Optimization of Traffic Flows in Multiservice Telecommunication Networks

Università degli Studi di Cagliari
Dottorato di Ricerca in Ingegneria Industriale

Davide Cherubini

Department of Electrical and Electronic Engineering

University of Cagliari

Doctor of Philosophy - SSd: ING-IND/31

01/03/2007

I would like to dedicate this thesis to my beloved parents...

Acknowledgements

This dissertation would not be possible without the guidance, encouragement and support from my supervisor Prof. Alessandra Fanni. The extensive discussion with her have helped me stay on the right track toward finishing this work.
I would also like to thank Prof. Antonio Frangioni for his ideas of finding the linear structure of the problems at hand. Thanks also for joining me in Paris last year!
A special thank goes to Giorgio Lembo, who carried me in the amazing real world of Internet technologies, for his invaluable time and for his friendship.
To Prof. Paola Zuddas who convinced me that a good method to solve a problem should also be a simple one, and for her care and love which I will never forget.
I would also like to thank Prof. Abdel Lisser and the whole staff of the Laboratoire de Recherche en Informatique of the Université Paris Sud who hosted me in Orsay last year.
To Francesco Pau and Ruben Cardella who helped me to setup and test ICELab in the Tiscali's headquarter.
To Prof. Maria Grazia Scutellà for her support in indicating interesting papers and for her time during my short visits in Pisa.
To Dr. Cristina Perra who introduced me to the telecommunication network optimization problem.
To my colleague (brother-in-"study") Cristian who shared with me this challenge!!
I have been also very lucky to work with an incredible group of students and researchers that I am honored to call my friends, including

Augusto, Barbara, Pietro, Mariangela, Francesca, Giuliana, Giuditta, Katiuscia, Manuela, Rita, Sara, Cristiana, and Daria.

Finally, my dissertation research has been financially granted by Tiscali and I would like to thank Mr. Paolo Susnik who believed in the significance of my work.

Abstract

This dissertation investigates routing optimization in IP telecommunication networks, under normal working conditions as well as under failure conditions.

The main objectives of the present optimization procedure are the minimization of the maximum link utilization in the network and to provide a configuration that guarantees a 100% survivability degree. Traditionally two different steps are used to achieve this goal. The first one aims to solve the well known “General Routing Problem (GRP)” [44] in order to find the optimal routing network configuration and, successively, a set of “optimal” backup paths is found in order to guarantee network survivability. Furthermore, traditional survivable techniques assume that the planning tasks are performed in a network control center while restoration schemes are implemented distributively in network nodes.

In this dissertation innovative linear programming models are presented that, making use of the Multi Protocol Label Switching (MPLS) techniques and IS-IS/OSPF IP routing protocol, melt routing and survivability requirements. The models are extremely flexible, thus it is possible to improve the objective function in order to fit itself to newer applications and/or traffic typologies.

The models presented in this dissertation help network service providers to optimize their network resources and to guarantee connectivity in case of failure, while still be able to offer a good quality of service.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

In the last few years, the Internet has exponentially expanded to a worldwide network connecting several millions of users.
The explosion of peer-to-peer (P2P) file sharing had a significant impact on available bandwidth of Internet Service Providers (ISP) networks. In fact, P2P applications provides a highly and cheaper accessible way for downloading or sharing multimedia contents. The resulting bandwidth congestion causes loss of performance and possible failure situations. Such events might be technically and economically harmful.

In the last decade, new techniques leading to find a near-optimal traffic routing scenario have been developed. Traffic Engineering (TE) enables ISPs to route network traffic in such a way that they can offer the best service to their users in terms of throughput and delay, moving traffic from congested links to less loaded areas of the network (load balancing). Subsequently, the design of survivable mesh based telecommunication networks has received considerable attention in recent years.

Network survivability techniques have been developed to guarantee seamless communication services in the face of network failures. Traditionally, two techniques have been proposed to make a network survivable, namely:

1. network design and capacity allocation

2. traffic management and restoration

Network design and capacity allocation try to mitigate system level failures, by designing the topology and determining the capacity of links in a backbone network so that the network can carry the projected traffic demand even if any one link is lost due to a failure.

Traffic management and restoration seeks to distribute the network load such that a failure has minimum impact when it occurs and such that flows affected by the failure are restored. Several previous works have been presented in literature, mainly focused on spare capacity allocation (SCA) [53] and on optimization of traffic flows with shortest path routing algorithm (such as OSPF and IS-IS) [44].

While the former might not be able to respond to a failure quickly, the latter obtains a suboptimal solution of the General Routing Problem only under normal working condition.

Recently, innovative approaches have been presented such as the use of meta-heuristic algorithms in order to solve the flow distribution problem under single failure condition in an IS-IS network.

The integration of a broken link situation in a single optimization model, as well as finding the optimal flow distribution in both normal working and failure condition, is a challenging task and it is addressed in this dissertation.

1.2 Objectives of the dissertation

The objective of this thesis is to study new optimization methods in order to minimize the congestion effects in a telecommunication network.

Specifically, the techniques are based on MPLS Traffic Engineering enhanced capabilities, such as explicit routing that permits a finer distribution of traffic flows.

Furthermore, an additional condition has been introduced in the models, in order to find the best network parameters configuration that guarantees the congestion avoidance, also in case of single link failure.

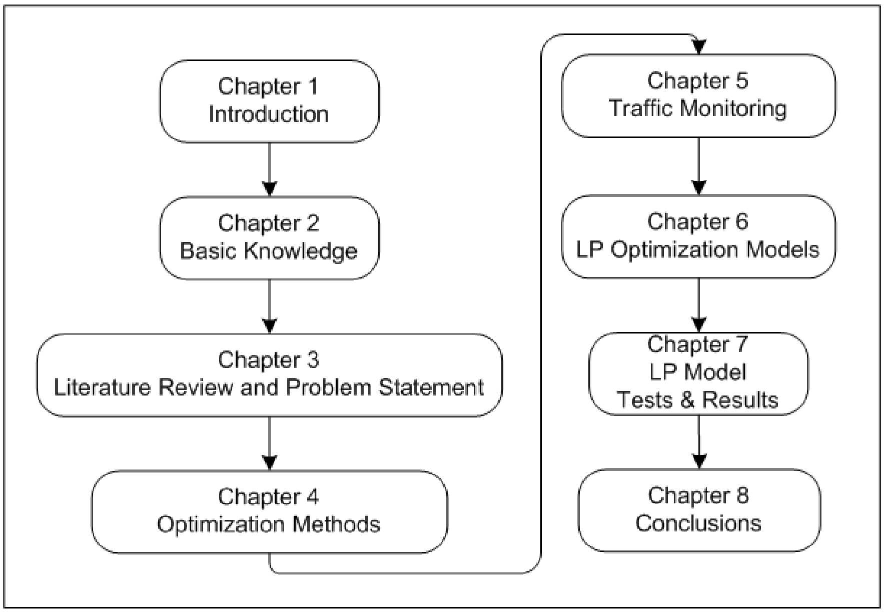


Figure 1.1: Structure of the thesis

1.3 Dissertation structure

This dissertation is organized as shown in figure 1.1. Next Chapter 2 describes the technologies/protocols generally used in a backbone network. It also classifies the failures and shows the different types of restoration schemes that could be used in a network topology affected by a fault condition.
Chapter 3 reviews the most significant works on network flows optimization and survivable techniques for backbone networks and, at the end of the chapter, the problem statement for the present thesis is given.
Chapter 4 provides the theoretical approaches to general optimization algorithms, namely: heuristic and deterministic.
In Chapter 5 the technique and the developed tool used to generate the traffic matrix are introduced.
Chapter 6 goes through the Linear Programming models that represent the core of the present dissertation.
Chapter 7 shows the tests that have been made and the results obtained using, as benchmark, both synthetic and real networks. Finally, Chapter 8 discusses the contributions of this work and concludes this dissertation.

Chapter 2

Basic knowledge

This chapter introduces the basic knowledge on IP routing protocols, MPLS technology, failures in a telecommunication network, and traditional restoration schemes.

2.1 Understanding IS-IS

What is what?

An autonomous system (AS) is a collection of IP networks and routers controlled by a single administrative entity, (by a common network administrator or by a group of administrators) that presents a common routing policy to the Internet. An autonomous system (sometimes referred to as a routing domain) has assigned a globally unique number, called Autonomous System Number (ASN).

The Open System Interconnection (OSI) protocol suite specifies two routing protocols, designed for the ASs, at the network layer: the End System-to-Intermediate System (ES-IS) and the Intermediate System-to-Intermediate System (IS-IS).

ES-IS protocol allows communication between end systems (hosts on a network) and intermediate systems (routers that are attached to other networks), while IS-IS is an interior (intradomain) routing protocol[1], designed to work within an autonomous system.

---

[1] The routing protocols used within an AS are called Interior Gateway Protocols (IGP).

OSI also defines an exterior (interdomain) routing protocol, that is the Interdomain Routing Protocol (IDRP[1]), designed to exchange routing information between autonomous systems.

How does it work?

IS-IS and OSPF, are the most used IP routing protocols within the backbone networks and the router packet-forwarding decision is taken using only the destination address specified in the packet header.

In order to determine the routes to all reachable destinations, IS-IS routers exchange link state information with their nearest neighbours. These topology information together with a metric value associated to every link, are flooded throughout the AS, so that every router within the AS has a complete knowledge of the topology of the AS.

Starting from the knowledge of the full topology of the “network”(the so-called link state database), each router routes the traffic toward a destination node, along shortest paths. In fact, based on the graph, each router constructs a tree consisting of the shortest paths to all destinations, and with itself as the root.

The calculation of the paths in IS-IS, and then the building of each routing table, is based on the “shortest path first” algorithm developed by Edgar W. Dijkstra [32] (see figure 2.1(a)). If the network topology changes, the protocol recalculates the routes, using the Dijkstra’s algorithm.

The Dijkstra’s Algorithm

- $C(i,\ j)$ = cost of link connecting nodes i and j
- $D(v)$ = current cost value of the path toward the destination node v
- $p(v)$ = predecessor node along the path toward v
- N = set of destination nodes with a known shortest path

---

[1] IETF introduced the Border Gateway Protocol (BGP) as exterior/border routing protocol that is widely used in in backbone networks as well as in the Internet.

- A = source node

Pseudocode of the algorithm

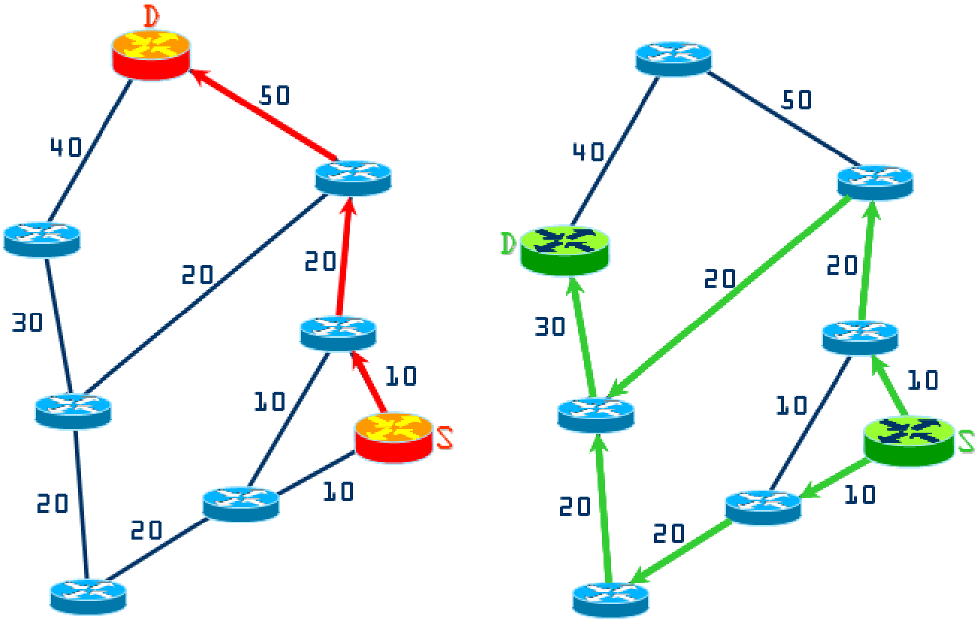
```
1. Initialization
    • $N = \{A\}$
2. For all nodes $v$
    • if $v$ is adjacent to $A$
        - then $D(v) = C(A, v)$
        - otherwise $D(v) = \infty$
3. Cycle ends when all nodes belong to $N$
    • find a node $w \notin N$ such that $D(w)$ is a minimum
    • add $w$ in $N$
    • update $D(v)$ for all $v \notin N$ adjacent to $w$
    • $D(v) = \min\{D(v), D(w) + C(w, v)\}$
```

Multiple paths

The IS-IS link metrics are constrained to be integers within the range from 1 to 65535. This bound increases the probability of obtaining equal cost paths as shown in figure 2.1(b). In this case, the traffic flows will be equally split among the shortest paths. Generally, the traffic splitting follows the mechanism of "per packet round robin", where each packet matching a given destination, is forwarded toward the egress node using the least recently used equal cost path.

Pros and Cons

The main advantage of a link state routing protocol is that the complete knowledge of topology allows routers to calculate routes that satisfy particular criteria. This can be useful for traffic engineering purposes, where routes can be constrained to meet particular quality of service requirements.



(a) IS-IS single path (b) IS-IS multiple equivalent paths

Figure 2.1: *Single and multiple paths*

The main disadvantage of a link state routing protocol is that it does not scale well, as more routers are added to the routing domain. In fact, by increasing the number of routers, the size and the frequency of the topology updates increase. In the meantime also time taken to calculate end-to-end routes increases. This lack of scalability means that a link state routing protocol is unsuitable for routing across the Internet at large. That is the reason why IGPs only route traffic within a single AS.

2.2 Understanding MPLS Technology

What is what?

MPLS (Multi-Protocol Label Switching) combines the speed and performance of packet-switched networks with the intelligence of circuit-switched networks to provide a best-of-breed solutions for integrating voice, video and data.
MPLS is an IETF standard built on the efforts of speeding up the management of packets into the inner nodes of the network, assigning most of the complex functions to the edge routers. Moreover, with its Traffic Engineering extension, MPLS is able to facilitate resource allocation and to realize particular type of services.

The MPLS architecture lies at an intermediate level of the OSI stack, placed between Layer 3 (Network) and Layer 2 (Data Link), usually named the “2.5 Layer”.
The term *multiprotocol* derives from the fact that MPLS is able to manage any layer 3 protocol and, typically, is applied to the IP protocols.

How does it work?

As shown in figure 2.2, a router operating in an MPLS domain is called Label Switching Router (LSR) and the one receiving/transmitting traffic from/to outside the domain is called Edge LSR (ELSR) or Label Edge Router (LER).
The main concept driving the MPLS idea is that packets arriving at the ingress LER have to be transported inside the MPLS domain, toward the egress LER.

For each destination address, or for each LER, a Forward Equivalent Class (FEC) is defined that aggregates packets needing the same treatment and directed to the same destination.
When an LER receives an IP packet, it classifies such packet in the correspondent FEC, basing on the information contained in the IP header. Moreover, it inserts a Label between the layer 2 header (e.g. Ethernet or ATM) and the layer 3 header (e.g. IP).

Labels are assigned and distributed by means of a protocol used for this

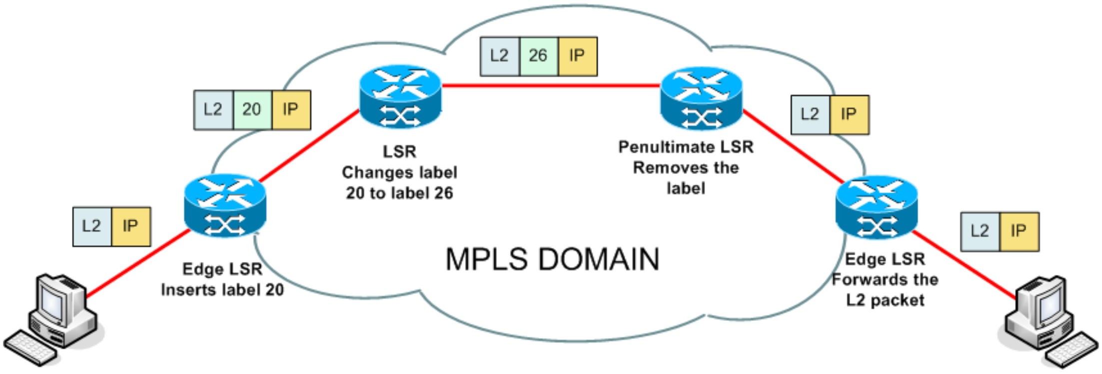


Figure 2.2: MPLS packet delivery mechanism

purpose[1]. All MPLS routers within the network regularly exchange label and reachability information to build a complete knowledge of the network, which is then used to determine paths and specify the new label to place onto the packet.

The label has a constant short length (20 bits) and identifies the FEC the packet is belonging to.

From the ingress LER to the egress LER, the whole set of forwarding operations will be realized only using the labels, while the IP header as well as the layer 2 header, will be hidden until the egress router is reached.

Each traversed LSR reads the incoming label, finds the corresponding FEC, looks up for the outgoing port, strips off the existing label, and applies a new label, which tells to the next hop LSR how to forward the packet.

The last LSR (penultimate hop) removes the label and forwards the packet to the egress LER, which will route such packet using a traditional IP protocol.

The path followed by packets belonging to the same specific FEC is called Label Switched Path (LSP), meaning that paths may be selected based on application requirements such as bandwidth required or maximum latency.

