDP: ${}^{rs}x_{pq}^{ij}$ and ${}^{rs}\mu_{p}^{ij}$, are basically auxiliary variables, used to limit the routing search space over the logical layer in order to keep consistency for tunnels' implementations.

Thus, given a feasible solution for ASP-MORNDP, it is immediate that copying values of variables τ_{pq}^{b}, y_{pq}^{ij} and ${}^{b}\eta_{pq}^{ij}$ preserves the cost and compliance for equations into groups (3.1) and (3.5). Given the complementary set of values for remaining variables of ASP-MORNDP: ${}^{rs}x_{pq}^{h}$, ${}^{rs}\mu_{p}^{h}$, ${}^{rs}\lambda_{pq}^{ij}$, ${}^{rs}\dot{z}_{pq}^{ij}$ and ${}^{rs}\ddot{z}_{pq}^{ij}$, for each $d_{rs} > 0$ let us define $LP(rs, h)$ as the set of physical links (ij) such that $y_{ij}^{pq} = 1$ for some logical link (pq) where ${}^{rs}x_{pq}^{h} = 1$. That is, $LP(rs, h)$ is the set of physical links used by the lightpaths of those logical links that support respectively the primary or secondary path for the tunnel corresponding to demand $d_{rs} > 0$.

Let us concentrate on how ${}^{rs}x_{pq}^{ij}$ and ${}^{rs}\mu_{p}^{ij}$ are defined. Given $d_{rs} > 0$ the rule of construction is the following:

$$ {}^{rs}x_{pq}^{ij} = \begin{cases} {}^{rs}x_{pq}^{1} \text{ when } (ij) \notin LP(rs, 1) \\ {}^{rs}x_{pq}^{2} \text{ when } (ij) \in LP(rs, 1) \end{cases} \tag{3.11} $$

$$ {}^{rs}\mu_{p}^{ij} = \begin{cases} {}^{rs}\mu_{p}^{1} \text{ when } (ij) \notin LP(rs, 1) \\ {}^{rs}\mu_{p}^{2} \text{ when } (ij) \in LP(rs, 1) \end{cases} \tag{3.12} $$

Since pair $({}^{rs}x_{pq}^{ij}, {}^{rs}\mu_{p}^{ij})$ matches either: ${}^{rs}x_{pq}^{1}$ and ${}^{rs}\mu_{p}^{1}$, or ${}^{rs}x_{pq}^{2}$ and ${}^{rs}\mu_{p}^{2}$, then equations (ii) to (vi) of (3.2) and (3.7) are paired and thus satisfied simultaneously. Regarding the equation (i) of (3.7), and given any physical and logical links: (ij) and (pq), let us guess that ${}^{rs}x_{pq}^{ij} = 1$. Hence, either ${}^{rs}x_{pq}^{1} = 1$ or ${}^{rs}x_{pq}^{2} = 1$, but no both simultaneously; otherwise, primary and secondary paths would share a logical link (pq), and thus a physical one, which is forbidden because of (3.3).

When ${}^{rs}x_{pq}^{1} = 1$ then $(ij) \notin LP(rs, 1)$ because of (3.11), and for every logical link $(\bar{p}\bar{q})$ it holds that ${}^{rs}\lambda_{\bar{p}\bar{q}}^{ij} = 1$, which in turns determines that ${}^{rs}\dot{z}_{pq}^{ij} = 1$ because of (i) in (3.4). Conversely, when ${}^{rs}x_{pq}^{2} = 1$ then $(ij) \in LP(rs, 1)$, and exists $(\bar{p}\bar{q}) \in L$ such that ${}^{rs}\lambda_{\bar{p}\bar{q}}^{ij} = 0$, which translates into ${}^{rs}\ddot{z}_{pq}^{ij} = 1$ because of (ii) in (3.4).

As a corollary, ${}^{rs}x_{pq}^{ij} = 1$ implies ${}^{rs}\dot{z}_{pq}^{ij} + {}^{rs}\ddot{z}_{pq}^{ij} = 1$, and the following inequality is immediate: $\sum_{rs:d_{rs}>0} d_{rs} \cdot {}^{rs}x_{pq}^{ij} \leq \sum_{rs:d_{rs}>0} d_{rs} \cdot ({}^{rs}\dot{z}_{pq}^{ij} + {}^{rs}\ddot{z}_{pq}^{ij}) \leq \sum_{b \in \hat{B}} b \cdot \tau_{pq}^{b}$, so equation (i) of (3.7) is also satisfied.

Remains now to check out the fulfillment of (3.8). If $y_{pq}^{ij} = 1$ and ${}^{rs}x_{pq}^{ij} = 1$, the value of ${}^{rs}x_{pq}^{ij}$ cannot be determined by the first entry in (3.11), because in such case (ij) would be in $LP(rs, 1)$; so it must be $y_{pq}^{ij} = 1$ and ${}^{rs}x_{pq}^{2} = 1$, which means that (ij) is in $LP(rs, 2)$. Since the valid construction entry of (3.11) is the second, it must also stand that (ij) is in $LP(rs, 1)$, but that cannot happen because (3.3) avoids physical implementations of primary and secondary paths intersect each other. □

Relaxations usually are a practical way to find approximate solutions for the original problem; of course to do so they must be easier to solve than the original. Unfortunately, this is not the case. The problem FRP-MORNDP was as hard to tackle down with the tested metaheuristics as ASP-MORNDP was.

Theorem 9. *The subproblem of routing the logical layer over the physical one, turns the problem FRP-MORNDP NP-Hard.*

Proof. The proof lies under reduction of 2ECSS (Two-Edge-Connected Spanning Subgraph) to FRP-MORNDP, and almost matches that described for Theorem 6, since once the demands become irrelevant because of the abundant capacity, the set of rules that mold both mappings ((3.1) and (3.5)) are the same. The only differences worthwhile to be mentioned in order to recreate the proof are the following:

- By the end of the direct part (⇒) of the proof of Theorem 6, we find out a feasible instance for ASP-MORNDP made up from a feasible instance of 2ECSS. This construction is also valid here because as we just saw (Theorem 8), FRP-MORNDP is a relaxation of ASP-MORNDP.

- By the end of the reciprocal part (⇐) of the proof of Theorem 6, we conclude that an instance of 2ECSS is feasible, up from the fact that a feasible solution to ASP-MORNDP determines pairs of physically independent paths between nodes, which conforms to Theorem 2.

  For this proof we can substitute the former argument by this: "Given any cut-set of P, this is also a cut-set for L because logical and physical constructions match. Thus it is a *multilayer bond*. Applying (3.9) (Lemma 5) and since both layers match, we get to the following inequality: $0 < \sum_{p\in V', q\in V''} d_{pq} \leq N(N-1)/2 \cdot (|bond_P| - 1)$, which results into $|bond_P| \geq 2$ for any physical bond $bond_P$. This fact together with Theorem 5 closes the two-edge-connectivity of the physical layer, and the feasibility of the construction."

The rest of the arguments remain unchanged. □

Theorem 10. *The subproblem of routing the logical layer over the physical one, turns NP-Hard the problem FRP-MORNDP.*

Proof. Most of the steps of this proof are analogous to those of Theorem 7, except for a minor detail into the direct part (⇒). Actually, this proof is easier than the former, because here, the Lemma 5 matches perfectly and it is not necessary an alternate construction of paths when some a_i is grater than $b_1/2$.

The idea is as follows (refer to the left half of Figure 3.8): "We are taking the partition which solves NPP as a basic assignment to balance paths between $(v_D v_{H_1})$ and $(v_D v_{H_2})$. Coming from v_D and once in v_{H_1} or v_{H_2}, the path is terminated directly into the corresponding node (some v_i). When a fault on any $(v_{H_x} v_i)$ affects a tunnel a *global detour* can be constructed switching paths assignment between $(v_{H_1} v_D)$ and $(v_{H_2} v_D)$. When the fault arises on $(v_{H_1} v_D)$ or $(v_{H_2} v_D)$, the affected paths can always be rebuilt across: $(v_D v_{A_1})$, $(v_{A_1} v_{H_2})$ and $(v_{H_2} v_i)$."

It is worth pointing out that this construction isn't possible for ASP-MORNDP, because requires three paths for tunnels. □

3.3 Summary

The set of protection mechanisms realizable over IP/MPLS technology is much richer than those of traditional transport networks. IP/MPLS allows to combine the rapid response of local protection (fast reroute) with the efficiency of a globally coordinated deployment (traffic engineering). We center our analysis upon two models, both sustainable over features widely supported by equipment providers.

Our first model (ASP-MORNDP) embeds the spirit of APS protection, although in this case active/standby resources are point-to-point provisioned, rather than over a tandem of locally protected rings. Conversely, the second model (FRP-MORNDP) grants absolute freedom for selecting paths, without imposing constraints upon them other than those coming from the capacities of links. In a sense, the second model extends the protection schemes of other models or technologies, and constitutes in fact a relaxation of the first one.

Both models are in general computationally hard to solve. Even so, on both cases exact solutions were found for a family of simple but useful instances, and also for a set of illustrative numerical examples. Besides, for more general cases a necessary condition was established, which proved to be very accurate upon real-world instances, capturing the essence of solutions (Chapter 5).

Overall results of our experiments with traditional methods discouraged us to use them for tackle down real-world applications. Instead, we implemented algorithms based on metaheuristics with promising results. The remaining of this document describes two metaheuristic implementations, suitable to find good solution for real applications as well as the numerical results obtained for them.

Chapter 4

Mastering Complexity

Contents

In Section 1.3.2 we summarized main characteristics of related works. Let us recall some remarkable elements of them as well as others proper of our own experience that guided these decisions afterwards:

- Existing models do not widely explore paths' constructions, neither for lightpaths nor tunnels; instead a set of candidates is pre-computed. Paths construction plays an essential role into our models, and constitutes the root of the intrinsic complexity for both of them (Theorem 6, Theorem 7, Theorem 9 and Theorem 10).

- Heuristics used to find solutions are based on traditional methods: Linear Relaxation, Branch-and-Cut, Branch-and-Bound, Lagrangian Relaxation or Optimizing Layers Separately. Except for a handful of cases, the size of the instances used upon referenced algorithms was below 15 nodes, and then are far from some of our needs.

- Our research group has good academic background on solving network design problems using metaheuristics ([Robledo 2005]).

Because of the previous facts, from the early stages of this work we perceived Metaheuristics as the tool to find solutions. We tried some of them before adopting definite ones; their main aspects and results are summarized next:

i) **Two Stages Approximation -** Instead of attempting to optimize the entire FRP-MORNDP problem at once, this approximation splits the problem into two non-independent stages. The first stage consists in constructing lightpaths for the logical links, whereas the second stage focuses on demands and how tunnels can be routed on failure scenarios. CPLEX was used as the central optimization engine for this heuristic. The approach found solutions for the simplest scenarios of ANTEL (Section 5.2). However, to do so network instances had to be simplified and split into two regions, which are optimized separately. Unfortunately, most remarkable improvements upon solutions (found with other methods) are lost after this process of simplification, so we do not go deeper upon details about this algorithm. For further information on this heuristic, see [Parodi 2011].

ii) **Genetic Algorithms -** The application of sequential and parallel evolutionary algorithms to ASP-MORNDP problem shows promising results. To be practical the model has to limit the number of logical routing configurations, and this is the reason why this implementation doesn't cover instances of FRP-MORNDP. This algorithm was capable of finding solutions for most ANTEL's scenarios and all those of RAU (Section 5.1). Additional details for this implementation are covered in Section 4.1.

iii) **Variable Neighborhood Search -** Unlike the Two Stages Approximation example, in this case, the metaheuristic is used to optimize the FRP-MORNDP problem as a whole (both layers simultaneously). However, to find solutions it was necessary a simplification process over instances as that depicted for i), i.e., splitting networks into regions. Thus this algorithm also lacks in finding major improvements for critical instances. We decided then to focus over other alternatives. For further information on the application of this metaheuristic to the problem see [Córez 2010].

iv) **Greedy Randomized Adaptive Search Procedure -** According on our benchmarks, this implementation proved -by far- to be the most suitable to find solutions for the real-world instances of FRP-MORNDP on both applications (RAU and ANTEL). Indeed, this was the only algorithm that found solutions for all the scenarios, and its solutions were those with the most important quality improvements, which it was expected because of its extended search-space (see Theorem 8). Additional details for this implementation are covered in Section 4.2.

4.1 Genetic algorithms

Within this section we describe the structure and abstract implementation for the family of metaheuristics, which according to our experience obtained the most promising results to find solutions for ASP-MORNDP instances. The algorithms described during this section were developed as a branch of the main research project

(carried out by this author). The additional branch was managed by Prof. Sergio Nesmachnow (Facultad de Ingeniería, Udelar), and carried out by an outstanding group of students: Leandro Gómez, Gastón Lasalt and Fernando Casalongue.

During Chapter 5, we refer to the results of these algorithms as obtained with Evolutionary Algorithms, although strictly speaking several complementary approaches were often used to find good quality solutions. The best result was always poked into the comparative for each instance, putting hence emphasis upon the performance of the model, rather than on details of the algorithm used. The metaheuristics implemented are: Evolutionary Algorithms (EA), Parallel Evolutionary Algorithms (PEA), Tabu Search (TS) and a hybrid of EA and TS. However, the TS that is used is indeed a mutation operator of the EA itself, only executed separately for very large instances (i.e. even indices of ANTEL's scenarios, Section 5.2), where the entire EA or even the PEA, could have taken too much time to complete.

All algorithms described into this section are implemented using MALLBA [Alba 2006], which is a framework with embedded algorithms for optimization that can deal with parallelism, in a user-friendly and efficient manner. Skeletons are implemented by a set of C++ classes that represent an abstraction of the entities participating in the resolution method: the *provided classes* (i.e. Solver and SetUpParams) implement internal aspects of the solver in a problem-independent way, and the *required classes* specify information specifically related to the problem. Each solver includes the required classes Problem and Solution, which encapsulate the problem-dependent entities needed by the resolution method. Using MALLBA allowed an efficient and reusable implementation of the EA and the parallel PEA applied to the ASP-MORNDP in this work.

4.1.1 Solution representation

The first thing we need to implement a *population based metaheuristic* -like an EA-, is a representation for its individuals. Since traditional encodings would require designing specific evolutionary operators to maintain the feasibility of solutions, a problem-dependent encoding was used to represent ASP-MORNDP solutions. For any instance of ASP-MORNDP, individuals are determined by: the capacities used to dimension logical links, the mapping of lightpaths over the physical layer, the primary and secondary paths of each tunnel over the logical layer.

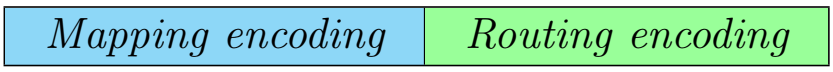


Figure 4.1: Encoding used for an individual

For these algorithms, we incrementally pick logical links up from the pool, on an on-demand fashion as tunnels are being assigned with paths. Once one of them is used, the only limit considered until the end of construction is that imposed by $b_{\bar{B}}$. On a final refinement, the capacity is assigned with the minimum necessary to fulfill demands assignment. Thus, capacities are inferred up from the configuration

for paths and lightpaths, and a solution is implicitly encoded by the routings of its terminal nodes. Two logical sub-encodings are used to represent each individual, as is schematized in Figure 4.1, i.e.: the *mapping encoding* and the *routing encoding*, and respectively correspond to paths selected either, for the implementation of lightpaths or for the primary and secondary paths of tunnels. For example purposes, let us a guess an instance where: $V = \{v_1, v_2, v_3, v_4\}$, $L = \mathcal{K}^4$, $P = L \backslash \{v_1 v_3\}$, $\hat{B} = \{1\}$, the only demands are $d_{13} = d_{24} = 1$, and $l_{ij} = 1$ for all physical links.

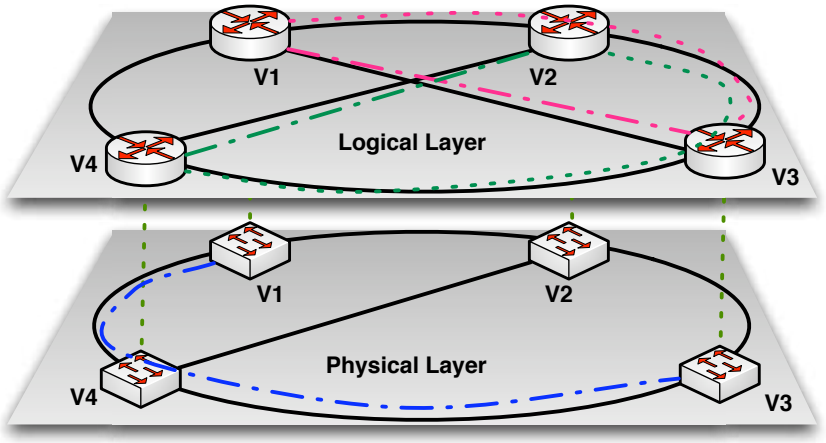


Figure 4.2: An abstract individual, valid for an instance

For such an instance, Figure 4.2 represents a feasible individual. On it and whenever possible, the mapping of lightpaths is direct. The only exception is for $v_1 v_3$, whose implementation is marked with a blue semi-dashed curve over the physical layer. Complementarily, purple and green semi-dashed curves over the logical layer, correspond to primary paths for demands d_{13} and d_{24} respectively. Dashed lines are used to indicate the path followed by the secondary path at each case. Since the logical link $v_1 v_4$ is not being used, we can dismiss it from the solution.

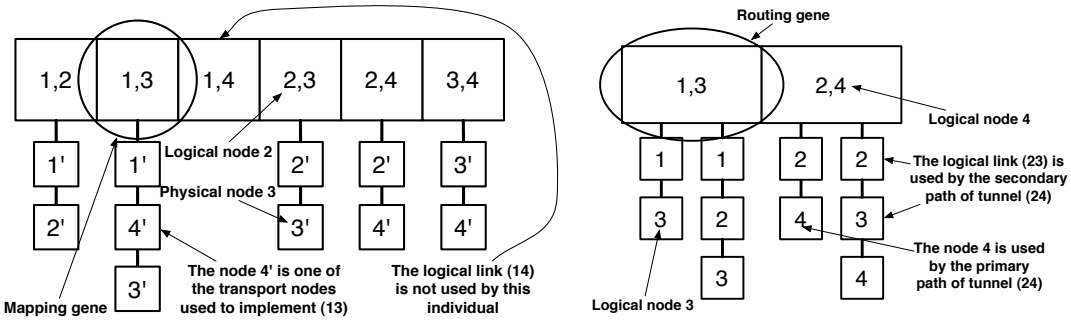


Figure 4.3: Encoding of an individual: *mapping* on left, *routing* on right

The Figure 4.3 shows the encoding corresponding to the individual described previously. The sequence of *mapping genes* appears on the left of the figure. Each mapping gene consists basically of an ordered list of the physical nodes used by the

lightpath, which is univocal regarding the physical links since this model does not allow multigraphs. From this encoding becomes immediate that the logical link v_1v_4 is not used in the construction. The sequence of *routing genes* appears on the right of the figure. For every non-null demand of the problem's instance, a pair of lists of logical nodes is attached, which represents the primary and secondary paths of the associated tunnel. Integrating both encodings we conclude that the individual is feasible. This is because active/standby pairs of routing genes are physically independent, and no single physical fault loads a logical link beyond 1, which is an available capacity.

4.1.2 Generating feasible solutions

Once established the encoding for each individual, the second thing every EA requires is a suitable population of them; just then, evolutionary operators can apply to produce an offspring.

The strategy chosen here consists in generating routing genes sequentially, taking demands in decreasing order of volume, and using a greedy randomized selection at each hop during the construction of primary and secondary paths. In the meanwhile -during the construction of the routing ones-, the creation of the mapping genes is indirectly triggered. The pseudo-code of Algorithm 6 corresponds to the procedure previously described.

Algorithm 6 Pseudo-code for createIndividual algorithm

Procedure createIndividual(V, L, P, $l : P \to \mathbb{R}_0^+$, $d : L \to \mathbb{R}_0^+$):

1: $mapping\_genes \leftarrow \emptyset$, $routing\_genes \leftarrow sortDemands(V, L, d)$;
2: $k \leftarrow 0$, $N_{gen} \leftarrow |routing\_genes|$;
3: **while** $k < N_{gen}$ **do**
4: $\quad rgene \leftarrow routing\_genes[k]$, $k++$;
5: $\quad$ **if** $incomplete(rgene)$ **then**
6: $\quad\quad attempts\_num \leftarrow 0$;
7: $\quad\quad$ **while** $incomplete(rgene)$ and $(attempts\_num < max\_attempts)$ **do**
8: $\quad\quad\quad buildRoutingGene(rgene, V, L, P, l, d)$;
9: $\quad\quad\quad attempts\_num++$;
10: $\quad\quad$ **if** $incomplete(rgene)$ **then**
11: $\quad\quad\quad resetTaintedRoutingGene(routing\_genes)$;
12: $\quad\quad\quad k \leftarrow 0$;
13: **return** $(mapping\_genes, routing\_genes)$.

The procedure $buildRoutingGene(rgene, V, L, P, l, d)$, attempts to build a path for both primary and secondary paths, and attach them to $rgene$. Primary path is built in first place. At each step during a path construction, this heuristic adds a new logical link such that: it supports the additional demand, it has not been used previously during this path construction; additionally, when the path corresponds to the secondary-path, physical independence is enforced. From all those physical

links satisfying previous constraints, one is picked up randomly. Probabilities are taken in inverse ratio to either the number of hops or the cost of the new link[1]. Whenever a yet unused logical link passes to integrate a logical path, a lightpath construction is launched, which we shall elaborate later on.

Optimality aside, the mere construction of feasible mappings and routings is a hard task. Let us recall that the mapping of lightpaths and the routings of tunnels, respectively constituted the core of proofs of Theorem 6 and Theorem 7. Taking demands in decreasing order helps to improve the effectiveness to build routings; the same idea guides our GRASP algorithm (see Section 4.2.2). Nevertheless, it would be naive to think that this is the only consideration to take care of. Algorithm 6 heeds the possibility that at certain step, paths cannot be built as a consequence of previous constructions. After trying $max\_attempts$ times to build paths for $rgene$ without success, those previously configured paths that blocked the way of $rgene$ most of the times are reset by $resetTaintedRoutingGene$ (i.e. their routing genes are erased), so the list $routing\_genes$ is scanned again all along. Mapping genes created because of paths of these routing genes are also reset. After repeating the while-loop (3) too many times, the process of creation is aborted.

Whenever called from within the procedure *buildRoutingGene*, the procedure *buildMappingGene* attempts to build a path for certain lightpath. The structure of this procedure matches the previous one, i.e., at each hop, physical links are filtered if they already are part of this path's construction. Additionally, while the secondary path is being built, links present into the primary one are discarded. Since longer lightpaths increase the objective cost, the weights for probabilities are in inverse ratio to the length of physical links considered. It is likely that a lightpath construction should be reset and started over, because a physical link chosen during early stages of the construction turns impossible the remaining of it. After repeating a lightpath implementation too many times, this process is aborted, which in turn aborts the routing gene construction (*buildRoutingGene*) that called it in the first place.

Hence, the construction of individuals is *recursive* rather than *iterative*, resembling a *backtracking* algorithm. It is worth pointing out that the greedy constructors described within this section, are also used to repair unfeasible solutions after applying the evolutionary operators.

4.1.3 Evolutionary operators

Now that we have a population, we only need evolutionary operators to wrap up the algorithm's design. The first one of them is a *fitness function*, necessary to compare the suitability of different individuals.

According on the problem we're dealing with (ASP-MORNDP), the immediate answer to this matter consists in using the cost as is determined by (3.5). Let us

[1]Both functions were used during the construction of solutions.

recall that this value is equal to the sum of the cost for each lightpath, which is in turn the product of the lightpath's length and a constant c_b, associated with the capacity chosen for the logical link.

The length of each lightpath is an immediate outcome of the mapping encoding, whereas the capacity necessary on each logical link, can be determined up from the maximum load supported on all physical link failures. Since active/standby paths are determined by the routing encoding, given any individual x, all of these values can be determined with a few computations, so does the total cost $C(x)$.

The fitness function f used within these algorithms is $f(x) = 1/(1+C(x))$. The selection is performed using a standard *roulette wheel*. The remaining operators deserve further detail.

4.1.3.1 Crossover

An ad-hoc crossover operator was designed, since the traditional ones do not assure feasibility for ASP-MORNDP solutions. The offspring is generated by copying a routing gene, and whenever possible all dependent mapping genes, from a randomly selected parent. The crossover verifies that the primary and the alternative paths are independent in the offspring (they could intersect, as some transport mappings in the offspring can be different to the original ones in the parent). This process is repeated until the routing gene is successfully copied or no more parents are left. If the routing gene insertion fails, the routing encoding will remain incomplete and the new solution will not be feasible. The *greedy randomized solution generator/corrector* (see Section 4.1.2) is then applied to complete the missing parts of the individual.

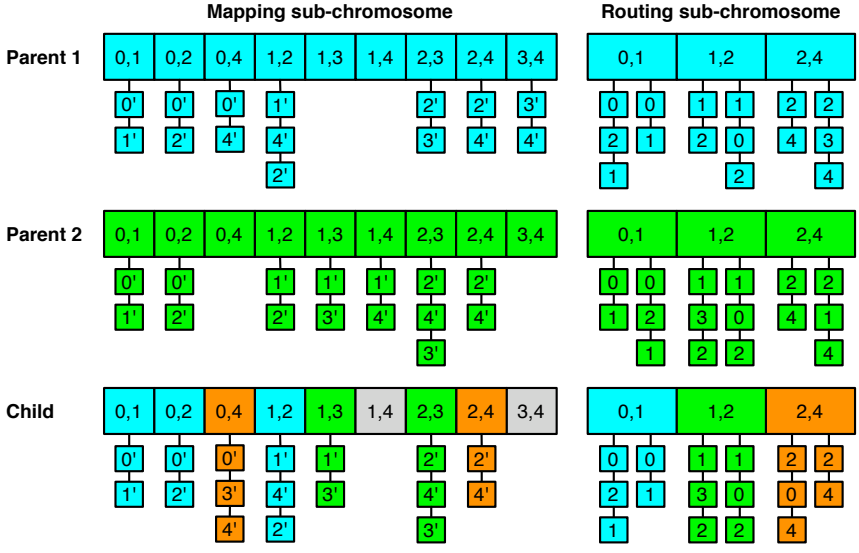


Figure 4.4: Crossover example

Figure 4.4 shows an example of the crossover operator. Routing gene [0,1] is copied from *Parent 1* into the outcome (*Child*), and with it, the mapping genes

associated to primary and secondary paths, i.e.: [0,1], [0,2] and [1,2]. All these genes are highlighted with cyan. Afterwards, an aleatory selection determines that routing gene [1,2] coming from *Parent 2* is selected for the *Child*. Mapping genes [1,3] and [2,3] are then copied into the new encoding, but [0,1] and [0,2] are inherited from the previous step. By fortune, former mapping encoding does not conflict with new routing ones. All those genes marked with green are inherited from *Parent 2*.

At this point and regardless of which parent conveys its routing gene [2,4] to the progeny, the construction would be infeasible. The reason is the following: both parents use [2,4] as the primary path, and on both encodings this logical link is implemented by the homologous physical link. Conversely, the first hop of both secondary paths is already encoded into the child -from previous iterations-, and on both cases using the physical link [2,4], which flaws the necessary physical independence. Hence, the routing gene [2,4] as well as mapping genes [0,4] and [2,4], are created using the generator/corrector greedy algorithm described in Section 4.1.2. All these *repaired genes* are highlighted in orange.

Mapping genes [1,4] and [3,4] are not inherited from parents nor are created by the repair process, so they remain empty in the progeny.

4.1.3.2 Mutation

Five mutation operators are used to introduce diversity and to improve the quality of the solutions of the EA. They are:

data layer mutation - It rebuilds a random gene of the routing encoding, mainly modifying the paths in the logical layer.

After selecting a routing gene, this mutation empties it and attempts to construct a new pair of primary/secondary paths, using the existing mapping of lightpaths for this individual. Primary and secondary paths are appended hop-by-hop by a greedy randomized process, which preserves physical independence at each step. Probabilities are taken in inverse ratio to either the number of hops or the cost of the new logical link (as in Section 4.1.2).

Mapping genes are not affected, unless they are solely used by the the gene that has been drawn to be rebuilt. In this case, the corresponding mapping genes are erased and the construction is carried out -from scratch- by the generator/corrector greedy algorithm described in Section 4.1.2.

transport layer mutation - It changes the mapping of lightpaths for some randomly selected mapping genes.

Once a mapping gene has been drawn, a portion -eventually the entire- of the path is taken, and it is hop-by-hop rebuilt by a greedy randomized process, which filters all those physical links whose insertion would affect physical independence of routing genes that use this mapping. Probabilities are in inverse ratio to the length of physical links considered.

tabu link mutation - It randomly selects a *tabu data link*, which is eliminated along with its associated genes from the solution. The randomized solution generator/corrector is then used to rebuild the missing parts of the solution, without using the tabu link.

best data layer mutation - It is a local search operator that uses the *data layer mutation* as a base for searching in the neighborhood of a solution. It modifies the current solution 30 times and the best result is returned.

best transport layer mutation - It is another local search operator that uses the *transport layer mutation* for the neighborhood search. Like the previous operator, it performs 20 changes and returns the best result found.

4.1.3.3 Parameters calibration

A calibration analysis was performed over six randomly generated small[2] ASP-MORNDP instances to determine the best parameter values for the GA operators. The studied parameters were: population size ($\#pop$), crossover probability (p_C), data layer mutation probability (p_{dM}), transport layer mutation probability (p_{tM}), tabu link mutation probability (p_{tlM}), best data layer mutation probability (p_{bdM}), best transport layer mutation probability (p_{btM}) and number of generations ($\#gen$).