LSP

The concept of LSP is quite simple, i.e. the traffic flows in one direction, from the head-end toward the tail-end using a specific path. Hence, duplex traffic requires two LSPs: one LSP to carry traffic in each direction.

---

[1] For example the Label Distribution Protocol (LDP)

This connection may be established for a variety of purposes, such as to guarantee a certain level of performance, to route around network congestion, or to create IP tunnels for network-based Virtual Private Networks (VPN).

In many aspects, LSPs are not so different than switched paths in ATM or Frame Relay networks, except that they are not dependent on a particular Layer 2 technology.

Traffic is assigned to LSPs based on pre-defined criteria; for example, all the high priority traffic generated from a critical application (e.g. bank trading) will be routed in a dedicated LSP.

2.2.1 MPLS Traffic Engineering (MPLS TE)

MPLS networks can use native Traffic Engineering (TE) mechanisms to minimize network congestion and improve network performance. Mapping efficiently the traffic streams to network resources can significantly reduce the occurrence of congestion and can improve quality of service in terms of latency, jitter, and packet loss. MPLS implementations can vary widely, from simple “best effort” data delivery to advanced networks, which guarantee delivery of information including re-routing to an alternate path (in case of a link or network failure) within 50 milliseconds.

Historically, IP networks relied on the optimization of underlying network infrastructure, or Interior Gateway Protocol (IGP), in order to perform TE.

On the contrary, MPLS extends existing IP protocols, such as IS-IS, and makes use of MPLS forwarding capabilities to provide native TE. In addition, MPLS TE can reduce the impact of network failures and increase service availability [17].

As mentioned in the previous paragraphs, an ingress LSR (or head end) can set up a TE LSP to an egress LSR (or tail end) through an explicitly defined path containing a list of intermediate LSRs (or midpoints).

IP routing protocols compute routing paths assigning a single metric per link and using destination-based routing not providing a general and scalable method for explicitly routing traffic.

In contrast, MPLS networks can support destination-based and explicit routing simultaneously. In the following of this dissertation these different kinds of LSPs will be distinguished as “Implicit LSP” and “Explicit LSP”, respectively.

The LSP can be a best-effort connection, in which case Label Distribution Protocol (LDP) or the earlier Tag Distribution Protocol (TDP) may be used. Alternatively, an LSP may request that bandwidth be reserved for its exclusive use.
Once allocated, MPLS guarantees that the bandwidth is available for the entire path. If the bandwidth is not available, then the connection request is refused. The LSP reserves bandwidth using either Resource Reservation Protocol with Traffic Engineering extensions (RVSP-TE) or Constraint-based Routing LDP (CR-LDP).

To enable network availability, MPLS provides mechanisms to quickly find an alternate path if the primary path is no longer available (typically due to a node or link failure). This Fast Re-Route (FRR) capability is critical for allowing service providers to offer high availability, high revenue SLAs.

An MPLS router can manage multiple paths to each destination. This technique is known as “liberal label retention” and, requiring more memory and processor cycles, it is most appropriate for edge routers where the unique important task is storing labels for connections which originate/terminate on that router.
Alternatively, the router may store only one path to each destination. This technique is called “conservative label retention” and, in the event of a failure, it requires that the signalling protocol determines a new optimal path. This process can take several seconds, since it may be necessary for the underlying IP routing protocols (typically iBGP, OSPF or IS-IS) to re-converge. To eliminate this delay, it is possible to pre-define alternative IP paths through the network.

2.3 Failure analysis

The backbone networks are usually well-engineered and properly provisioned, leading to very low packet losses and negligible queuing delays. This robust network design is one of the reasons why the occurrence and impact of failures in these networks have received little attention.

However, failures occur almost everyday [49] and an understanding of their characteristics, properties and effect is extremely valuable.

Accordingly with [59], failures are classified in two main groups: those resulting from scheduled maintenance activities and those generated by unplanned failures. While the events belonging to the former set can be easily tackled, those included in the latter set must be identified as the shared link failures (further distinguished among those that share IP routers and those that share optical devices) and the single link failures.
The distribution in time and space of the faults affecting a backbone network shows that:

- 20% of all failures can be attributed to scheduled network maintenance activities
- 80% of all failures can be attributed to unplanned failures, where:
  - 30% are identified as shared link failures. Half of them are deriving from router problems, which are hardware dependent faults, while the remaining half are directly connected to the optical infrastructure;
  - 70% are classified as single link failures

The present dissertation is mainly focused on finding the optimal configuration of the network parameters, which guarantees the delivery of all traffic flows even in case of single link failure (100% survivability level). As will be shown in Chapter 6, the proposed network configuration is a combination of IS-IS routing and MPLS LSP.

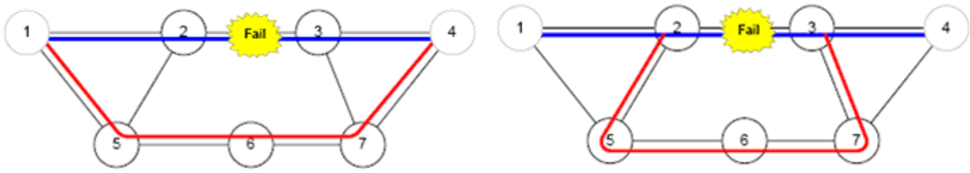
2.4 Restoration Schemes

Restoration schemes are classified as link restoration and path restoration as shown in Figures 2.3, where the blue line represents the primary path followed by flow 1 → 4 while the red line is the backup one.

As shown in figure 2.3(a), in path restoration, the origin/destination nodes of the traffic traversing the failed link, initiate the rerouting process.

In contrast, in link restoration, the end nodes of the failed link are responsible for rerouting the affected traffic flows as depicted in figure 2.3(b).
Path restoration will require less spare capacity, but is more complex to implement with respect to link restoration, as many more nodes are involved in the restoration process, and it is also slower in the speed of restoration as compared to link restoration.

In the present dissertation will be presented models that make use of the link restoration as well as a preliminary model with the path restoration scheme.



(a) Path restoration (b) Link restoration

Figure 2.3: *Failure restoration schemes*

Chapter 3

Literature Survey & Problem Statement

This chapter summarizes the most relevant works on optimization of telecommunication networks using IS-IS (or OSPF) routing, or MPLS techniques, or a combination of the two. Furthermore, studies on survivability of a telecommunication network will be widely discussed highlighting the contribution of the present dissertation. In the end of the chapter the problem statement constituting the core of this work will be introduced.

3.1 Optimization of IS-IS/OSPF routing

The IS-IS/OSPF routing optimization problem can be summarized with the following sentence [43] *"Can a sufficiently clever weight settings make OSPF routing perform nearly as well as optimal general/MPLS routing?"*

It has been demonstrated that the answer to this question is negative.
[43] is the first and the most important contribution in analyzing and solving the stated problem. In this paper the authors introduced the well-known concept of "General Routing Problem" (GRP), that is a Multicommodity Minimum Cost Flow (see Chapter 6) representation of the search for the optimal traffic flow distribution in a graph.

With this formulation, a piece-wise linear cost function is associated to each arc, as shown in figure 3.1, stating that it is "cheaper" to send additional flow over an arc with a small utilization rather than over a highly occupied link. The objective function to be minimized is the sum over all arcs, of the associated cost functions.

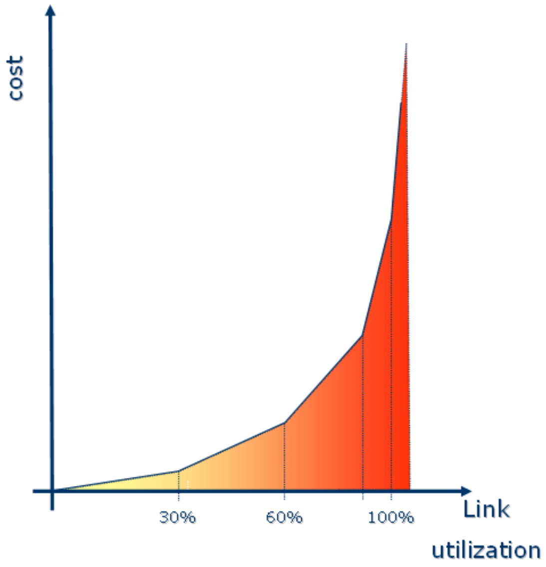


Figure 3.1: Cost function depends on link utilization

The knowledge of the optimal solution for the GRP is an important benchmark for evaluating the quality degree of solutions obtained when the OSPF (or IS-IS) routing constraints are introduced within the GRP model.
To solve the linear formulation of the GRP, the authors used CPLEX [7] solver, while the Simulated Annealing algorithm has been used by the author in order to solve the NP-Hard Mixed Integer problem representing the model in the real world.

In the last few years different techniques have been used to solve this problem such as Tabu Search [44], Genetic Algorithm [57] [36], and tailored heuristics presented in [52] and in [48].
Results obtained with all the different approaches have shown that it is possible to find a "clever" weight settings that provides a nearly optimal flow distribution, even when compared with those achieved with the MPLS technology.

In [64] an adaptive and distributed algorithm that balances the link load in an OSPF network is proposed. The basic idea is to find a method that gradually modifies the utilization of the network making small changes in the traffic splitting ratios, operating directly in the routers configuration. This approach could generate instability of the network, with a consequent loss of performances, when the traffic matrix rapidly changes.

3.2 Optimization of MPLS routing

From the introduction of the MPLS technology in the telecommunication world, many works to optimize the traffic engineering techniques have been proposed. The main objective of these papers is to find the optimal LSPs configuration that minimizes different objective functions, such as the total congestion of the network, the total mean delay, or the maximum number of hops in an LSP.

In [58] several heuristics used to search the optimal LSPs distribution minimizing the maximum utilization of the network are compared.

In [65] the authors use the same approach of [64] in order to minimize the maximum link occupation or the total mean delay of the network. Basically, the link loads are periodically measured and the traffic is gradually dislocated from the congested part of the network toward the less congested one. The adaptive algorithm has been applied to small networks and does not consider the impact of the link failure.

In [26], Tabu Search is used to find a layout of MPLS paths with the minimum number of hops, while in [39] the routing problem for MPLS networks is represented as an off-line multiobjective Mixed Integer Programming that looks for the best trade-off between the minimal routing delay, the optimal load balancing, and the minimal splitting of traffic trunks.

In [38] and in the following paper by the same authors [23] a multipath adaptive traffic engineering mechanism is introduced. The approach aims to avoid network congestion by adaptively balancing the load among multiple paths based on measurement and analysis of path congestion level.

In [63] the author presents a novel algorithm based on Simulated Annealing to optimize OSPF link metrics and, in a successive step, two Mixed Integer Programming (MIP) models, to setup complementary MPLS paths, are proposed. Basically, the majority of traffic is routed along the OSPF shortest paths and it has been demonstrated that already a small number of LSPs greatly improve the network congestion level.

3.3 Survivability problem

All the aforementioned papers do not take into consideration the problem of survivability of a telecommunication network.

A general formulation of the survivability problem, as a linear programming problem, has been addressed in [24], which proposed a cutting plane algorithm based on the concept of analytic center to manage the huge size of the problem. This method has been applied to networks with up to 60 nodes and 120 links, but it does not take into account any routing constraint.

The survivability problem applied to an OSPF routed networks has been recently addressed in [44], [61],[66] , [60], and in [28].

In [44] and [61] the authors adapt the heuristic used in [43] in order to find a robust weights set that guarantees the delivery of traffic (no congestion) taking into account all the single link failure scenarios as alteration of the cost function introduced in [43]. Due the fact that the problem is NP-hard, and then too time consuming, the authors selected a critical set of failure scenarios representative of all the scenarios and, in order to solve such problem, they used the Tabu Search metaheuristic.

A similar approach has been introduced in [60], but, thanks to a highly efficient Tabu Search implementation, the whole set of failure scenarios have been considered.

In [28], the same problem has been solved using an evolutionary algorithm, but also the failure of a router is considered.

A complete different method has been proposed in [66], where a bicriteria approach is used. In particular, the objective function considers both the network utilization of the normal state and that of the failure states, but the impacts are separately considered. Basically, the authors use a local search algorithm derived from the Tabu Search, where the objective function “drives” the search toward Pareto optimal solutions.

Obtained results with the different approaches, show that it is possible to significantly reduce the congestion in case of single failure in an IS-IS/OSPF network, by allowing a slightly higher maximum occupation in normal conditions. The drawback of all presented methods is the huge amount of time required in order to find a feasible solution.

Network survivability with MPLS technology has been mainly formulated as Spare Capacity Allocation (SCA) problem [54], [55], [53]. In this case the authors try to mitigate the impact of a single link failure by placing sufficient capacity in the network. A possible limitation of such an approach is that it strongly depends on the network topology, because it is necessary to create an LSP backup path entirely disjointed from the primary working LSP and, in particular topologies, this goal is unachievable [25].

3.4 Problem statement

Based on the above literature review, the problems addressed in this dissertation are:

1. Is it possible to obtain a robust configuration of the network using the combination of IS-IS routing and MPLS-TE techniques?
2. Is it possible to formulate the question addressed in 1. as a pure LP problem?
3. Is it possible to obtain the optimal configuration in a reasonable time?

Chapter 4

Optimization Methods

This chapter describes different typical approaches to optimization methods, namely heuristic and deterministic algorithms. At the end of the chapter the techniques used in the present dissertation are briefly introduced. A wide and detailed explanation will be furnished in the next chapters.

4.1 Heuristic Algorithms

The term heuristic is used for those algorithms that look for feasible solutions among all possible ones, but that do not guarantee the optimality of the solution found. In fact, a heuristic method usually finds good solutions lying very close to the optima and runs reasonably quickly. Unfortunately, there is no argument that this will always be the case.

With the term meta-heuristic a class of general heuristic methods is indicated, which can be applied to a wide range of computational problems by combining heuristics. Meta-heuristics are generally applied to problems for which it does not exist a specific algorithm or heuristic, or simply when it is not useful to implement such a method.

The goal of heuristic and meta-heuristic optimization algorithms, is to find the combination of input parameters (state) among all possible solution (search space), which minimizes (or maximizes) a specific function (objective or goal function). Variants and "hybrids" of heuristic techniques have been proposed in literature, applying them to specific complex problems. As shown in Chapter 3, meta-heuristics techniques have been widely used in traffic network optimization problems, but the running time closely depends on the dimension of the problem and this may be a fundamental factor in particular class of decision problems.

Some well-known meta-heuristics are:

- Greedy Algorithm
- Best-first Search
- Simulated Annealing
- Ant Colony Optimization
- Tabu Search
- Genetic Algorithm

In the following Tabu Search, Genetic Algorithm, Simulated Annealing, and the optimization method derived from the observation of social insects (Ant Colony), are briefly described.

4.1.1 Tabu search (TS)

Local search heuristic starts with an initial solution and moves from neighbour to neighbour decreasing the objective function value.

The main problem with this strategy, common to other minimization methods, is to escape from local minima where the search process is not anymore able to find any further neighbourhood solution that decreases (or increases) the objective function value.

Different strategies have been proposed to solve this troublesome problem, and one of the most efficient strategies is the Tabu Search.

The basic concept of Tabu Search (TS), as described by one of its author (F. Glover) in 1986 [47], is “a meta-heuristic superimposed on another heuristic”.

As aforementioned, the overall approach is to avoid short term cycles by forbidding, in the next iteration, moves which may take the proposed solution to points previously visited (hence named “Tabu”) and preventing to get trapped into a local minimum. Each encountered configuration is stored in one or more tabu lists, forming the so-called Tabu Search Memory.

The Tabu search method does not choose the next step (or move) randomly, but following a specific strategy.

4.1.2 Genetic Algorithm (GA)

"Genetic algorithms are based on a biological metaphor: they view learning as a competition among a population of evolving candidate problem solutions. A 'fitness' function evaluates each solution to decide whether it will contribute to the next generation of solutions. Then, through operations analogous to gene transfer in sexual reproduction, the algorithm creates a new population of candidate solutions." [56].

From this definition, typical genetic algorithm requires two factors to be defined:

1. a "genetic" representation of the solution domain,
2. a "fitness function" to evaluate the solution domain.

A standard genetic representation of the solution is as an array (bits, integer, real, or other types work essentially in the same way) because this data structure, especially if it has fixed length, facilitates the crossover operation. The fitness function is always problem-dependent, it is directly defined over the genetic representation, and it measures the quality of the solution.
The main steps followed by the Genetic Algorithm are:

1. **Initialization**
   Initially many individual solutions are randomly generated to form an initial population over the search space (solution domain).

2. **Selection**
   During each successive step (called "epoch"), a certain proportion of the existing population is selected to breed a new generation. Individual solutions are selected using fitness function, which may be a stochastic function defined in order to select a small portion of less fit solutions. This helps

to keep a certain diversity of the population, preventing premature convergence on poor solutions.

3. **Reproduction**
   The third step of the Genetic Algorithm is to generate a next generation of population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation. A pair of "parent" solutions is picked out, from the set of solution previously selected, for "breeding" and to generate a new solution ("child") using the methods of crossover and mutation. The "child" typically shares many of the characteristics of its "parents". Selecting new parents for each child, the process continues until a new proper population of solutions is generated.
   As well as in biology only the "chromosomes of best organisms" from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

4. **Termination**
   This generational process is repeated until a terminating condition has been reached, such as:

   - A solution is found that satisfies minimum criteria
   - Maximum number of generations reached
   - Maximum computation time reached
   - Successive iterations no longer produce better results
   - Manual inspection
   - Combinations of the above

4.1.3 Simulated Annealing (SA)

The Simulated Annealing (SA) is the oldest among the metaheuristics and also one of the first algorithms that has an explicit strategy to avoid to be trapped into local minima, and it was first presented as a search algorithm for CO (Carbon-Oxide) problems in [51] and [29].