The best results were obtained when using this configuration of parameters: $\#pop = 50$ individuals, $p_C = 0.60$, $p_{dM} = 0.01$, $p_{tM} = 0.01$, $p_{tlM} = 0.01$, $p_{bdM} = 0.70$, $p_{btM} = 0.70$ and $\#gen = 2000$.

From each generation to the following, a standard roulette wheel is used to select pairs of individuals (parents), over which previous operators are applied until the population raises from 50 to 200 individuals (300%). Before going into the next generation another roulette wheel is applied to select only 50 individuals. Probabilities on both roulette wheels are proportional to the fitness function described in the begging of Section 4.1.3.

4.1.4 Derived algorithms

As we mentioned at the begging of this section, besides of the sequential EA itself, already described, several algorithms were implemented using this EA as a skeleton. They are: Tabu Search (TS), a hybrid of EA and TS, and a Parallel Evolutionary Algorithm (PEA).

The TS uses most of the structure of EA, including the routing and mapping encodings. The local search used by the TS is the *best data layer mutation* (Section 4.1.3.3). The generator/corrector greedy algorithm (Section 4.1.2) is also used. The novel component integrated here is the *tabu list*, where forbidden solutions are kept.

[2] Of around 15 nodes each.

After a calibration process, the size of this list was set to 100 (ts_{tl}), that of the neighborhood to 50 (ts_{nh}), whereas the number of iterations was set to 5000 (ts_{it}).

Complementarily, a hybrid of EA and TS was tested. This algorithm simply integrated an extra mutation within the offspring construction, which essentially is the TS previously commented. However, due to the relative higher computational cost of this mutation a parameters re-calibration was issued. Those parameters that changed from former EA and TS versions are: $p_{bdM} = 0.90$, $p_{btM} = 0.50$, $ts_{tl} = 10$, $ts_{nh} = 40$ and $ts_{it} = 60$. An additional parameter *tabu mutation probability* (p_{tbM}) was introduced and its setting is $p_{tbM} = 0.005$.

Finally, we are giving the main characteristics of the PEA implementation. Regarding the classification of paradigms to design PEAs (i.e.: master-slave, distributed subpopulations and cellular, see Section 2.1.3.1), and since the fitness function is very cheap to compute, the *distributed subpopulations* approach was chosen. The number of isles/demes to distribute the population was set to 4, with 13 per-deme-individuals, and a per-deme-offspring size of 39. The migration of individuals between demes is realized synchronically, after three iterations. Only one individual from each deme is migrated, selected after a tournament. The remaining parameters within each deme, are copied from the sequential EA.

Table 4.1 summarizes some performance metrics for the parallel model, through the values for the *speedup* and *computational efficiency* metrics. The speedup (S_P) is defined as the ratio between the execution time of the sequential and the parallel GA, when using P computational resources. Conversely, the computational efficiency (e_P) is a normalized value of the speedup, which considers the number of computational resources ($e_P = S_P/P$).

scenario	*speedup* (S_4)	*efficiency* (e_4)
07	3.14	0.79
03	3.38	0.85
11	2.37	0.59

Table 4.1: Performance metrics for the PEA on ANTEL's scenarios

This table reports the speedup and the computational efficiency of the PEA, when using four computing resources. Test instances correspond to representative ANTEL's scenarios (Section 5.2). The results in Table 4.1 indicate that the parallel GA has a sub-linear speedup behavior, but very good efficiency values (i.e. near 0.8) when using four computing resources. These efficiency values suggest that using a parallel GA model is an efficient and effective approach to solve the ASP-MORNDP.

To conclude this section we record that all experimental analysis commented here, were performed in Quad core Xeon E5430 servers at 2.66 GHz, with 8 GB RAM and Linux CentOS, from the Cluster FING infrastructure (see `http://www.fing.edu.uy/cluster`).

4.2 GRASP

We decided to use a metaheuristic algorithm based on GRASP to find good quality solutions for real-world instances of FRP-MORNDP. A high level block-diagram of our algorithm is shown in Figure 4.5.

As for every GRASP implementation, this algorithm has a loop with two phases: the *construction phase* builds a *randomized feasible solution*, from which a local minimum is found during the *local search phase*. This procedure is repeated $MaxIter$ times while the best overall solution is kept as the result. Further information and details in GRASP algorithms was described on Section 2.1.3.2.

The *initialization phase* performs computations whose results are invariants among iterations, like the shortest path and distance over the physical layer between each pair of nodes.

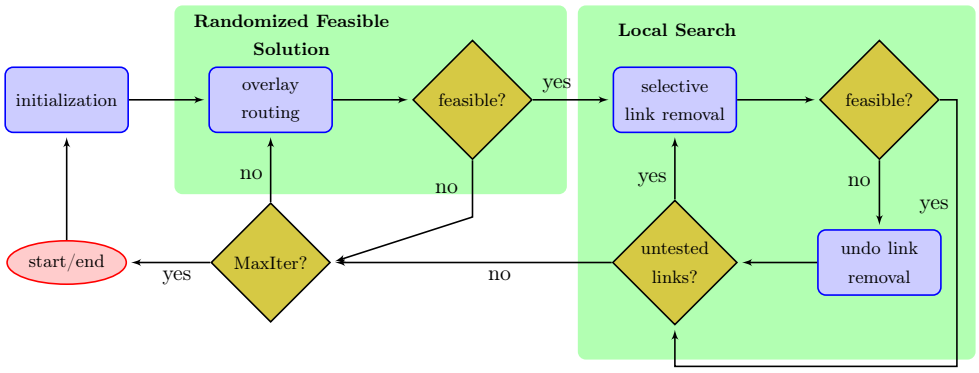


Figure 4.5: Block-diagram of the GRASP implementation used.

The other two blocks are more complex and deserve an elaborated analysis.

4.2.1 Construction Phase

The *randomized feasible solution phase* performs a heuristic low cost balanced routing of the logical layer over the physical one. The goal of this phase is finding a path for every lightpath, minimizing the number of physical links intersections. It is also desirable that the total cost is as low as possible but as a second priority.

Before going into details, let us highlight some main characteristics of this block. First of all, we decided to decrease the problem's complexity by eliminating the multiple capacity assignment for links (i.e. $|\hat{B}|$ reduces to 1). This simplifies considerably the construction of solutions[3] without mostly affecting the usefulness of them. We are designing backbone networks after all, so using the maximal possible capacity on links was desirable for both counterparts (ANTEL and RAU). In spite of that, finding solutions for FRP-MORNDP is still a challenging task. It suffices

[3]Whenever $|\hat{B}| = 1$, finding minimum cost for (3.5) is equivalent to find a deployment with the minimum total length for lightpaths mapping.

recalling that complexity analysis of Section 3.2.3 also integrated this premiss into its proofs. This block selects a path for the lightpath of each logical link. The task of determining which logical links are dispensable relies on the following stage (Section 4.2.3). Remains to be seen under which criteria are these lightpaths mappings constructed. Let us observe that as a consequence of Theorem 9 is expected for this problem to be hard in general. The proof of that property is based on the fact that lightpaths cannot intersect, but this is because of the reduction chosen, in which both layers match. A more general analysis for this sub-problem, covered by [Oellrich 2008], shows that finding a minimal length mapping of physically disjoint lightpaths over a network is in general an NP-Hard problem.

On other cases where the number of logical links is necessarily higher than the number of physical links, intersection of lightpaths is unavoidable. That would be the case for Lemma 7 or Corollary 1. However, even during the construction of feasible solutions onto both proofs, lightpaths mapping was chosen to minimize the number of physical intersections. Furthermore, the proof itself of the necessary condition for existence of solutions (Lemma 5) embeds the idea that lightpaths must be balanced over physical links on each bond (pigeonhole principle). Thus we adopted this as the primary goal during lightpaths construction.

Algorithm 7 Pseudo-code for overlayRouting algorithm

Procedure overlayRouting($(V, L), (V, P)$**,** $d : V \times V \to \mathbb{R}_0^+$**):**

1: $sol(e) \leftarrow \emptyset$, $pd(e) \leftarrow 1$, $\forall e \in L$;
2: **while** exists not-processed(v) **do**
3: Select v randomly;
4: $prob(vw) \leftarrow \frac{1}{d(v,w)}, \forall (vw) \in L \,/\, sol(vw) = \emptyset$;
5: Normalize *prob* such that: $\sum_{e \in L} prob(e) = 1$;
6: **while** exists $w \in V$ such that $((vw) \in L$ AND $sol(vw) = \emptyset)$ **do**
7: Draw such $w \in V$ randomly weighted by $prob(vw)$;
8: *shlp* $\leftarrow$ the shortest lightpath for (vw) without repeating physical links;
9: **if** $(shlp{=}\emptyset)$ **then**
10: $\mathrm{pd}(v, w) \leftarrow (1 + \sum_{e \in L} |sol(e) \cap \{(vw)\}|)^p$ for all $(vw) \in P$;
11: Restart *repeated physical links control window*;
12: **else**
13: $sol \leftarrow sol \cup \{shlp\}$;
14: **return** $sol : L \to 2^P$.

Besides and since the total length of lightpaths matches the objective to minimize in FRP-MORNDP[4], is desirable for this to be the lowest possible. We decided to used a heuristic for this challenging sub-problem. The algorithm implemented in module *overlay routing* is detailed in Algorithm 7. The parameters of this function are: the logical graph (V, L), the physical graph (V, P), and the minimum distance over the physical layer to connect each pair of nodes -computed during the ini-

[4]The factor $c_{b_{\hat{B}}}$ is common to all terms of (3.5) because $|\hat{B}| = 1$.

tialization phase-. The output is a mapping between logical links and the subset of physical links used by their lightpaths. In abstract, the strategy chosen in the heuristic described in Algorithm 7 is the following:

1) Nodes are taken randomly (e.g.: uniformly), and for each node its logical links are also taken randomly but with probabilities in inverse ratio to the minimal possible length of their lightpaths over the physical layer.

2) The heuristic iterates along a sequence of *control windows*. Within each one of them, new physical intersections are not allowed.

3) Instead of using the real distances of the physical links (l_{ij}), from this point on and until the next window, pseudo-distances $\bar{l}_{ij}$ will be used for all (ij) in P.

4) Prior to start routing lightpaths, all these pseudo-distances are set to 1. According to these new weights, logical links are routed following the minimal distance without repeating physical links among them (i.e. a classical CSPF).

5) Usually, after routing some lightpaths the set of not-yet-used physical links empties, and it is necessary to start over a new *control window* by filling again the not-yet-used set. Prior to do this, the pseudo-distances are updated using the following rule: $\bar{l}_{ij} = (1 + n_{ij})^p$ for some fixed penalty p, where n_{ij} is the number of lightpaths that are making use of physical link (ij) up to the moment.

For instance, let us assume that the logical and physical layers are those sketched on the upper and lower parts of Figure 4.6, respectively.

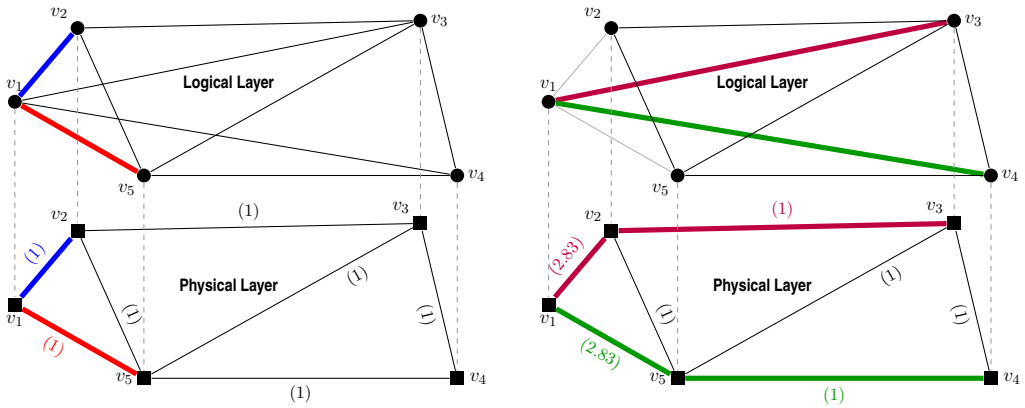


Figure 4.6: Balanced routing applied to logical links $(1,2),(1,5)$ and $(1,3),(1,4)$

The logical links to implement are: $(12),(15),(13),(14),(23),(35)$ and so on. For example purposes this is also the order in which they are drawn. The left half of Figure 4.6 shows the implementation of logical links (12) and (15) over the physical layer (respectively bold blue and red lines on both layers).

At this point and in order to find paths for (13) and (14) we need to update the pseudo-distances and restart the window. If $p = 1.5$ and since $n_{12} = n_{15} = 1$,

then $\bar{l}_{12} = \bar{l}_{15} = 2^{1.5} \approx 2.83$ during the next window. The next two logical links are then routed using these updated values. Their lightpaths are represented with purple and green lines on the right half of Figure 4.6.

The link (23) is the following in the list and it can be routed into two hops within this control window (cyan line in Figure 4.7). A window restart is necessary to route the lightpath of (35), as it can be seen on the right half of Figure 4.7.

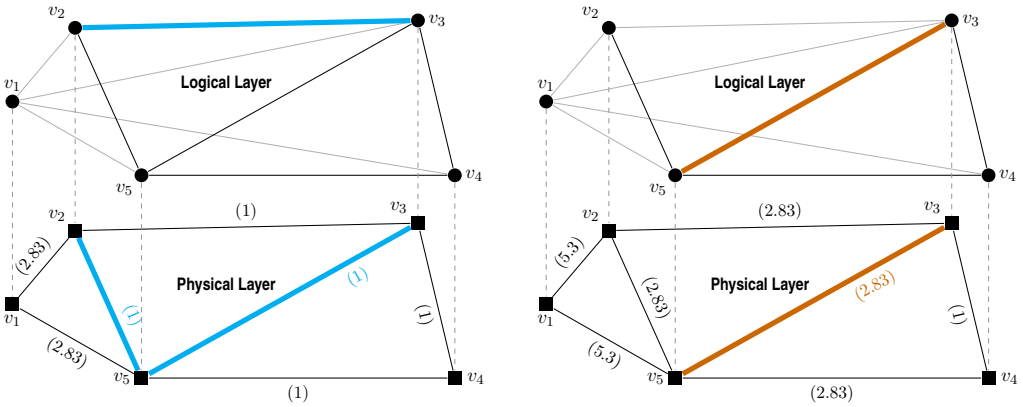


Figure 4.7: Lightpaths for logical links (23) and (35)

This construction is greedy in many senses. As long as it can be held, each lightpath is set-up without repeating links in the minimal number of hops. A global minimization, which coordinates hops and intersections, is not attempted because of its complexity. The core engine for this heuristic is a pretty clear variant of CSPF, which is in turn a very well known greedy polynomial algorithm.

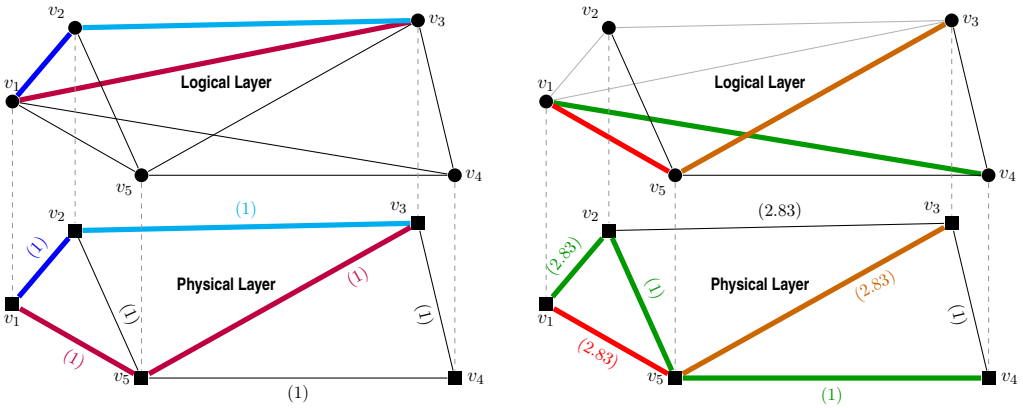


Figure 4.8: Lightpaths when links drawn are: (23), (12), (13), (15), (14) and (35)

Besides, the construction of lightpaths may be highly randomized. The mere order in which nodes and logical paths are drawn shapes its physical implementation. Figure 4.8 shows how an alternate order for selecting paths impacts on lightpaths. Higher values of p increase diversity but also affect stability of the result.

It is worth mentioning that lightpaths set-up during early iterations of this construction usually have paths of better quality. As a rule of thumb: the earlier the settling of a lightpath, the higher its influence over paths of remaining ones. Aware of this characteristic, Algorithm 7 promotes the earlier selection of shorter paths. Premature setting for intrinsically long paths affects negatively the cost of shorter paths, which are usually higher in number. Nevertheless, this promotion is based on a randomized selection, from which results an important diversity, necessary for this metaheuristic to work properly.

The result of the *randomized feasible solution phase* should be a candidate feasible configuration for the route of each lightpath over the physical network; but we did not make use yet of the traffic and capacity information, so we cannot be sure this is so.

4.2.2 Determining whether a solution is feasible

The next issue is determining whether the configuration found previously is feasible or not. The answer to this question is far from being easy, since this sub-problem is NP-Complete and it is in fact the core of our proof of Theorem 10. So we do not intend to generate an exact answer for this question. Instead we are using a heuristic, and accepting thereby some lack of precision.

Our heuristic is inspired in a very well known heuristic for NPP: *choosing up sides for a ball game*, which is in turn a cherished custom of childhood all over the world. The mechanism is more or less the following: "two chief bullies of the neighborhood would appoint themselves captains of the opposing teams, and then they would take turns picking other players. On each round, a captain would choose the most capable player from the pool of remaining candidates, until everyone present had been assigned to one side or the other. The aim of this ritual was to produce two evenly matched teams". [Hayes 2002] analyzes in detail how this idea can extend to solve NPP problem and under what premisses this heuristic works properly. For construction purposes, the underlying idea is taking earlier care of how bigger numbers are arranged, whereas smaller ones are finally used for fine-tuning purposes.

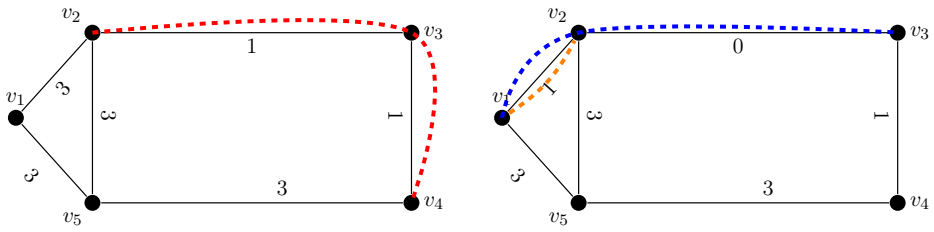


Figure 4.9: Paths for the tunnels (24), (12) and (13) over a Logical Layer.

The heuristic in our case is the following: *demands are taken in decreasing order of volume (d_{pq}) and the tunnel associated with each one is constructed over the logical layer using the minimal number of hops, while at every step only are considered links*

with enough free capacity to allocate the new demand. The strategy is inspired into the following fact: *the lower the number of hops, the lower utilization of global capacity resources upon the network*. This idea was previously used during the proof of Lemma 7.

For instance, Figure 4.9 shows an example logical topology where all links' capacities are 3. Let demands be: $d_{24} = 2$, $d_{12} = 1$, $d_{13} = 1$ and $d_{23} = 1$. The paths followed by each tunnel are sketched in Figure 4.9 and Figure 4.10 using curves of different colors: *red* for (24), *orange* for (12), *blue* for (13) and *cyan* for (23). The remaining capacity in every link after routing each tunnel -two tunnels on the right of Figure 4.9- is also represented.

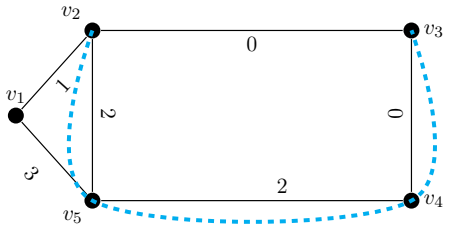


Figure 4.10: Path for the tunnel (23) over a Logical Layer.

This flavor of CSPF algorithm is very straightforward and inherits Dijkstra's algorithm performance. However an efficient implementation is quite complicated because of the following fact: *to be sure a solution is feasible this algorithm must be repeated for every failure scenario*. Furthermore, the function *isFeasible* is used in both: *construction* and *local search* phases. It is actually used several times within the same iteration in the *local search* (see Figure 4.5). In order to improve overall efficiency: paths cache, optimized data structures and several others low-level programming techniques are used. Once the function *isFeasible* has checked out a candidate solution, we are sure this configuration is valid. Conversely, the opposite doesn't hold and this heuristic exposes to *false negatives*[5].

The heuristic looks pretty similar to what capacity constrained-based routing can achieve within a pure IP/MPLS control plane (see Figure 2.25). The first noticeable difference comes from the order in which paths are established, which is seldom controlled onto a distributed operating environment. From the practical point-of-view this can be easily solved setting strictly the *primary paths* for LSPs and allowing an automatic secondary one based on traffic engineering.

However, we should introduce the concept of *bumping* to fully reproduce the constructions found with this heuristic onto an operative network. In the context of traffic engineering of paths, *bumping* means that a path is reconfigured to release capacity in order for other to be accommodated, i.e., not because of a proper issue.

[5]Otherwise it would constitute a polynomial complexity algorithm for finding solutions for an NP-Hard problem.

For example, let us consider a network instance whose coarse structure as it viewed from a distance is represented in Figure 4.11. There is a set of demands: $d_{p_1q_1} = 1$, $d_{p_2q_2} = 2$, $d_{p_3q_3} = 2$ and $d_{p_4q_4} = 3$, to fulfill between *subnet1* and *subnet2*. Nominal paths assignment for demands results into: $d_{p_1q_1}$ and $d_{p_2q_2}$ deployed over **(a)**, $d_{p_3q_3}$ deployed over **(b)** and $d_{p_4q_4}$ over **(c)**. Under this configuration a failure on link **(c)** in a standard traffic-engineered control plane of a network, would leave tunnel for $d_{p_4q_4}$ out of service, because IGPs do not reroute paths that aren't directly affected by a topology change.

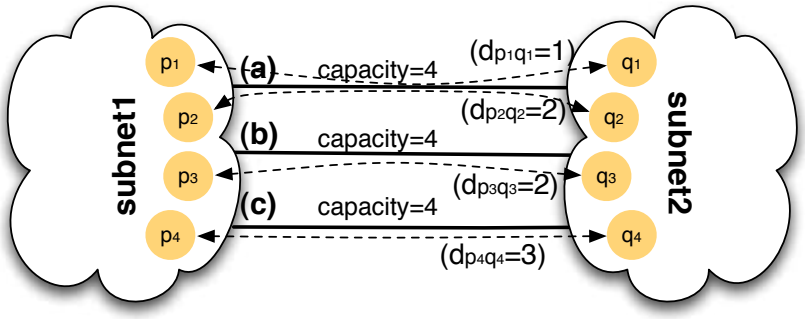


Figure 4.11: How bumping can improve routing quality

However the surviving network actually counts with enough capacity to accommodate demands, which could fit in if they were arranged so: $d_{p_1q_1}$ and $d_{p_4q_4}$ deployed over **(a)**, while $d_{p_2q_2}$ and $d_{p_3q_3}$ are over **(b)**. In an attempt to prevent this, the heuristic used to implement the function *isFeasible*, tries to rearrange the affected tunnels in first place, and just when this fails, starts over again from scratch.

As a trade-off, it happens that sometimes, solutions validated as positive are not technologically implementable by state of the art IGPs. This last fact is proper of the model rather than the algorithm. The choice of *bumping* presumes a centralized NMS interacting with control plane of nodes, which is usually resisted by maintenance staffs. In spite of the previous cautions, on our test instances (see Chapter 5), either bumping is seldom required or it can be reduced by adding some logical links, so we are confident that the heuristic is then representative of what practical IP/MPLS networks can do. Actually, this heuristic integrates the added value of being quasi-replicable over standard protocols.

4.2.3 Local search

Up to this point we have a feasible configuration for every lightpath, but we are still using all of the initial logical links and the input network is very likely to be over-engineered. Moreover, in the *construction phase* we attempted to distribute the routes of the lightpaths uniformly over the physical layer, forcing the usage of links multiple times when logical density is higher than physical. It is likely then that too many logical links fail simultaneously because of some single physical link

failure. Thus, it is very likely that many of these *redundant links* may be disposed of, if they are not really adding useful spare capacity.

It is worth mentioning that from this point on and until the next iteration of GRASP, the costs of lightpaths are revealed because we have their lengths -from the configuration of their routes- and there is only one possible capacity.

Through the *local search phase* we intend to remove the most expensive and unnecessary logical links off the current configuration. The process is the following: logical links are taken in decreasing order of cost of their lightpaths, each one is removed and the feasibility of the solution is checked out again. If the solution remains feasible the logical link under analysis is permanently removed, otherwise it is reinserted and the sequence follows to the remaining logical links. This subroutine is very straightforward and its scheme is highlighted as *local search* in Figure 4.5.

Once this processes is finished a minimal/lower-cost solution is achieved. Clearly, our local search approach is inspired in the *first-improvement* strategy rather than on the *best-improvement* (see Section 2.1.3.2). Strictly speaking it is possible to find a better quality solution by widely exploring the neighborhood, but the computational cost of *isFeasible* turns prohibitive doing so.

4.2.4 Stability issues

The algorithm specified in Algorithm 7 works satisfactorily when logical layer is not overcrowded of links respect to the physical one. According to the characteristics of each instance, some logical links should be included into the input data-set while others are forbidden. In order to count with the greatest possible search-space, it is tempting to include all non-forbidden logical links into the input data-set. Unfortunately, stability of the *Randomized Feasible Solution* phase, degrades when too many logical links are present.

Besides mandatory logical links, the remaining ones are included to satisfy the constraints coming from Lemma 5, Lemma 6 and Lemma 7. For this purpose, the *faces* of the plain representation of the physical layer are analyzed one by one. For determining how many logical links should be included within each face, the following rules are used:

i) Preserve traffic demands between nodes of the same face.

ii) Distribute uniformly the traffic terminating outside of the current face among nodes contained along its border, i.e., nodes splitting the current face from adjacent ones.

iii) Set up logical links according to Lemma 6 and Lemma 7, but over-dimensioning capacity by a rate, usually around 50%.

For instance, the left half of Figure 4.12 presents an hypothetical case where thin-black lines represent physical links and thick-grey ones correspond to logical links between nodes of the same face.

Let us consider the face determined by $\{v_1, v_2, v_3, v_4, v_5\}$. If $d_{17} = 3$ then this demand is temporarily omitted and the remaining demands are updated so: $d'_{12} = d_{12} + 1$, $d'_{13} = d_{13} + 1$ and $d'_{14} = d_{14} + 1$. All internal links of the face delimited by v_1, v_2, v_3, v_4 and v_5 are included because the number of nodes upon it is odd, and after computing new demands -step ii) of the transformation-, some of them are close -above 60%- to $b_{\bar{B}}/2$ (see Lemma 7). A similar situation arises from $\{v_4, v_3, v_7, v_{10},$- $v_{12}, v_{13}\}$, although in this case some demand is near $b_{\bar{B}}/3$. The diagonals were omitted to follow the construction guidelines of Lemma 7, because this face has an even number of nodes.

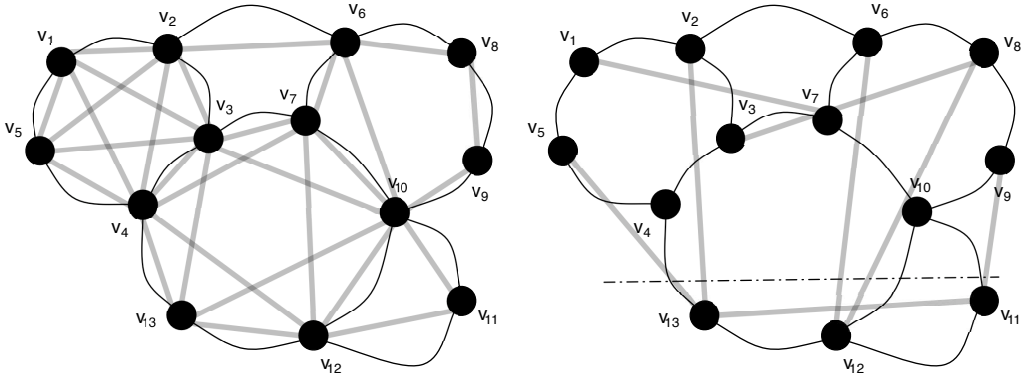


Figure 4.12: Logical links internal to faces and between them.

In the case of faces delimited by v_3, v_2, v_6, v_7 and v_{10}, v_{11}, v_{12}, the demands were below $4b_{\bar{B}}/n^2$ and $4b_{\bar{B}}/(n^2 - 1)$ respectively, corresponding to limits established in Lemma 6 for the even and odd rings; thus the cycle was used as the reference topology. The same happens for $\{v_7, v_6, v_8, v_9, v_{10}\}$ but the gap in this case was narrow. That issue, together with the fact that the number of neighbor faces for this case is above the average, led us to add an extra link (v_6, v_{10}) to facilitate detouring of traffic across this face in failure scenarios.