As the name say, the SA simulates the process of tempering of a metal (e.g. steel) and glass, and assumes each point within the search space as the state of a physical system. The objective function is interpreted as the internal energy at each state and the algorithm tries to carry the system from an arbitrary initial state, to the state with minimum possible energy.

In order to escape from local minima the algorithm permits the so-called "uphill moves" where the resulting solution has a worse value . As will be shown in the following, the probability of accepting uphill moves generally decreases and is controlled by two factors: the difference of the objective functions and the value of a global time varying parameter called "temperature".
The main steps followed by the Simulated Annealing Algorithm are:

1. **Generation of the initial solution**
   The initial solution can be either randomly or heuristically produced. During this phase the temperature parameter is initialized.

2. **Fundamental iteration**
   At each step the SA algorithm compares a randomly sampled solution from the neighbourhood of the current solution x. This new value x' is accepted as new current solution with a probability that is generally evaluated following the Boltzmann distribution $e^{(-(f(x')-f(x))/T)}$, where $f(x)$ and $f(x')$ are the values of the objective function at state x and x' respectively, and T is the temperature parameter.
   Analogously to the physical annealing process, the temperature decreases as the simulation proceeds, thus at the beginning of the search the probability of accepting uphill moves is high and the algorithm explores the search space (random walk).
   In a second phase, when the probability of uphill moves decreases (it is inversely proportional to T), the method becomes an iterative improvement algorithm converging to a global (or a local) minimum. This step is repeated until the termination condition is satisfied.

3. **Convergence to optimum**
   As theoretical result, it can be shown that for any given finite problem, the

probability that the simulated annealing algorithm finds the global optimal solution approaches 1 as the annealing schedule is extended. However, this condition will usually exceed the time required for a complete search of the solution space.

4.1.4 Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a metaheuristic approach proposed by M. Dorigo et al. in [34], [33], and [35].

The inspiring source of ACO is the foraging behaviour of real ants that consents to find shortest paths between food sources and the nest. In fact, while walking from food sources to the nest and vice versa, ants release a chemical substance (the pheromone) on the ground, and the direction chosen by the following ants is the path marked with a stronger pheromone concentrations.

ACO algorithm is based on a particular parametrized probabilistic model, called by the authors the pheromone model. “Artificial” ants increasingly construct solutions by adding opportunely defined solution components to a partial solution under consideration.

The ACO algorithm has been successfully used in the telecommunication routing problem, with the name of AntNet, using two sets of homogeneous mobile agents, called forward and backward ants. Agents belonging at each set have the same structure, but they can “sense” different inputs and produce different independent outputs.
The AntNet algorithm can be briefly described as follows:

- Let $\mathcal{G} = (\mathcal{C}, \mathcal{L})$ be a completely connected graph, whose vertices are the solution components $\mathcal{C}$ and the set $\mathcal{L}$ are the connections. This graph is commonly called the construction graph.

- At regular intervals, from every network source node s, a mobile agent (the forward ant) is launched, with a randomly selected destination node d. The agent owns a memory stack and a dictionary structure in which the information about the elapsed time to reach the visited node and the identifier associated to the node are stored.

- If a cycle is detected, that is, if an ant is forced to return to an already visited node, the cycle's nodes are popped from the ant's stack and all the memory about them is destroyed.
- When the destination node d is reached, the agent becomes a different agent (backward ant) transferring to it all the contents of its memory.
- The backward ant takes, in the opposite direction, the same path as that of its corresponding forward ant. At each traversed node along the path it pops its stack to know the next hop node.
- At each node reached, the backward ant updates two data structures held and updated by every node, that are a routing table, organized as in vector-distance algorithms, and a list representing a "memory" of the network as seen by the reached node in probability terms.

The update of the routing table happens using the trip time experienced by the forward ant, that gives a clear indication about the goodness of the followed route from a physical point of view (number of hops, transmission capacity of the used links, processing speed of the crossed nodes) and from a traffic congestion point of view.

4.2 Deterministic Algorithms

In deterministic models good decisions bring about good outcomes. Given a particular input, it will always produce the same correct output, and the underlying machine will always pass through the same sequence of states. Therefore, the outcome is deterministic (e.g. risk-free).
Deterministic algorithms are by far the most studied and familiar kind of algorithm, as well as one of the most practical, since they can be run on real machines efficiently.

One simple model for deterministic algorithms is the mathematical function; just as a function always produces the same output given a certain input. The difference is that algorithms describe precisely how the output is obtained from the input, whereas abstract functions may be defined implicitly.

Examples of particular abstract machines which are deterministic include the deterministic Turing machine and deterministic finite automaton.

4.3 Linear Programming Approach

In the present work Linear Programming (LP) optimization methods have been used to perform the optimization process. The in-depth investigation of the LP techniques is presented in Chapter 6.

Chapter 5

Traffic Monitoring

This chapter introduces the concept of Traffic Matrix. Moreover, the methods used to obtain it from data collected by a proprietary tool will be widely described.

5.1 Traffic Matrix

Basically, a traffic matrix describes the amount of data traffic transmitted between every pair of ingress and egress points in a network. It is a $(N-1) \times (N-1)$ matrix, with N = number of nodes, with zeros in its main diagonal. The generic term a_{ij} indicates the flow originated by node i directed to node j.

Obtaining the traffic matrix is not an easy task. Generally an empiric approximation of the matrix is generated, based on network operator's sensibility. During the meeting with the employers of an important Italian ISP (Tiscali [20]), a criterion has been identified in order to generate the traffic matrix directly from the aggregated flows traversing the links of the Tiscali's Italian backbone network.

In fact, it is easy to observe and collect data from the interfaces of a router regarding the total amount of traffic flowing on it, with no regard about the source of the destination of the flow. Moreover, using this kind of measure, the load of the router's CPU is basically insignificant.

The network operators have anticipated that the traffic is usually subdivided as follows:

- 60% of the total traffic is directed from Italy toward foreign countries;
- 30% of the total traffic is exchanged between different Italian ISPs;
- 10% of the total traffic is flowing inside the ISP backbone.

This criterion gives a rough approximation of the traffic matrix, since the number of links in a network is typically much smaller than the number of node pairs. Thus, a finer method is needed in order to generate a more trustworthy picture of the flows within the network. To achieve this task in the present work a traffic measurement tool, named ICEFlow, has been developed in collaboration with Tiscali.

5.2 Traffic monitoring

Many of the decisions of the IP network operators depend on how the traffic flows in their network.

Generally, as mentioned above, the support in routers for measuring traffic matrices is poor and operators are often forced to estimate the traffic matrix from other available data, typically link load measurements and routing configurations. The link loads are readily obtained using the Simple Network Management Protocol (SNMP), which is part of the Internet Protocol (IP) suite, and which consists of a set of standards for network management defined by the Internet Engineering Task Force (IETF).

However, more sophisticated tools exist that permit to collect a more granular data regarding the traffic managed by a specific router.
Cisco NetFlow is the most used instrument by traffic monitoring specialists and, in particular by the operators of the ISP we used as benchmark case. For this reason it is addressed in the following.

5.2.1 Cisco's NetFlow

Each packet flowing through a router is analyzed using its own set of IP attributes. The so-called IP Packet attributes used by NetFlow, as reported in the Cisco documentation [6], are:

1. the IP source address;
2. the IP destination address;
3. the Source port ;
4. the Destination port ;
5. the Layer 3 protocol type ;
6. the Class of Service;
7. the Router interface.

All packets with the same characteristics are assembled into a single flow and then packets and bytes tallied. The Cisco's NetFlow is a part of the Cisco Internetwork Operating System (IOS) and enables routers to condense all these information in a cache memory called the NetFlow cache as shown in Figure 5.1[1]. It is possible to set the sampling rate of the packets in order to avoid the loss of performance of the router.

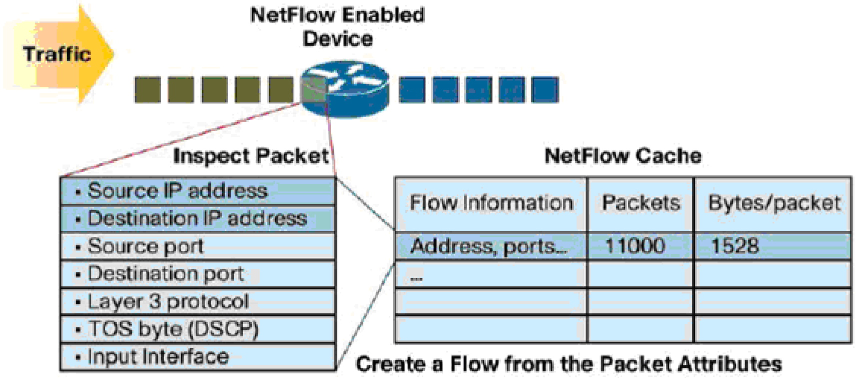


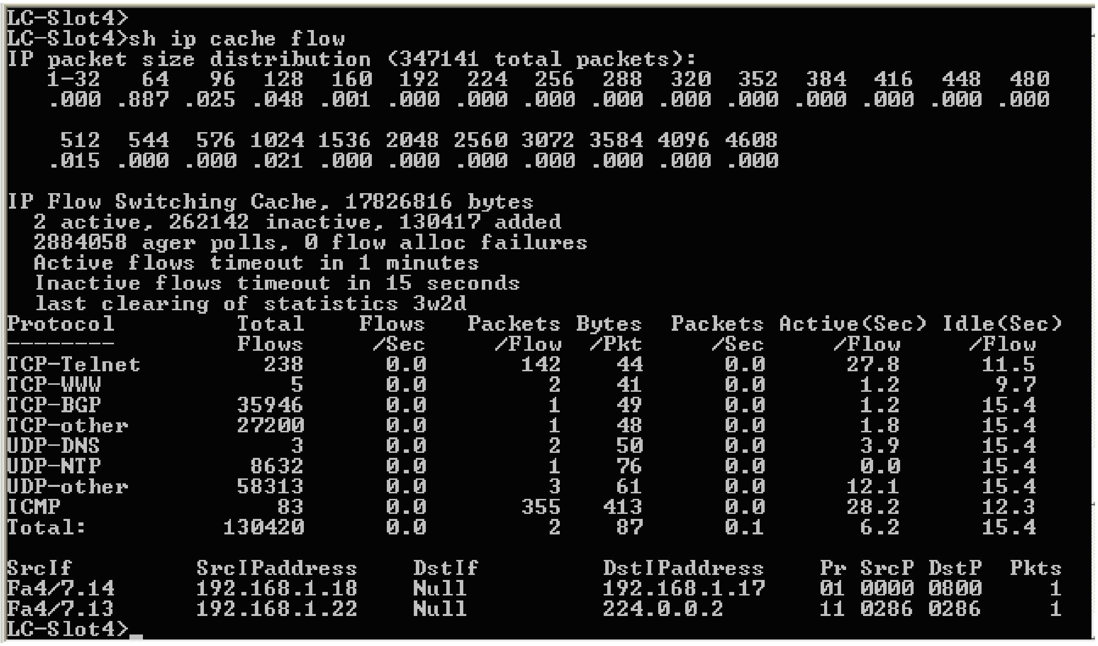
Figure 5.1: Exporting IP attributes to NetFlow cache

Two primary methods to access and analyze NetFlow data exist, which are the Command Line Interface (CLI) with its "*show*" commands, as shown in Figure 5.2, or utilizing a reporting tool running in a server.

---

[1] Courtesy of Cisco Systems

Exporting the cache memory contents to an external device assures a smaller load of the router in terms of occupied memory and CPU working time. Moreover, the NetFlow collector is in charge of assembling and understanding the periodically received information (UDP packets) and of producing reports underlying the desired data characteristics.



```
LC-Slot4>
LC-Slot4>sh ip cache flow
IP packet size distribution (347141 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .887 .025 .048 .001 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .015 .000 .000 .021 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 17826816 bytes
  2 active, 262142 inactive, 130417 added
  2884058 ager polls, 0 flow alloc failures
  Active flows timeout in 1 minutes
  Inactive flows timeout in 15 seconds
  last clearing of statistics 3w2d
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet         238      0.0       142    44      0.0      27.8      11.5
TCP-WWW              5      0.0         2    41      0.0       1.2       9.7
TCP-BGP          35946      0.0         1    49      0.0       1.2      15.4
TCP-other        27200      0.0         1    48      0.0       1.8      15.4
UDP-DNS              3      0.0         2    50      0.0       3.9      15.4
UDP-NTP           8632      0.0         1    76      0.0       0.0      15.4
UDP-other        58313      0.0         3    61      0.0      12.1      15.4
ICMP                83      0.0       355   413      0.0      28.2      12.3
Total:          130420      0.0         2    87      0.1       6.2      15.4

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Fa4/7.14      192.168.1.18    Null          192.168.1.17    01 0000 0800     1
Fa4/7.13      192.168.1.22    Null          224.0.0.2       11 0286 0286     1
LC-Slot4>_
```

Figure 5.2: NetFlow cache

5.2.2 NetFlow tools

During the scouting exercises performed to identify and select the best tool for the present work, several open source instruments have been analyzed and tested. In particular, CFlowd [5] and Flow-Tools [9], as collecting and analyzing tools, and FlowScan [10] and its variant JKFlow [13] as visual reporting tools have been examined.

Figure 5.3 shows the typical architecture used in order to collect and analyze the UDP NetFlow packets sent by the Cisco Routers. The Web Server permits to access the visual or textual reports generated by the analyzer. Usually, the collector, the analyzer, and the Web Server are condensed in a single hardware device.

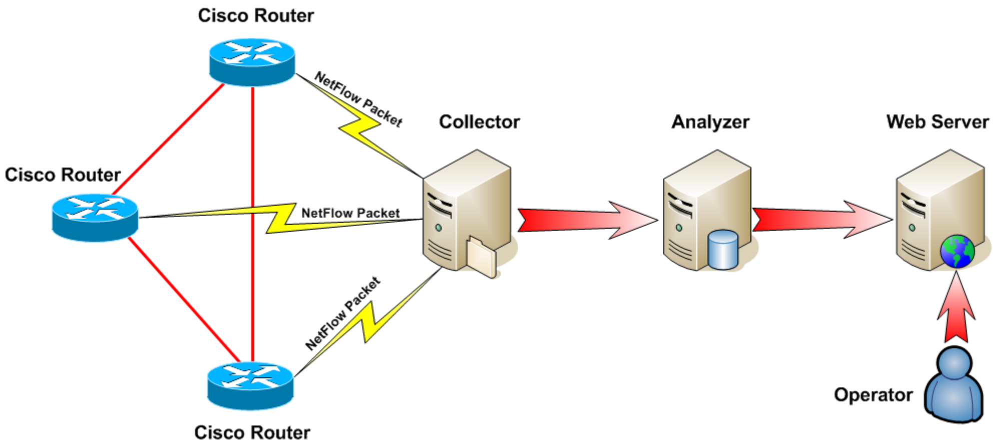


Figure 5.3: NetFlow collector architecture

5.2.2.1 CFlowd

CAIDA's CFlowd is a flow analysis tool released to enable ISPs to collect data from NetFlow routers. This analysis package is composed by three programs: *cflowdmux*, *cflowd*, and *cfdcollect*.
cflowdmux accepts Cisco flow-export packets arriving from the NetFlow routers as UDP packets and saves them in shared memory buffers.
Directly from these buffers, *cflowd* creates tabular data to be inputted to *cfdcollect*, which will store these data in different files.

Currently, CFlowd is no longer supported by the CAIDA team, and this is the reason that pushed this work to consider a different tool, that is Flow-Tools.

5.2.2.2 Flow-Tools

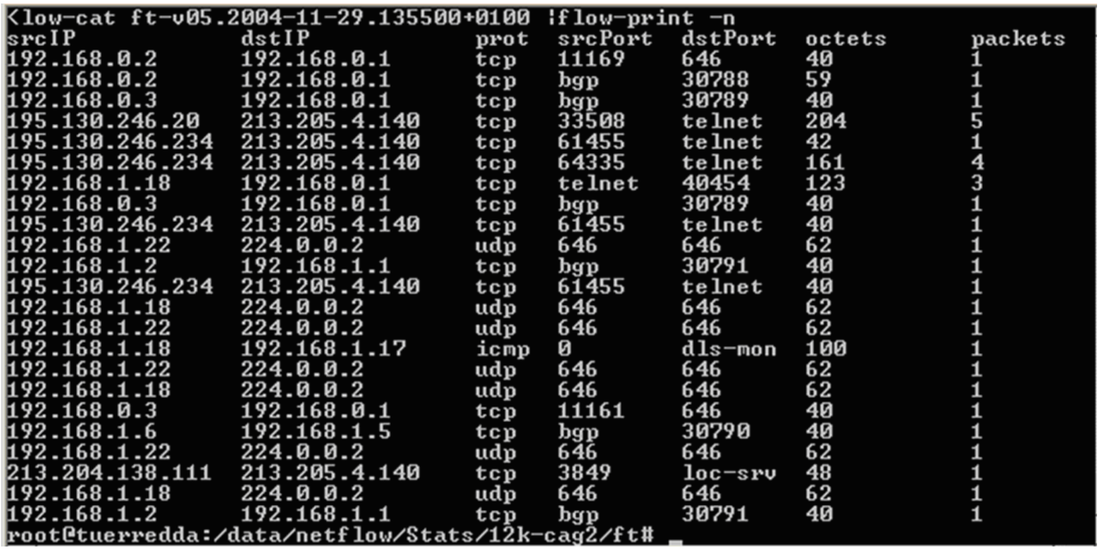
Flow-Tools is a set of programs for processing and generating reports from NetFlow data. The tools can run in a single server as well as in multiple servers for large collecting networks and it is compatible with several versions of NetFlow.