Besides logical links between nodes of the same face, links between faces were added to increase the diversity and robustness of the solutions. In order to do that, the bond condition (3.9) is applied to several selected bonds, like for instance those marked upon the right of Figure 4.12. In this case links are added until the gap between terms of (3.9) applied to the bond represented with a semi-dashed line, surpasses 50% of the necessary condition. The process is repeated for several bonds. It is worth pointing out that this potential capacity to move traffic amid faces, is complementary to that previously reserved when intra-faces links were defined, because nodes placed along the borders of faces can also relay such traffic.

Complementarily, we remark the importance of having certain level of consistency between lengths of physical hops. The heuristic *Randomized Feasible Solution* assigns paths using a dynamical metric which is based in the usage of links and the number of hops. Higher stability and better performance are achieved when

deviation among physical length of hops is minimized. They can easily be balanced by adding intermediate nodes on abnormally large physical links.

Although possible, we haven't automatized the previous process. Doing it manually is relatively easy. Moreover, this had the added-value of allowing integrating intuitive decisions coming from network designers, taking advantage of their experience. This work was always aimed as a tool to assist decisions, not as a substitute of designers. All adjustments over the input data-set described during this section are manually realized during first stages of each application case. Once tuned, this data-set is used to run several scenarios without further concern.

4.2.5 Boosting performance

Within this section we analyze the overall complexity of the GRASP implementation previously described. Besides, some improvements either implemented or planned are analyzed.

Theorem 11. *The GRASP implementation presented upon this section, has polynomial complexity.*

Proof. This proof consists in surveying the complexity of each component of Figure 4.5 separately. Here, we are only establishing the intrinsic polynomial complexity of this algorithm, not the best possible bound/worst-case-performance. All the computations realized during the *initialization phase* are of polynomial complexity, being the shortest physical path and distance between nodes the most expensive in terms of machine cycles. The only subroutine which eventually could jeopardize this complexity (i.e. finding structural physical bonds), is planned as *future work*, and it is a promising performance booster rather than a threat, as we shall see later in this section. Additionally, this phase is only run once.

Following the idea introduced in Figure 3.4, Section 3.1.3, we assume that each instance is represented by three graphs: $PL = (V, P)$, $LL = (V, L)$ and $DL = (V, D)$, which correspond to: *physical*, *logical* and *demand* layers respectively. Physical and demand graphs are weighted. Given any physical link $(ij) \in P$, it is known its length $l_{ij} > 0$. Complementarily, given any demand link $(rs) \in D$, the associated demand $d_{rs} > 0$ is also known. For purposes of this algorithm, there is only one capacity to dimension logical links. Besides, during this proof we don't assume the usage of optimized data structures, other than that explicitly mentioned (i.e. Fibonacci heap). As we shall see, the performance is determined by the structure of these graphs.

The *overlay routing* subroutine (see Algorithm 7), randomly takes logical links and finds the *minimum physical pseudo-distance* path for each one of them. The best case is that where the *control window* is not restarted, which matches Dijkstra's algorithm's complexity[6], that is, $O(|P| + |V|log(|V|))$. However, this requires

[6]We use the implementation based on a *min-priority queue* as a baseline in this proof, implemented using a *Fibonacci heap*, and due to Fredman and Tarjan on 1984.

$|P| \geq |L|$, which is unusual on real-world application cases. Whenever $|P| < |L|$ some logical node has a higher degree than its physical homologous, thus a window restart becomes unavoidable. A worst-case would take a separate window (a different Dijkstra's instance) for each logical link, and then complexity would rise to $O(|L|(|P| + |V|log(|V|)))$.

A subroutine widely used is *isFeasible*. As we have seen in Section 4.2.2, determining whether a solution is feasible or not for a lightpaths mapping, is in itself a difficult task. Besides, it must be repeated as many times as the number potential physical failures ($|P|$). Given a fault scenario $(ij) \in P$, the heuristic takes demands in decreasing volume, and runs a CSPF over the surviving logical layer, whose only constraint is the logical capacity. If demands are narrow when compared with logical capacities (i.e. $\sum_{(rs)\in D} d_{rs} \leq b_{\bar{B}}$), constraints never apply, all of the logical links are available for routing tunnels and the proof reduces again to Dijkstra's complexity. Under these premisses the complexity to check on all scenarios would be $O(|P|(|L| + |V|log(|V|)))$. Conversely, when each single demand d_{rs} saturates the logical links it uses (i.e. $d_{rs} \geq b_{\bar{B}}/2$), the usable logical topology changes after routing each tunnel, and the order increases to at most $O(|P||D|(|L|+|V|log(|V|)))$.

The *isFeasible* function is used once during the *randomized feasible solution phase*, and $|L|$ times during the *local search phase*. Hence, the computational complexity of each iteration of this GRASP should be placed somewhere between: $O(|P||L|(|L| + nlog(n)))$ and $O(|P||L||D|(|L| + nlog(n)))$, being $n = |V|$. All limits are of polynomial complexity in n since multigraphs are forbidden, and therefore the number of elements in P, L and D is bounded to at most $n(n-1)/2$. □

The previous result deserves further analysis before entering into performance improvements. At first glance the complexity might look close to $O(n^8)$, when the structure of all layers is close to the $\mathcal{K}^n$. Although polynomial, this complexity is quite bad in terms of computational efficiency. Luckily, this bound is exaggerated. We already know that when physical and logical topologies match, the optimal logical mapping copies the underlaying topology (see proof of Theorem 6). It is worth pointing out that under these circumstances, Algorithm 7 always finds the optimal configuration in time of order $O(n^2)$ (it turns deterministic), which is an extra value of the heuristic. Complementarily, on actual physical networks most nodes are of degree 2, and the construction of the input data-set described in Section 4.2.4, not only contributes to stability but also to keep the gap between complexities of logical and physical layers between limited boundaries. Hence, the expected complexity of Algorithm 7 for real-world instances should be between $O(nlog(n))$ and $O(n^2log(n))$.

The portion of the algorithm that could compromise the performance the most, is the *local search phase*. The worst case considered within the proof is unrealistic, because when there is demand between each pair of nodes, and each one of them saturates links (i.e. $d_{rs} \geq b_{\bar{B}}/2$), the instance is unfeasible no matter what are the physical and logical topologies[7]. As we will see in Chapter 5, by adding virtual

[7]Even when $LL = PL = \mathcal{K}^n$, the logical links of any bond would be saturated, and there would

nodes we may keep demands values in a proper ratio with links' capacities. Besides, the matrices of demands are sparse, with just a fistful of hub nodes, rather than dense and balanced. So for practical purposes, and for each failure scenario, the performance of the CSPF is similar to that of a regular Dijkstra's algorithm.

Moreover, important portions within physical networks of our instances are basically 2-connected, so many parts of these networks are sequences of nodes of degree 2 (see for instance Figure 4.14), that is: H-paths (Theorem 3). Logical links affected by physical link failures along each H-path, are similar if not identical. Since actual physical networks have many equivalent fault scenarios, we decided to include *routes caches* to save CPU cycles during paths computation. In a sense, trying to rearrange only affected logical paths after a failure (see Section 4.2.2), reuses most of the existing routes. However, when this is not sufficient, remembering some already used paths helps to increase the performance. The effect of these caches in the overall performance is impressive. Figure 4.13 shows with red dots, experimental times (expressed in msecs) taken by our *isFeasible* heuristic to determine if logical network constructions are feasible.

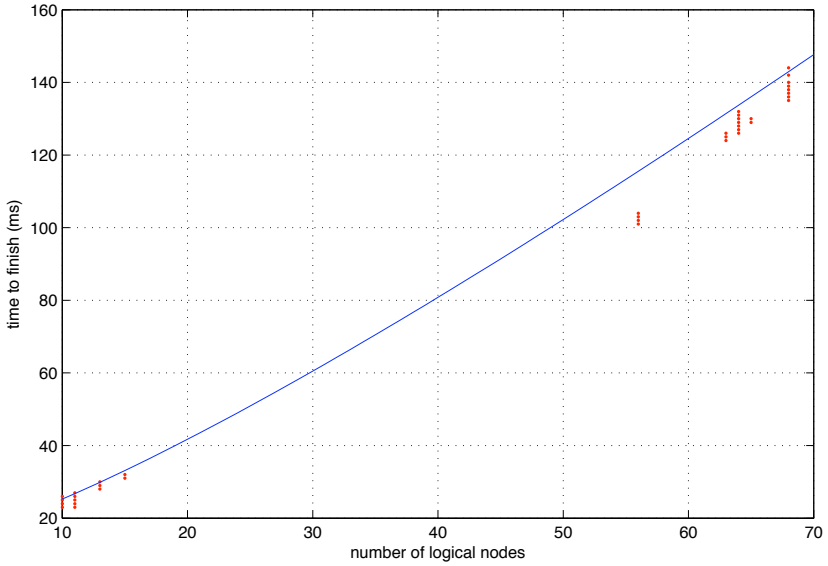


Figure 4.13: Time expended by *isFeasible* heuristic as a function of $|V|$

The number of logical nodes ranges from 10 to 68, but unfortunately in our instances there is a gap in the middle. The number of physical links spanned by these instances ranges from 121 to 190, but its effect into the result is minimal, which is consistent with described H-paths structure. The blue curve corresponds to a minimum squares regression of the function $a+b\cdot|V|log(|V|)$, adjusted to serve as an upper limit. Only instances classified as *feasible* by the heuristic were considered, which positions them as the hardest to compute, as we shall see immediately. From these experimental results, we conclude that the order of this heuristic when applied to real-world instances is much closer to $n^2log(n)$ than to the worst case

be no room to fulfill demands after a physical failure.

considered into the theorem's proof. Computations were performed in Intel Core 2 Duo processors at 2.4 GHz, with 2 GB of RAM memory (a MacBook laptop). The scope of the cache used was limited to each single call of the *isFeasible* function. It is very likely that a cache whose scope spans every iteration within the *local search phase*, can go even further to improve performance of the metaheuristic.

Using *routes caches* and other computational tricks helps indeed to boost performance, but there are further enhancements rather sustained on theoretical results. One we analyze here is based on Lemma 5. The *initialization* phase performs computations whose results are invariant among iterations. We already mentioned the shortest physical path and distance between each pair of nodes. Finding a base for *structural bones* could be another important task performed during this phase. As we have seen, the subroutine most intensive in terms of computations is the *feasibility check* (Section 4.2.2), especially when is applied into the *local search* phase. The cornerstone of this performance boosting relies on the following fact: *discarding solutions is much easier than validate them.* This is because to validate a candidate solution, a different configuration of tunnels' paths must be found to route traffic in every failure scenario, whereas for discarding it, is only necessary to find one scenario where capacity or connectivity is no sufficient. According in our experience, whenever solutions are discarded this takes place just after a few iterations.

The application of Lemma 5 into this matter is very direct. Given a candidate mapping for lightpaths (i.e. immediately after exiting the *overlay routing* module, Figure 4.5), *multilayer bonds* are taken in sequence and for each one of them, a subset of logical links traversing it, is randomly chosen to minimally satisfy (3.9), which is almost free in computations when the sequence has only a few elements. If this construction is not feasible, other logical links are reinserted until regain feasibility. The efficiency of the local search phase is increased afterwards. Of course, this only makes sense if the time expended on refining the construction is lower than that to redeem onto the following stage. Let us recall that: "bonds of a planar graph are those whose edges form a cycle into the dual graph" (Lemma 2 and Figure 2.4). Since programming a cycle generator is simple, programming a bonds generator is also simple for a planar physical layer. Unfortunately, finding all cycles in a graph, even for a planar and cubic graph, is NP-Hard. This follows from [Garey 1979] where it was proved that the decision problem: "to find out if a planar, cubic and 3-connex graph is Hamiltonian", is an NP-Hard problem.

Luckily in some of our instances the degree of most nodes in the physical network was only two, so we decided to simplify the structure before computing the dual graph. The result is still NP-Hard but since it is reduced to a fistful of nodes, the complexity turns manageable. Figure 4.14 shows how this idea is applied to a portion of the physical network (the left part of the image), leading to a much simpler problem. The structural physical network counts 8 nodes, whereas the original one has 53. The reductions of size on our real instances are around this ratio too.

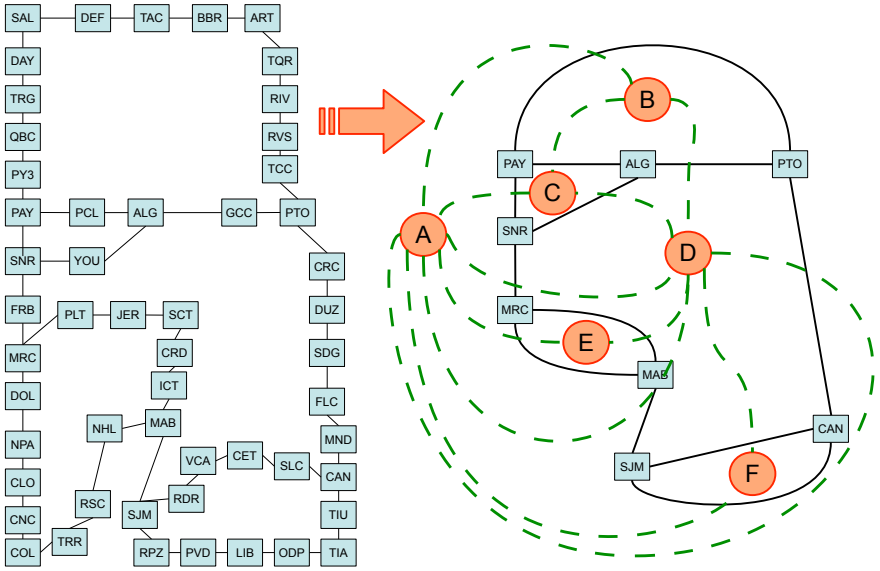


Figure 4.14: Structural bones defined by dual cycles.

The second half of this picture also represents the dual graph, with one node per face: A, B, C, D, E, F, and one link between every pair of nodes whose associated faces are adjacent. Through this example is easy to show how bonds are determined from cycles. For instance, there are three cycles that span nodes A, C, D. They are determined by the unique links between A and C and between C and D, combined with the three different links that connect A with D. Each one of these cycles defines a structural bond; following the example and for the commented cycles, they would correspond to: {PAY-SNR, ALG-YOU, PTO-CRC}, {PAY-SNR, ALG-YOU, FRB-MRC} and {PAY-SNR, ALG-YOU, MAB-SJM}.

Although experiments with this idea issued promising results, this extra step into the *construction phase* is not integrated to the algorithm yet, but figured as a line of future work. The reason is the following: besides of actual/real nodes, some instances must add *virtual nodes* to integrate characteristics proper of the scenario to represent (see Chapter 5), and after adding them, planarity is often lost. However, this idea was successfully used for the VNS algorithm ([Córez 2010]), because instances tackled down with it, used a reduced physical network that did not suffer from this issue.

The following is a much more standard improvement. After $MaxIter$ iterations the best solution found is chosen to be the output of the algorithm. Since the construction procedure privileges the nodes drawn earlier to shape the paths of the lightpaths, we presume that by adding path-relinking to the mapping of lightpaths we could improve the quality of the result, if the initial lightpaths routes of the elite solutions are prioritized to explore new solutions. We are also considering checking this presumption into future work. For further information in path-relinking enhancement to GRASP, please refer to: [Resende 2003] and [Glover 1996].

Chapter 5

Application Cases

Contents

This chapter covers in detail the two application cases over which algorithms detailed in Chapter 4 were applied. Both comprise respectively, the most important Academic (RAU) and Commercial (ANTEL) IP networks of Uruguay.

5.1 RAU

The existence of interconnected *academic networks* is an essential asset for the actual academic activity. Academic networks offer services and fulfill needs far beyond of what commercial Internet is intended to accomplish. In fact Internet was born as the infrastructure to interconnect academic networks, long before anyone could think about its massive applications to business and commerce (see Section 1.1.2).

Academic proceeding is essentially collaborative and modern communications sustain activities such as: education, research and development. The RAU (Red Académica Universitaria/Universitary Academic Network) inaugurated Internet in Uruguay by 1988 and was institutionally incorporated by 1990 under the operation of SeCIU (Servicio Central de Informática Universitario), UdelaR (Universidad de la República). Subsequently, several other academic institutions joined to RAU: UCUDAL (Universidad Católica Dámaso Antonio Larrañaga), Universidad ORT, ANII (Agencia Nacional de Investigación e Innovación), INIA (Instituto Nacional de Investigación Agropecuaria), Institut Pasteur de Montevideo among others. Academic transversality potentiates capabilities beyond of what single areas can do as isolated units. Suffice to mention the reconstruction of a DNA sequence, physically run on

the Pasteur but processed onto the High Performance Cluster of FING (Facultad de Ingeniería, UdelaR). In addition to a communications medium for sharing information, academic networks are called to be the platform onto which new technologies are to be developed. Not only new telecommunications protocols or standards, but also collaborative applications such as: telemedicine, with critical requirements, some of which cannot be adequately supported upon commercial Internet.

The last important technological upgrade for RAU was Clara[1], by 2003, when several regional academic networks started to interconnect directly or with European counterparts[2]. This upgrade was conceived as an overlay over an ATM layer like that sketched in Figure 1.2. Currently, the RAU is being redesigned to take full advantage of the evolution of technologies. This new network, baptized as RAU2 is precisely the object of the application covered into this section.

5.1.1 Network structure

The RAU2 is intended to connect 108 points, scattered all over the country but with a notorious higher density on Montevideo. Figure 5.1 indicates with red dots the points of presence to connect. Many points are too close to others and appear as overlapped into the picture.

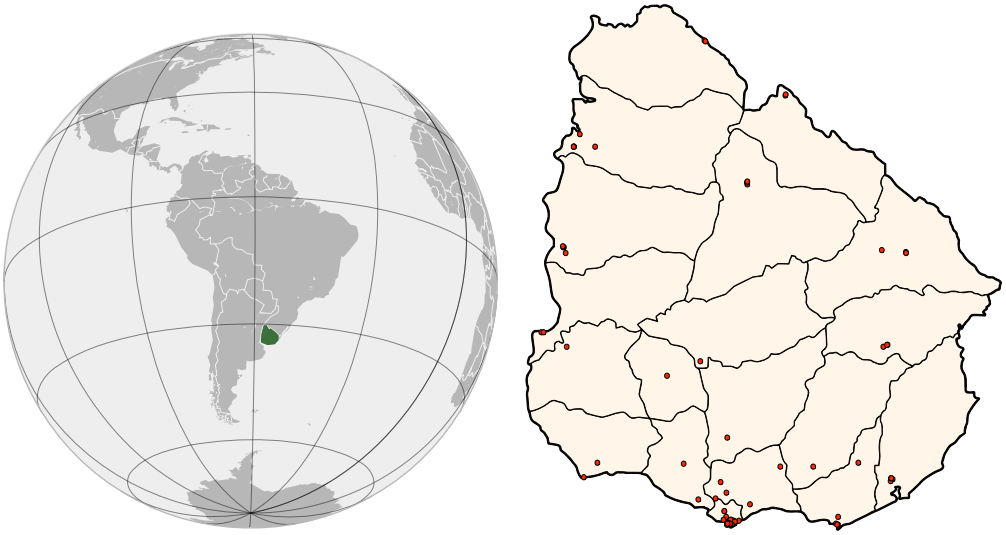


Figure 5.1: Map of Uruguay and points of RAU2

Ten of these points were selected as POPs for the backbone of the network. The choice was based on the importance of each center rather than on the proximity with other points. We identify these nodes by the following labels: *RIV* (Rivera), *SAL* (Salto), *TCC* (Tacuarembó), *PAY* (Paysandú), *ROC* (Rocha), *MLD* (Maldonado), *SeCIU*, *FQuim* (Chemical Faculty), *HoCli* (Hospital de Clínicas - University Hospital) and *FInge* (Engineering Faculty).

[1] CLARA: Cooperación Latino Americana de Redes Avanzadas.
[2] ALICE: América Latina Interconectada con Europa.

The POPs: RIV, SAL, TCC, PAY, ROC and MLD are located into interior departments of Uruguay. The remaining four (SeCIU, FQuim, HoCli and FInge) are into Montevideo (Capital City). Figure 5.2 roughly marks the location of each POP. The POP *SeCIU* is where most network services and connections are actually installed. The scheme of interconnections over which the current RAU is developed, is a typical *Hub and Spoke* architecture, i.e., remote points are connected with a common central node through point-to-point connections (ATM/Frame Relay PVCs or LAN2LAN Ethernet), resembling a star or a spoke wheel.

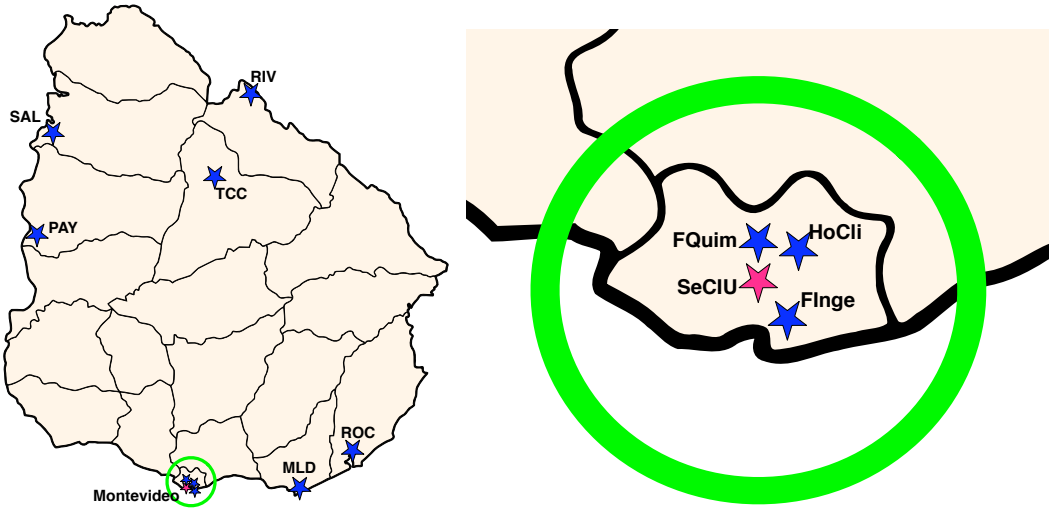


Figure 5.2: Points of presence of RAU2's backbone

Not coincidentally, the first drafts of RAU2 inherited this particular architecture. Figure 5.3 shows the logical topology chosen in the final draft. It can be appreciated how connections are essentially deployed around SeCIU.

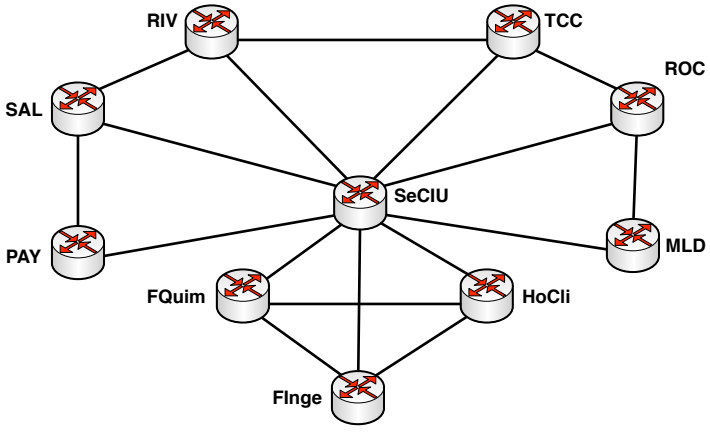


Figure 5.3: Draft for RAU2's logical topology

Recalling elements of analysis commented in Chapter 2, we notice some issues

on this topology. First of all, there is one node per POP. The node SeCIU is an articulation point in this graph, so a fault on it would disconnect the network into two parts. Since some connections are centralized into SeCIU (e.g. connections to Internet and CLARA), it is very likely that a fault onto SeCIU leaves most network services off-line. We haven't commented it yet, but we anticipate here that there are two outgoing physical connections from every node of Montevideo towards other physical nodes. On the upcoming Section 5.1.3 we show that under some demand scenarios and when applied from SeCIU to the remaining nodes of Montevideo, this configuration violates necessary conditions established by Lemma 3 and Lemma 5. In consequence a fault upon a single physical link in this design necessarily leaves some demands unfulfilled. We can certainly state thereby that this design is hazardous in terms of availability. As we shall see, it is also inefficient in terms of costs.

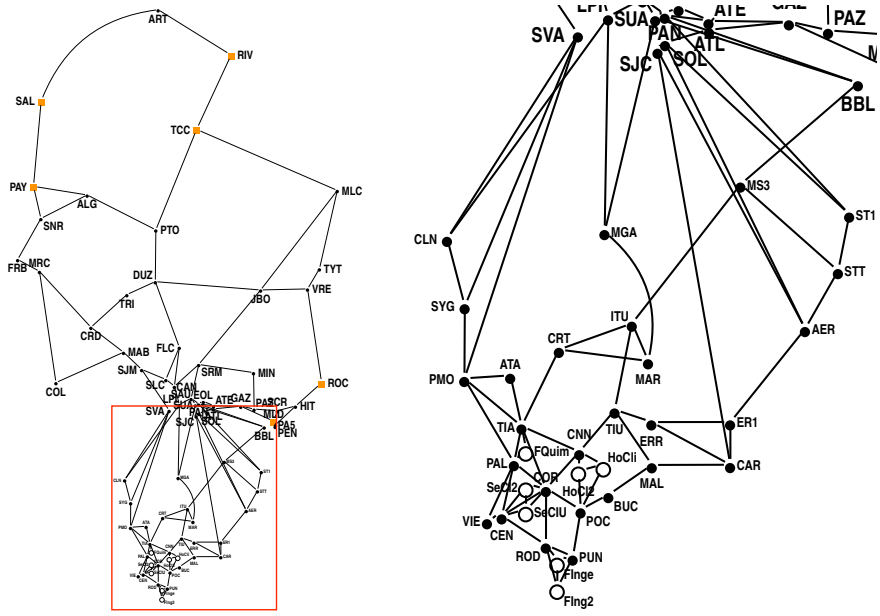


Figure 5.4: Physical network considered for RAU2's design

Regarding the physical structure, the left part of Figure 5.4 shows a representation of the entire physical network considered. This is the result of merging portions of the optical infrastructure of: ANTEL (Adminsitración Nacional de TELecomunicaciones), UTE (administración nacional de Usinas y Transmisiones Eléctricas) and other potential optical fiber providers. The length of each link roughly matches the geographical distance between points, except for those squared within red. Digging trenches, deploying conduits and fibers is much cheaper when is made on *rural areas* than on *metropolitan* ones. According on our information a factor of ten is a pretty good rule-of-the-thumb approximation. Thus, to keep consistency between lengths and costs, metropolitan lengths in Montevideo and its nearby areas are ten times larger than the geographical distances, and this portion looks as seen through a magnifying glass, which is represented on the right of the figure.

5.1.2 Demands to fulfill

Up to these days the RAU brings services mainly to professors, researchers and administrative staff of some faculties. Students have no access to RAU's services in general, and on many centers the academic staff has no access either. Expanding the universe of users of RAU is one of the premisses of this analysis. The next consists in increasing the effective bandwidth and the overall performance.

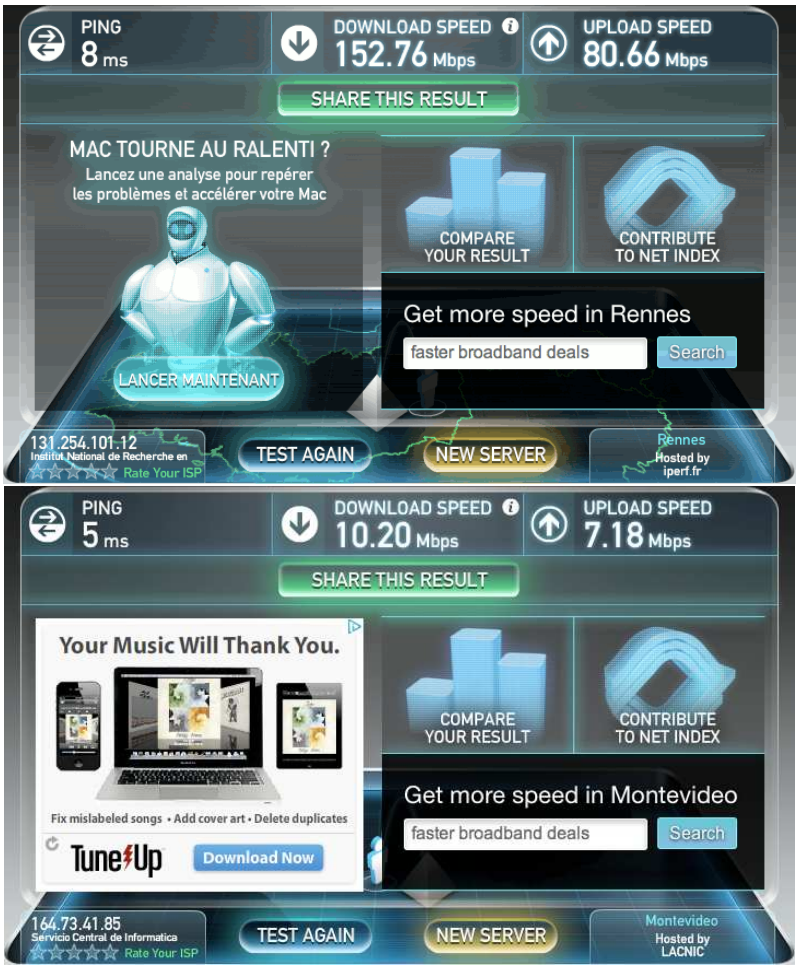


Figure 5.5: Overall performance for connecting to Internet from INRIA and FING

The tasks comprised into this thesis were carried out into two research centers: FING/UdelaR (Montevideo, Uruguay) and INRIA (Rennes, France). Figure 5.5 shows results for a well-known *Internet speed-test* run from inside of both academic networks. The spread between them is beyond the order of magnitude.

So our design aims on bringing world-class services to: 11.000 professors, 7.000 members of administrative staffs and 140.000 students on all academic centers. Above 91% of the students and above 94% of academic or administrative human resources are located at Montevideo, which is overwhelming. However, the traffic

terminating outside of the national boundaries (Internet, Clara, Alice) might be better routed across connections with nodes near to Argentina and Brazil, and this increases the relative importance of those nodes located at the interior departments.

Besides of increasing the number of users, we aim on raising the bandwidth they can make use of. In terms of targets, we established 10Mbps for students and 100Mbps for employees. In other words, students can get a performance -through WiFi connections installed in faculty buildings- in RAU2 similar to that professors already have in RAU; whereas members of either academic or administrative staffs -making use of UTP Ethernet connections on their desktops- can achieve a performance comparable to that of their European counterparts.

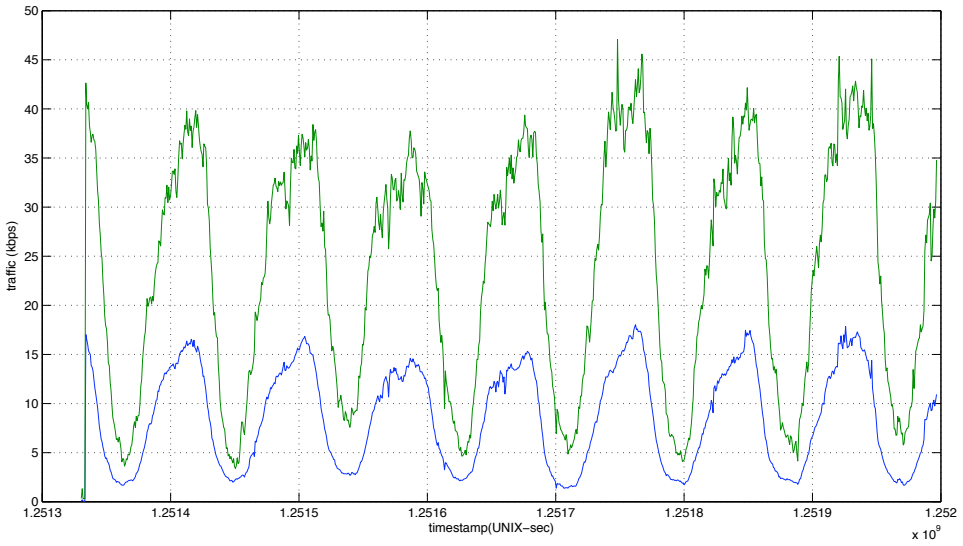


Figure 5.6: Expected traffic along a week for a regular Internet user

To estimate traffic we have based on experience and statistical information. Users were separated into two groups: *heavy* and *regular/light* users. Along a certain period some users download a volume of information from the network, which is proportional to the available bandwidth. The *heavy users* group comprises those users with such a profile of usage. They represent around 10% of the total.

The remaining 90% are *light users*; the volume of information they download from the network is regular, i.e., they do not intensify their habits of usage because of the extra performance of a better connection.

The Figure 5.6 shows a typical contour for the traffic of a *light user* along a period of one week. The green curve represents *downstream*[3] whereas the blue one corresponds to *upstream*. Since traffic in our models is symmetrical, at the rush hour and into the backbone we consider some 40kbps as the expected contribution of each *light user*, regardless of it is student or employee of a faculty.

Conversely, the traffic of a *heavy user* is tightly bonded to bandwidth, so on

[3]Traffic coming from the network towards the user.

a per-person base, it is teen times higher for employees than for students, and on either case is much bigger than that of a single *light user*. Fortunately, it can be controlled limiting access to certain kinds of applications or sites.

We defined two bandwidth scenarios according on the existence -or not- of such kind of *content filter*. The summarized information of Internet traffic expected under these premisses on both scenarios is the following:

traffic scenario	*offered bandwidth*	*traffic from HU*	*traffic from LU*	*total traffic*	*total demand*	*per-user demand*	*overbooking factor*
Low	3,200,000	-	6,320	6,320	7,022	0.002	456
High	3,200,000	31,360	5,688	37,048	41,164	0.013	78

Table 5.1: Internet traffic and demands information, expressed in Mbps

The number of users totalizes 158.000 and the offered bandwidth is of 3200Gbps on both scenarios. Differences start up from this point on and they are summarized in Table 5.1.

On traffic scenario *low* all users are *light users* and then the expected traffic across the backbone at the rush hour is 6320Mbps. We are providing a gap of 10% between demand and expected traffic as a mean for offering a level of certainty, i.e., to reduce the probability for this stochastic process (the traffic process) to overpass demand limit. Thus, total demand to reserve rises to 7022Mbps, only 0.22% of the offered bandwidth. That is, the *overbooking* between offered and moveable traffic is of 456. This value is higher than normal and it would be very optimistic for a regular ISP. However, when we consider that a percentage of these students do not attend daily their courses on many centers, the value turns realistic for RAU2 application.

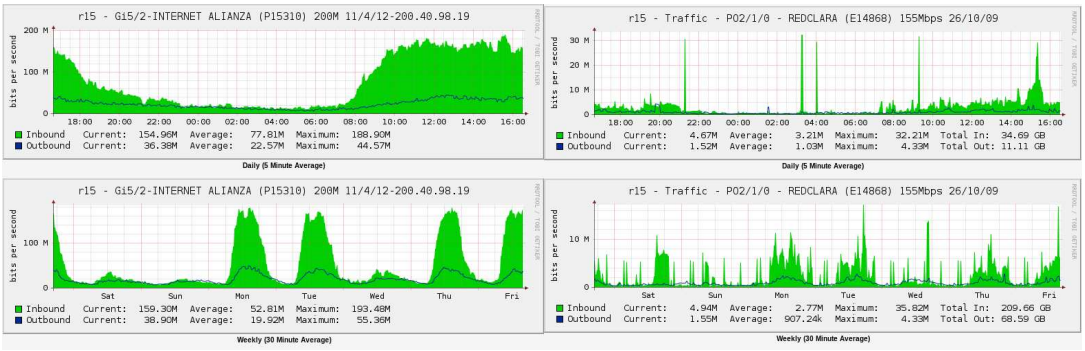


Figure 5.7: Traffic statistics for Internet and Clara connections

On traffic scenario *high* all kinds of traffic are allowed, *heavy users* can do as they wish and traffic volume increases significantly. It is worth pointing out that overbooking factor here (78) is more similar to industrial standards. We may think about this case as a *worst case* for traffic.

For completing the demand information, we should add to Internet traffic that

ending on Clara Network and the traffic internal to the network itself. We agreed with SeCIU counterparts to apply the same ratio of expected growing for Internet to others kinds of traffic. After monitoring traffic on appropriate interfaces we determined that the current peek of traffic with Internet is of 200Mbps (see Figure 5.7) and with Clara is of 30Mbps. Traffic between nodes of RAU required monitoring a much greater number of interfaces, which we are not showing here.

Thus the factors to apply for scenarios *low* and *high* are 35 and 206 respectively. Each one of the 98 points that are not backbone nodes is assigned to the closest node across the physical layer (Figure 5.4) and its traffic is integrated into it.

traffic scenario	*Internet demand*	*Clara demand*	*Internal demand*	*total demand*
Low	7,022	1,050	412	8,484
High	41,164	6,180	2,425	49,769

Table 5.2: Demands to Internet, Clara and Internal, expressed in Mbps

Table 5.2 summarizes *on-line* demands for all destinations on each traffic scenario. By *on-line* we mean that this traffic is generated by interactive applications, such as: web browsing, files downloading, e-mail, voice and video applications, etc. That is, the kind of applications that are currently making use of the RAU.

At the first stages of this analysis, we considered traffic coming from all applications as on-line traffic. However, this premiss was relaxed later because it would result into a too-expensive/out-of-the-budget network but also and fundamentally because it is not critical. Most big-data applications (e.g. DNA processing) are *batch/off-line* rather than interactive. On-line traffic presumes the presence of human users, and they are present at daytime leaving most bandwidth unused during nighttime (Figure 5.6). We rely on the existence of a Content Delivery Network (CDN) that moves during nighttime this huge volume of information among points.

Therefore, the only demands considered for dimensioning the network here, are those summarized in Table 5.2, which are on-line demands. As a second stage -i.e. after finding the optimal design for each instance- we check the effective capacity and the total movable volume between points, to assess the feasibility to attend off-line demands through a CDN over this construction.

5.1.3 Best solutions found

Demand is a key requirement to establish instances but it is not the only one. Before going into details about aspects considered for scenarios, we are briefing some economical characteristics of current RAU. Actually, total annual leasing cost for RAU connectivity is of USD 1,238,500.00 and it can be decomposed into three portions: access-and-bb (USD 573,500.00/year), Internet connectivity (USD 450,000.00/year) and Clara connectivity (USD 260,000.00/year).

It is difficult to differentiate backbone from access because most connections are point-to-point circuits from the served point towards SeCIU. All mentioned connections are supported by ANTEL. The new design for RAU2 uses cheaper technologies to connect sites and also rationalizes resources usage by putting additional backbone nodes closer to points to serve.

Since early drafts of the project for RAU2, ten POPs are defined for the backbone (see Figure 5.3). The total budget for this new access-plus-backbone project is of 1,006,500.00/year and it decomposes approximately into: 84% backbone and 16% access-transport. The access portion comprises dedicated optical fibers from access points to POP of ANTEL's optical transport network (DWDM). Once in an optical POP, connections are routed over point-to-point optical citcuits towards the closest node of RAU's backbone.

Placing additional backbone nodes by the middle of the country might help to decrease transport costs, but SeCIU refused this due to operational issues. We analyze the backbone taking its presence as fixed, and making focus on what was established by SeCIU as critical aspects to determine, that is:

- How many nodes should be installed into each POP?
- What is the loss of efficiency due to centralizing connections into SeCIU?
- Which are the optimal points for sharing traffic with other networks?
- What connections are necessary for doing all this resiliently and efficiently?

5.1.3.1 Coarse-grain scenarios

To answer these questions we agreed on setting instances for representative scenarios. Each configuration is analyzed on both extreme traffic cases (low and high). The coarse-tuning comprises four scenarios where only Internet traffic is considered. Two of them are essentially feasibility checks over drafts of SeCIU and ANTEL, whereas the following two are actual optimizations.

demand	*HoCli*	*FInge*	*FQuim*	*SAL*	*ROC*	*PAY*	*MLD*	*RIV*	*TCC*	**Total**
low	1,799	1,692	855	177	97	89	57	28	10	**4,804**
high	10,548	9,919	5,012	1,036	571	523	333	163	57	**28,162**

Table 5.3: Demands from backbone nodes to SeCIU, expressed in Mbps

Table 5.3 shows values for demand from all POPs of the backbone towards Internet, that is, towards node SeCIU.

It is worth pointing out that *total* values -highlighted with bold letters- do not totalize Internet traffic (as in Table 5.1); this is because the Internet traffic from the POP SeCIU (the most important in volume) does not need to traverse the backbone to get to the local border router.

The capacity chosen to dimension logical links on all scenarios is of 10Gbps. Hence and since the total traffic for the *low* demand case is of 4804Mbps (its double fits into one single link), suffices to construct a 2-edge-connected logical network where links are physically independent to get a feasible solution for both ASP-MORNDP and FRP-MORNDP. This is more or less what EA and GRASP found (see Figure 5.9). On the second case (high demand) even a more complex network is not always sufficient. The draft for the logical network has two clearly defined components articulated by SeCIU (Figure 5.3). It is pretty clear that if the logical design cannot handle demands for one of these subcomponents, then it cannot fulfill demands for the whole either. Let us forget for a while of those nodes located at interior departments.

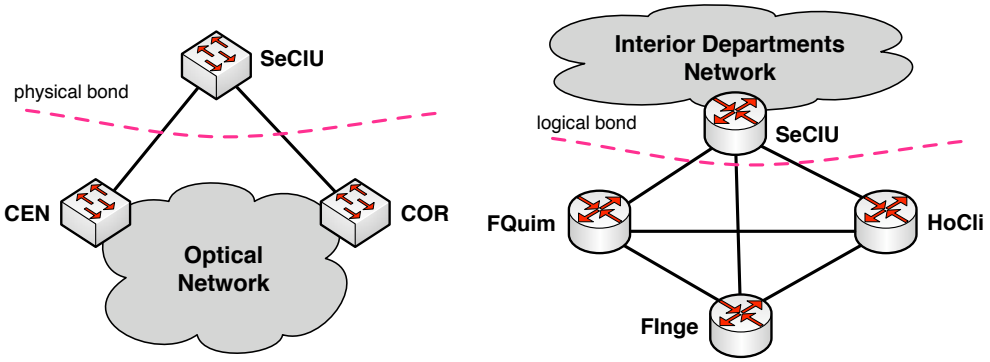


Figure 5.8: Critical multilayer bond in drafts of RAU

Each one of the metropolitan points (SeCIU, HoCli, FInge and FQuim) counts two independent physical connections with nodes of the optical network. Since the aggregated demand between SeCIU and the rest of metropolitan nodes raises to 25.479Gbps (green cells on Table 5.3), the bond of Figure 5.8, which detaches SeCIU from the rest of the optical network violates (3.9) - Lemma 5:

$$\sum_{p\in V',q\in V''} d_{pq} = 25.479\text{Gbps} \nleq b_{\bar{B}} \left\lfloor \frac{|bond_{\bar{L},P}|(|bond_P|-1)}{|bond_P|} \right\rfloor = 10\text{Gbps} \left\lfloor \frac{3\cdot(2-1)}{2} \right\rfloor$$

Thus, under some physical link failures the drafts of SeCIU and ANTEL are not capable of handling traffic without congesting links for FRP-MORNDP, neither for ASP-MORNDP -this is a relaxation of the previous-.

This cannot be fixed only by extending the quantity of logical connections. The mere demand between HoCli and SeCIU (10.548Gbps) solely cannot fit within a logical link of 10Gbps, so in order to find solutions, it is necessary to include at least an extra node into the POP HoCli. These nodes are called: HoCli and HoCl2.

The drafts of ANTEL and SeCIU were useful to tune other low-level parameters, such as: the number of intermediate hops to append into physical links (Section

4.2.4). In any case and due to the reduced number of nodes, modifications are minimal.

scenario index	*demand level*	*total demand*	*ASP-MORNDP #links & kms (EA)*		*FRP-MORNDP #links & kms (GRASP)*	
AL	low	7,022	–	–	17	6,133
AH	high	41,164	–	–	17	6,133
01	low	7,022	11	3,022	10	2,939
02	high	41,164	15	3,799	17	4,976

Table 5.4: Best solutions found with EA and GRASP for first scenarios

Table 5.4 shows the number of logical links and equivalent kilometers[4] for the scenarios 01 and 02. It also shows information for the draft network (AL and AH), although in this case links are fixed and they are just implemented using the minimum number of kilometers[5]. As we mentioned there is no feasible solution for AH and the row is highlighted with yellow. Figure 5.9 shows a representation of logical networks constructed by EA (ASP-MORNDP) and GRASP (FRP-MORNDP) for scenarios 01 and 02. None of them resembles the shape of Figure 5.3.

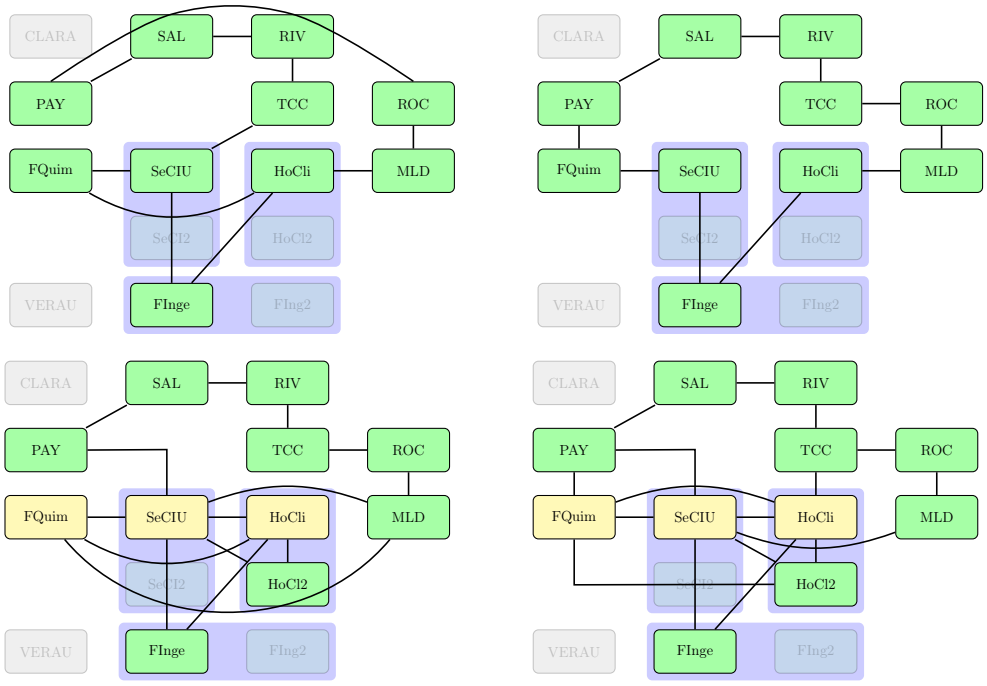


Figure 5.9: Logical networks constructed by EA (ASP-MORNDP, left side) and GRASP (FRP-MORNDP, right side) for scenarios 01 (top) and 02 (bottom)

The performance of GRASP under scenario 02 (bottom-right) was poorer than

[4]For rural areas this value matches the actual length, whereas for the metropolitan area the length is multiplied by 10 as in Figure 5.4.

[5]There is no optimization other than this.

EA (bottom-left). EA used fewer links (15 vs. 17), and above all used shorter links than GRASP. This is unexpected because FRP-MORNDP is a relaxation of ASP-MORNDP. The root cause lies on stability issues arising from the over-assignment of the only two physical links from SeCIU towards other nodes, which affects Algorithm 7 more seriously than Algorithm 6. Instead of filtering logical links following guidelines of Section 4.2.4 we increased the number of nodes on some POPs. This is necessary anyway (see Section 5.1.3.2), and after doing so, the GRASP algorithm behaves much better.

5.1.3.2 Fine-detail scenarios

Remaining scenarios are defined for the whole traffic (i.e. Internet et al) as it figures in Table 5.2. Instead of a matrix of demands, the Figure 5.10 represents the values of demands between POPs for *high demand* scenarios, using lines of different colors.

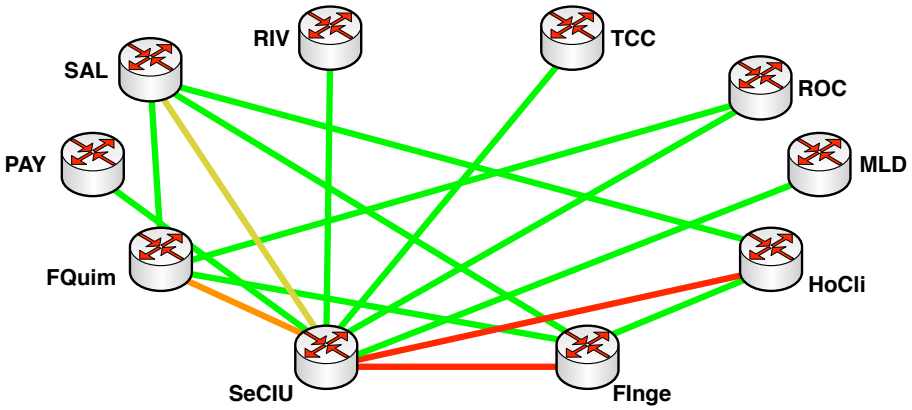


Figure 5.10: Demands between POPs of RAU2's backbone

Demands below 50Mbps are prorated amid the remaining demands. Green lines correspond to demands into the range [50 to 999]Mbps. The unique yellow and orange lines respectively indicate a demand within the range [1 to 5]Gbps and [5 to 10]Gbps. Red lines correspond to demands above 10Gbps, impossible to be fulfilled between single nodes with this capacity for links. We decided then to include extra nodes into the POPs: SeCIU, FInge and HoCli. These nodes are labeled as: SeCI2, FIng2 and HoCl2 (already commented on Section 5.1.3.1).

The ten scenarios analyzed for RAU's application are summarized in Table 5.5. We already have seen main characteristics for scenarios: AL, AH, 01 and 02. The first two only are included as baselines, i.e., as a reference of what experimented human beings would have determined manually.

Scenarios 01 and 02 were included to tune the input data-set. Their main characteristics were already analyzed. Scenarios 03 and 04 are equivalent to 01 and 02, expect for the following facts: all on-line traffic is considered for these scenarios (Internet, Clara and Internal), and some extra nodes are added to the logical layer. SeCIU remains as the only point of interchange (peering) with other networks.

scenario index	*demand type*	*classes of traffic*	*nodes considered*	*border nodes*	*additional comments*
AL	low	Internet	originals	SeCIU	baseline configuration
AH	high	Internet	originals	SeCIU	baseline configuration
01	low	Internet	originals	SeCIU	data-set tuning
02	high	Internet	double node at HoCli	SeCIU	data-set tuning
03	low	on-line	double nodes at: SeCIU, HoCli y FIng	SeCIU	extra nodes improve the quality of constructions
04	high	on-line	double nodes at: SeCIU, HoCli y FIng	SeCIU	extra nodes improve the quality of constructions
05	low	on-line	same double nodes plus VERAU and CLARA	Backbone	simulates a change of peering scheme
06	high	on-line	same double nodes plus VERAU and CLARA	Backbone	simulates a change of peering scheme
07	low	on-line	originals	Points	gets Internet out of the RAU
08	high	on-line	double node at HoCli	Points	gets Internet out of the RAU

Table 5.5: Summary of scenarios characteristics and comments

The results are really impressive. Just by adding some nodes the overall behavior of the GRASP implementation improved significantly. Table 5.6 shows the main numerical results to instances of both demands levels and for both models.

scenario index	*demand level*	*total demand*	*ASP-MORNDP #links & kms (EA)*		*FRP-MORNDP #links & kms (GRASP)*	
03	low	8,484	16	3,536	15	3,250
04	high	49,769	23	3,978	18	3,585

Table 5.6: Best solutions found with EA and GRASP for scenarios 03 and 04

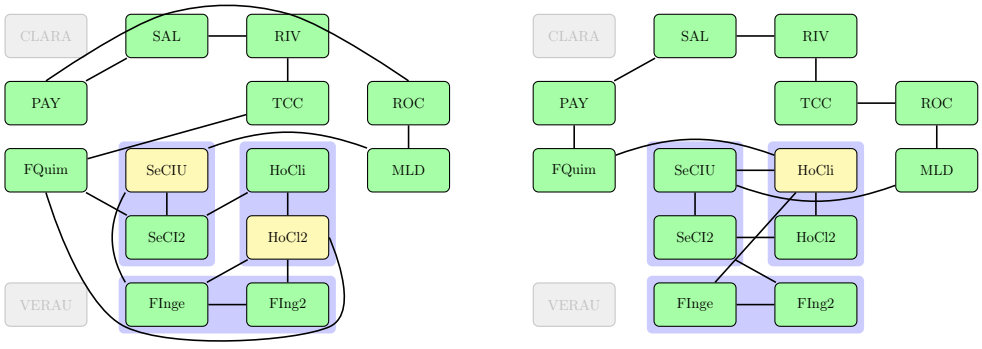


Figure 5.11: Logical networks constructed by EA (ASP-MORNDP, left side) and GRASP (FRP-MORNDP, right side) for scenario 03

Figure 5.11 and Figure 5.12 show solutions found for scenarios 03 and 04 re-

spectively. Solutions for ASP-MORNDP (EA) are always on the left side of the images. Despite some minor differences, they are pretty similar. After adding extra nodes the solutions found for FRP-MORNDP are of lower cost than those of ASP-MORNDP, which should be expected in general.

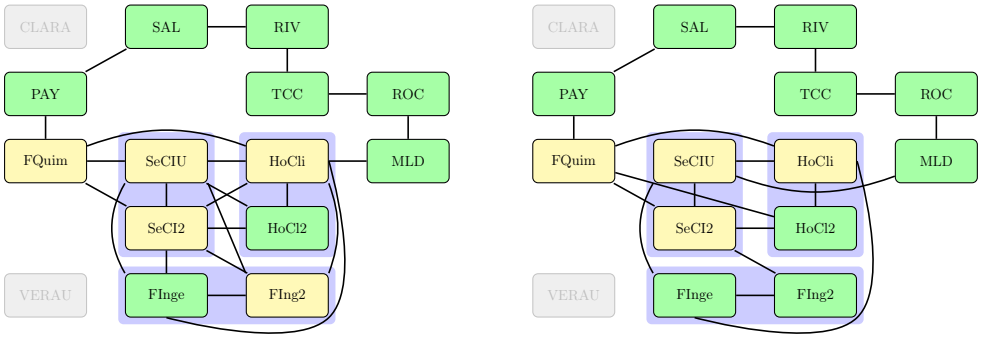


Figure 5.12: Logical networks constructed by EA (ASP-MORNDP, left side) and GRASP (FRP-MORNDP, right side) for scenario 04

As it is indicated in Table 5.5 scenarios 05 and 06 also duplicate nodes on SeCIU, HoCli and FInge. These scenarios are designed to relax the uniqueness of SeCIU as gateway for the external traffic. Instead, we assume the existence of two virtual nodes: CLARA and VERAU.

All *Internet traffic* formerly terminating into SeCIU is now redirected to node VERAU. *VERA Universidades* is the brand of the newer, cheaper and faster Internet access products, recently launched by ANTEL for academic institutions. By adding virtual links from all nodes of the Backbone to this new node, and marking it as the destination for Internet into the matrix of demands, we are indirectly letting the algorithm to find out the best way to get the traffic off the network. The resiliency, intrinsic component of ASP/FRP-MORNDP models is thereby inherited by the output design. The idea must be implemented with care to be sure we are not introducing spurious solutions with it. The list of precautions is the following:

Cost - In order for constructions to be cost-consistent, virtual distances should match real costs. VERAU services have a common list of prices along the entire country, whose entries are only determined by the leased bandwidth.

Hence, for a connection speed of 10Gbps the cost of every virtual physical link, which connects any node of the backbone with the node VERAU, matches the cost of an optical path of 847km.

Lightpaths - After adding virtual physical links to an instance, we are exposing constructions to make use of them for any purpose. Particularly, it turns possible that such virtual links might be used for a lightpath other than those ending upon VERAU node.

This is very unlikely because of the cost of these virtual links. Although the largest equivalent distance between physical points of RAU's optical network is of 1100km (ART-HoCli). Using VERAU as an intermediate point for a lightpath would add another 1694kms for going back and forth. Furthermore, actual constructions of algorithms rarely use lightpaths longer than 750km, minimizing then the risks of incurring in such a spurious construction.

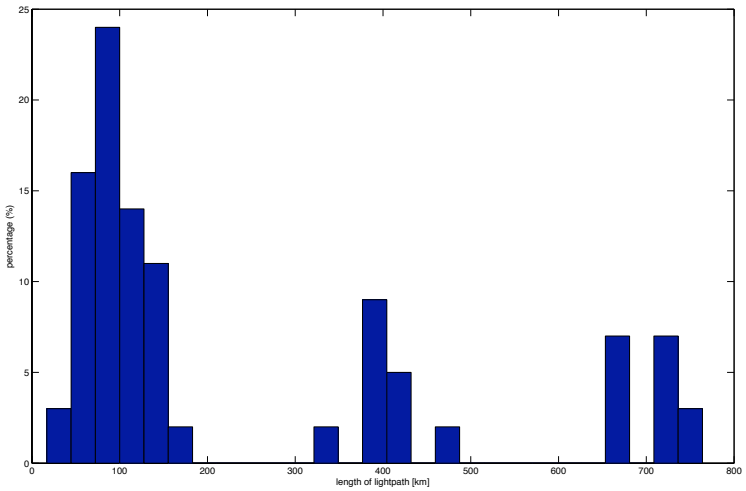


Figure 5.13: Relative weight for length of lightpaths (experimental)

Figure 5.13 shows a histogram representing the relative statistical weight of lightpaths' lengths, found by the GRASP implementation over 10 different instances. The values are clustered because of the physical structure of the network (see Figure 5.2). Largest lightpaths are those between North and South. This is unavoidable because of the lack of intermediate nodes. Shortest lightpaths are those between nodes of Montevideo and MLD-ROC, RIV-TAC and SAL-PAY. The intermediate cluster is composed of east-west links (SAL-RIV, MLD-SeCIU, et al). Most lightpaths are below 200km because a higher logical density is necessary among nodes of Montevideo, which operate as the core of this network.

Besides betting to statistical behavior, we checked out that such spurious lightpaths are not present into the output of the algorithms.

Fake tours - Even if lightpaths are consistent, tunnels might not be, since new virtual logical connections could be used as a shortcut for other paths.

For instance, on the left network of Figure 5.14, VERAU could be used as an intermediate hop for the tunnel between SeCI2 and PAY. This is unrealizable; it would mean that some private traffic traverses Internet to go back latter.

Luckily this seldom happens. Internet and Clara traffics are the most important in volume; so they are arranged at first place, bottoming available capacity on these virtual links, and forcing reaming tunnels to follow other

paths. This rule really worked well, except for a handful of tunnels of low demand, which could make they way over such narrow capacities. Nevertheless, we isolated such tunnels for solutions found and checked out the existence of capacity onto real alternative paths before validating solutions. Upon the end of Section 5.1.3.3 we comment a more effective alternative.