NetFlow data is collected, by default, every 30 seconds and stored in portable files every 5 minutes. These files may be analyzed by the following programs that are included in the Flow-Tools distribution (the list is directly reported from the Flow-Tools documentation).

- **flow-capture** - Collect, compress, store, and manage disk space for exported flows from a router.
- **flow-cat** - Concatenate flow files. Typically flow files will contain a small window of 5 or 15 minutes of exports. Flow-cat can be used to append files for generating reports that span longer time periods.
- **flow-fanout** - Replicate NetFlow datagrams to unicast or multicast destinations. Flow-fanout is used to facilitate multiple collectors attached to a single router.
- **flow-report** - Generate reports for NetFlow data sets. Reports include source/destination IP pairs, source/destination AS, and top talkers. Over 50 reports are currently supported.
- **flow-tag** - Tag flows based on IP address or AScnumber. Flow-tag is used to group flows by customer network. The tags can later be used with flow-fanout or flow-report to generate customer based traffic reports.
- **flow-filter** - Filter flows based on any of the export fields. Flow-filter is used in-line with other programs to generate reports based on flows matching filter expressions.
- **flow-import** - Import data from ASCII or cflowd format.
- **flow-export** - Export data to ASCII or cflowd format.
- **flow-send** - Send data over the network using the NetFlow protocol.
- **flow-receive** - Receive exports using the NetFlow protocol without storing to disk like flow-capture.
- **flow-gen** - Generate test data.
- **flow-dscan** - Simple tool for detecting some types of network scanning and Denial of Service attacks.

- **flow-merge** - Merge flow files in chronoligical order.
- **flow-xlate** - Perform translations on some flow fields.
- **flow-expire** - Expire flows using the same policy of flow-capture.
- **flow-header** - Display meta information in flow file.
- **flow-split** - Split flow files into smaller files based on size, time, or tags.
- **flow-print** - Display on screen information requested using, for example, flow-cat.

Figure 5.4 shows an example of the using flow-print/flow-cat. It is possible to retrieve information about the source/destination IP address, the type of protocol, the source/destination port, the dimension and the number of packets constituting the flow.



```
<low-cat ft-v05.2004-11-29.135500+0100 |flow-print -n
srcIP            dstIP            prot  srcPort  dstPort  octets      packets
192.168.0.2      192.168.0.1      tcp   11169    646      40          1
192.168.0.2      192.168.0.1      tcp   bgp      30788    59          1
192.168.0.3      192.168.0.1      tcp   bgp      30789    40          1
195.130.246.20   213.205.4.140    tcp   33508    telnet   204         5
195.130.246.234  213.205.4.140    tcp   61455    telnet   42          1
195.130.246.234  213.205.4.140    tcp   64335    telnet   161         4
192.168.1.18     192.168.0.1      tcp   telnet   40454    123         3
192.168.0.3      192.168.0.1      tcp   bgp      30789    40          1
195.130.246.234  213.205.4.140    tcp   61455    telnet   40          1
192.168.1.22     224.0.0.2        udp   646      646      62          1
192.168.1.2      192.168.1.1      tcp   bgp      30791    40          1
195.130.246.234  213.205.4.140    tcp   61455    telnet   40          1
192.168.1.18     224.0.0.2        udp   646      646      62          1
192.168.1.22     224.0.0.2        udp   646      646      62          1
192.168.1.18     192.168.1.17     icmp  0        dls-mon  100         1
192.168.1.22     224.0.0.2        udp   646      646      62          1
192.168.1.18     224.0.0.2        udp   646      646      62          1
192.168.0.3      192.168.0.1      tcp   11161    646      40          1
192.168.1.6      192.168.1.5      tcp   bgp      30790    40          1
192.168.1.22     224.0.0.2        udp   646      646      62          1
213.204.138.111  213.205.4.140    tcp   3849     loc-srv  48          1
192.168.1.18     224.0.0.2        udp   646      646      62          1
192.168.1.2      192.168.1.1      tcp   bgp      30791    40          1
root@tuerredda:/data/netflow/Stats/12k-cag2/ft# _
```

Figure 5.4: flow-print example

5.2.2.3 FlowScan

NetFlow data reports, generated with Flow-Tools, can be managed and displayed also using different open source instruments that are freely downloadable from

the Internet.
FlowScan is a set of PERL scripts and modules that binds together:

1. the flow collector (Flow-Tools);
2. a high performance Round Robin Database [18];
3. a visualization tool.

From the files created by Flow-Tools, FlowScan maintains counters reflecting what was found. These counter values are stored using the Round Robin Database and visualized in a friendly graphical fashion. FlowScan only permits a rough analysis of the data and, in the last few years, several modules with different levels of granularity have been developed, such as CampusIO [3], SubnetIO [19], Autofocus [2], CarrierIn [4], CUFlow [8] and, mainly, JKFlow [13].

5.2.2.4 JKFlow

The FlowScan modules enlisted in the previous subsection, have multiple distinct functionalities (e.g. CampusIO can report flows per services, while CUFlow provides excellent reporting on services for different routers) but do not have the capability to report site-to-site flows. JKFlow [13] is able to solve that problem is an easy and flexible to configure XML-FlowScan module, with the following basic concepts:

- **Sites / Subnets** - define source/destination subnets (e.g. a Country and its subnets)
- **Directions** - selects flows matching specific source/destination, sites/subnets with the possibility to define traffic pattern to be monitored, such as applications, services, protocols, and the total traffic.
- **Outbound traffic** - matches source address matching “from” sites/subnets and destination address with “to” sites/subnets. Inbound traffic - matches otherwise.
- **Multiple Directions** - it is possible to monitor multiple directions and, within each of them, it is possible to specify different traffic patterns.

- **Sets** - are grouping of traffic patterns to be observed. They can be defined over multiple directions.

JKFlow may appear as a complete FlowScan module but it has some drawback. For example, the subnets are defined by mean of a list of IP addresses and, in order to define a "direction" between two routers a high number of IP addresses are needed.

In other words, JKFlow isn't able to perform an AS-to-AS report[1]. This is a fundamental capability of the monitoring tool needed for the present work, mainly because the Italian backbone is organized per AS. To solve this problem we decided to develop a new module, directly form JKFlow, called ICEFlow.

5.2.2.5 ICEFlow

ICEFlow is a network traffic flow visualization, analysis, and reporting tool based on open source tools, namely RRD-Tools, Flow-tools and JKFlow, and has been developed by myself and C. Murgia.

ICEFlow is able to collect, send, process, and generate reports from NetFlow format (indigenous to Cisco routers) and to identify and distinguish the traffic per single protocol (ICMP, TCP, UDP,) and per single service/application (peer-to-peer applications, FTP, HTTP, e-mail,). Furthermore, its main characteristics are:

- **Granularity** - it can collect and analyze traffic flowing (in both the directions) in a single interface of a router, or the flows incoming/outgoing to/from a group of routers, as well as in a subnet or in an Autonomous System.

- **Robustness** - it can easily tackle network configuration changes and the insertion of new protocols and applications

The results are displayed in a graphical fashion or accessing downloadable textual files, with variable time-basis (by default from 1 day to 1 year) and different scale of traffic (Mbps, packets, flows).

---

[1] In the last version of JKFlow the capability of monitoring AS-to-AS traffic has been implemented. This add-on has been done after that the phase of collecting traffic for the present work was finished.

5.2.3 The network LAB

Before using ICEflow in the Tiscali's Italian backbone, the tool has been tested in a laboratory network created in the ICELab (Information & Communication Engineering Lab) located within the Tiscali's headquarter.

The network is composed by 5 different models of Cisco router subdivided in 3 virtual Autonomous Systems, connected as shown in the logical architecture of Figure 5.5.

In order to simulate the behaviour of a real network and to generate different types of traffic, 10 virtual LAN have been created. The routers are connected each other with a Fast Ethernet @ 100Mbps and the link between the Cisco 5300 and the Cisco 3600 has been realized with a point-to-point @ 2Mbps.

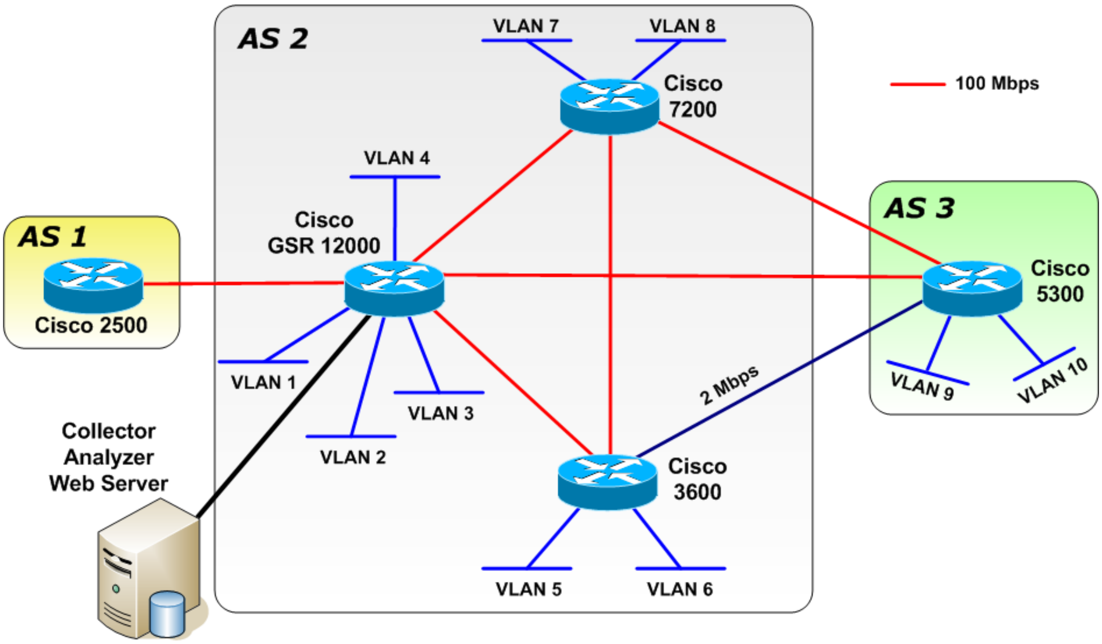


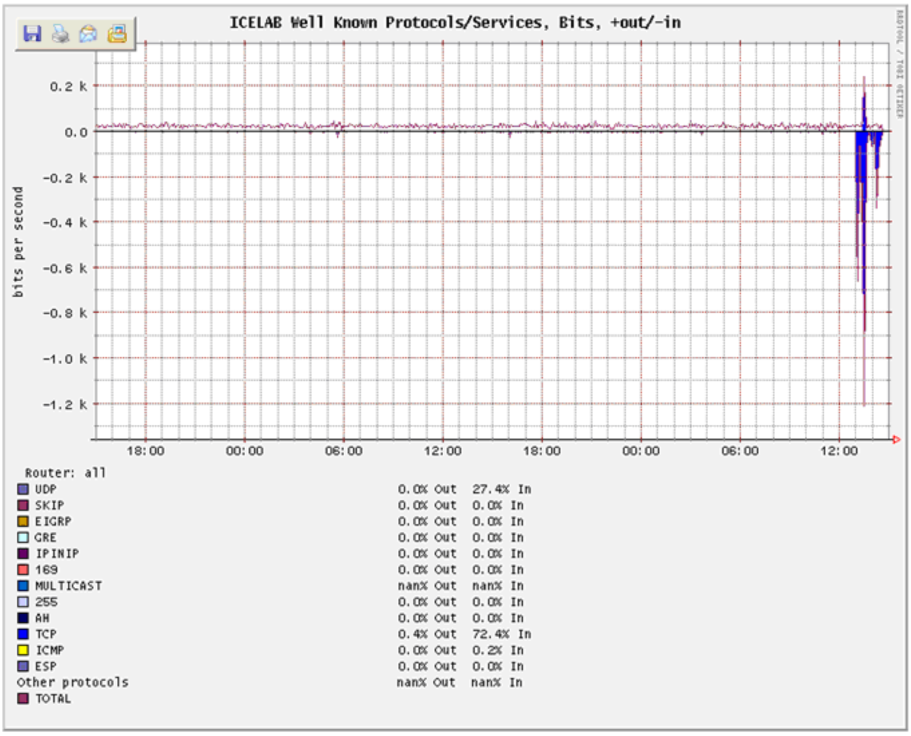
Figure 5.5: Logical topology of the ICELab benchmark network

The NetFlow collector, the analyzer, and the Web Server have been condensed in a single workstation that presents the following characteristics:

- **CPUs**: Dual Processor @ 1GHz
- **RAM Memory**: 1 GByte @ 400 MHz

- **Hard Disks**: 4 disks in RAID-0 configuration with capacity of 40 GByte
- **Operating System**: Linux Slackware 10.0
- **Web Server**: Apache HTTP Server Version 2.0
- **NetFlow software**: Flow-Tools, FlowScan, and ICEFlow

During the tests, the routers run NetFlow Version 5 with a sample rate of 1 packet over 10 and traffic generated by different size of ICMP (e.g., the ping command), TCP, and UDP flows have been tested. The collector has received the data sample and the ICEFlow analyzer has correctly interpreted the different types of traffic. A screenshot of the graphical view taken during the test phase is shown in Figure 5.6.



```
Router: all
■ UDP                                  0.0% Out   27.4% In
■ SKIP                                 0.0% Out   0.0% In
■ EIGRP                                0.0% Out   0.0% In
□ GRE                                  0.0% Out   0.0% In
■ IPINIP                               0.0% Out   0.0% In
■ 169                                  0.0% Out   0.0% In
■ MULTICAST                            nan% Out   nan% In
□ 255                                  0.0% Out   0.0% In
■ AH                                   0.0% Out   0.0% In
■ TCP                                  0.4% Out   72.4% In
■ ICMP                                 0.0% Out   0.2% In
■ ESP                                  0.0% Out   0.0% In
Other protocols                        nan% Out   nan% In
■ TOTAL
```

Figure 5.6: ICEFlow screenshot of the traffic flowing in test network

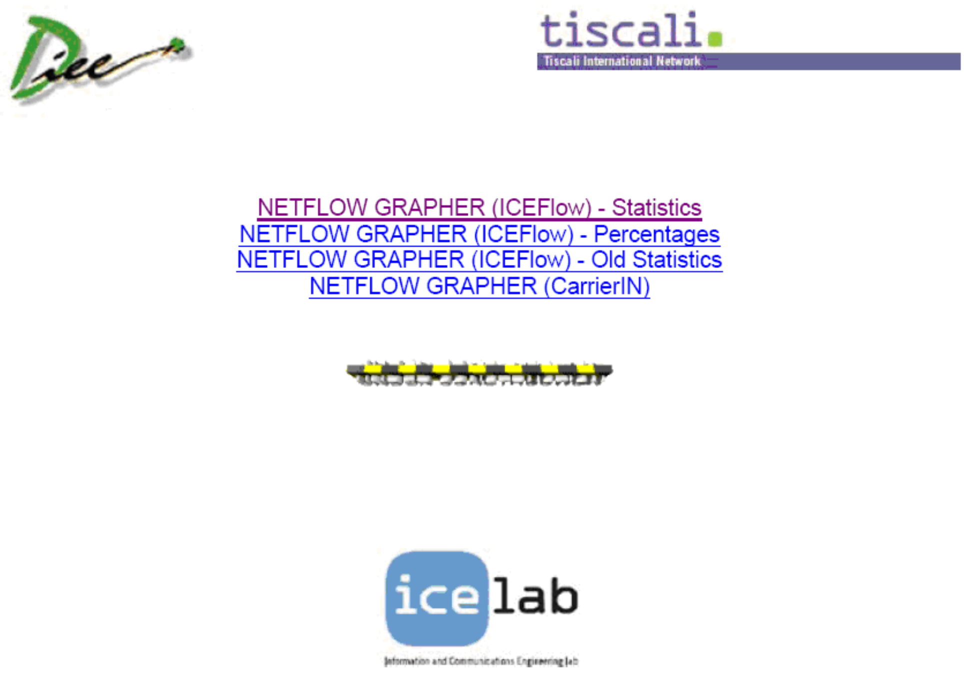
5.2.4 Monitoring the Italian Backbone

The successful tests of ICEFlow in monitoring the benchmark network, pushed us to gradually integrate the tool in the real backbone network.

In several backbone core routers (Cisco GSR 12000), NetFlow version 5 export has been activated with a sampling rate of 1 packet over 10000, and the data have been exported to the Workstation used in the benchmark phase, with a time rate of 30 seconds.

Files containing the traffic data of 5 minutes have been stored and, using the Round Robin Database and the configuration of ICEFlow, the valuable information have been encased in predefined folders.

In the following a series of screenshots taken from the ICEFlow visualization tool will be shown. Figure 5.7 shows the main page of the ICEFlow tool, where it is possible to select between statistics expressed in amount of traffic (Mbps) or as percentages.