A similar idea applies to Clara traffic. This case is much simpler because the points of peering with other Clara nodes (on Argentina and Brazil) are more specific: PAY, SAL and RIV.

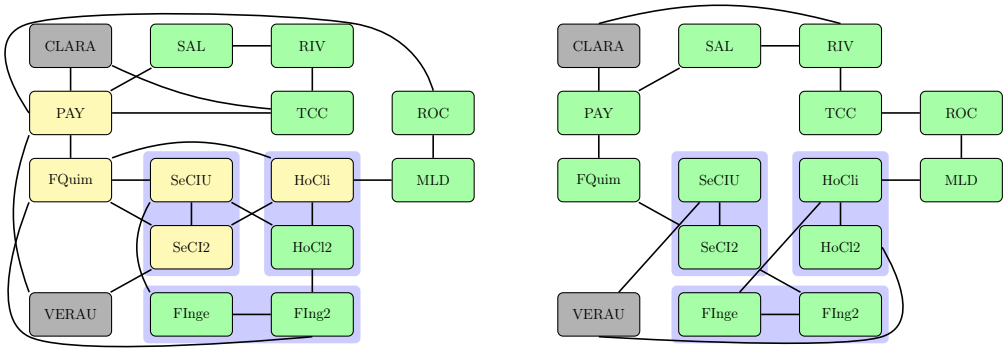


Figure 5.14: Logical networks constructed by EA (ASP-MORNDP, left side) and GRASP (FRP-MORNDP, right side) for scenario 05

As for previous scenarios, Figure 5.14 and Figure 5.15 show network schemes for solutions found for scenarios 05 and 06 for both models. Unlike scenario 02, the solution found here by EA is notoriously worse than that of GRASP.

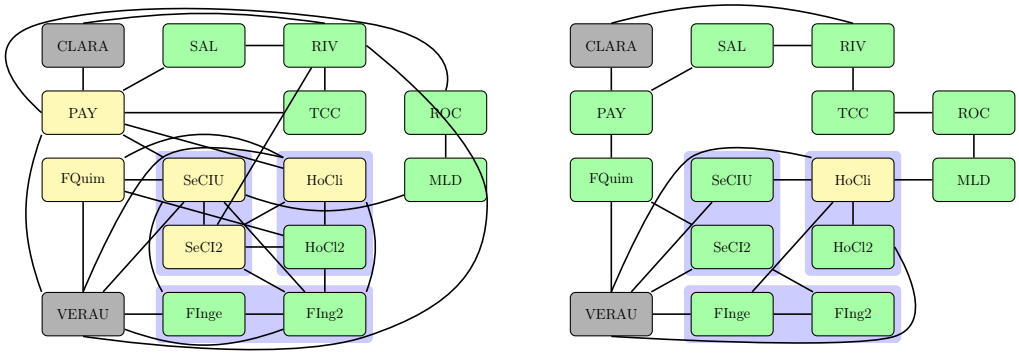


Figure 5.15: Logical networks constructed by EA (ASP-MORNDP, left side) and GRASP (FRP-MORNDP, right side) for scenario 06

The overall information about the performance of both instances is shown in Table 5.7. Unlike previous tables, this one shows two values for lengths on each model: the usual total of equivalent kilometers, and a refined value (enclosed by

parenthesis) where virtual lengths have been excluded. If we hadn't done so, the cost on scenarios 05 and 06 would comprise: backbone, Internet and Clara costs, which is unfair when comparing performance against previous values.

scenario index	*demand level*	*total demand*	*ASP-MORNDP* *#links & kms (EA)*		*FRP-MORNDP* *#links & kms (GRASP)*	
05	low	8,484	23	6,488 (5,893)	17	5,498 (3,022)
06	high	49,769	32	10,888 (4,177)	21	8,945 (3,081)

Table 5.7: Best solutions found with EA and GRASP for scenarios 05 and 06

In addition to refining the values, a fair assessment of quality and cost should somehow integrate the fact that over scenarios 05 and 06, Internet and Clara connections are now protected against physical failures as much as internal connections were on the previous instances.

5.1.3.3 Putting it all together

There are two final entries in Table 5.5 to describe. Scenarios 07 and 08 correspond to instances where VERAU services -dimensioned with the appropriate capacity- are installed into all points of presence (108 points of Figure 5.1), to get Internet's traffic off the RAU as soon as possible. SeCIU remains as the point of interchange for Clara's traffic.

Under these premisses the total demand to be handled by the RAU2's backbone, on *light demand* case (scenario 07) limits to 897Mbps, turning feasible for both models the solution found by GRASP for scenario 01. Under *high demand* case (scenario 08), the traffic onto the backbone is of 6.512Gbps. The solution found for ASP-MORNDP on scenario 02 is still valid onto 08. For FRP-MORNDP however, the instance was recomputed to avoid instabilities of scenario 02.

scenario index	*backbone cost* *ASP*	*& FRP*	*VERAU cost* *ASP*	*& FRP*	*CLARA cost* *ASP*	*& FRP*	*total cost* *ASP*	*& FRP*
AL	–	845,460	–	163,988	–	34,851	–	1,044,299
AH	–	845,460	–	961,323	–	205,122	–	2,011,905
01	416,613	405,171	163,988	163,988	34,851	34,851	615,452	604,010
02	523,730	685,991	961,323	961,323	205,122	205,122	1,690,175	1,852,436
03	487,473	448,045	163,988	163,988	34,851	34,851	686,312	646,884
04	548,407	494,228	961,323	961,323	205,122	205,122	1,714,852	1,660,673
05	812,409	416,613	163,988	163,988	13,035	11,313	989,432	591,914
06	575,841	424,747	672,926	576,794	76,720	66,585	1,325,487	1,068,126
07	405,171	405,171	369,277	369,277	34,851	34,851	809,299	809,299
08	523,730	430,950	893,271	893,271	205,122	205,122	1,622,123	1,529,343

Table 5.8: Comparative of costs (USD/year) for all scenarios of RAU2

The Table 5.8 presents a full comparative of costs on all scenarios. The scenarios AL and AH only are *cost baselines*. AH is not even feasible and is highlighted with

yellow. Scenarios of odd indices correspond to *low demand*. Conversely, even indices correspond to *high demand* scenarios.

The second group of columns express the leasing cost of backbone in dollars-per-year, for solutions found for both models (ASP-MORNDP and FRP-MORNDP). These values use the commercial offer of ANTEL[6] as an economical basis.

VERAU cost columns resumes costs for connecting to Internet. Although designing the backbone imposes a fixed capacity (of 10Gbps) in the process, once known the traffic load supported on each logical interface, the optimal value for VERA's cost was picked up from the price list for dimensioning connections with Internet[7]. For those scenarios where SeCIU is the gateway point (AL to 04), Internet cost was doubled reflecting the duplicity of physical independent connections necessary to provide a resiliency comparable to that of scenarios 05 and 06.

Since Internet connectivity on these scenarios is unaware of the protection mechanism deployed in the backbone, cost for ASP and FRP models match. Differences begin from scenario 05 onwards, at least in theory. VERAU costs for scenario 05 match on both models because both constructions used two connections from the backbone to VERAU (Figure 5.14), and because the cost of VERAU services is independent of the geographical point. With only two connections there is no room for cost spread, other than a better design of the backbone itself, which now delivers Internet traffic off it in a better way. Hence, 163.988 is the best cost on all odd scenarios up to 05, i.e., all scenarios where Internet demand is of 7.022Gbps and it is terminated onto two physically independent connections.

Conversely, scenario 06 shows the lowest value for Internet connectivity and the lowest value for backbone deployment on all high demand scenarios (even indices). This backbone construction even improves the cost of 08, which is weird since Internet traffic doesn't move into the backbone on this scenario. After thoroughly analyzing this case, we conclude that this is just because GRASP was able to find a slightly better solution for this instance. Handling Internet traffic is not a big issue in this scenario, because against a fault there is usually a neighbor to deliver Internet traffic to. CLARA costs on 05 and 06 are also lower because connections to Clara are directly deployed from points near or onto international borders (PAY,RIV and TCC), reducing then the length of connections when compared with those deployed from Montevideo.

The final columns show the total cost in each case and model. ASP-MORNP/EA found the best solution in one of them (02). The scenario 07 was a tie between models. In all instances but one, the solutions found by FRP-MORNP/GRASP were the best. Furthermore, the most cost-effective solutions for low demand (odd scenarios) and high demand (even scenarios) were both found by FRP-MORNP/GRASP, and are 05 and 06 respectively (highlighted with blue).

[6]Integrated with the original draft of the project sketched in Figure 5.3.
[7]A similar procedure was followed with Clara connections.

The *cost spread* compared with manually designed solutions is respectively of: 61.4% and 88.4%. This is outstanding when we consider that one of them wasn't even feasible in the first place. We must conclude then that the usage of cheaper technologies combined with an efficient design, allows SeCIU to construct a network 200 times faster than current with a lower budget.

We extend now on aspects other than numerical results. This application case was actually a branch of a separated project: *the construction of the first open-source academic network of the world*. This faculty is designing a backbone entirely composed of PC-like hardware and open-source software. With this goal in mind, it is important to check out that the performance requirements of nodes can be covered with the open-source implementation. We are certain that a mere software router can handle up to 10Gbps on its forwarding backplane. This value can raise up to 40Gbps with hardware acceleration. Nodes in Figure 5.9, Figure 5.11, Figure 5.12, Figure 5.14 and Figure 5.15 were colored to reflect these limits. Green nodes are those where traffic traversing node's backplane was below 10Gbps on all physical faults. Conversely, yellow nodes are those where backplane load sometimes overpassed the 10Gbps limit, but never the 40Gbps one. Finally, on red nodes the 40Gbps limit is violated under some failure scenario, but these nodes are not part of constructions and we can be sure the project is theoretically realizable.

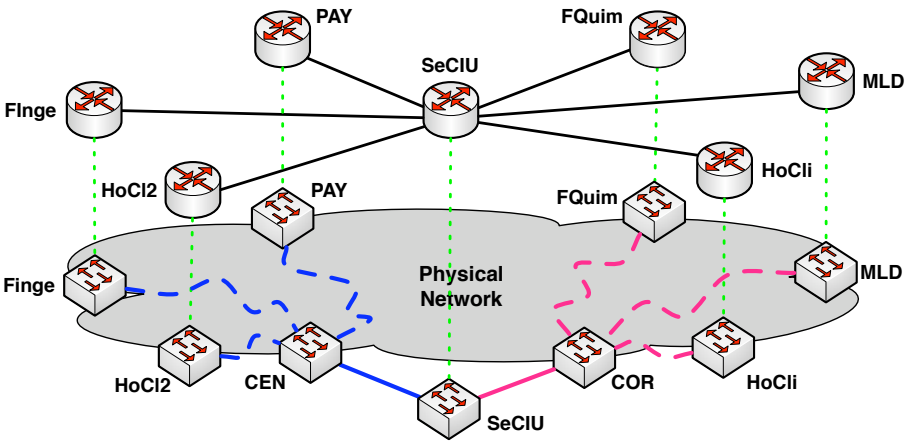


Figure 5.16: Paths distribution in scenario 02

To conclude this synthesis we analyze the effect of *bumping* in the constuctions. Unlike ASP-MORNDP, solutions found by FRP-MORNDP might not be directly realizable. The reason is that standard IGPs capabilities are below of those used by our *isFeasible* heuristic (see Section 4.2.2). Setting up tunnels in decreasing order of volume is realistic, even in an operational environment. Unfortunately, the bumping process, sometimes necessary to start over again the paths assignment after a failure, is delicate and requires the intervention of an external NMS.

However, after analyzing the solutions found by our GRASP for RAU's instances, we conclude that bumping is only necessary on scenarios 02 and 06. The root cause

is different for both cases. The problem with scenario 02 is that the hub/central POP only counts one node (SeCIU), and six logical connections, which in turn are grouped into two physically independent groups. Physically, SeCIU is connected to CEN and COR, and from them lightpaths make they way to their correspondent logical nodes. Figure 5.16 shows with blue and purple paths, how lightpaths are arranged over (SeCIU,CEN) and (SeCIU,COR) upon the solution found for scenario 02. The total demand towards SeCIU is of 28.162Gbps. When the heuristic starts up -all logical links are operational- tunnels are roughly uniformly distributed over the six logical links. Nevertheless, when either (SeCIU,CEN) or (SeCIU,COR) fail, the initial arrangement of paths over the surviving logical links (which sum 30Gbps), is incompatible with the demands to be detoured over them. In fact, the case is pretty similar to that described in Figure 4.11.

This can be easily solved by adding an extra node into SeCIU (as in the following scenarios), and some extra connections from it to other metropolitan nodes. After doing this, bumping is no longer necessary. The cost of such a change only increments in 4% the cost found by GRASP for that scenario.

The other case were *bumping* was necessary is scenario 06. The root cause here is much easier to explain. The heuristic used by the *isFeasible* function, doesn't use costs to calculate distances; it only considers the number of hops. Thus, when available, the path (for instance): SeCIU - VERAU - SeCI2 is preferred to route the tunnel (SeCIU, SeCI2) (see Figure 5.15). Situations of such type happen often, saturating then the capacity to Internet with spurious traffic, which must be rearranged after a failure to make room to genuine demands. This kind of situations was anticipated by us (see *fake tours* on page 145). It is actually a side effect of including virtual nodes and links. Fortunately, it can be easily fixed on real world applications. First of all, internal traffic wouldn't use connections to Internet to make its way, but even so, it is possible (usual actually) setting costs for links over standard IGPs. Adjusting such costs, most of the undesired behaviors can be prevented.

We considered these automatic adjustment of costs of logical links, as a future improvement to our algorithm. For applications analyzed during this work, such adjustment wasn't necessary.

5.2 ANTEL

ANTEL (Administración Nacional de Telecomunicaciones/National Administration of Telecommunications) is the Uruguayan government-owned telecommunications company, solely licensed for commercializing landline telephony services. Since the unbundling process never took place in Uruguay, ANTEL also holds in fact the monopoly for selling fixed data services in the country. Complementarily, it provides mobile phone services and radio connections for Internet access, although both are in competition (with Claro, Movistar and Dedicado). Cable operators are no licensed to bring other services than Cable TV.

Regarding the market-share, ANTEL holds 100% (over 1 million) of the fixed telephony lines, 48% of the 5.13 million mobile phone services, around 95% of the 653000 broadband Internet access connections, and over 70% (USD 860 million) of the total incomes of non-broadcast telecommunications services, being then by far the most important ISP of this country, and that with the largest/widest backbone network.

On the following sections we will show how this research contributed to improve the quality of ANTEL's backbone design. Unlike the application described into Section 5.1, where we are allowed to give full information on any topic, many ANTEL's details are covered by a Non-Disclosure Agreement (NDA), which prevents us from using real-values. Instead, we show information through referential values, which capture the essence of the improvements.

5.2.1 Drivers of the change process

By the late 2000's, ANTEL had the typical mix of legacy technologies of a traditional TELCO (Figure 1.1), and just as most of them, was on its way to take the next technological leap. Besides of a TDM/PSTN network for sustaining phone services, ANTEL had a widely deployed ATM backbone network, whose primary function was *aggregation*, i.e., gathering Internet's traffic from access points towards an *IP core*. This architecture of specialized components within the backbone was described in Section 2.2.1 (see Figure 2.16). By that time when ATM was the technology for the aggregation function, the existence of BRASes was more a necessity than a choice, since ATM nodes do not embed the IP protocol natively.

PSTN and ATM networks were both overlays of an SDH layer. Hence, the SDH network also spanned the entire country. The *IP core* on the other hand had a limited circumscription, with presence only into four POPs within Montevideo. Its connections were supported over a metro-ethernet network, and protected by Spanning Tree Protocol (STP): an Ethernet native mechanisms to restore connectivity against a fiber cut, very poor in terms of performance.

Several forces operate into this process of change; some of them are opposite. Just like on RAU's application, the primal force here is the demand for higher bandwidths, although conversely there is an aggressive national policy for keeping prices as low as possible. This policy consists in: freezing prices for Internet access products, performing periodic updates of access line-rates for existing services, which are paired with the releasing of cheaper access products intended to reach new marked segments iteratively. Along a decade, this strategy has pushed up to Uruguay to a leading position of Internet density and performance.

To sustain this commercial policy, the links of the ATM backbone implemented through SDH interfaces, needed to be substituted by cheaper point-to-point Ethernet connections, which are natively supported as a transport media for MPLS frames, but not for ATM cells. Luckily, the recently standardized *pseudo-wire em-*

ulation[8] for Ethernet edge-to-edge connections, turned viable the transparent replacement of one technology by the other. Hence, the progressive substitution of ATM nodes by IP/MPLS ones was scheduled for the aggregation network.

Within Montevideo's metropolitan area, where plenty of optical fibers were laid, this substitution took place swiftly. Conversely, in the rest of the country optical fiber resources were scarce, so a thorough planning was required.

The evolution of the aggregation network was accompanied by an update of the IP core. Although easy to understand and deploy, STP is perhaps the best worst-example for efficiency on resource utilization. At any time and in order to avoid forwarding loops, STP builds a tree over a 2-edge-connected graph, wasting thereby links intensively and avoiding in fact any kind of traffic engineering[9]. Besides, after a failure on a link, STP requires several seconds to reconstruct a new tree, which is absolutely inappropriate for backbone standards.

Thus, the future of the IP core was pretty clear: pure IP nodes should evolve to IP/MPLS nodes, the metro-ethernet layer should be suppressed and instead, several optical point-to-point connections would be established between nodes. However, a mere substitution of technologies wasn't enough to be rise to the occasion.

The sustained growth of international traffic (mostly Internet's traffic) increases the frequency of changes in agreements with other ISPs. Upon the former IP core, these changes were set-up one-by-one and each change included two separate portions: one for the international connection, leased to International Carriers from Montevideo towards the counterpart NAP (Network Access Point)[10], and a complementary portion corresponding to the connection with the AS (Autonomous System) of the peer ISP.

Coordinating such changes with two independent counterparts (Carrier and ISP) was very hard and time consuming. It is much easier when both portions can be maintained separately, that's the reason why NAPs exist. Having presence into a NAP separates both kinds of connections and allows a faster and more flexible response to support growing of Internet demand. Hence, ANTEL decided to extend its IP core to span to *Terremark's NAP of the Americas*, an important NAP and Data Center in the US territory (Miami).

There is only a pair of additional elements to complete the big picture. Up to the present days, SDH prevails as the technology to transparently transport connections across several providers/domains. So the connections between the portion of the IP core at Montevideo and the new POP at Miami, could be implemented either by installing new IP POPs onto the borders with Argentina and Brazil[11], or by potentiating the national SDH network to transparently connect Montevideo with

[8] Martini draft: IETF's PWE3, Luca Martini, Cisco Systems.
[9] Over a tree, there is a unique path to connect pairs of point.
[10] Also known as IXPs, for Internet eXchange Points.
[11] By those days they were the only way to reach submarine cables of International Carriers.

these borders. The second choice was taken.

Therefore, the optical transport network into the interior departments should serve as a foundation for two overlays of increasing importance: the new IP/MPLS Aggregation Network and a newer/potentiated SDH Network, commended to connect Montevideo with the rest of the world. The pressure for having more and more optical connections forced to deploy DWDM massively into the interior departments. Finally, the excessive percentage of Internet traffic ending outside of its AS, pushed ANTEL to assess the convenience of installing local Data Centers of highest tiers, to promote the existence of local content.

5.2.2 Reduction to scenarios

Just like in the case of RAU's project, the strategy chosen here to assist in decisions, consists in creating problem instances to quantitatively assess the economical suitability of the different alternatives/scenarios.

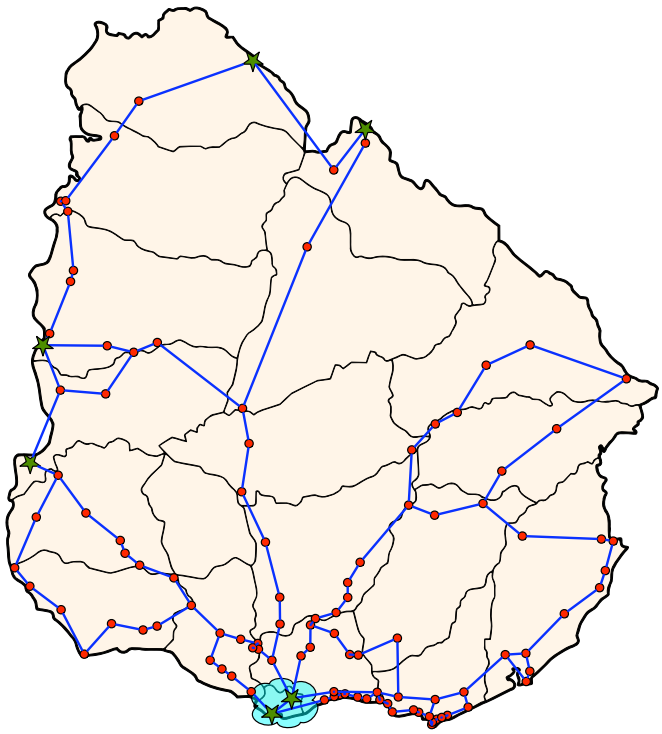


Figure 5.17: Important points of ANTEL's optical network

Nodes in Figure 5.17 represent the existing DWDM infrastructure. Except for a bunch of villages[12], these points cover the entire population of the country.

The metropolitan area of Montevideo contains around 67% of the total population of Uruguay. Its geographical presence is marked with a pale cyan cloud in

[12]Which are hanged from some of these POPs.

Figure 5.17. The project of deployment of the IP/MPLS aggregation network within this area was advanced by the time when this work started (2010), so we excluded it from the analysis.

However, the traffic demand coming from this area is actually considered, because international traffic traverses this network before leaving the country. Those points through which Uruguay is connected to neighbor countries (Argentina and Brazil), are marked with green stars in Figure 5.17, so are the two POPs into Montevideo where metropolitan and interior networks connect to each other.

It is worth pointing out that physical structure of the optical transport network, matches criteria of design proper of a network intended to course telephonic traffic.

Although accurate in geographical terms, the Figure 5.17 is hard to read because some nodes overlap. Instead and as Harry Beck did with the London Underground Tube map, we shall use Figure 5.18, an alternate representation that dispenses of the geographical distances.

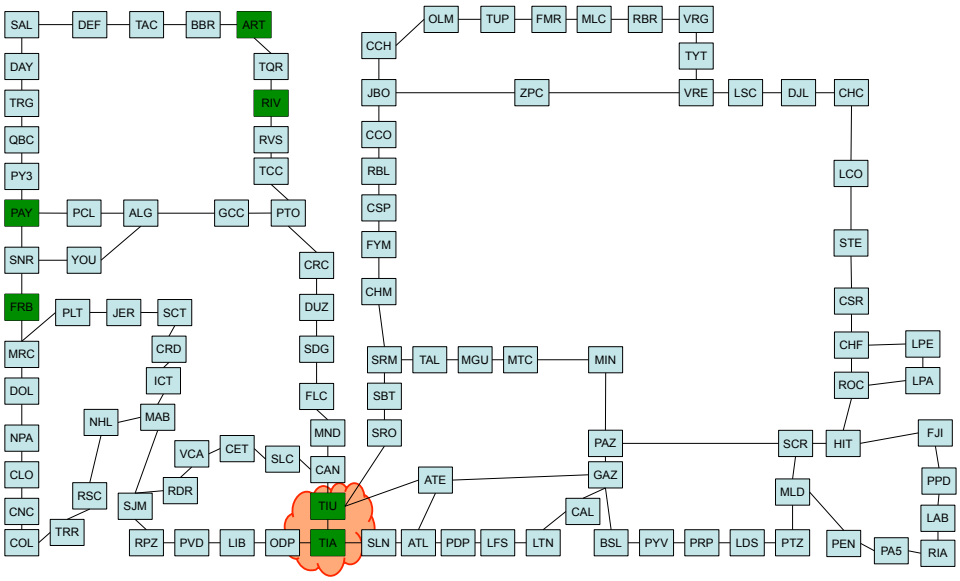


Figure 5.18: A non-geographical representation of ANTEL's network

Combining some macro elements commented in Section 5.2.1 with physical details previously depicted, we should conclude that in abstract terms, the main subcomponents of ANTEL's design problem are those sketched in Figure 5.19.

Topologically speaking, Uruguay comprises three zones: the *eastern territory*, the *western territory* and the *metropolitan area*. The *aggregation network*, geographically dispersed all over the country, is responsible of gathering and delivering the traffic of the customers towards the metropolitan *public IP core*. The deployment of the portion of the *aggregation network* situated at the metropolitan area was advanced by 2010, so we focus on the eastern and western regions. There are no points of the aggregation network outside of the national boundaries.

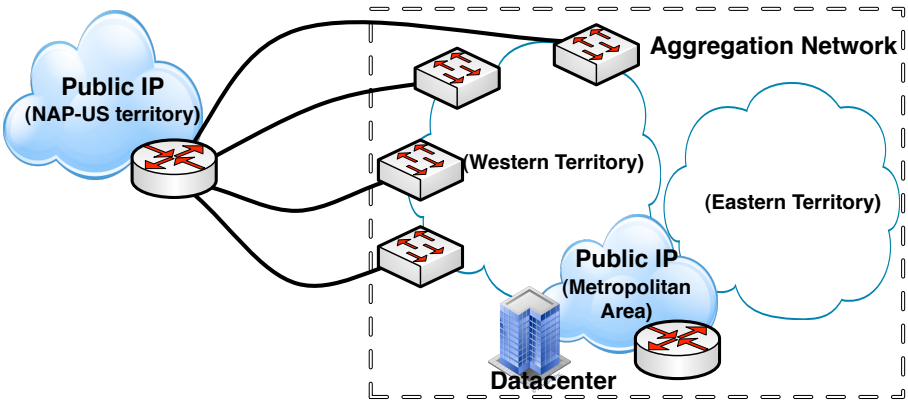


Figure 5.19: Network portions of ANTEL's design problem

The public IP core, where the AS of this ISP is implemented, centralizes the international connections with other ISPs as well as those to Data Centers of local content providers. The public IP core is geographically concentrated and only has POPs in Montevideo and into a NAP in Miami (blue clouds of Figure 5.19).

By 2010, all international connections between the NAP and the national boundaries should be routed through four points (ART, FRB, PAY and RIV), all of them located at the western territory.

Several planning concerns arise from the situation exposed:

- Is the current architecture convenient? or Would it be better to merge both IP/MPLS networks?

  When an ATM network performs the aggregation function, the existence of a separate IP network is inevitable. But now that both share the same technology this duality of functions deserves to be revised.

- Are the IT infrastructure investments necessary to increase the percentage of local content profitable?

  For a country so far away from the main sources of content, international connections represent an important component of the total expenditures.

  They can be reduced by installing local Data Centers and by signing agreements with major content providers (e.g. Google, Akamai) to install their: mirrors, caches or other forms of CDNs on these Data Centers.

  The investments necessary to deploy and maintain Data Centers are predictable and controllable. Conversely, savings in network costs caused by the reduction of international traffic, are far from being straightforward.

- Which would be the optimal network to fulfill every demand requirement at lowest cost possible?

- How sensitive is the cost to changes in demands?

  On a regular business plan, costs and expenditures are accurately foreseen; on a telecommunications company however, they are not that predictable. The structure of costs of an Internet access service comprises elements of diverse nature, which involve users' behavior among others.

  Being able to anticipate costs coming from different alternative marketing plans, is a valuable tool for commercial staffs.

Traffic forecasts on this application follow similar criterion to those of RAU's. However, unlike RAU's case, filtering traffic in an ISP's network is unkind and unusual. Rather, *heavy users* are controlled by indirect means. For instance, one alternative is counting with a cheaper portfolio of access products, with a per-month limited volume of Gigabytes to move.

To estimate future demand, we received from the marketing team three products portfolios and followed a users segmentation process similar to that of RAU's forecast. The details of these portfolios are no relevant. Their demand forecasts for Internet's traffic are detailed in Table 5.9. The aggregated demands for other kinds of services (i.e. phone service, VPNs and IPTV) sums 13050Mbps.

traffic scenario	*offered bandwidth*	*traffic from HU*	*traffic from LU*	*total traffic*	*Internet demand*	*per-user demand*	*overbooking factor*
Low	1,896,300	20,486	15,803	36,289	40,412	0.082	47
Medium	1,084,860	26,615	16,567	43,182	45,243	0.092	24
High	1,896,300	56,789	17,893	74,682	79,159	0.162	24

Table 5.9: Internet traffic and demands information, expressed in Mbps

Traffic scenarios of Table 5.9 are associated with different combinations of access rates and commercial plans for tolling downloaded volume. These traffic forecast were assembled on 2010 for an expected universe of 500.000 broadband Internet users on 2014. Actual values are higher (620.000) because on 2011 ANTEL released *Universal Hogares*, an Internet access service free of charge for those customers which hold a landline phone service. However, the volume of moveable data for each one of these services is limited to 1GB-per-month, slightly affecting then the total demands of Table 5.9 by a factor lower than 5%.