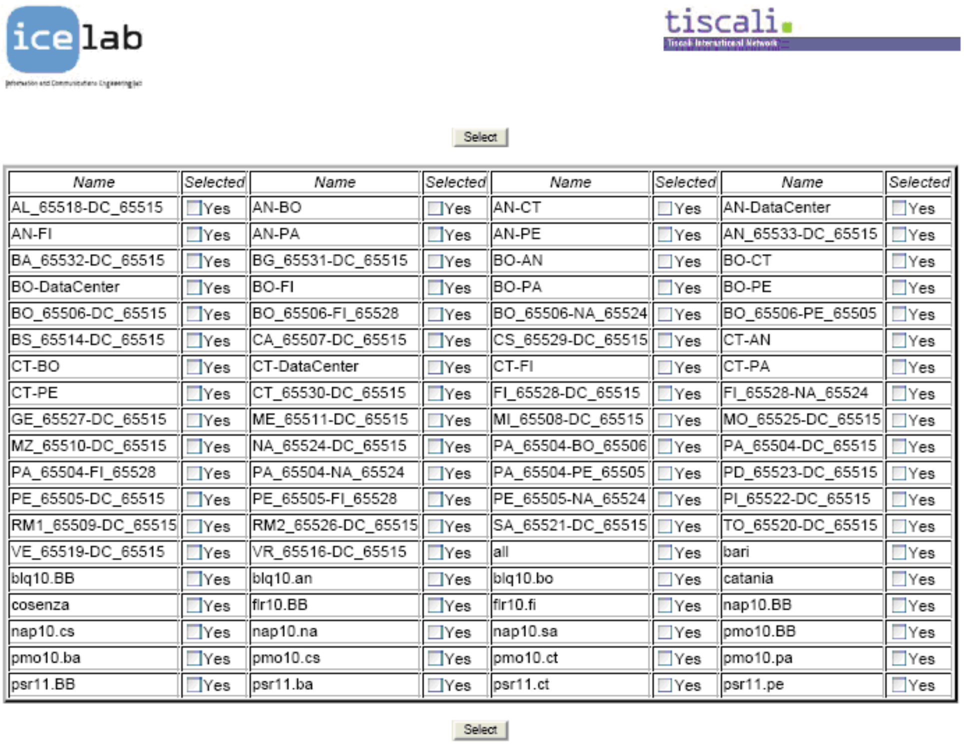


NETFLOW GRAPHER (ICEFlow) - Statistics
NETFLOW GRAPHER (ICEFlow) - Percentages
NETFLOW GRAPHER (ICEFlow) - Old Statistics
NETFLOW GRAPHER (CarrierIN)

Figure 5.7: ICEFlow index page

It is also present the possibility to recall old statistics and to analyze the traffic with a different tool (CarrierIN), in order to validate the correct meaning of the obtained results.

Selecting the first option (the others conduct to a similar page) it is possible to choose the desired flows/directions to be monitored, as shown in Figure 5.8. Multiple choices are allowed.

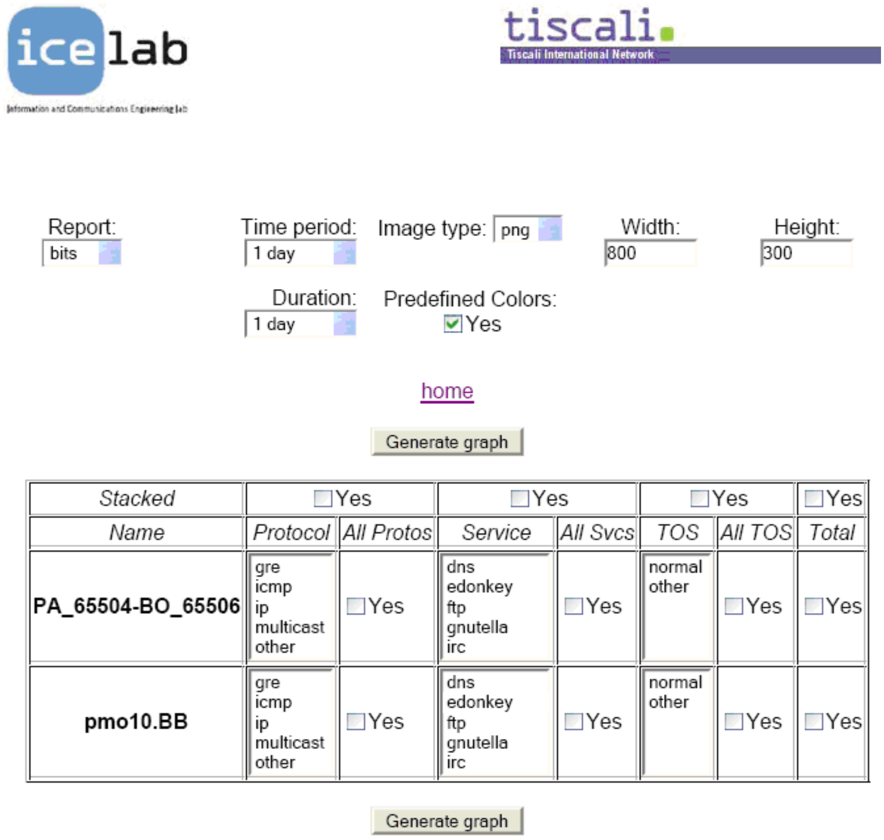


Name	Selected	Name	Selected	Name	Selected	Name	Selected
AL_65518-DC_65515	☐Yes	AN-BO	☐Yes	AN-CT	☐Yes	AN-DataCenter	☐Yes
AN-FI	☐Yes	AN-PA	☐Yes	AN-PE	☐Yes	AN_65533-DC_65515	☐Yes
BA_65532-DC_65515	☐Yes	BG_65531-DC_65515	☐Yes	BO-AN	☐Yes	BO-CT	☐Yes
BO-DataCenter	☐Yes	BO-FI	☐Yes	BO-PA	☐Yes	BO-PE	☐Yes
BO_65506-DC_65515	☐Yes	BO_65506-FI_65528	☐Yes	BO_65506-NA_65524	☐Yes	BO_65506-PE_65505	☐Yes
BS_65514-DC_65515	☐Yes	CA_65507-DC_65515	☐Yes	CS_65529-DC_65515	☐Yes	CT-AN	☐Yes
CT-BO	☐Yes	CT-DataCenter	☐Yes	CT-FI	☐Yes	CT-PA	☐Yes
CT-PE	☐Yes	CT_65530-DC_65515	☐Yes	FI_65528-DC_65515	☐Yes	FI_65528-NA_65524	☐Yes
GE_65527-DC_65515	☐Yes	ME_65511-DC_65515	☐Yes	MI_65508-DC_65515	☐Yes	MO_65525-DC_65515	☐Yes
MZ_65510-DC_65515	☐Yes	NA_65524-DC_65515	☐Yes	PA_65504-BO_65506	☐Yes	PA_65504-DC_65515	☐Yes
PA_65504-FI_65528	☐Yes	PA_65504-NA_65524	☐Yes	PA_65504-PE_65505	☐Yes	PD_65523-DC_65515	☐Yes
PE_65505-DC_65515	☐Yes	PE_65505-FI_65528	☐Yes	PE_65505-NA_65524	☐Yes	PI_65522-DC_65515	☐Yes
RM1_65509-DC_65515	☐Yes	RM2_65526-DC_65515	☐Yes	SA_65521-DC_65515	☐Yes	TO_65520-DC_65515	☐Yes
VE_65519-DC_65515	☐Yes	VR_65516-DC_65515	☐Yes	all	☐Yes	bari	☐Yes
blq10.BB	☐Yes	blq10.an	☐Yes	blq10.bo	☐Yes	catania	☐Yes
cosenza	☐Yes	flr10.BB	☐Yes	flr10.fi	☐Yes	nap10.BB	☐Yes
nap10.cs	☐Yes	nap10.na	☐Yes	nap10.sa	☐Yes	pmo10.BB	☐Yes
pmo10.ba	☐Yes	pmo10.cs	☐Yes	pmo10.ct	☐Yes	pmo10.pa	☐Yes
psr11.BB	☐Yes	psr11.ba	☐Yes	psr11.ct	☐Yes	psr11.pe	☐Yes

Figure 5.8: Selecting the desired direction to be monitored

The next page, depicted in Figure 5.9, permits the selection of several options:

- The type of report (bits, packets, flows)
- The time period and duration (by default one day)
- The format of the graphic (PNG or GIF)
- The size/resolution of the graphic
- The selection of protocols, applications, type of services, and/or the total amount of traffic



Report: bits

Time period: 1 day

Image type: png

Width: 800

Height: 300

Duration: 1 day

Predefined Colors: ☑ Yes

home

Generate graph

Stacked	☐ Yes		☐ Yes		☐ Yes		☐ Yes
Name	Protocol	All Protos	Service	All Svcs	TOS	All TOS	Total
PA_65504-BO_65506	gre icmp ip multicast other	☐ Yes	dns edonkey ftp gnutella irc	☐ Yes	normal other	☐ Yes	☐ Yes
pmo10.BB	gre icmp ip multicast other	☐ Yes	dns edonkey ftp gnutella irc	☐ Yes	normal other	☐ Yes	☐ Yes

Generate graph

Figure 5.9: Selecting the desired options

Finally, Figure 5.10 shows a graphical example of the traffic exchanged between the node of Palermo and the Internet, limited to peer-to-peer application and web destination (HTTP) traffic.

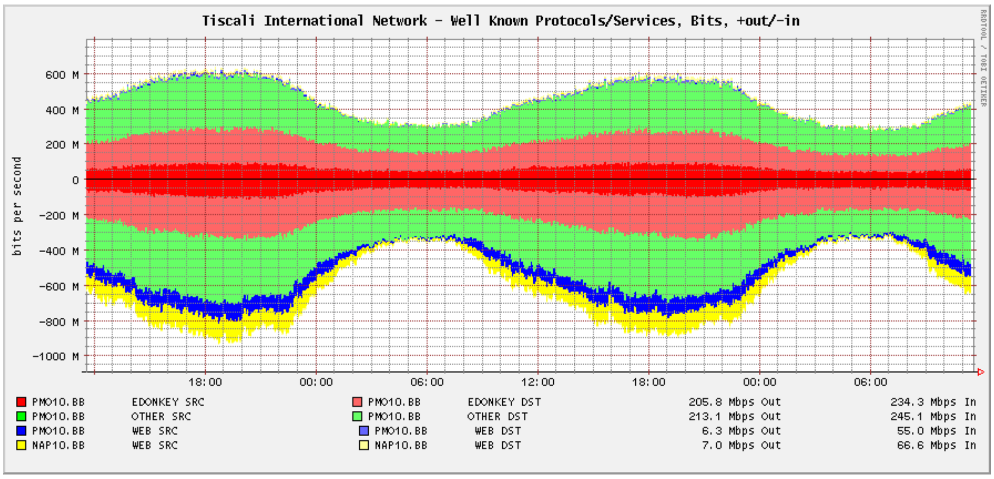


Figure 5.10: Graphical view of the traffic

Chapter 6

LP Optimization Models

This chapter introduces the basic concepts of Linear Programming Optimization, the flow optimization problems, and their extension to Multicommodity flow problems in order to formulate three optimization models for the traffic flow in a backbone telecommunication network.

6.1 Introduction

A specific class of mathematical problems, where a linear function has to be minimized (or maximized), and subject to given linear constraints, is called the class of Linear Programming (LP) problems. A Linear Programming is a problem that can be expressed, in its Standard Form, as follows:

$$\min \quad cx \tag{6.1}$$

$$\text{subject to} \quad Ax = b \tag{6.2}$$

$$x \geq 0 \tag{6.3}$$

where x is the vector of unknown variables, A is a $m \times n$ matrix of known coefficients (with m representing the number of constraints and n being the number of variables), and c and b are vectors of known coefficients.

The expression cx represents the objective function, while the equations $Ax = b$ are called the constraints. The matrix A is generally not square, but has more columns than rows ($n > m$, underdetermined), leaving great latitude (degrees of freedom) in the choice of x. In geometric terms, it can be shown that the optimal value of cx lies on the boundary of the polytope defined by the intersection of the hyperplanes $Ax = b$.

Two families of solution techniques are widely used today, both visiting a progressively improving series of trial solutions, until a solution is reached that satisfies the optimal conditions.

The **Simplex algorithm**, devised by George B. Dantzig in 1947 [31], is an algebraic, iterative method that identifies an initial “basic solution” (called the Corner Point) and then systematically moves to an adjacent basic solution improving the value of the objective function. The algorithm starts and remains on the boundary of the polytope, searching for an optimal point.
Conceptually the method may be outlined in 5 steps:

1. Determine a starting basic feasible solution setting $(n - m)$ non-basic variables to zero.
2. Select an “entering” variable from the non-basic variables, which gives the maximum improvement of the value of the objective function. If none exists then the optimal solution has been found.
3. Select a “leaving” variable from the current basic variables and set it to zero (such variable becomes non-basic)
4. Make the “entering” variable a basic variable and determine the new basic solution
5. Return to Step 1.

The *Interior-point* method, by contrast, visits points within the interior of the feasible region. This method derives from techniques developed in the 1960s by Fiacco and McCormick for nonlinear programming [40], but its application to linear programming dates back only to Karmarkar in 1984 [46].
Karmarkar remarked the fact that moving through the interior of the feasible region of a linear programming problem, using the negative of the gradient of the objective function as the movement direction, may trap the search into corners of the polytope. In order to avoid this “jamming”, the negative gradient is balanced

with a particular "centering" direction that, in the Karmarkar version of the algorithm, is based on the concept of analytic center.

The related problem of Integer Linear Programming (ILP) requires some or all the variables to be integer values. A subset of the ILP is defined when the variables x assume boolean values $[0, 1]$. These programs have often the advantage of matching better the "reality" than LPs, but they are much harder to solve. The most widely used techniques for solving integer programming problems use the solutions of a series of Linear Programs searching for integer optimal solutions.

Linear and integer programming proved to be valuable for modelling many and diverse types of problems in planning, routing, scheduling, assignment, and design. Furthermore, LP and its extensions have been successfully applied to transportation, energy, telecommunications, and manufacturing.

6.2 Graphs and Network Flows

Most of the optimization problems can be modelled by means of a graph [25]. This structure often directly derives from the intrinsic nature of the problem (e.g., the urban transport network, power system, hydric pipelines), while in other cases it arises from the model. Generally, in Operations Research, the term "network" denotes a weighted graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ where the "weights" are numeric values associated to nodes and/or arcs of the graph.
More precisely:

- At each node a real value is associated, which may be
    - o a positive value, representing the amount of "good" (or commodity) exiting from the node (surplus);
    - o a negative value, which represents the amount of commodity entering in the node (deficit);
    - o a zero value; in such case the node is called transit node.
- At each arc the following features are associated

- o a cost denoting the “price” to be paid by a single commodity unit to traverse the arc;
- o an upper and a lower capacity constraint denoting the maximum and the minimum amount of commodity that can be carried by the arc.

A graph can be undirected if its arcs have no direction (edges or lines), or it can be a directed graph if each of its arcs is directed from a node x to a node y (arcs, directed edges, or arrows). In this case y is called the head and x is called the tail of the edge; y is said to be a direct successor of x, and x is said to be a direct predecessor of y.

It is useful to define the concept of Backward Star and Forward Star. The former represents the set of edges entering in a node, while the latter is the set of edges outgoing from a node.
The incidence matrix E for directed graphs is a $[n \times m]$ matrix, where n and m are the number of nodes and arcs respectively, such that:

$$E_{ij} = \begin{cases} -1 & \text{if the arc } a_j \text{ leaves the node } n_i \\ 1 & \text{if the arc } a_j \text{ enters the node } n_i \\ 0 & \text{otherwise} \end{cases}$$

It is possible to define three different types of problem over a graph, which are:

1. the minimum cost flow problem
2. the maximum flow problem
3. the shortest path problem

6.2.1 The Minimum Cost Flow Problem

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a directed graph and $x \in \mathcal{R}^m$ be a vector, where m is the number of edges. x is said to be a flow for $\mathcal{G}$ if it verifies the flow conservation equation constraint:

$$\sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} = b_i \qquad i \in N \tag{6.4}$$

Where BS and FS represent the Backward Star and the Forward Star respectively. The matrix formulation of this equation can be directly derived using the definition of Incidence Matrix:

$$Ex = b \tag{6.5}$$

A flow x_{ij} is feasible if it verifies the capacity constraints:

$$l_{ij} \leq x_{ij} \leq u_{ij} \tag{6.6}$$

where l_{ij} and u_{ij} are the lower and upper values of capacity for arc (i, j). The lower bound is often set to zero.
The objective function can be written as:

$$cx = \sum_{(i,j)\in A} c_{ij} \cdot x_{ij} \tag{6.7}$$

Equations 6.7, 6.5 and 6.6 define the Minimum Cost Flow Problem (MCF) as:

$$\begin{aligned} \min \quad & cx & \qquad (6.8) \\ & Ex = b & \qquad (6.9) \\ 0 \leq \quad & x \leq u & \qquad (6.10) \end{aligned}$$

or, in its extended formulation:

$$\begin{aligned} \min \sum_{(i,j)\in A} c_{ij} \cdot x_{ij} & & \qquad (6.11) \\ \sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} = b_i & \qquad i \in N & \qquad (6.12) \\ 0 \leq x_{ij} \leq u_{ij} & \qquad (i, j) \in A & \qquad (6.13) \end{aligned}$$

The MCF problem can be easily solved using the classic Linear Programming techniques. However, as shown in the next sections, the MCF can represent either a Maximum Flow problem or a Shortest Path problem allowing the use of the specific algorithms developed for those classes of problems.

6.2.2 The Maximum Flow Problem

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a directed graph, $u = [u_{ij}]$ be a vector, representing the upper capacity of the arcs, and s and t be two distinct nodes, which are source and destination of a flow, respectively.
The Maximum Flow Problem consists on determining the maximum flow that can be sent from source s to destination t.
A mathematical formulation of such class of problems is:

$$\max \quad v \tag{6.14}$$

$$\sum_{(j,s)\in BS(s)} x_{js} - \sum_{(s,j)\in FS(s)} x_{sj} + v = 0 \tag{6.15}$$

$$\sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} = 0 \qquad i \in N \backslash \{s,t\} \tag{6.16}$$

$$\sum_{(j,t)\in BS(t)} x_{jt} - \sum_{(t,j)\in FS(t)} x_{tj} - v = 0 \tag{6.17}$$

$$0 \leq x_{ij} \leq u_{ij} \qquad (i,j) \in A \tag{6.18}$$

where BS and FS are the Backward and the Froward Star respectively, and v is the flow. Equations 6.15 and 6.17, and 6.16 represent the flow balancing equations at source node s, destination node t, and in a genreic transit node i, respectively.