Table 5.10 shows the main characteristics of the twelve scenarios arising from ANTEL's design concerns. They are clustered into three groups according on traffic forecasts associated with commercial portfolios (Table 5.9). Each one of these clusters is split into four instances, which are determined by combinations of boolean values: *local content* and *merged networks*. When *local content* is marked as *high* (H), it means that Data Centers are installed into the metropolitan area, rising the percentage of local content (i.e. traffic ended up locally) to 25%. Conversely, when *local content* is marked as *low* (L), all Internet's traffic consists of international traffic. This is the reason why columns *Internet demand* and *international demand*

scenario index	*traffic scenario*	*local content*	*merged networks*	*Internet demand*	*international demand*	*total demand*
01	Low	H	N	40,412	30,309	53,462
02	Low	H	Y	40,412	30,309	53,462
03	Low	L	N	40,412	40,412	53,462
04	Low	L	Y	40,412	40,412	53,462
05	Medium	H	N	45,243	33,932	58,293
06	Medium	H	Y	45,243	33,932	58,293
07	Medium	L	N	45,243	45,243	58,293
08	Medium	L	Y	45,243	45,243	58,293
09	High	H	N	79,159	59,369	92,209
10	High	H	Y	79,159	59,369	92,209
11	High	L	N	79,159	79,159	92,209
12	High	L	Y	79,159	79,159	92,209

Table 5.10: Scenarios for representative ANTEL's design concerns

match on rows where local content is low, whereas on the others the second column only represents 75% of the first.

The column *merged networks* of Table 5.10 indicates whether the Aggregation and IP Core of Figure 5.19 were integrated or not into one single IP/MPLS network. They remain as separate entities on odd scenarios, which means that the aggregation network to design, only has to deliver traffic to the metropolitan area. Conversely, on even scenarios: the metropolitan sources, the NAP POP and their connections are part of the instance to solve. To reflect this situation we used a NAP node, which resembles the VERAU node of RAU's application on scenarios 05 and 06.

The *total demand* column adds Internet and non-Internet demands. Traffic types other than Internet (e.g. VoIP, VPNs and IPTV) are served exclusively by the IP/MPLS aggregation network in all scenarios.

5.2.3 Assessing costs of decisions

Just as we did for RAU's case, ANTEL's scenarios detailed in Table 5.10 were translated into problem instances of ASP-MORNDP and FRP-MORNDP, to use metaheuristics to find quality solutions for them. Besides, by 2010 the bit-rate chosen to be used onto backbone's interfaces was 10Gbps, matching RAU's application. A couple of years latter some interfaces of ANTEL's core were updated to 40Gbps, but this change doesn't affect our instances because ANTEL's core circumscribes to Montevideo and we are leaving this area out of the scope. Many characteristics of both applications are very much alike. There are also some important differences we should mention.

Although the portion of ANTEL's backbone to design here does not span the metropolitan area, there are plenty of virtual metropolitan nodes to add, either

for those scenarios where metropolitan traffic traverses the backbone network (even indices), or for those where the traffic from interior departments ends up into the IP core, located at Montevideo (odd indices).

To avoid stability issues many virtual nodes were added into the two POPs where traffic is interchanged with Montevideo and surroundings (i.e. TIA and TIU), just as we did for: SeCIU, HoCli and FInge on RAU's case. An identical approach was taken into POPs: ART, FRB, PAY and RIV, because they are the points through which international traffic is routed. As a result, the number of virtual nodes into this application rises up to 34 on some instances, over a total of up to 68 nodes.

Another remarkable difference with RAU's case is how Internet cost is calculated. On RAU's case the calculation reduces to search for an appropriate entry into the list of prices for the required bandwidth. On ANTEL's case, this cost has two components: the cost for leasing a transit connection with other ISP in Miami is also as clear as a lookup into a list of prices, but there is also a complementary cost to connect Uruguay with Miami. As we shall see, this is in fact the most significant.

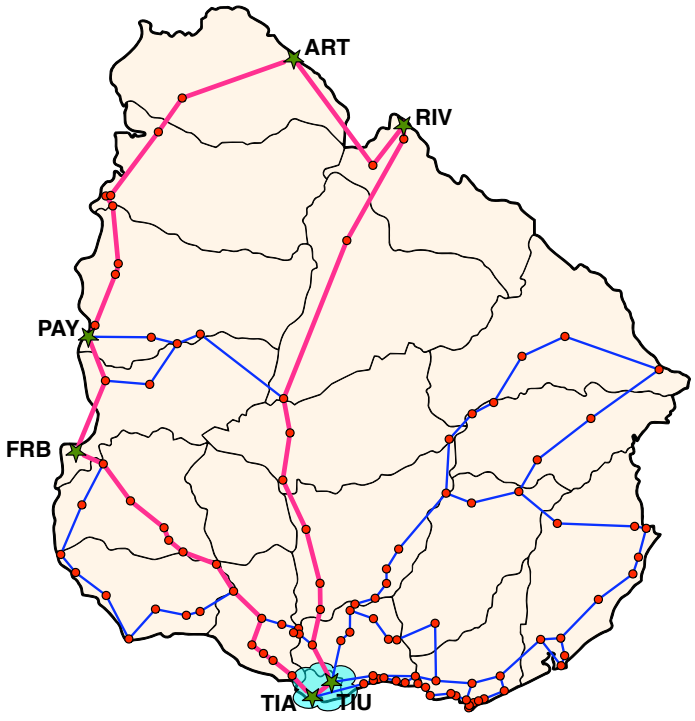


Figure 5.20: Optimal ring for protecting international connections on SDH

For even scenarios, where the IP/MPLS backbone fulfills a double function of aggregation and Internet routing, the NAP in Miami is a POP of the backbone and the international connections are part of the problem to solve. After translating costs to kilometers -as in RAU's case-, a connection between the national boundaries and Miami matches an optical circuit of 10.000km, which is approximately the submarine

length[13] from Uruguay to Florida (USA). For odd scenarios, where Internet traffic is routed from Montevideo to Miami using SDH connections, we use the optimal "1+1" protection scheme of SDH as a cost basis for connecting Montevideo with Miami. The cost for the international portion of this connection is known, whereas the national portion is pretty simple and can be computed manually, because there is a cycle that spans: ART, FRB, PAY, RIV, TIA and TIU using the minimum length. This cycle is highlighted with purple in Figure 5.20 and its cost increases the international connection by 10%.

Table 5.11 shows the comparative of results for both models on all instances. These costs are only referential, i.e., the ratio among them is real, but values themselves are not. Scenarios are clustered by their volume of demand.

scenario index	*Internet cost*	*backbone cost*		*international cost*		*total cost*	
		ASP	*FRP*	*ASP*	*FRP*	*ASP*	*FRP*
01	916	287	265	3,716	3,716	4,919	4,897
02	916	700	369	2,554	2,554	4,170	3,839
03	1,198	287	265	4,645	4,645	6,130	6,108
04	1,198	559	419	3,406	2,980	5,163	4,597
05	987	277	265	3,716	3,716	4,980	4,968
06	987	723	383	2,980	2,554	4,690	3,924
07	1,339	277	265	4,645	4,645	6,261	6,249
08	1,339	573	484	3,832	2,980	5,744	4,803
09	1,691	311	313	5,574	5,574	7,576	7,578
10	1,691	740	617	5,109	3,406	7,540	5,714
11	2,255	311	313	7,432	7,432	9,998	10,000
12	2,255	846	725	6,386	4,683	9,487	7,663

Table 5.11: Comparative of costs for all scenarios of ANTEL

The column *Internet cost* corresponds to the cost of leasing Internet capacity within the NAP of Miami. Entries on this column are proportional to the column *international demand* in Table 5.10. This cost is unaware of how the aggregation network or the IP core is designed, so there is a single per-column-entry.

The column *backbone cost* shows the best cost for deploying the IP/MPLS backbone into the interior departments for each instance and model. For even scenarios, where aggregation network and IP core are merged, the total cost has been disaggregated into national and international to compare solutions fairly. Within each cluster, values on odd rows match. These matchings occur on scenarios: 01 and 03, 05 and 07, 09 and 11, and they are highlighted with pale: blue, yellow and purple respectively. On each one of these pairs of scenarios the demand level is the same; and on all of them the IP core is a separate metropolitan network. The only difference comes from the fact that on scenarios: 01, 05 and 09 a fraction of the

[13] That of the path followed by submarine optical cables.

Internet traffic is terminated into local Date Centers rather than flowing towards the NAP. This fact passes unnoticed to the aggregation network, so the resulting designs match, regardless of what percentage of traffic ends up into Miami.

The column *international cost* shows the cost necessary to connect Uruguay with Miami. On odd scenarios these connections are deployed from Montevideo across SDH connections, so costs between models match because there is no further optimization possible. The costs on rows associated with even scenarios, correspond to the connections from the national boundaries towards Miami. They don't have to match because here the algorithms do have room to improve the quality.

Anyhow, the backbone cost on each even scenario is always higher than in its precedent, because on all of them the IP/MPLS backbone is also responsible of moving the metropolitan traffic towards the national boundaries, whereas on odd scenarios this cost is integrated to the *international cost* column. To globally evaluate the cost of constructions we must account it as a whole. The column *total cost* synthesizes the best costs on each component of ANTEL's backbone for each instance. The best solution for each instance is highlighted on Table 5.11. Conversely, the best solution found for the same level of demand (cluster) is marked with cyan.

The performance of FRP-MORNDP(GRASP) surpasses ASP-MORNDP(EA)'s on all scenarios but 09 and 11. The spread between values rounds 11.7%, reaching 32% in one instance (scenario 10). Conversely, on those instances when ASP defeats FRP, it only does it by 0.02%. Best solution on each cluster is always found for FRP, and the spread with the worst cost within these clusters ranges from 60% to 75%, impressive in terms of absolute costs.

ASP					**FRP**				
scenario index	*Internet cost (%)*	*backbone cost (%)*	*international cost (%)*	*total cost (%)*	*scenario index*	*Internet cost (%)*	*backbone cost (%)*	*international cost (%)*	*total cost (%)*
02	22.0%	16.8%	61.2%	4,170	02	23.9%	9.6%	66.5%	3,839
06	21.1%	15.4%	63.5%	4,690	06	25.2%	9.7%	65.1%	3,924
01	18.6%	5.9%	75.5%	4,919	04	26.1%	9.1%	64.8%	4,597
05	19.8%	5.6%	74.6%	4,980	08	27.9%	10.1%	62.0%	4,803
04	23.2%	10.8%	66.0%	5,163	01	18.7%	5.4%	75.9%	4,897
08	23.3%	10.0%	66.7%	5,744	05	19.9%	5.3%	74.8%	4,968
03	19.5%	4.7%	75.8%	6,130	10	29.6%	10.8%	59.6%	5,714
07	21.4%	4.4%	74.2%	6,261	03	19.6%	4.3%	76.1%	6,108
10	22.4%	9.8%	67.8%	7,540	07	21.5%	4.2%	74.3%	6,249
09	22.3%	4.1%	73.6%	7,576	09	22.3%	4.1%	73.6%	7,578
12	23.8%	8.9%	67.3%	9,487	12	29.4%	9.5%	61.1%	7,663
11	22.6%	3.1%	74.3%	9,998	11	22.6%	3.1%	74.3%	10,000
	21.7%	**8.3%**	**70.0%**	**6,388**		**23.9%**	**7.1%**	**69.0%**	**5,862**

Table 5.12: Per-model/per-component cost for all scenarios of ANTEL

However, comparing solutions only by their cost could not make sense. The value for the lowest cost solution of each cluster surpasses the precedent, but it also does it the demand to fulfill. The prices of products portfolio 1 (low) are lower than those of 2 (medium), and these in turn are lower than those of 3 (high). A serious economical assessment should integrate incomes and outcomes, but this is out of

the scope of this work. Table 5.12 presents an alternative view for the results of Table 5.11. In it, instances have been ordered by increasing *total cost*. The relative weight of sub-components is presented for each model. In general, those scenarios with lowest values of *international demand* in Table 5.10 (cyan rows: 01, 02, 05 and 06), tend to appear over first places, whereas those with highest values (yellow rows: 09, 10, 11 and 12) concentrate at the end. This is strict for ASP, not so for FRP. Besides, on all cases the relative weight of components: *Internet*, *backbone* and *international* is regular, and they respectively represent around: 22.7%, 7.7% and 69.6%. Solutions found for ASP are in average 9% more expensive than those of FRP, which makes sense due to the relaxation relationship between them.

The sum of all of the costs to connect to other ISPs -including the complementary international connections- rounds 92.3% of the total. Thus, it isn't surprising that instances associated with scenarios where the percentage of local content is higher (01, 02, 05, 06, 09 and 10) result in solutions of lower cost than those where all of the traffic is international (03, 04, 07, 08, 11 and 12). However, an accurate evaluation of the quality of these solutions must also integrate the costs to build the Data Centers, which are out of the scope of this work either.

At first glance these results might look discouraging, since unlike RAU's case, the global weight of the primary objective to optimize here (the interior aggregation network) only represents from 3% to 17% of the total. Combinations of: demand volume, percentage of international traffic and the existence of specialized components inside of the backbone, define the scenarios in Table 5.10. The first two of these variants were already commented. We expected some savings from merging aggregation and IP core, because when both are distinct, the traffic from the western region is moved first towards Montevideo (the IP core) to go back latter towards national boundaries along the same conduits. However, the review of the numbers brought no great expectations. The Internet traffic terminated into the western territory is around 18% of the total, while both: the *national component of the international connection* and the *aggregation network* represent less than 15% of the total, thus, global savings expectable from this action should be negligible; and yet savings coming from merging networks are impressive.

scenario	*backbone cost*				*international cost*				*total cost*			
indices	*ASP*		*FRP*		*ASP*		*FRP*		*ASP*		*FRP*	
01 - 02	-413	-143.9%	-104	-39.2%	1,162	31.3%	1,162	31.3%	749	15.2%	1,058	21.6%
03 - 04	-272	-94.8%	-154	-58.1%	1,239	26.7%	1,665	35.8%	967	15.8%	1,511	24.7%
05 - 06	-446	-161.0%	-118	-44.5%	736	19.8%	1,162	31.3%	290	5.8%	1,044	21.0%
07 - 08	-296	-106.9%	-219	-82.6%	813	17.5%	1,665	35.8%	517	8.3%	1,446	23.1%
09 - 10	-429	-137.9%	-304	-97.1%	465	8.3%	2,168	38.9%	36	0.5%	1,864	24.6%
11 - 12	-535	-172.0%	-412	-131.6%	1,046	14.1%	2,749	37.0%	511	5.1%	2,337	23.4%

Table 5.13: Cost variations (savings) resulting from merging networks

Table 5.13 shows the per-component savings from such fusion. Up from the scenarios definitions (Table 5.10), this reduces to compare: 01 with 02, 03 with 04

and so on. The relative change, resulting from moving to the following scenario is shown for each component and for each model. Let us note that on both models, the merging always reduces the international cost, although at the expenses of the aggregation network, which is consistent with a previous observation.

Besides, the merging always reduces the total cost, although now the behavior amid models changes. For ASP instances, the relative reduction decreases, as the demand gets bigger. Conversely, for FRP instances, the relative reduction is sustained and it is higher than ASP's on all instances. These savings are concentrated upon international connections. In fact, for FRP model they are much higher than the cost of the national backbone itself (compare Table 5.11 with Table 5.13). To analyze these savings, we should broaden our outlook upon network structure.

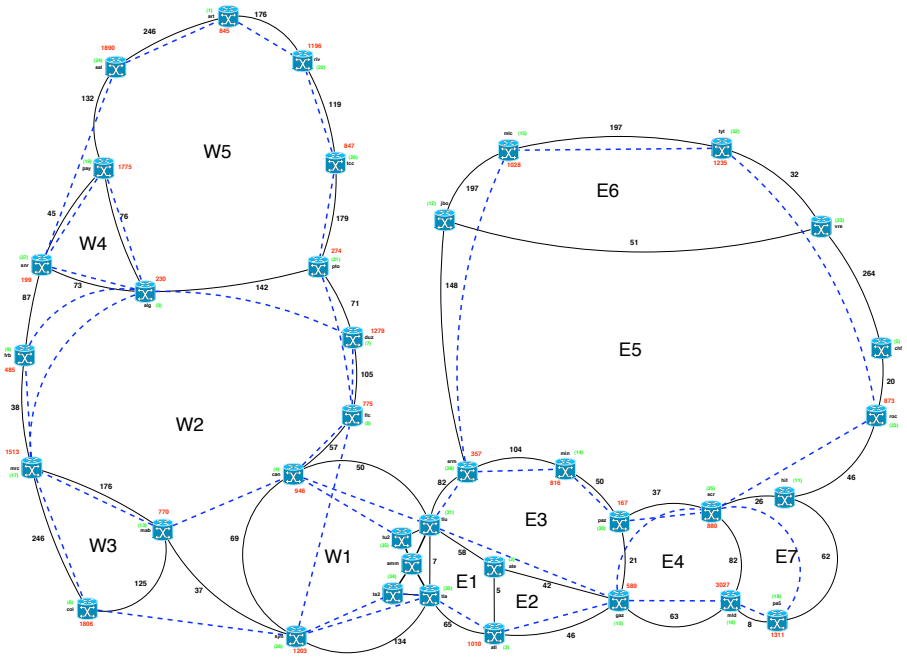


Figure 5.21: Solution found for instances 11 of ANTEL

Figure 5.21 and Figure 5.22 show a simplified representation of solutions found for FRP to instances 11 and 12, i.e., those with the highest level of international demand. Physical faces have been labeled to facilitate subsequent explanations. Those images corresponding the the other scenarios are in Chapter 7. Let us observe that configurations found on eastern faces for both instances match. This is consistent with the fact that regardless of which is the following hop to Internet, the traffic coming from the eastern region have to flow towards Montevideo in first term. Conversely, all faces onto the western area are more intensively used in instance 12, which is also consistent, because here the international traffic traverses them before leaving the country. The additional number of optical circuits required from physical links, limits to two or three. That's it in general, expect for faces W1 and W2, where the number increases substantially.

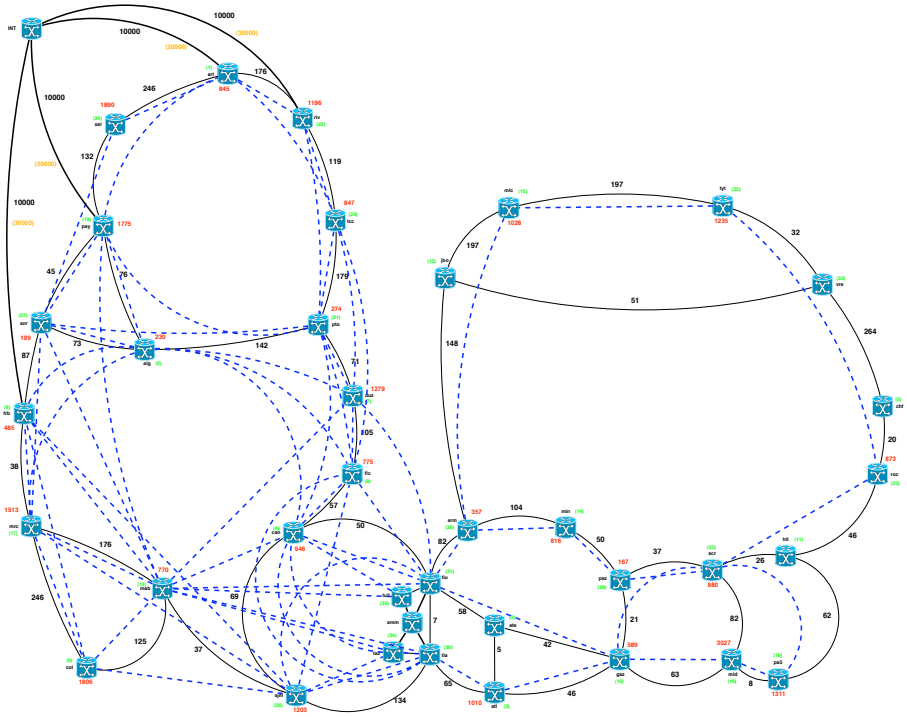


Figure 5.22: Solution found for instances 12 of ANTEL

Taking a closer look to Figure 5.18, we can isolate structural issues that are the root cause of this behavior. The face W2 contains the most critical bonds of this network. They are marked in Figure 5.23 with dashed-red curves.

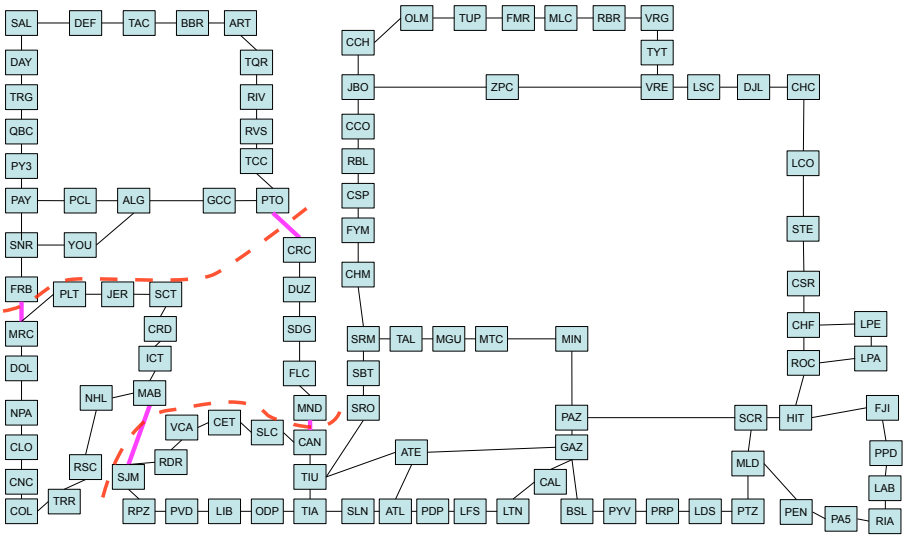


Figure 5.23: Critical bonds for ANTEL's optical network

The size of both physical bonds is 2, then the minimal logical capacity necessary to satisfy (3.9) (Lemma 5) doubles the demand between components. If we add

into the equation the fact that this face separates the most important sources of traffic from all international connections, the overpopulation of logical links becomes evident. Before the fusion, all the international connections are duplicated due to the standard SDH "1+1" protection. This mechanism demands doubling the costs upon the most expensive connections. After merging networks and while traffic-engineering engines may count with the degree of freedom proper of the FRP-MORNDP model, more cost effective solutions can be found.

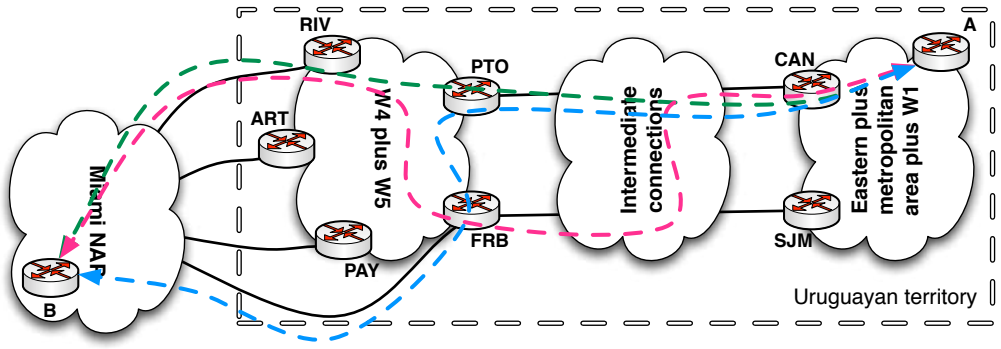


Figure 5.24: Abstract structure and How FRP-MORNDP operates

Figure 5.24 presents an abstract/modular representation of the optical network. In a nutshell, four zones are demarcated: eastern plus metropolitan areas, an intermediate connections zone, an international gateway zone (W4 + W5), and the POP at Miami. The subnetwork that spans W4 and W5 is connected with Miami through four independent and expensive connections. The eastern plus metropolitan areas generate over 85% of the total international demand, and most of it should cross the intermediate zone to make its way towards Miami.

Whenever a fault affects the active path of a tunnel, it is possible to find a national detour when enough local spare capacity is provided. For instance, in Figure 5.24 the active path to connect A and B is highlighted with a green-dashed curve. A fault that leaves inoperative the link from PTO to the intermediate zone, can be worked around by routing this path across FRB to follow up latter its original path (see purple-dashed curve). In an analogous way, a tunnel can recover from a fault over international connections between RIV and Miami detouring though FRB without affecting initial hops of the path. Both detours are decoupled, i.e., they can operate independently.

Thus, to build cost-effective solutions the algorithm simply populates critical bonds with national/cheaper logical links, rather than international and more expensive ones. This is realizable because FRP-MORNDP allows the construction of a sequential/bond-after-bond scheme of protections, all along the way.

Due to Lemma 5 the theoretical limit of resources efficiency over such a bond of size 4 (the international bond) is of 75% (3/4), rather than the standard 50% of the SDH protection. In other words, to protect 100Gbps from Montevideo to

Miami, SDH requires 200Gbps of international capacity, whereas the FRP flavor of IP/MPLS only requires 100Gbps/0.75 ≈ 133Gbps, saving then a 33.3% of international capacity, which explains concisely the spread seen in Table 5.13.

ASP-MORNDP cannot do so because its protection scheme is provisioned extreme-to-extreme, underusing then the extra degree of international connectivity, which is limited by an intermediate physical bonds of size 2. In other words, the bond condition Lemma 3 on this case applies upon the worst-case (bonds of Figure 5.23), instead of bond after bond as in FRP-MORNDP model. Hence, the active/standby (ASP-MORNDP) model turns basically as inefficient as SDH is to protect international traffic, and cannot sustain the spread of performance achieved on first scenarios, which is attached to savings upon the national portion of international (Montevideo to Miami) connections.

Before finishing this section, it is mandatory to give an overall idea upon the improvements with respect to ANTEL's design. The truth is that we do not have enough details for a full comparative. What we do know is how the network of ANTEL is actually designed, and from it we can guess what we want.

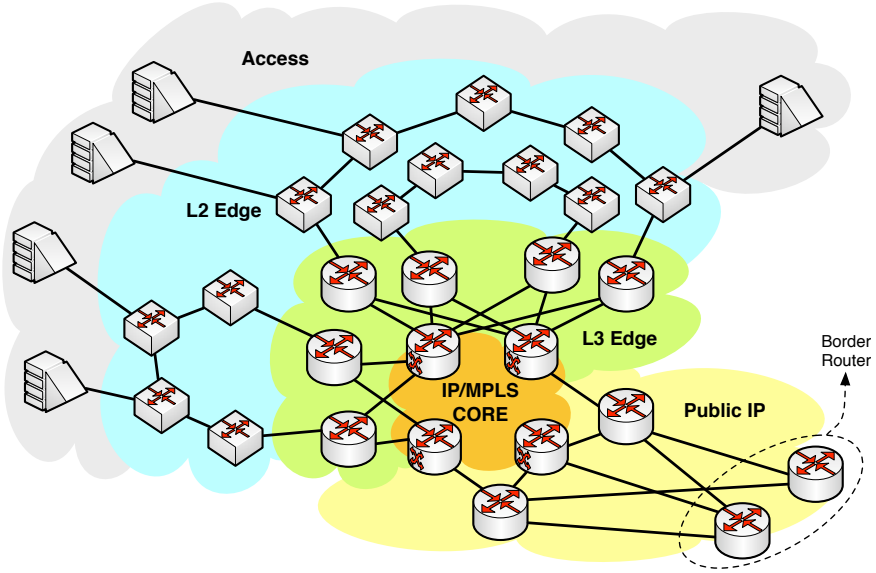


Figure 5.25: Scheme of ANTEL's convergent levels

The idea is more or less as follows: the topology of the logical network is structured in specialized sub-components of nested levels, in such a way that Internet traffic is forced to make its way towards the *core of the network*, and from it to the *public IP network*. Hence, traffic from *access nodes* goes to a first level of IP/MPLS nodes (L2 edge nodes), which are chained towards a higher hierarchy of IP/MPLS nodes (L3 edge). From these nodes, the traffic is routed according on the service,

i.e., Internet traffic goes towards BRASes and from them towards the *public IP network*. Conversely, private (VPN) traffic is routed by the L3 edge nodes, without the intervention of other components. All the traffic between L3 and IP nodes is routed through the *core*. The Figure 5.25 sketches this structure.

This design dispenses with traffic-engineering, because the logical topology itself is shaped to avoid it. This strategy is absolutely utilitarian in terms of design and operation. Guaranteeing physical independence between paths from one level to the following, and balancing/distributing customers to keep bottomed the traffic load over logical links, this arrangement of nodes should work fine. The coordination required between the groups that bring support to the logical and the physical networks is minimal. The cost of course is the over-engineering associated with this structure. Maximally, when we consider that the core is placed within Montevideo, when the constructions found by EA and GRASP, lead us to think that it should be placed between the metropolitan area and those points where international connections ingress into the country.

Anyhow, the scheme of two physically independent paths, and the existence of a separate public IP network, resembles us the odd indices instances of Table 5.11 for the ASP-MORNDP model, although without optimization at all. Thus, we are confident that the fusion of networks and the extensive usage of traffic engineering can reduce the total cost by a value far above the 30%.

5.3 Summary

This chapter detailed two real-world application cases, over which were applied the models and algorithms depicted in Chapter 3 and Chapter 4. These application cases respectively correspond to the most important academic and commercial networks of our country.

Both applications were driven by different objectives. The RAU case is spurred by a technological upgrade, i.e., the replacement of ATM by IP/MPLS technologies. Although when we started this project, the substitution was already in progress, the design criteria were still guided by former paradigms this worked helped to review. From them we remark: a logical hub and spoke topology, with a centralized scheme of connections towards other networks.

The application for ANTEL was also driven by optimization purposes, intending to assess the cost of some commercial or technological decisions. Additionally to primary objectives to optimize, we found many issues associated to the structure of the network inherited as a part of the legacy infrastructure. Just to mention two of them: an optical network intended to serve a traffic completely different (i.e. telephonic), a structure of components geographically misplaced (e.g. a metropolitan core). Because of the parallelisms amid evolution of different TELCOs, we are confident that this kind of situations should replicate to other ISPs.

Additionally to the numerical results, both applications were complemented with theoretical elements of analysis. Without them, many of the mayor improvements wouldn't have been found.

Quantitatively speaking, the potential improvements are outstanding. For the RAU's case they range from 61% to 84%. Furthermore, we have shown how the usage of updated and cheaper technologies, combined with an efficient network design, would allow SeCIU to construct a network 200 times faster than current with the same budget. We don't know for sure what they are for ANTEL's, but we are pretty sure that they are far above 30%.

Neither RAU nor ANTEL designs were concerned with traffic engineering details. Conversely, traffic engineering is totally integrated to ASP-MORNDP and FRP-MORNDP models. In fact, the only essential difference between both of them, relies precisely upon the strategy chosen to set the paths for IP/MPLS tunnels. The solutions found by the ASP flavor are directly realizable over industrial standards, although to become plenty practical, it would be useful to count with the support of some external network manager, to set-up and maintain the paths found by the metaheuristic. Conversely, after fixing bumping issues (minor changes according to our experience) the solutions found by FRP can be automatically reproduced, over the dynamic routing protocols that are actually working upon operative IP/MPLS networks. With an appropriate network design and a minimal of configurations, the traffic engineering capabilities of industrial routing protocols, can reach a performance pretty close to what our models can achieve.

Finally, the paths for lightpaths, which was a concern into our models, wouldn't be a key issue during these applications. In fact, on most cases the route built by the metaheuristic was close -when not identical-, to that a human designer would choose manually. The only exception is in the solution of ASP-MORNDP for scenario 06 on RAU's application, but this certainly explainable by the lack of stability of the EA for this instance, rather than on other causes.

Chapter 6

Conclusions

The first part of this work summarizes aspects of the evolution of communications services and technologies, relevant to understand the context of the thesis, upon which the rest of the work is founded. Telecommunications technologies have been evolving sustainedly along a century. During most of this time the telephone service leaded the way, progressively integrating more and more users. In parallel, the grade of these services achieves previously unknown levels, which comprised not only the quality of the voice but also an impressive availability. These extraordinary levels-of-service were sustained by the digitalization of the communications and the massive deployment of optical fiber. By the time when the telephone service ruled this market, the engineering was highly specialized over decoupled groups. The interaction amid those responsible of laying optical fiber, laying the cooper local-loop, deploying the TDM transport networks (SDH/SONET) and the TDM telephone exchange (PSTN switches), was minimal. Technology and design models were focused upon these segments and their concerns. As a consequence there was a lack of global optimization.

Only 30 years ago, the IT industry was basically monopolistic, the offer of services was stalled, and the future envisioned by a few TELCOs, hardware and software providers, was a world with a fistful of *mainframes* accessed from *dumb terminals*, which only needed low speed connections of 16Kbps. A few years later, the disruptive appearance of Internet shattered these plans to pieces. Today, residential access connections of 100Mbps are quite frequent, and a mere *smart phone* counts with a computing power and a set of available applications, far beyond of what legacy mainframes ever had. This sequence of rapid changes in paradigms pushed organizations through a stressing process of transformation. In the middle of it, several overlay networks and technologies emerged, many of them transitorily. Although organizations have been evolving to accompany these changes, some practices still look like placed a step behind of technological evolution. One of them is the lack of coordination among the design of different components. The primary goal of this work always was to aid counterparts to determine how to get the maximum benefits from an existing infrastructure, using state-of-the-art technologies in conjunction with a globally coordinated design. We agreed with our two partner organizations (RAU and ANTEL) to use two-layers models, with a logical IP/MPLS network deployed over a DWDM optical transport network.

Parts of this work required an almost encyclopedic review of technical documents, some of them related to the concrete application cases. After reviewing the

related literature we decided to develop new models. Many of the previous resilient multi-layer network models were inspired by TDM rather than IP/MPLS. Others are indeed aware of IP/MPLS capabilities, but their cost structures put the hardware of nodes and the connections between them, in the same order of importance, whereas in our applications the relative importance of the first one is negligible. In accordance with this, we decided to give to our models the highest degree of freedom to design the physical connections between nodes, since they form the cost.

The IP/MPLS technology allows a vast spectrum of applications, whose analysis constitutes a challenging task by itself. We developed two complementary models to reflect extreme practical implementations for IP/MPLS protection mechanisms. The first of them (ASP-MORNDP) resembles legacy SDH/SONET protection schemes, although it provides logical protection from point to point, instead of along a sequence of locally protected rings. Its results are directly implementable over actual operative networks. The second one (FRP-MORNDP) relaxes the problem allowing a much wider space of configurations. As a drawback, these constructions require further analysis prior to validate them as *practically feasible*. For the results for our instances however, these changes when necessary were minimal.

We proved that problems arising from both models are computationally hard to solve (NP-Hard) in general. Complementarily, we examined theoretical properties of them, many of which were used afterwards during the construction of algorithms. From early stages of this research, we decided to use metaheuristics to find good quality solutions to real-world instances. Although several were surveyed (GRASP, EA, TS, VNS among others), into this document we limited ourselves to describe concisely the two branches that benchmarked the best, that is: EA for ASP-MORNDP and GRASP for FRP-MORNDP. These algorithms constituted an essential tool to find solutions, but they were not the only ones.

Together with the metaheuristics and their numerical contributions, both projects were blended with abstract elements of analysis, without whom many results had been bypassed or had passed unnoticed. Some of these elements were purely theoretical; others integrated details about the technology or the casuistic on networks evolution. Such concerns were translated into instances of the problems, feeding this way our algorithms, and up from the obtained results, newer instances and concerns were isolated, iteratively, until getting to the definite results. In this document, we restricted the description to comment on most significative instances.

In our opinion, the most important product of this work lies on the real world applications and the analysis of the main elements for their improvements. Some of these results are supposed to have a strong impact. After adopting changes in the network architecture and optimizing the usage of resources, we conclude that the original design of RAU2 can be improved in around 70%. For ANTEL's case this number isn't known for sure, but we are certain that it is well above 30%. These spreads are intrinsic to inherited practices rather than to an inappropriate design. Weaknesses on RAU's case were caused by a *hub and spoke* logical configuration, and

the existence of a centralized point for traffic interchange; both typical for overlay networks upon legacy ATM technology. Conversely, on ANTEL's case the main improvements arise from abandoning a structure of highly specialized components, a change that allowed better workarounds to structural deficiencies of the legacy optical fiber network. This network, carefully designed to attend the telephone service, doesn't fit well with the now dominant traffic of Internet. The substantial changes in traffic requirements compel to revise the placement of backbone, core, as well as the points of interchange with other peers.

Additionally, to fully exploit the advantages of newer technologies, it is mandatory an update on the design practices and their related models. Legacy models, which focused into a single layer, are no longer appropriate, neither is the disaggregation of network design into loosely coordinated layers. Although there are other multi-layer models, we are confident that this of ours surpasses existing ones for these applications. In the first place, this comes because of the size of the instances tackled down, which in some cases reached 70 nodes. Putting focus onto the really important sources of costs is another specific attribute. Compared with the cost of connecting logical nodes among them, the cost of the logical nodes themselves was below 10% on all instances. In a final place, the most important results were gotten from the FRP-MORNDP model, due to the high degree of freedom it has to build logical paths. Most other models just preset the paths for the tunnels, loosing this finesse to build solutions of good quality.

Conversely, in the light of the outcome, we agree with existing models that pre-computing lightpaths is a fair enough trade-off, which doesn't compromise the quality of the results, whereas indeed contributes to improve the simplicity of the models and the performance of the algorithms. This is perhaps the main change we would introduce into future versions of MORNDP models. We also consider further improvements upon the algorithms. One of them is using intra-interations routes caches during the local search phase of GRASP. Furthermore, the existence of a pre-computed pool of lightpaths with revealed/fixed paths probably would help to implement inter-iterations caches and path-relinking for the GRASP algorithm, as well as easier operators for the EA one, avoiding two levels of nested constructions. These changes together with the application of the necessary condition (bonds condition) to speed-up the algorithms, are stated as *future work*.

It is worth mentioning that solutions found with both models, are realizable over operational networks. Moreover, when we consider the constructions of FRP-MORNDP model, we conclude that by designing a network as the algorithm suggests to do, possibly with some minor changes (e.g. setting different costs on some links), we can be sure that the very same traffic engineering protocols that are actually running upon the IP/MPLS networks, can manage themselves to recreate the logical paths. In other words, *the implementation of pseudo-optimal actual networks is technologically at hand*, what we believe constitutes another important result for the technical community.