In order to solve this kind of problems many methods have been developed and, the most important are listed in Table 6.1.

Algorithm	**Complexity**
Ford - Fulkerson [42]	$O(A \cdot \mathit{maxflow})$
Edmonds - Karp [37]	$O(N \cdot A)$
Relabel-to-front [30]	$O(A^3)$

Table 6.1: Algorithms to solve the Max Flow Problem

As aforementioned, the Maximum Flow problem can be seen as a Minimum Cost Flow problem if a “virtual” arc (called the return arc) is added to the graph that goes from the destination t to the source s with a flow value of v.

6.2.3 The Shortest Path Problem

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a directed and weighted graph, where a cost c_{ij} is associated at each arc (i, j). For each path P in $\mathcal{G}$, the total cost $C(P)$ is the sum of the costs of the arcs constituting P:

$$C(P) = \sum_{(i,j)\in P} c_{ij} \tag{6.19}$$

Let us consider two nodes r and t, and let $\mathcal{P}$ be the set of paths connecting r to t, then the corresponding shortest path problem is defined as:

$$\min\{C(P) : P \in \mathcal{P}\} \tag{6.20}$$

It is possible to formulate such problem as a Minimum Cost Flow Problem, with arc capacity $= +\infty$ and the costs taken from the original shortest path problem. Furthermore, the source node r sends a single unit of flow received from the destination node t.

$$\min \quad cx \tag{6.21}$$

$$Ex \quad = b \tag{6.22}$$

$$x \quad \geq 0 \qquad x \text{ integer} \tag{6.23}$$

where E is the incidence matrix, and

$$b_i = \begin{cases} -1 & \text{if } i = r \\ 1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases}$$

The described problem (single-source shortest path) can be extended to find the shortest paths for every pair of nodes in the network (all-pairs shortest path problem).

The most important algorithms for solving this problem are listed in Table 6.2.

The shortest path algorithms are used in telecommunication in order to find the best route for the traffic flow. As mentioned in chapter 2, the Dijkstra's algorithm is applied when the IS-IS or the OSPF routing protocol is used, while the Bellman-Ford's algorithm is used in distance-vector routing protocols such as

Algorithm	**Complexity**
Dijkstra [32]	$O(N^2)$
Bellman-Ford [27]	$O(N \cdot A)$
A^* Search [1]	Polynomial
Floyd-Warshall [41]	$O(N^3)$
Johnson [50]	$O(N^2 \log(N) + N \cdot A)$

Table 6.2: Algorithms to solve the Shortest Path problem

the Routing Information Protocol (RIP). Moreover, the shortest path problem can be applied in a huge number of applications [25], such as:

- Robot navigation
- Urban traffic planning
- Optimal pipelining of microelectronic chips
- And many others

6.3 Multicommodity Flow Problems

Sometimes, the networks are dedicated to the transport of a single commodity (e.g., water). More often, the edge capacities are shared by different flows representing multiple commodities, where one commodity will always remain the same without any transformation in a different commodity (e.g., an apple will not ever become a pear!).

In mathematical terms this means that, at each vertex, each commodity has its own flow conservation constraint, and the total flow through each arc cannot exceed the maximum capacity.

Important examples of multicommodity flow problems arise in transportation, manufacturing networks, and telecommunication, where a separate commodity per class of traffic and origin/destination pair can be identified. In a multicommodity flow problem, either a Min-Cost flow or a Max Flow problem, each commodity has its own single or multiple sources s_i and its own single or multiple destinations t_i.

In the integer multicommodity flow problem, the capacities and flows are restricted to be integers. Unlike the single commodity flow problem, for problems with integral capacities and demands, the existence of a feasible fractional solution to the multicommodity flow problem does not guarantee a feasible integral solution.

In typical telecommunication systems model an extra constraint, which may be imposed, is to restrict each commodity to be sent along a single path or to be split in equal parts and sent along multiple equivalent paths as seen in chapter 2. Moreover, adding a further constraint representing the survivability of the network in case of failure makes the model NP-hard.

In that case, a new formulation is needed, and the present work is focused on the development of Multicommodity Min-Cost Flow models that match the problem statement introduced in chapter 3 and that reduce the complexity.

6.3.1 Multicommodity Min-Cost Flow

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a directed graph with n nodes and m arcs, and $x = [x_1, \cdots, x_k]$ be a vector, representing the multicommodity flow, of k distinct flow vectors [45].

A linear Multicommodity Min-Cost Flow problem can be formalized as an extension of the single commodity formulation as follows:

$$\min \sum_{h \in K} \sum_{(i,j) \in A} c_{ij}^h x_{ij}^h \tag{6.24}$$

$$\sum_{j:(i,j) \in A} x_{ij}^h - \sum_{j:(j,i) \in A} x_{ji}^h = b_i^h \qquad \forall i \in N, \forall h \in K \tag{6.25}$$

$$0 \leq x_{ij}^h \leq u_{ij}^h \qquad \forall (i,j) \in A, \forall h \in K \tag{6.26}$$

$$\sum_{h \in K} x_{ij}^h \leq u_{ij} \qquad \forall (i,j) \in A \tag{6.27}$$

with $K = \{1, \cdots, k\}$.

This representation is called *node-arc formulation* and describes an LP problem; equation 6.24 states that the k commodities have to be “routed” over the graph $\mathcal{G}$ with the minimal cost (objective function to be minimized), respecting the constraints declared by the next two equations defined per single commodity, which are the flow conservation 6.25 , the capacity constraint 6.26, and the last equation 6.27 indicating the mutual capacity constraint stating that the total sum of all commodities flowing through arc (i, j) have to be less than its maximum total capacity.

Relaxing these last constraints, it is possible to formulate the problem in a different way simply noting that the k resulting problems are Shortest Path problems.
This new formulation is called the *arc-path formulation*:

$$\min \sum_{p \in P} c_p x_p \tag{6.28}$$

$$\sum_{p \in P(h)} x_f = d_h \qquad \forall h \in K \tag{6.29}$$

$$\sum_{p:(i,j) \in f} x_p \leq u_{ij} \qquad \forall (i,j) \in A \tag{6.30}$$

$$x_p \geq 0 \qquad \forall p \in P \tag{6.31}$$

where $P(h)$ is the set of possible paths for each commodity h and $P = \cup_h P(h)$. The two formulations are basically equivalent and the choice of one of them is strictly dependent on the problem at hand.

6.4 LP models

In the following, three LP models that aim to minimize the maximum traffic load in a telecommunication network and, in the meantime, to avoid congestion in case of failure (single failure is considered, as it is the most probable) are presented. These models are structured in two layers: the first one, common to all models, solves the well-known “General Routing Problem” (GRP) [43], which is basically a Multicommodity Min-Cost Flow problem, using the IS-IS routing protocol and a complementary set of LSP tunnels.

The second layer introduces the survivability constraints, which are expressed in different ways, depending on the used restoration technique. Furthermore, these constraints are presented as capacity constraints in order to guarantee a 100% survivability level avoiding possible congestion scenarios.

For each scenario, the occupation level of all links is evaluated and the maximum is considered as objective function to be minimized. This situation represents an upper bound of the global optimization process and a better objective function can be defined.

As shown in chapter 3, an optimal or at least sub-optimal configuration of the IS-IS metrics is the underlying condition for all the models. Such a set can be found using metaheuristic approaches presented in literature [43], [44],and [63].

6.4.1 Model 1

The basic idea of the first model is to find an optimal set of complementary explicit LSP tunnels, to be used in combination with conventional IS-IS routing protocol, in order to reduce the occupation level of the network under single link failure condition.

The link restoration technique is used and the backup paths are realized using implicit LSPs. It is advisable to recall that, for an explicit LSP the used links have to be specified by the network operator, while an implicit LSP automatically determines its path using the IS-IS metrics.

In terms of network management, this model specifies the following steps:

1. Setup of the (sub)optimal IS-IS routing metrics previously calculated.
2. Setup of the explicit LSPs determined by the model.
3. No intervention of the network operators is required **to setup the restoration configuration**.

The model is presented with the arc-path formulation as well as with the node-arc formulation.

6.4.1.1 Arc-path Formulation

Notation

Data

$\mathcal{N}$ - Node set

$\mathcal{A}$ - Edge set

$\mathcal{F}$ - Commodity set

c_{ij} - Capacity associated with link (i, j)

$\mathcal{P}(h)$ - Set of edges which belong to path h

$\mathcal{P}(f)$ - Set of paths for commodity f

d^f - Effective bit rate of flow f

x^f_{ij} - Share of flow f carried by IS-IS and traversing link (i, j)

Variables

u_{max}	Maximum utilization in the network - to be minimized
is^f	Share of flow f carried by ISIS
p_h	Share of flow f which uses path h with MPLS technology

The General Routing Problem can be specified as follows:

$$z = \min(u_{max}) \tag{6.32}$$

$$\sum_{f\in\mathcal{F}} is^f \cdot x^f_{ij} + \sum_{h:(i,j)\in\mathcal{P}(h)} p_h \leq u_{max} \cdot c_{ij} \quad \forall (i, j) \in \mathcal{A} \tag{6.33}$$

$$\sum_{h\in\mathcal{P}(f)} p_h = d^f - is^f \quad \forall f \in \mathcal{F} \tag{6.34}$$

$$p_h \geq 0 \quad \forall h \in \mathcal{P}(f) \tag{6.35}$$

$$is^f \in [0, d^f] \qquad \forall f \in \mathcal{F} \tag{6.36}$$

Equation 6.32 is the objective function to be minimized, while the left hand side (LHS) of capacity constraints in equation 6.33, specifies the total amount of traffic traversing link (i, j) and it is composed by the traffic routed by IS-IS (first sum) and by the LSP-carried traffic. Obviously, for each link (i, j) this value has to be smaller than variable umax times the link capacity c_{ij}.

Equation 6.34 determines the amount of traffic carried by MPLS and represents a sort of flow conservation defined for each commodity instead of node.

It is important to pay attention to constraint 6.36 because it states that variable is^f is a real variable making the model entirely linear and, as shown in the previous sections and subsections, a (pure) linear program is "easy" to solve.

In fact, changing the definition of variable is^f, for example considering a "binary" variable $\{0, d_f\}$ by specifying that every commodity f is carried only by IS-IS or only by MPLS, the problem becomes a Mixed Integer Program (MIP) because it integrates linear constraints and linear objective function with an integer constraint. MIPs are known to be harder to solve with respect to LPs.

The solution of GRP gives the optimal distribution of LSPs that minimizes the maximum bandwidth occupation and this is also an important result because it represents a lower bound during the optimization process of IS-IS metrics [43], in normal working conditions.

Survivability Constraints

The survivability constraints introduced in the model are formalized as follows:

$$\sum_{f \in \mathcal{F}} is^f \cdot x_{ij}^{f,l} + \sum_{h:(i,j) \in \mathcal{P}(h)} p_h + x_{ij}^l \left(\sum_{h:l \in \mathcal{P}(h)} p_h \right) \leq u_{max} \cdot c_{ij} \quad \forall (i,j),\ l \in \mathcal{A} \tag{6.37}$$

Where the constant x_{ij}^l specifies the share of MPLS traffic, flowing along link l that, in case of failure of such link, is rerouted by IS-IS along link (i, j), while $x_{ij}^{f,l}$ is the share of flow f carried by IS-IS and traversing link (i, j), when the link l fails.

Note that, if $x_{ij}^{f,l}$ is less than x_{ij}^l the constraint specified by equation 6.33 is redundant.

6.4.1.2 Node-arc Formulation

The node-arc formulation of the first model makes use of a different set of data and its MPLS flow variable is not anymore a path but a flow traversing a specific link. This formulation has been preferred during the tests phase because it presents a better algebraic structure.

Data

$\mathcal{N}$ - Node set

$\mathcal{A}$ - Edge set

$\mathcal{F}$ - Commodity set

c_{ij} - Capacity associated with link (i, j)

d^f - Effective bit rate of flow f

x_{ij}^f - Share of flow f carried by ISIS and traversing link (i, j)

Variables

u_{max}	Maximum utilization in the network - to be minimized
is^f	Share of flow f carried by ISIS
$flow_{ij}$	Share of flow f carried by MPLS and traversing link (i, j)

The mathematical model for the GRP using the node-arc formulation, is formalized as follows:

$$z = \min(u_{max}) \tag{6.38}$$

$$\sum_{f\in\mathcal{F}} x_{ij}^f \cdot is^f + \sum_{f\in\mathcal{F}} flow_{ij}^f \le u_{max}\cdot c_{ij} \quad \forall (i,j)\in\mathcal{A} \tag{6.39}$$

$$\sum_{j:(j,i)\in\mathcal{A}} flow_{ji}^f - \sum_{j:(i,j)\in\mathcal{A}} flow_{ij}^f = \begin{cases} -d^f + is^f & \forall i\in\mathcal{N}: i = I(f) \\ d^f - is^f & \forall i\in\mathcal{N}: i = E(f) \\ 0 & \text{otherwise} \end{cases} \tag{6.40}$$

$$flow_{ij}^f \ge 0 \quad \forall (i,j)\in\mathcal{A},\ \forall f\in\mathcal{F} \tag{6.41}$$

$$is^f \in [0, d^f] \quad \forall f\in\mathcal{F} \tag{6.42}$$

While the objective function 6.38 remains unaltered, the capacity constraints 6.39 and the flow conservation equations 6.40 become different. While the former is really close to 6.33 because it is defined for each arc and only the MPLS "variable" has changed, the latter assumes a brand new form, being defined for each node.

Commodity per source aggregation

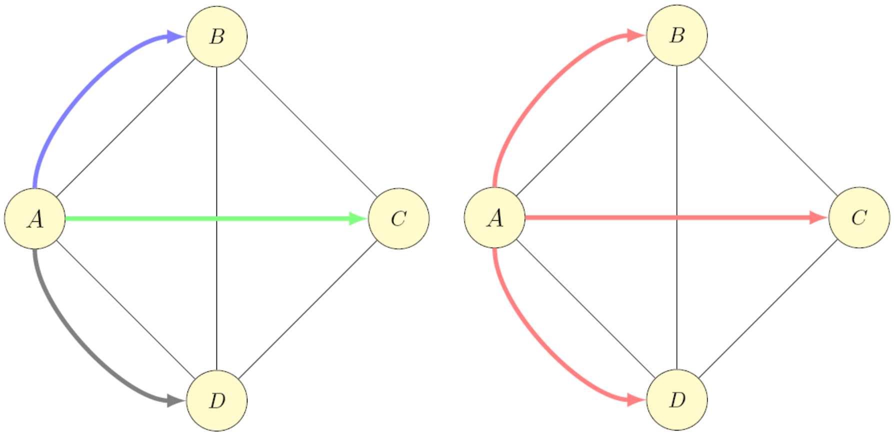
In order to reduce the number of variables and to make the model easier to solve, the commodities have been grouped based on their source node as follows:

$$flow_{ij}^h = \sum_{f:I(f)=h} flow_{ij}^f \tag{6.43}$$

Where $I(f)$ identifies the ingress node for commodity f.

Let us consider the example in figure 6.1 where the commodities $A \to B$, $A \to C$, and $A \to D$ are replaced by a single commodity "A" which is the sum of them.

Considering $F(h)$ as the set of commodities having as ingress the generic node h,



(a) Disaggregated flows (b) Aggregated flows

Figure 6.1: *Aggregating flows per source node*

it is possible to rewrite the GRP as follows:

$$z = \min(u_{max}) \tag{6.44}$$

$$\sum_{f \in \mathcal{F}} x_{ij}^f \cdot is^f + \sum_{h \in \mathcal{N}} flow_{ij}^h \leq u_{max} \cdot c_{ij} \quad \forall (i,j) \in \mathcal{A} \tag{6.45}$$

$$\sum_{j:(j,i)\in\mathcal{A}} flow_{ji}^h - \sum_{j:(i,j)\in\mathcal{A}} flow_{ij}^h = \begin{cases} -\sum_{f \in F(h)} d^f + is^f & i = h \\ d^f - is^f \quad if\ i \neq h, i = E(f), f \in F(h) \\ 0 \quad \text{otherwise} \end{cases} \tag{6.46}$$

$$flow_{ij}^h \geq 0 \quad \forall (i,j) \in \mathcal{A},\ \forall h \in \mathcal{N} \tag{6.47}$$

$$is^f \in [0, d^f] \quad \forall f \in \mathcal{F} \tag{6.48}$$

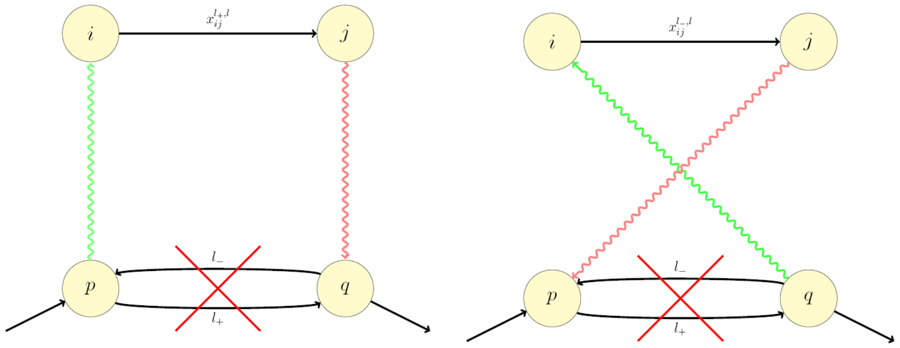
$$\tag{6.49}$$

Survivability Constraints

The survivability constraints with flow aggregation can now be introduced with the example of figure 6.2.