Chapter 7

Appendix

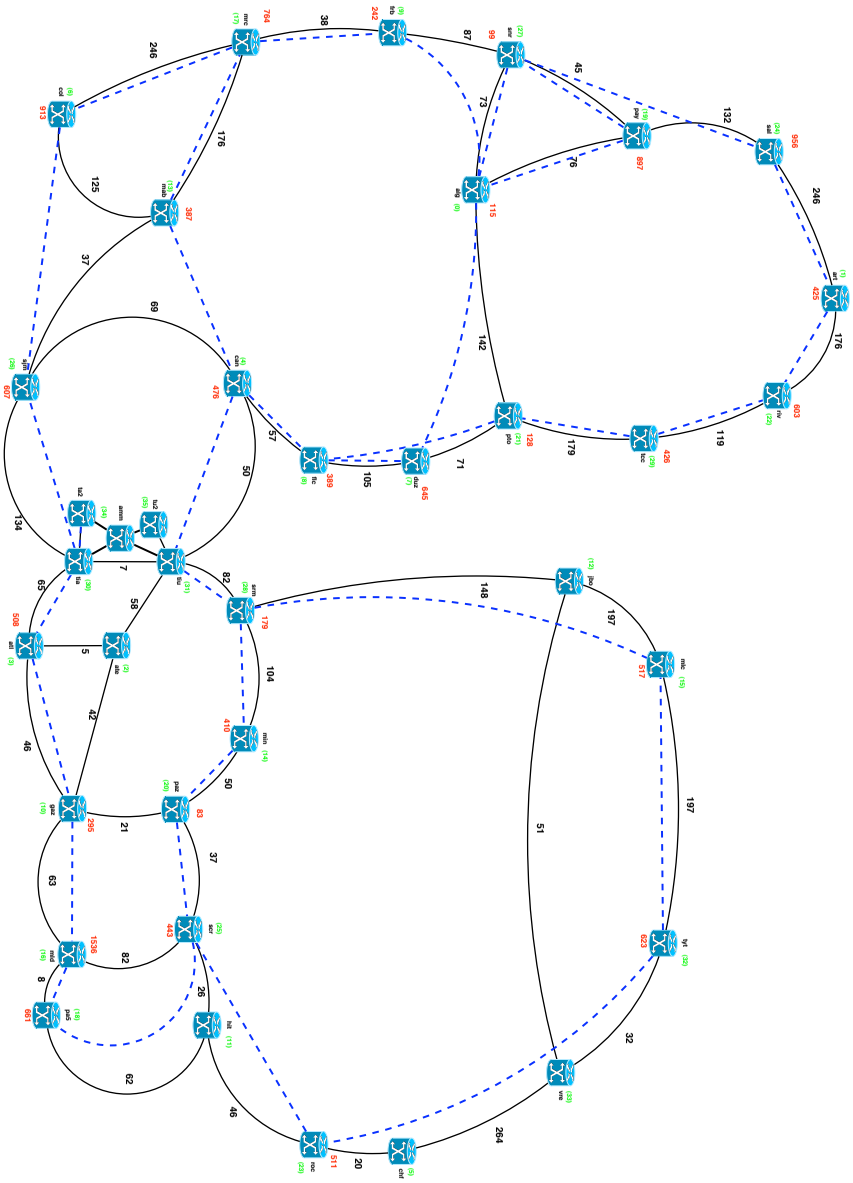


Figure 7.1: Solution found for instances 01 of ANTEL

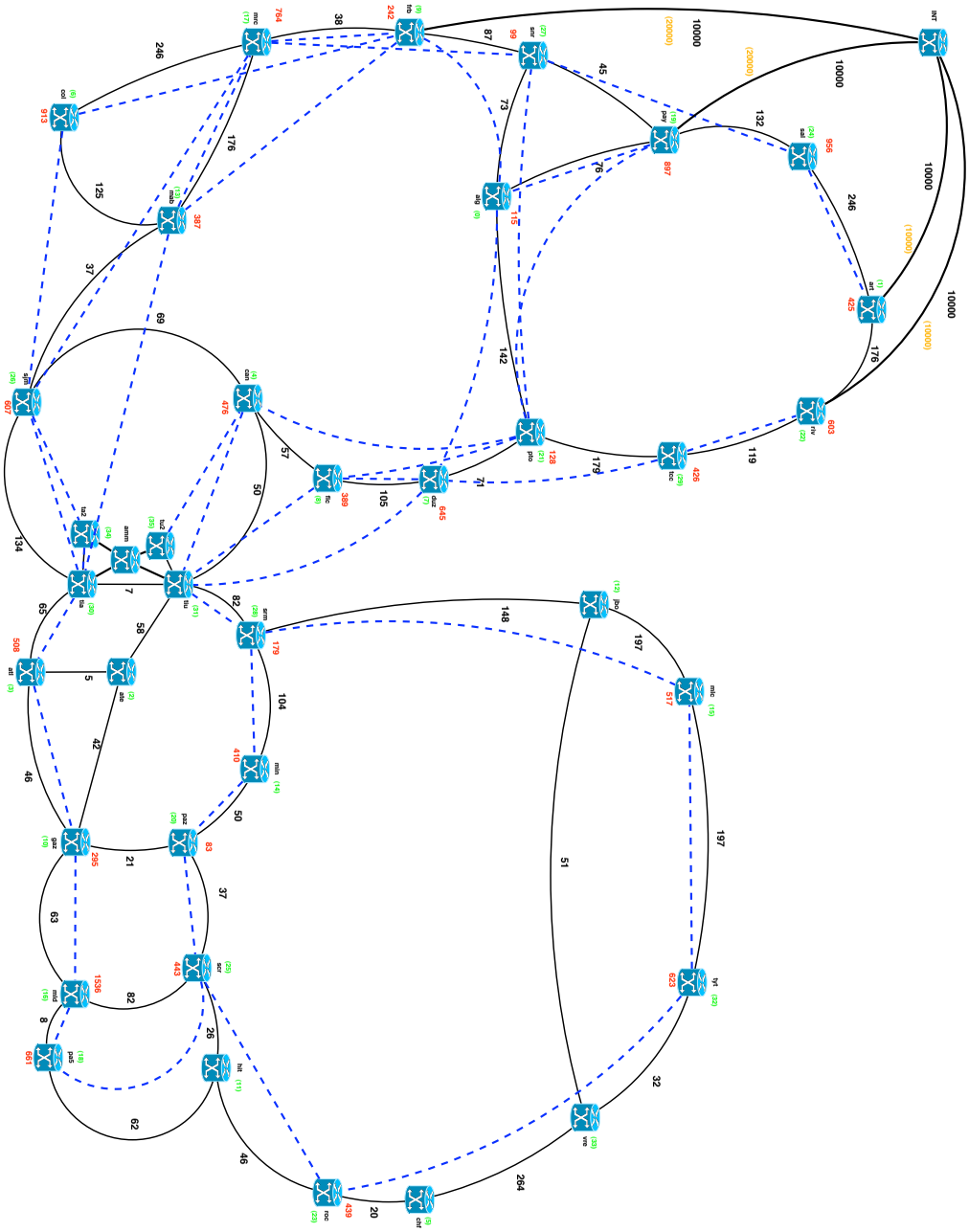


Figure 7.2: Solution found for instances 02 of ANTEL

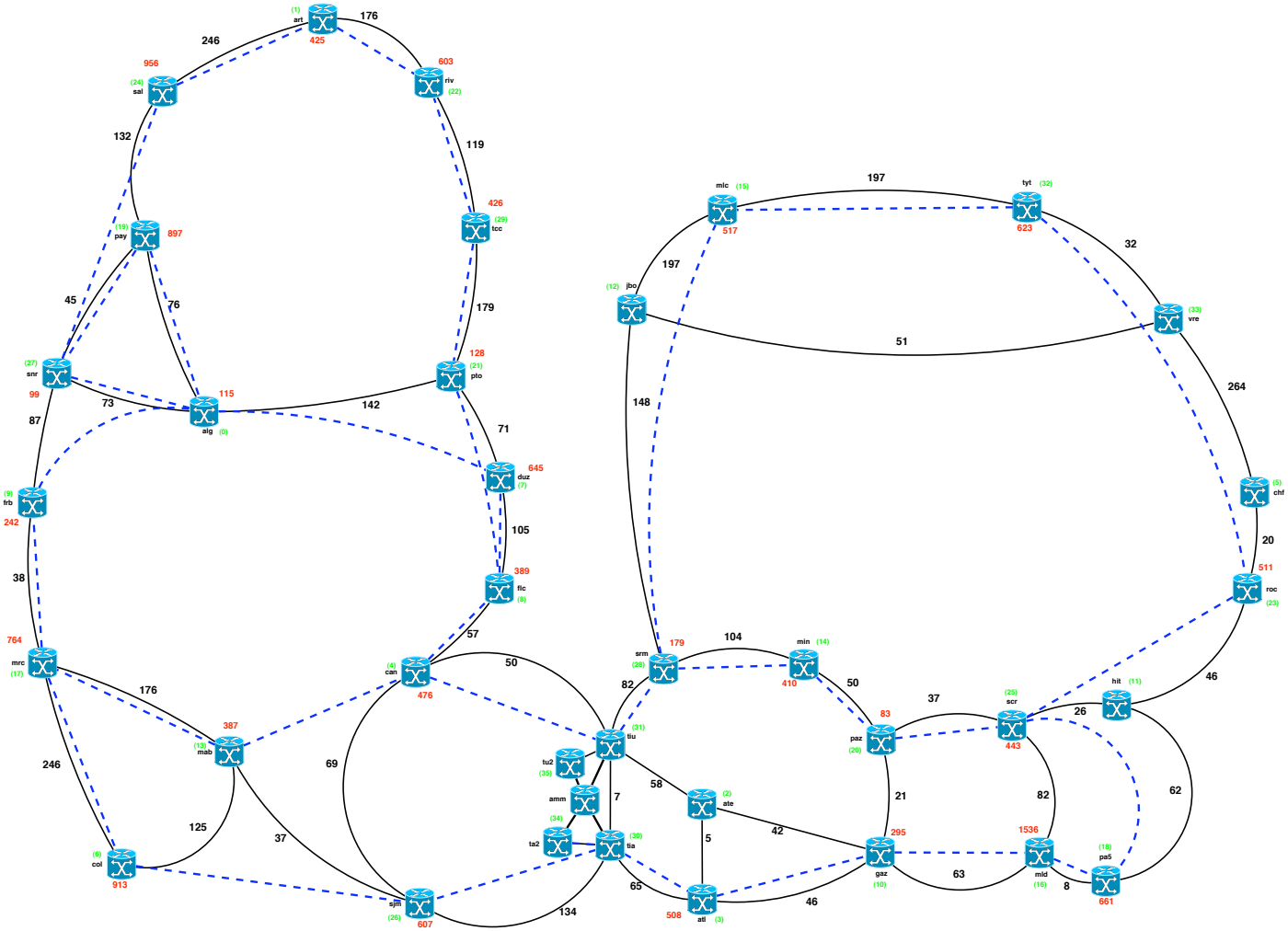


Figure 7.3: Solution found for instances 03 of ANTEL

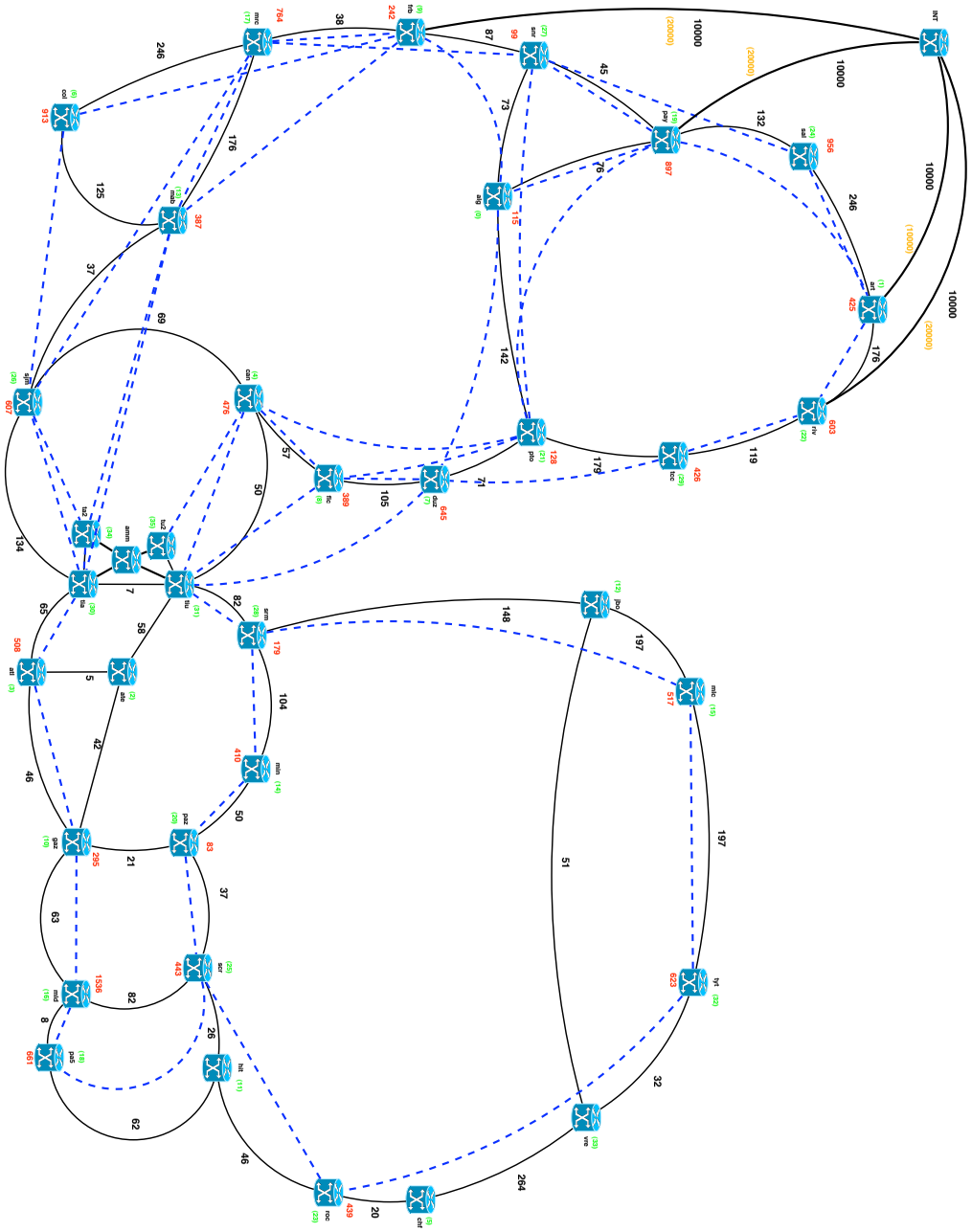


Figure 7.4: Solution found for instances 04 of ANTEL

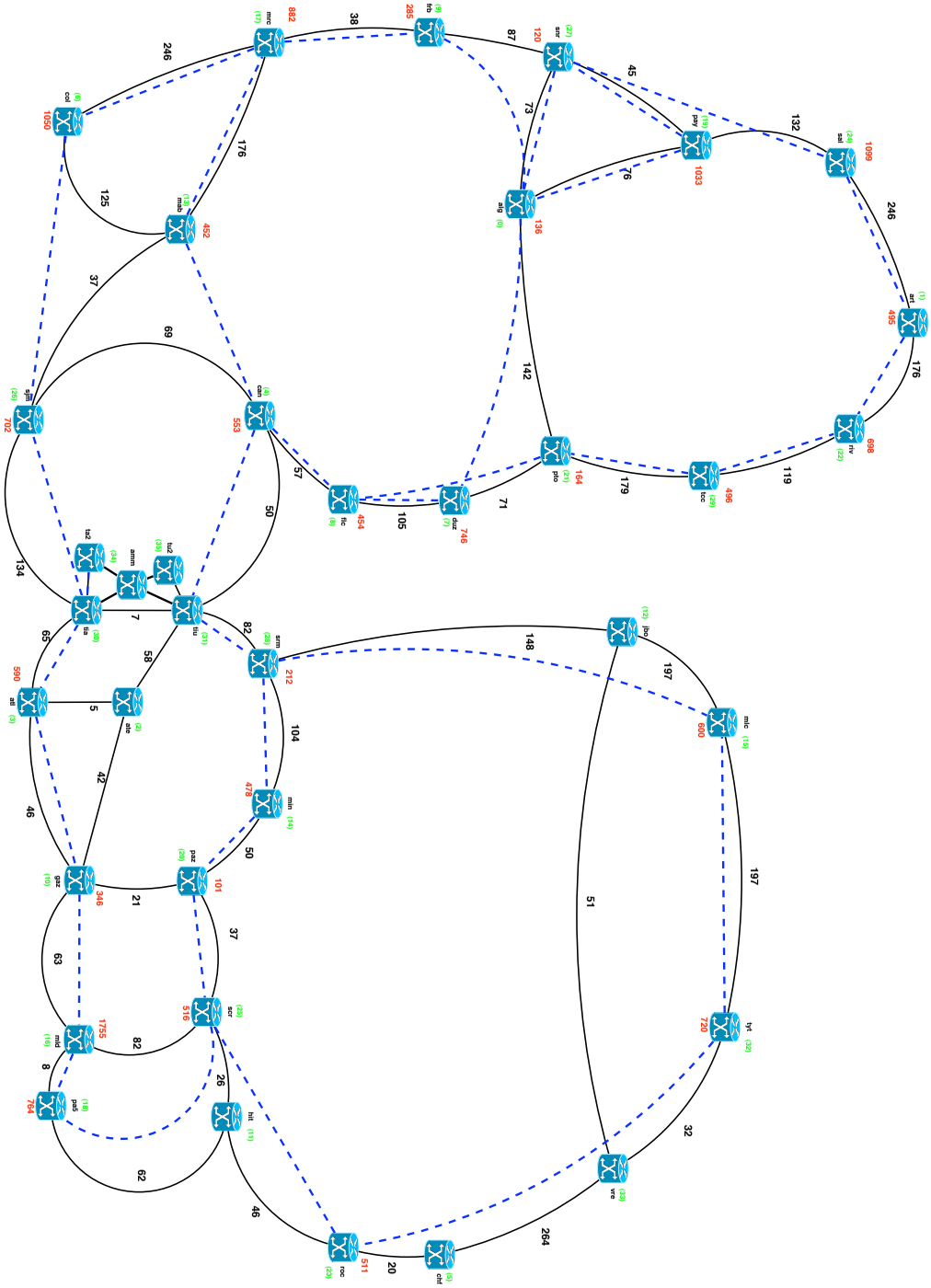


Figure 7.5: Solution found for instances 05 of ANTEL

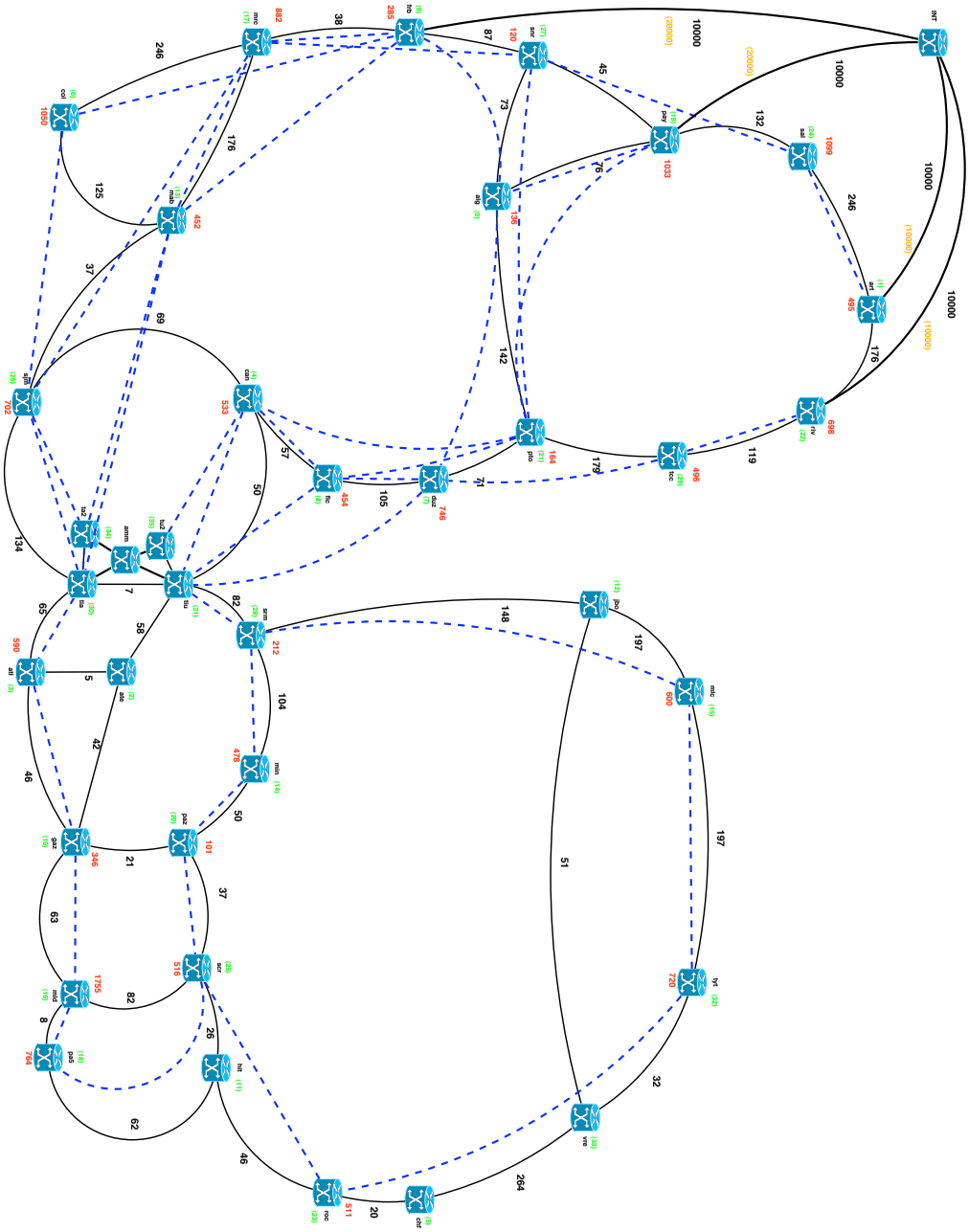


Figure 7.6: Solution found for instances 06 of ANTEL

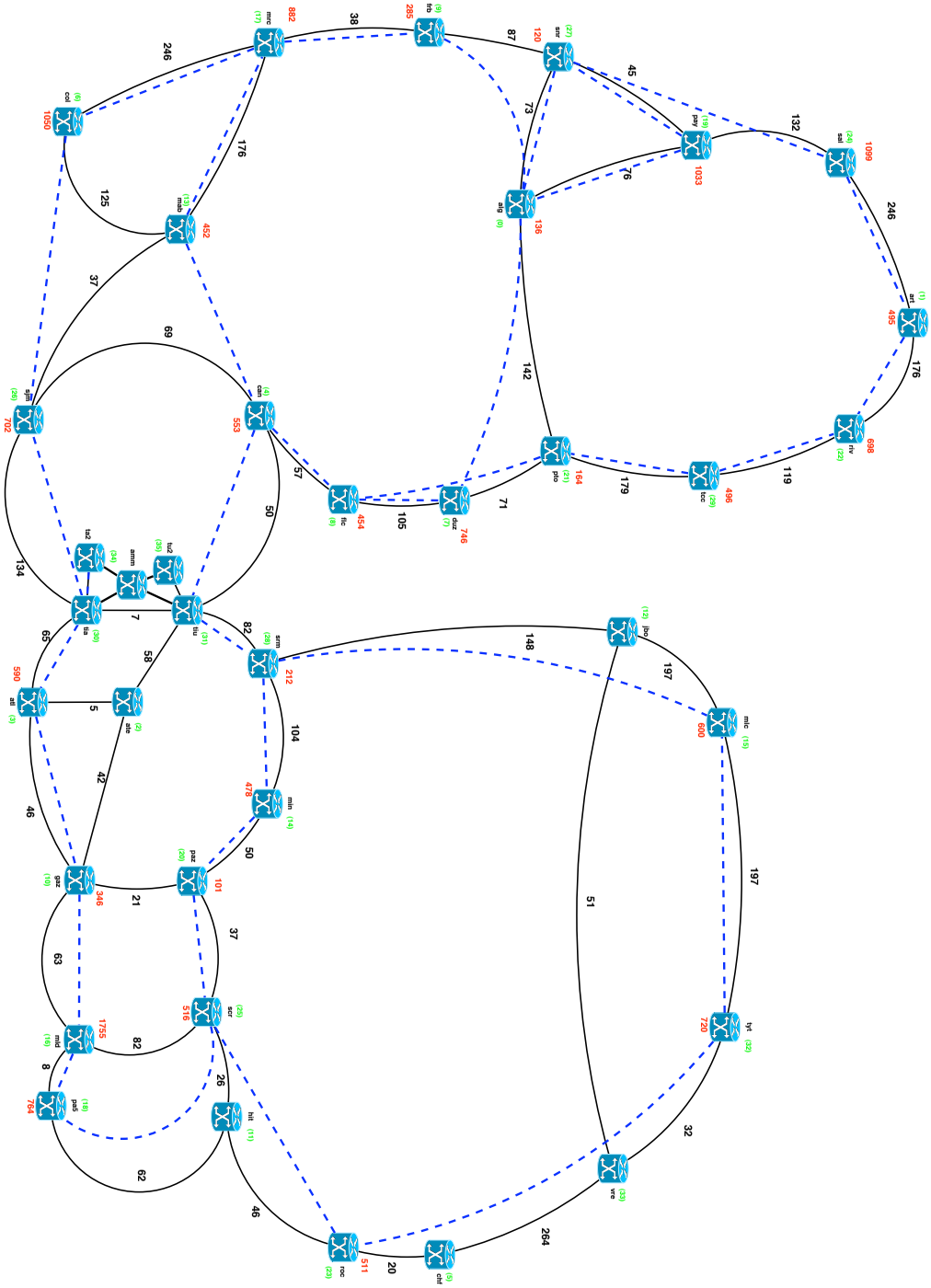


Figure 7.7: Solution found for instances 07 of ANTEL

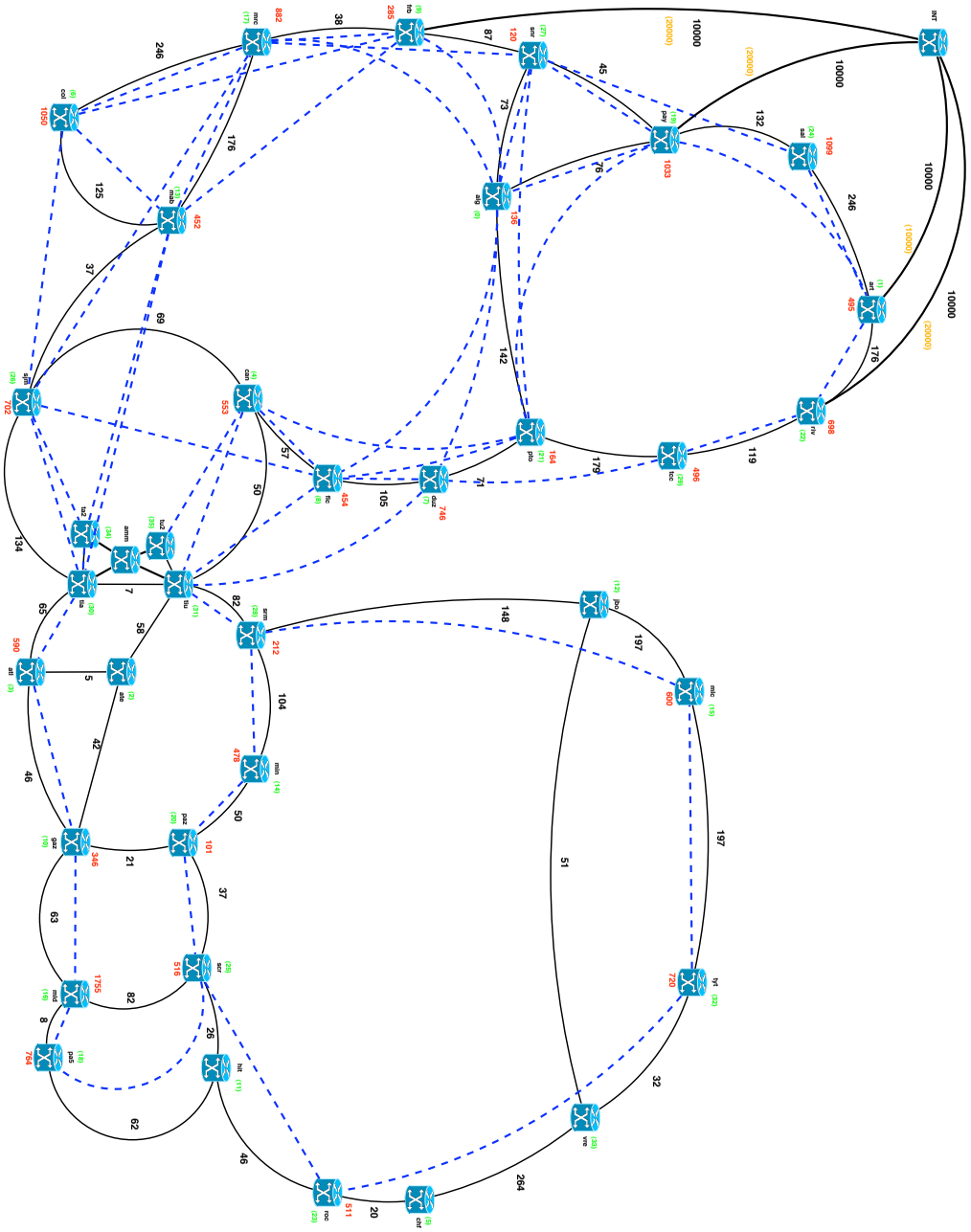


Figure 7.8: Solution found for instances 08 of ANTEL

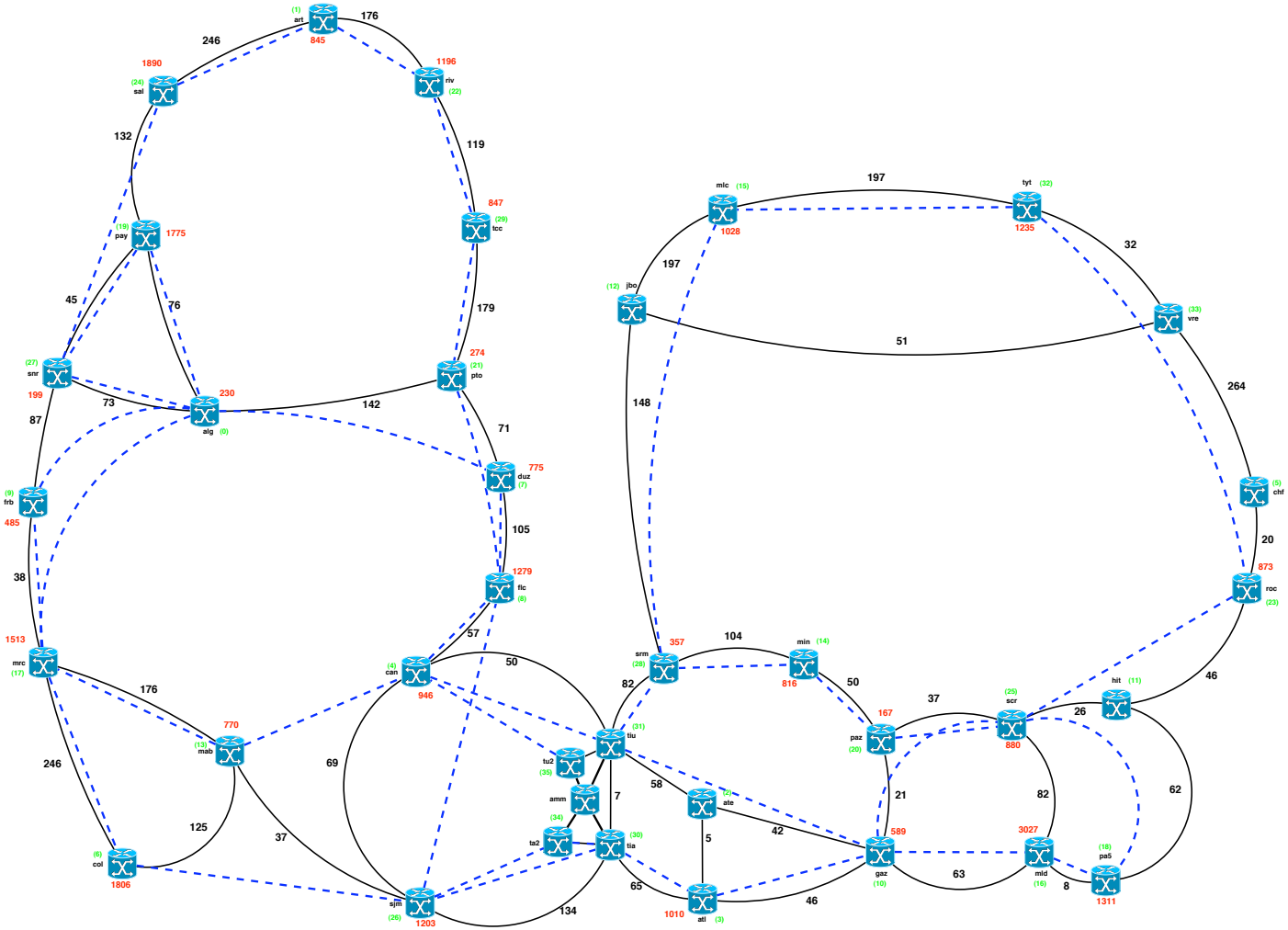


Figure 7.9: Solution found for instances 09 of ANTEL

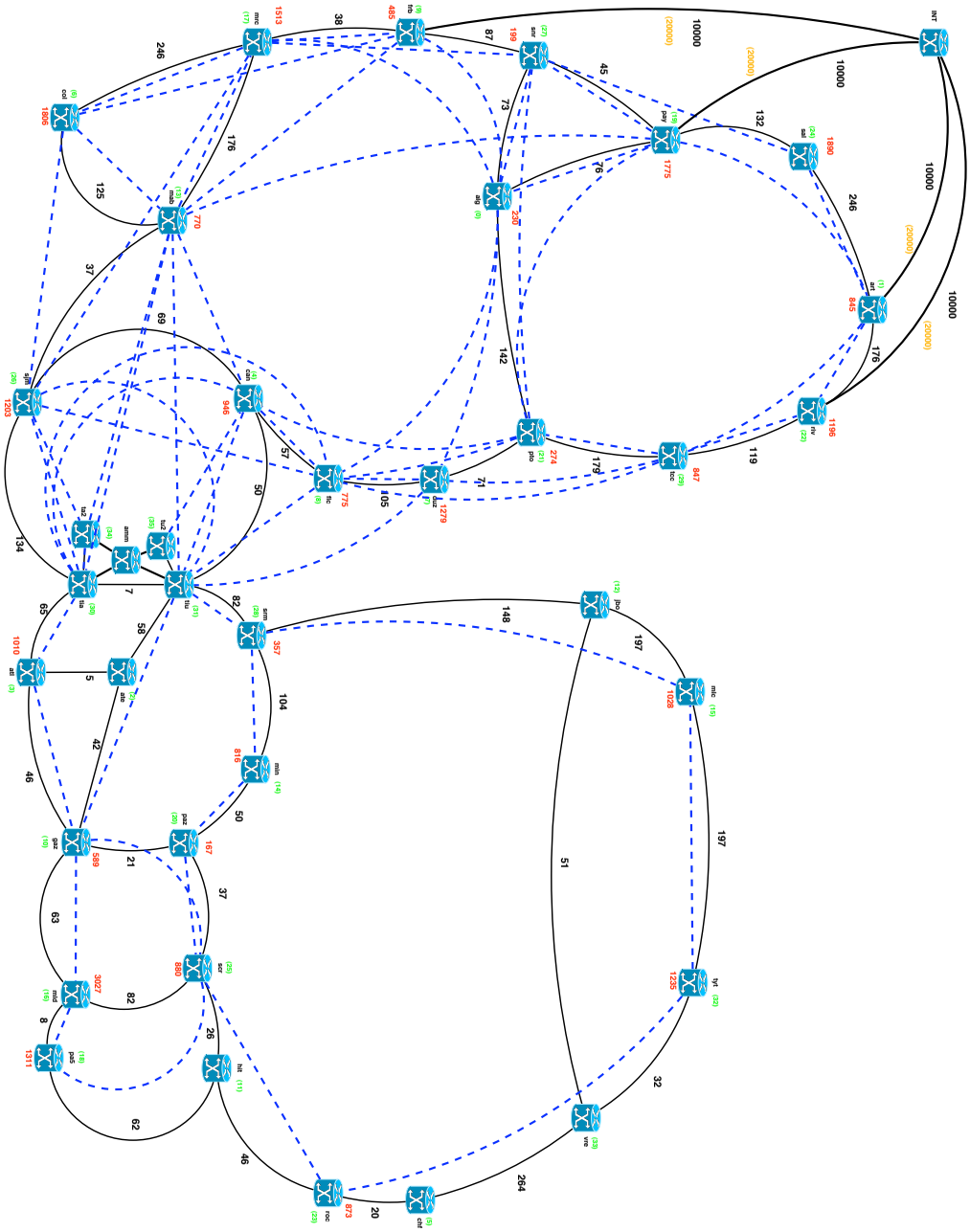


Figure 7.10: Solution found for instances 10 of ANTEL

Bibliography

[Agrawal 1995] Ajit Agrawal, Philip Klein and R. Ravi. *When trees collide: An approximation algorithm for the generalized Steiner problem on networks.* SIAM Journal on Computing, vol. 24, no. 3, pages 440–456, 1995. 23

[Alba 2002] E. Alba and M. Tomassini. *Parallelism and evolutionary algorithms.* IEEE Transactions on Evolutionary Computation, vol. 6, no. 5, pages 443–462, 2002. 50

[Alba 2006] E. Alba, F. Almeida, M. Blesa, C. Cotta, M. Díaz, I. Dorta, J. Gabarró, C. León, G. Luque, J. Petit, C. Rodríguez, A. Rojas and F. Xhafa. *Efficient parallel LAN/WAN algorithms for optimization. The MALLBA project.* Parallel Computing, vol. 32, no. 5-6, pages 415 – 440, 2006. 109

[Alba 2013] E. Alba, G. Luque and S. Nesmachnow. *Parallel Metaheuristics: Recent Advances and New Trends.* International Transactions in Operational Research, vol. 20, no. 1, pages 1–48, 2013. 50

[Alevras 1996] Dimitris Alevras, Martin Grötschel and Roland Wessäly. *A network dimensioning tool.* In Preprint SC 96-49, Konrad-Zuse-Zentrum fur Informationstechnik, 1996. 24

[Bäck 1997] T. Bäck, D. Fogel and Z. Michalewicz, editeurs. Handbook of evolutionary computation. Oxford University Press, 1997. 49

[Baïou 1996] Mourad Baïou. *Le problème du sous-graphe Steiner 2-arête connexe: approache polyédrale.* PhD thesis, Université de Rennes 1, 1996. 21

[Balakrishnan 1994a] Anantaram Balakrishnan, Thomas L. Magnanti and Prakash Mirchandani. *A Dual-based Algorithm for Multi-level Network Design.* Manage. Sci., vol. 40, no. 5, pages 567–581, May 1994. 30

[Balakrishnan 1994b] Anantaram Balakrishnan, Thomas L. Magnanti and Prakash Mirchandani. *Modeling and Heuristic Worst-case Performance Analysis of the Two-level Network Design Problem.* Manage. Sci., vol. 40, no. 7, pages 846–867, July 1994. 30

[Balakrishnan 1998] Anantaram Balakrishnan, Thomas L. Magnanti and Prakash Mirchandani. *Designing Hierarchical Survivable Networks.* Oper. Res., vol. 46, no. 1, pages 116–136, January 1998. 30

[Bienstock 1990] Daniel Bienstock, Ernest F. Brickell and Clyde L. Monma. *On the Structure of Minimum-weight K-connected Spanning Networks.* SIAM J. Discret. Math., vol. 3, no. 3, pages 320–329, May 1990. 21

[Canale 2009] E. Canale and F. Robledo. *Designing Backbone Networks using the Generalized Steiner Problem.* In IEEE Computer Society, editeur, Proceedings of 7th IEEE International Workshop on the Design of Reliable Communication Networks, 2009. 23

[Córez 2010] A. Córez. Multi-overlay network planning by applying a variable neighbourhood search approach. Master's thesis, Engineering Faculty, UdelaR, 2010. 108, 130

[Cruz 2003] F.R.B. Cruz, G.R. Mateus and J. MacGregor Smith. *A Branch-and-Bound Algorithm to Solve a Multi-level Network Optimization Problem.* Journal of Mathematical Modelling and Algorithms, vol. 2, no. 1, pages 37–56, 2003. 30

[Davis 1991] L. Davis. Handbook of genetic algorithms. van Nostrand Reinhold, New York, 1991. 49

[de Aragão 2001] M. Poggi de Aragão, C.C. Ribeiro, E. Uchoa and R.F. Werneck. *Hybrid local search for the Steiner problem in graphs.* In Extended Abstracts of the 4th Metaheuristics International Conference (MIC 2001), pages 429–433, 2001. 51, 55

[Diestel 2012] Reinhard Diestel. Graph theory. Springer-Verlag, Heidelberg, 2012. 44

[Eswaran 1976] Kapali P. Eswaran and R. Endre Tarjan. *Augmentation Problems.* Society for Industrial and Applied Mathematics (SIAM), vol. 5, no. 4, pages 653–665, 1976. 86

[Feo 1989] T.A. Feo and M.G.C. Resende. *A probabilistic heuristic for a computionally difficult set covering problem.* Operations Research Letters, vol. 8, pages 67–71, 1989. 55

[Feo 1995] T.A. Feo and M.G.C. Resende. *Greedy Randomized Adaptive Search Procedures.* Journal of Global Optimization, vol. 6, pages 109–133, 1995. 51, 55

[Festa 2004] P. Festa and M.G.C. Resende. *An annotated bibliography of GRASP.* Rapport technique TD-5WYSEW, AT&T Labs Research, 2004. 51

[Fouilhoux 2011] P. Fouilhoux, O. Ekin Karasan, R. Mahjoub, O. Özkök and H. Yaman. *Survivability in hierarchical telecommunications networks.* Networks, vol. 59, no. 1, pages 37–58, November 2011. 30

[Garey 1979] M.R. Garey and D.S. Johnson. Computers and intractability: A guide to the theory of np-completeness. New York: W.H. Freeman, 1979. 48, 129

[Glover 1996] Fred Glover. *Tabu search and adaptive memory programing – Advances, applications and challenges.* In Interfaces in Computer Science and Operations Research, pages 1–75. Kluwer, 1996. 130

[Gough 2004] Clare Gough. CCNP BSCI exam certification guide. Cisco Press, 2004. 74

[hark Chung 1992] Sung hark Chung, Young soo Myung and Dong wan Tcha. *Optimal design of a distributed network with two-level hierarchical structure.* European Journal of Operational Research, vol. 62, no. 1, pages 105–115, 1992. 30

[Hayes 2002] Brian Hayes. *The Easiest Hard Problem.* American Scientist, vol. 90, no. 2, page 113, March-April 2002. 89, 121

[Hucaby 2004] David Hucaby. CCNP BCMSN exam certification guide. Cisco Press, 2004. 74

[Kan 2009] D. Kan, A. Narula-Tam and E. Modiano. *Lightpath routing and capacity assignment for survivable IP-over-WDM networks.* Design of Reliable Communication Networks (DRCN 2009), pages 37–44, October 2009. 29

[Kerivin 2005] Hervé Kerivin and A. Ridha Mahjoub. *Design of survivable networks: A survey.* Networks, vol. 46, no. 1, pages 1–21, June 2005. 22, 23

[Koster 2008] Arie Koster and Sebastian Orlowski. *Single-layer Cuts for Multi-layer Network Design Problems.* In American University University of Maryland, editeur, Selected proceedings of the 9th INFORMS Telecommunications Conference, volume 44 of *Operations Research/Computer Science Interfaces*, pages 1–23. Springer Science+Business Media, LLC, 2008. 25

[Kubilinskas 2005a] Eligijus Kubilinskas and Michal Pioro. *An IP/MPLS over WDM network design problem.* Proceedings INOC, page 6, March 2005. 27

[Kubilinskas 2005b] Eligijus Kubilinskas and Michal Pioro. *Two design problems for the IP/MPLS over WDM networks.* IEEE Xplore, 2005. 27, 28

[Martins 2000] S.L. Martins, M.G.C. Resende, C.C. Ribeiro and P.M. Pardalos. *A parallel GRASP for the Steiner tree problem in graphs using a hybrid local search strategy.* Journal of Global Optimization, vol. 17, pages 267–283, 2000. 51, 55

[Mavridou 1998] T. Mavridou, P.M. Pardalos, L.S. Pitsoulis and M.G.C. Resende. *A GRASP for the bicuadratic assignment problem.* European Journal of Operational Research, vol. 105, pages 613–621, 1998. 51

[Monma 1990] ClydeL. Monma, BethSpellman Munson and WilliamR Pulleyblank. *Minimum-weight two-connected spanning networks.* Mathematical Programming, vol. 46, no. 1-3, pages 153–171, 1990. 21

[Oellrich 2008] Martin Oellrich. *Minimum Cost Disjoint Paths under Arc Dependences. Algorithms for Practice.* PhD thesis, Technische Universität Berlin, 2008. 26, 118

[Okamura 1981] Haruko Okamura and P.D. Seymour. *Multicommodity flows in planar graphs.* Journal of Combinatorial Theory, vol. 31, no. 1, pages 75–81, August 1981. 23, 24, 97

[Orlowski 2006] S. Orlowski, Arie M. C. A. Koster, C. Raack and R. Wessäly. *Two-Layer Network Design by Branch-and-Cut featuring MIP-based Heuristics*, 2006. 25

[Osborne 2002] Eric Osborne and Ajay Simha. Traffic engineering with MPLS. Cisco Press, 2002. 74

[Pardalos 1999] P.M. Pardalos, T. Qian and M.G.C. Resende. *A Greedy Randomized Adaptive Search Procedure for the feedback vertex set problem.* Journal of Combinatorial Optimization, vol. 2, pages 399–412, 1999. 51

[Parodi 2011] C. Parodi. Integer optimization applied to the design of robust minimum cost multi-layer networks. Master's thesis, Engineering Faculty, UdelaR, 2011. 108

[Pepelnjak 2001] Ivan Pepelnjak and Jim Guichard. MPLS and VPN architectures. Cisco Press, 2001. 74

[Pepelnjak 2003] Ivan Pepelnjak and Jim Guichard. MPLS and VPN architectures II. Cisco Press, 2003. 74

[Resende 1998a] M.G.C. Resende. *Computing approximate solutions of the maximum covering problem using GRASP.* Journal of Heuristics, vol. 4, pages 161–171, 1998. 51

[Resende 1998b] M.G.C. Resende and C.C. Ribeiro. *A GRASP for graph planarization.* Journal of Heuristics, vol. 4, pages 161–171, 1998. 51

[Resende 2003] M.G.C. Resende and C.C. Ribeiro. *Greedy Randomized Adaptive Search Procedures.* Rapport technique, AT&T Labs Research, 2003. 51, 54, 55, 130

[Ribeiro 2002] C.C. Ribeiro, E. Uchoa and R.F. Werneck. *A hybrid GRASP with perturbations for the Steiner problem in graphs.* INFORMS Journal on Computing, vol. 14, no. 3, pages 228–246, 2002. 51, 55

[Risso 2012] Claudio Risso, Sergio Nesmachnow and Franco Robledo. *A Parallel Evolutionary Algorithm for Multilayered Robust Network Design.* In 3PGCIC'12 - 7th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pages 291–296, Victoria, BC, Canada, November 2012. 32

[Risso 2013a] Claudio Risso, Eduardo Canale, Franco Robledo and Gerardo Rubino. *Using metaheuristics for planning resilient and cost-effective multilayer networks.* In RNDM'13 - 5th International Workshop on Reliable Networks Design and Modeling (RNDM'13), pages 90–96, Almaty, Kazakhstan, September 2013. 32

[Risso 2013b] Claudio Risso and Franco Robledo. *Using GRASP for designing a layered network: a real IP/MPLS over DWDM application case.* International Journal of Metaheuristics, vol. 2, no. 4, pages 392–414, December 2013. 32

[Risso 2013c] Claudio Risso, Franco Robledo and Pablo Sartor. *Optimal Design of a Multi-Layer Network. An IP/MPLS Over DWDM Application Case.* Current Developments in Optical Fiber Technology, pages 3–20, June 2013. 32

[Risso 2014] Claudio Risso, Eduardo Canale and Franco Robledo. *Optimal design of an IP/MPLS over DWDM network.* Pesquisa Operacional - Special Issue from CLAIO/SBPO 2012, 2014. 32

[Robledo 2005] Franco Robledo. *GRASP heuristics for Wide Area Network design.* PhD thesis, Université de Rennes 1, 2005. 107

[Rosseti 2001] I. Rosseti, M. Poggi de Aragão, C.C. Ribeiro, E. Uchoa and R.F. Werneck. *New benchmark instances for the Steiner Problem in Graphs.* In Extended Abstracts of the 4th Metaheuristics International Conference (MIC 2001), pages 557–561, 2001. 51

[Ruiz 2011] M. Ruiz and O. Pedrola. *Survivable IP/MPLS-Over-WSON Multilayer Network Optimization.* Optical Communications and Networking, vol. 3, no. 8, pages 629–640, August 2011. 28

[Stoer 1993] Mechthild Stoer. Design of survivable networks. Lecture Notes in Mathematics. Springer-Verlag, February 1993. 22, 23

[West 1995] Donald B. West. Introduction to graph theory. Prentice Hall Professional Technical Reference, 1995. 44

[Winter 1986] Pawel Winter. *Generalized Steiner Problem in Series-parallel Networks.* J. Algorithms, vol. 7, no. 4, pages 549–566, December 1986. 23

[Winter 1987] P. Winter. *Steiner Problem in Networks: A Survey.* Netw., vol. 17, no. 2, pages 129–167, April 1987. 23

[Zhang 2013] Xiaoning Zhang, Kun Li and Lin Bian. *Towards the maximum resource sharing degree for survivable IP/MPLS over WDM mesh networks.* Optical Switching and Networking, no. 0, 2013. 28