The failure of the undirected edge $l = p - q$ involves two directed arcs $l_+ = (p \rightarrow q)$ and $l_- = (q \rightarrow p)$.

The IS-IS protocol will reroute the corresponding flows affected by the fail through the network and part of them might traverse the link (i,j) as shown in figure



(a) Rerouting flow $p \rightarrow q$ (b) Rerouting flow $q \rightarrow p$

Figure 6.2: *Rerouting flows in case of edge failure*

6.2(a) and 6.2(b).

The survivability constraint on total flow traversing the directed link (i, j), after an event of failure over a generic edge l, is:

$$\sum_{f\in\mathcal{F}} x_{ij}^{f,l}\cdot is^f + \sum_{h\in\mathcal{N}} flow_{ij}^h + \sum_{h\in\mathcal{N}} (x_{ij}^{l_+,l}\cdot flow_{l_+}^h + x_{ij}^{l_-,l}\cdot flow_{l_-}^h) \le u_{max}\cdot c_{ij} \quad \forall (i,j) \neq l_+, l_- \in \mathcal{A} \tag{6.50}$$

In the first term the constant $x_{ij}^{f,l}$ specifies the share of flow f routed by IS-IS along the link (i, j), when the edge l fails (calculated over the new graph obtained removing edge l), while the second term is the flow carried by explicit MPLS LSP along link (i, j).

In the third term, the constants $x_{ij}^{l_+,l}$ and $x_{ij}^{l_-,l}$ specify the share of MPLS traffic flowing through the link l, from $p \rightarrow q$ and from $q \rightarrow p$ respectively, that in case of edge l failed, is rerouted along link (i, j) as shown in figure 6.2(a) and 6.2(b) respectively. As aforementioned, the solution obtained aims to guarantee 100% level of survivability in case of single link failure optimizing, automating, and speeding up the traffic engineering decision process.

Aggregated Flow Decomposition

The solution of the previously presented model gives as result the portion of aggregated per source node flows traversing every link. These flows must be

decomposed in order to find the explicit LSP to be imposed in each router.
For the sake of clarity, in figure 6.3 the solution of the presented model for a benchmark network is shown.

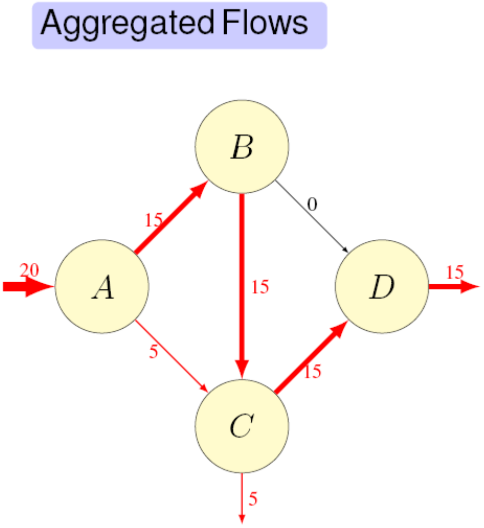
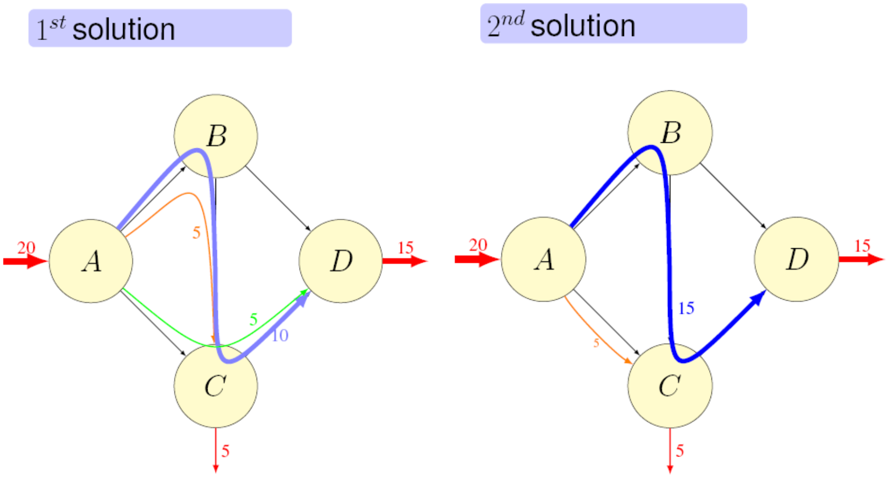


Figure 6.3: Example of solution

At ingress node A the aggregated flow (deficit) is specified, which is equal to the sum of traffic outgoing from nodes C and D. These values are the given commodities directly obtained from the traffic matrix.
Furthermore, for each arc the solution obtained from the model is indicated.
Two possible equivalent solutions of the decomposition algorithm are shown in figures 6.4(a) and 6.4(b).

In the first solution 3 flows have been obtained, while in the second the initial aggregated flow has been decomposed in 2 flows.
In other words, it is possible to obtain multiple equivalent solution while the number of LSP paths to be imposed is significantly different.

Unfortunately, a path decomposition algorithm that furnishes a unique optimal solution for these class of problems does not exist in literature and then it is only possible to obtain multiple solutions, which have to be evaluated in order to fit further explicit requirements of the problem (e.g. minimize the maximum number of LSP).



(a) 1^{st} solution (b) 2^{nd} solution

Figure 6.4: *Flow decomposition solutions*

6.4.2 Model 2

The second model pursues the same objective of the first model using IS-IS and complementary explicit LSPs. However, in this case, the technique used to guarantee the network to be survivable is the **path restoration** and the backup path is realized using implicit LSPs (obtained from the IS-IS metrics).

The data and the variables, as well as the GRP, are essentially identical to those of the first model, while the survivability constraint is quite different.

For completeness the whole model is reported and its arc-path formulation is given.

Notation

Data

$\mathcal{N}$ - Node set

$\mathcal{A}$ - Edge set

$\mathcal{F}$ - Commodity set

c_{ij} - Capacity associated with link (i, j)

$\mathcal{P}(h)$ - Set of edges which belong to path h

$\mathcal{P}(f)$ - Set of paths for commodity f

d^f - Effective bit rate of flow f

x_{ij}^f - Share of flow f carried by ISIS and traversing link (i,j)

Variables

u_{max}	maximum utilization in the network - to be minimized
is^f	Share of flow f carried by ISIS
p_h	Share of flow f which uses path h with MPLS technology

Model for GRP

$$z = \min(u_{max}) \tag{6.51}$$

$$\sum_{f \in \mathcal{F}} is^f \cdot x_{ij}^f + \sum_{h:(i,j) \in \mathcal{P}(h)} p_h \leq u_{max} \cdot c_{ij} \quad \forall (i,j) \in \mathcal{A} \tag{6.52}$$

$$\sum_{h \in \mathcal{P}(f)} p_h = d^f - is^f \quad \forall f \in \mathcal{F} \tag{6.53}$$

$$p_h \geq 0 \quad \forall h \in \mathcal{P}(f) \tag{6.54}$$

$$is^f \in [0, d^f] \qquad \forall f \in \mathcal{F} \tag{6.55}$$

Survivability Constraints

In this case, the survivability constraints are the sum of two distinct terms.
The first term is the part of traffic traversing the link (i,j) when the edge l fails. This flow is composed by the former IS-IS traffic and the MPLS traffic affected by the failure rerouted by IS-IS.
The second term is the MPLS flow unaffected by the failure of the edge l.

$$\sum_{f\in\mathcal{F}} x_{ij}^{f,l}\left(is^f + \sum_{h\in\mathcal{P}(f),\ l\in\mathcal{P}(h)} p_h\right) + \sum_{h:(i,j)\in\mathcal{P}(h),\ l\notin\mathcal{P}(h)} p_h \leq u_{max}\cdot c_{ij} \quad \forall(i,j),\ l\in\mathcal{A} \tag{6.56}$$

Once again, a 100% survivability level is guaranteed.

6.4.3 Model 3

Finally, the third model finds the optimal set of explicit LSP to be used in combination with the (sub)optimal IS-IS metrics configuration, thus the data of the problem, the variables, and the GRP arc-path formulation are unaltered.

Notation

Data

$\mathcal{N}$ - Node set

$\mathcal{A}$ - Edge set

$\mathcal{F}$ - Commodity set

c_{ij} - Capacity associated with link (i,j)

$\mathcal{P}(h)$ - Set of edges which belong to path h

$\mathcal{P}(f)$ - Set of paths for commodity f

d^f - Effective bit rate of flow f

x_{ij}^f - Share of flow f carried by ISIS and traversing link (i,j)

Variables

u_{max}	Maximum utilization in the network - to be minimized
is^f	Share of flow f carried by ISIS
p_h	Share of flow f which uses path h with MPLS technology

Model for GRP

$$z = \min(u_{max}) \tag{6.57}$$

$$\sum_{f\in\mathcal{F}} is^f \cdot x_{ij}^f + \sum_{h:(i,j)\in\mathcal{P}(h)} p_h \leq u_{max} \cdot c_{ij} \quad \forall (i,j) \in \mathcal{A} \tag{6.58}$$

$$\sum_{h\in\mathcal{P}(f)} p_h = d^f - is^f \quad \forall f \in \mathcal{F} \tag{6.59}$$

$$p_h \geq 0 \quad \forall h \in \mathcal{P}(f) \tag{6.60}$$

$$is^f \in [0, d^f] \qquad \forall f \in \mathcal{F} \tag{6.61}$$

Survivability Constraints

In this model the 100% degree of survivability is realized by means of explicit LSPs, found as solution of the model, and implemented in the router configuration as backup secondary paths. Each backup path, represented by a new variable q_h, is common for all the working primary paths obtained for each commodity f.
The recovery scheme used is the path restoration technique.

As can be seen from the equations 6.62, 6.63, 6.64, 6.65, and 6.66 the survivability constraints are quite complex in this case and the model requires particular solving techniques such as Column Generation [62].

$$\sum_{h\in\mathcal{P}(f),\ l\in\mathcal{P}(h)} p_h \leq \sum_{h\in\mathcal{P}(f),\ l\notin\mathcal{P}(h)} q_h \quad \forall f \in \mathcal{F},\ \forall l \in \mathcal{A} \tag{6.62}$$

$$\sum_{f\in\mathcal{F}} is^f \cdot x_{ij}^{f,l} + \sum_{h:(i,j)\in\mathcal{P}(h),\ l\notin\mathcal{P}(h)} (p_h + q_h) \leq u_{max} \cdot c_{ij} \quad \forall (i,j),\ l \in \mathcal{A} \tag{6.63}$$

$$q_h \leq d^{f(h)} \cdot y_h \quad \forall h \in \mathcal{P}(f) \tag{6.64}$$

$$y_h \in \{0,1\} \qquad \forall h \in \mathcal{P}(f) \tag{6.65}$$

$$\sum_{h\in\mathcal{P}(f)} y_h \leq 1 \qquad \forall f \in \mathcal{F} \tag{6.66}$$

Equation 6.62 states that the flow to be rerouted (p_h) has to be smaller than the capacity of the backup path (q_h).
Constraint 6.63 represents the flow conservation equation for arc (i,j). The first term is the IS-IS traffic flowing along the arc when the edge l fails, while the second

term is the MPLS part composed by the primary working path (unaffected by the failure) and the backup path.
Following constraints 6.64 and 6.65 state if the backup path is used or not.

Chapter 7

LP Model Tests & Results

The first optimization model presented in the last chapter, makes use of IS-IS routing protocol in combination with explicit MPLS LSPs in order to find the best traffic flow distribution. The objective function minimization process involves also additional survivability constraints that guarantee the delivery of the traffic in case of single link failure, using the link restoration technique performed by the underlying IS-IS protocol.
In this chapter the results of tests performed over this model are shown and, as benchmark networks, three different topologies have been used.
Two of them are synthetic networks while the third is a real backbone network.
At the beginning of the chapter a short description of the implementation and of the software used to solve the LP model is given.

7.1 Model implementation

The pre-testing phase has required the implementation of a C++ code reproducing the behaviour of the IS-IS routing protocol. The instances of the problem have been prepared using the Jones Lustig format [14] and have been read using a service class for Multicommodity Min Cost Flow solvers called "Graph" [11]. A C++ function has been also implemented, which prepares the matrix representing the constraints of the MMCF problem passing it to an Open source solver called OsiSolver [16]. A further C++ class to decompose the aggregated per source flows has been implemented.

7.2 The highly meshed network

The first meaningful test of the model has been performed over a randomly generated synthetic network composed by $n = 8$ nodes and $m = 36$ directed arcs, each of them having a maximum capacity of 1000 Mbps. The total number of commodities is $n \cdot (n-1) = 56$ flows and the corresponding model involves 345 variables and 422 constraints.
The characteristics described are summarized in Table 7.1, while the logical topology is depicted in Figure 7.1.

Nodes	8
Arcs	36
Flows	56
Variables	345
Constraints	422

Table 7.1: Characteristics of the synthetic network

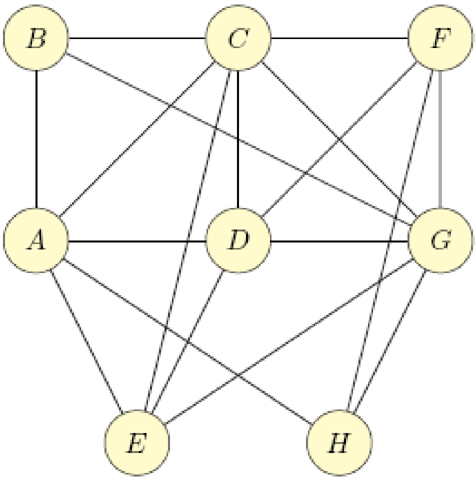


Figure 7.1: Logical topology of the synthetic network

The traffic matrix has been randomly generated aiming to slightly "stress" the network in case of single link failure.
As "default" configuration, the solely IS-IS routing protocol has been considered.

A unit link metrics value has been assigned to all arcs (default configuration suggested by the major router vendors). Note that in that case, the shortest paths are those with the minimum number of hops.

The obtained results have been also compared with those of the IS-IS metrics optimization performed using a metaheuristics algorithm (in particular a Tabu Search algorithm implemented by the research group of University of Cagliari [60]), which considers the failure scenarios in the search process.

The LP model have been applied starting from the conditions above described and, as shown in Table 7.2, the objective function value has been evaluated and compared in both "normal condition" (solution of the GRP) and in "failure condition" (adding the survivability constraint).
The last column of Table 7.2 shows the number of explicit LSPs determined by the model.

Phase	Normal Condition	Failure Condition	# LSP
IS-IS Default Routing	$u_{max} = 34\%$	$u_{max} = 70\%$	0
IS-IS optimization	$u_{max} = 29\%$	$u_{max} = 47\%$	0
MPLS optimization	$u_{max} = 23\%$	$u_{max} = 35\%$	55
Joint IS-IS/MPLS opt.	$u_{max} = 23\%$	$u_{max} = 35\%$	48

Table 7.2: Comparing results for the synthetic network

Results discussion

The obtained results show that the application of the model starting either from the "default" configuration or the "IS-IS optimal configuration", produces the same maximum link utilization. Only the number of LSPs needed changes and this may depend on the fact that (likely) the IS-IS optimization already identifies some best paths.
It is important to note that, in case of failure, the maximum link occupation has been halved with respect to the "default" configuration.

7.3 The IBCN European network

The second benchmark network is part of the Zuse Institute Berlin's (ZIB) SNDlib [22], that is a library of test instances for Survivable fixed telecommunication Network Design.

The problem was originally defined as a capacity planning problem and has been provided by the INTEC Broadband Communication Networks research group (IBCN) which is a research institute founded by the Flemish Government, focusing on information & communication technology (ICT) in general, and applications of broadband technology in particular [12].

During the tests, the links capacity has been fixed and the demands values have been scaled in order to obtain a slight congestion in case of failure when the "default" configuration is used.

The topology contains $n = 37$ nodes widely spread in the European territory and $m = 114$ directed arcs, The characteristics described are summarized in Table 7.3, while the network topology is shown in Figure 7.2.
The traffic matrix is given as part of the instance with $n \cdot (n-1) = 1332$ commodities (flows), thus the resulting problem has 5551 variables and 5813 constraints.

Nodes	37
Arcs	114
Flows	1332
Variables	5551
Constraints	5813

Table 7.3: Characteristics of the IBCN European network

Routing Optimization

As preliminary step, the optimal solution of the GRP has been found using the LP model, with all constraints but the survivability one, and results have been compared with those obtained by the "default" configuration as well as by the IS-IS metrics optimization process.

Figure 7.2: The IBCN network topology

Phase	u_{max}	% Gain (Def)	% Gain (IS-IS)	# LSP
Default	$u_{max} = 70.64\%$	–	–	0
IS-IS opt.	$u_{max} = 53.71\%$	23.71%	–	0
MPLS opt.	$u_{max} = 40.43\%$	42.77%	24.73%	822

Table 7.4: GRP solution for the IBCN network

Table 7.4 shows that, using “default” metrics values, the maximum occupation is already above the 70% of the maximum link capacity. Conversely, using the metrics obtained with the Tabu Search IS-IS metrics optimization, the maximum link occupation reduces to 53.71% with a gain of 23.71%.

The LP model for the GRP finds a LSP distribution that pull down the maximum link utilization to 40.43% of the links capacity. This result corresponds to a “gain” of capacity equal to 42.77% with respect to the “default” configuration and a reduction of occupation of 24.73% with respect to the IS-IS optimization. The only “price to pay” is that 822 LSPs have to be set. Note that such operation can be done in a supervised automatic way.

Graphically, results can be summarized using histograms where in the x-axis the arc is indicated and the y-axis shows the correspondent link utilization.

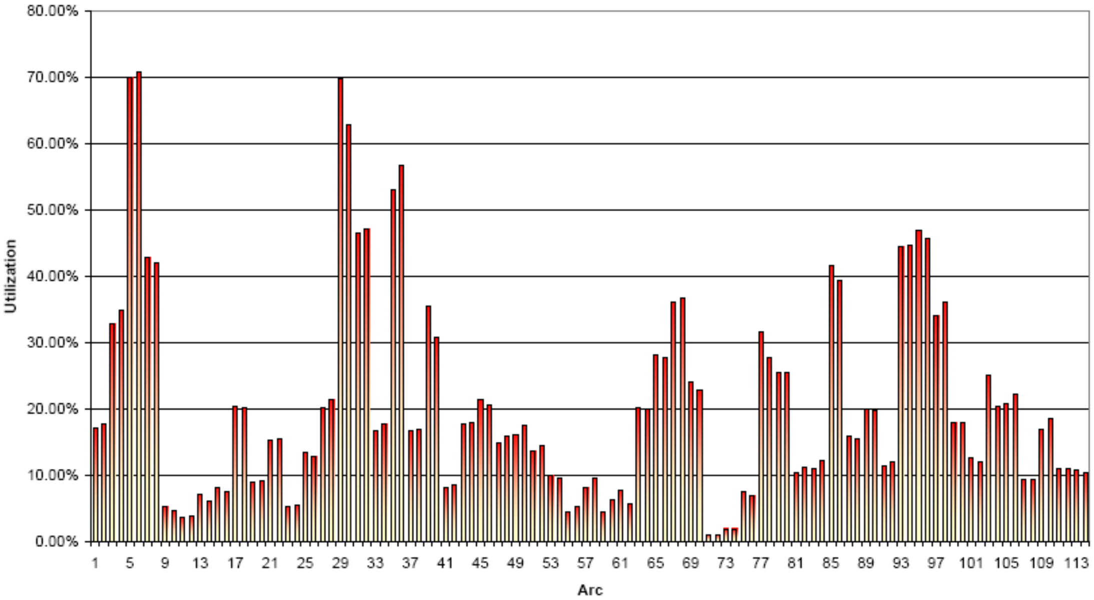


Figure 7.3: Default metrics under normal condition

Observing the histograms it is possible to note that with respect to the initial situation (figure 7.3), the IS-IS optimization process only reduces the peaks of traffic (figure 7.4).

In the other hand, the MPLS optmization technique (figure 7.5) “equalizes” the occupation of the links, by distributing the flows over the whole network, reducing the waste of network resources.

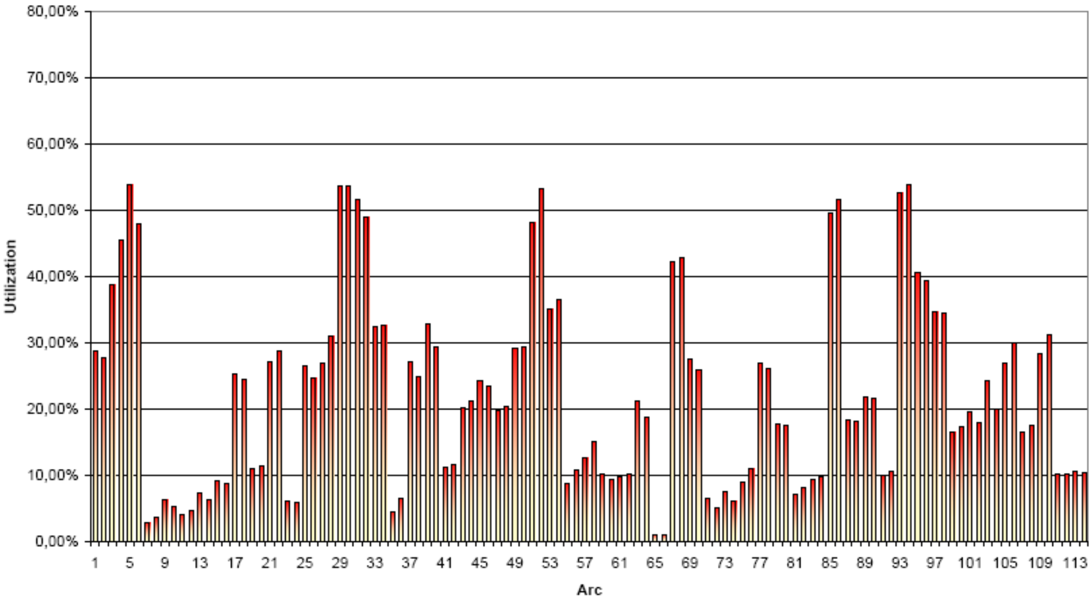


Figure 7.4: IS-IS optimized metrics under normal condition

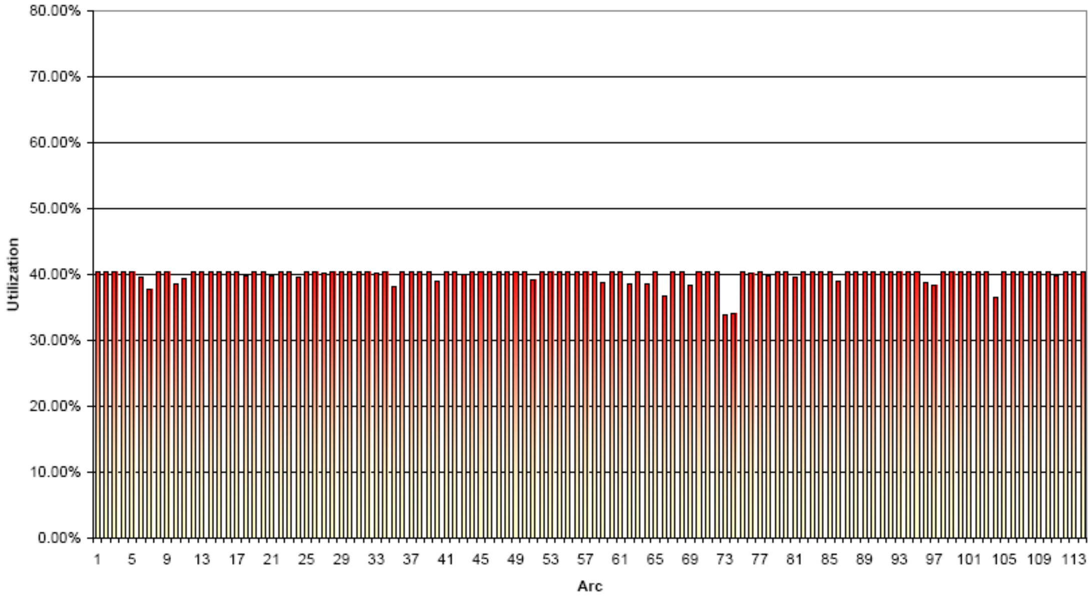


Figure 7.5: MPLS optimization under normal condition

Survivability optimization

Introducing the survivability constraints, the obtained results are significantly different, and a detailed discussion is needed. Table 7.5 shows the following statements:

- in case of single link failure, "default" metrics produce the forecasted congestion;
- the IS-IS metaheuristic metrics optimization reduces the objective function value by 26.45% carrying it to the 74.06%. Even if this could seems a good result, as the network is always under the congestion level, it is still above the empiric threshold of the 70% link occupation.
- The MPLS LP model reduces the maximum link occupation by 36.38% with respect to the "default" metrics and by 13.50% with respect to the IS-IS optimized metrics. The resulting maximum occupation, in case of failure of any one link, assumes a value of 64.06%.

Phase	u_{max}	% Gain (Def)	% Gain (IS-IS)	# LSP
Default	$u_{max} = 100.7\%$	–	–	0
IS-IS opt.	$u_{max} = 74.06\%$	26.45%	–	0
MPLS opt.	$u_{max} = 64.06\%$	36.38%	13.50%	543

Table 7.5: Survivability optimization results for the IBCN

The following figures show in the y-axis the objective function value in case of failure of the edge indicated in the x-axis.

In Figure 7.6, it can be seen that, when edge 15 fails the network suffers a congestion event with a consequent loss of performance. Moreover, when edge 34 fails the maximum occupation is over 90%.

Figure 7.7 shows the same histogram obtained when the IS-IS metrics have been optimized using the metaheuristic procedure, while in Figure 7.8 the results of the LP model are depicted. In this latter case, it is important to underline the smoothness (variance close to zero) of the objective function in case of failure of one of any of the edge of the network.

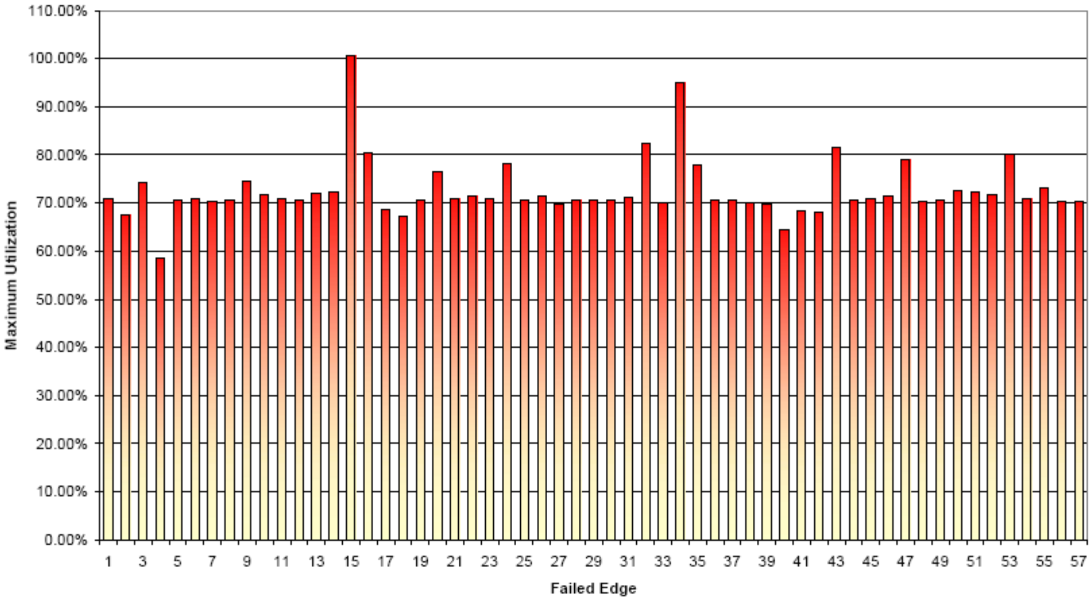


Figure 7.6: Objective function value in case of failure (default metrics)

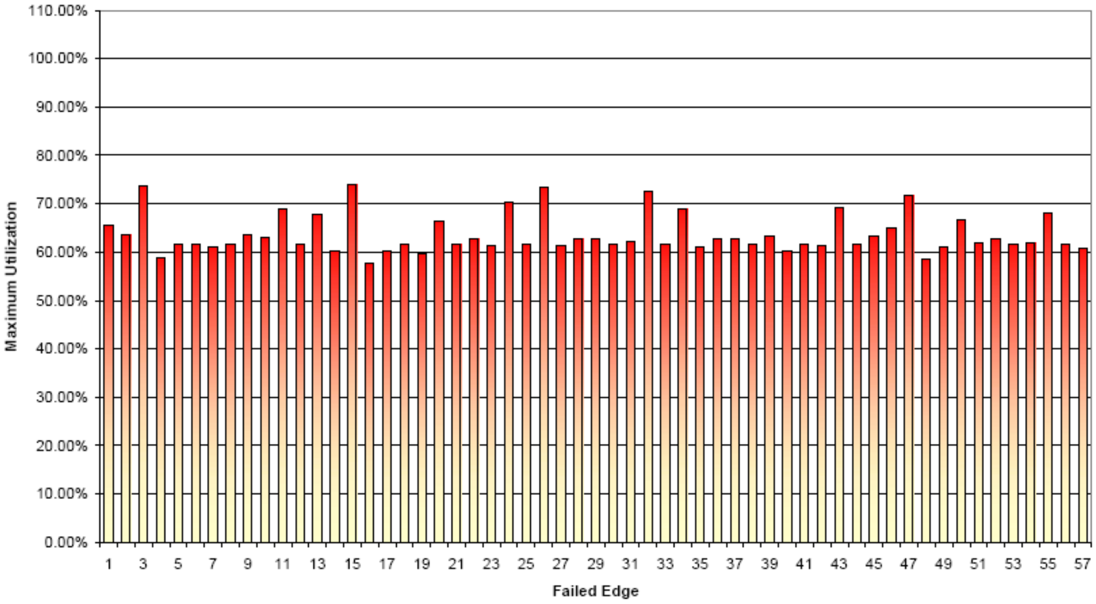


Figure 7.7: Objective function value in case of failure (IS-IS optimized metrics)

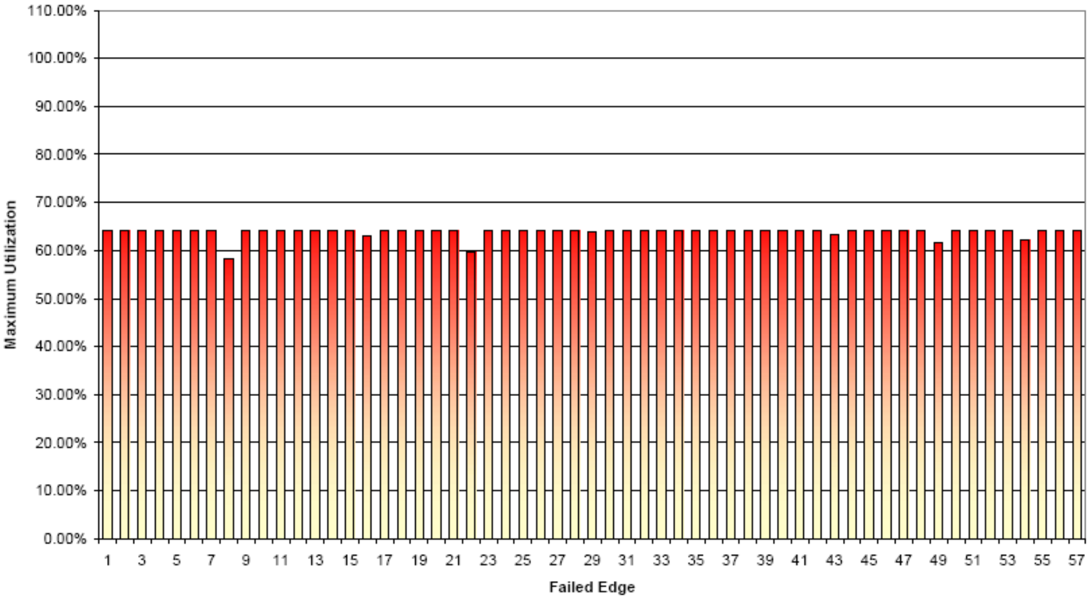


Figure 7.8: Objective function value in case of failure (LP model)

7.4 The Real ISP's Italian Backbone Network

The final test of the LP optimization model has been performed over the Italian part of Tiscali International Network (TINet) [21].

The network is composed of 18 nodes and 54 arcs with different maximum capacity that, at the moment of tests, ranges from 1 Gbps to 2.5 Gbps.

The traffic matrix, composed by 306 flows, has been built thanks to the willingness and the collaboration of the ISP's network operators and managers.

The resulting model is made of 1279 variables and 1402 constraints and it has been solved either as GRP or taking care of the survivability constraint. The obtained results have been compared with the "default" configuration, with the ISP's IS-IS metrics configuration, those used by the network operator in normal working condition, and with the IS-IS metrics obtained from the metaheuristic optimization process.

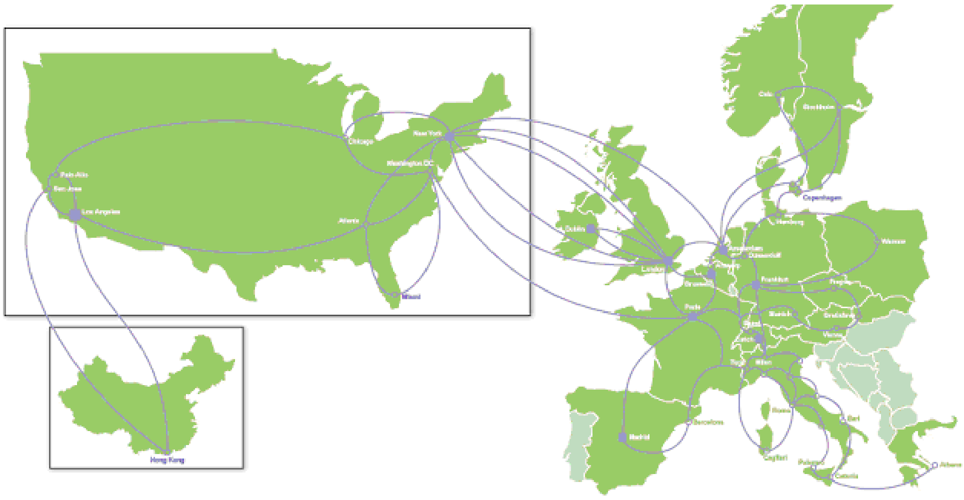


Figure 7.9: Tiscali International Network

The characteristics of the TINet network are summarized in Table 7.6, while the whole Tiscali International Network logical topology is depicted in Figure 7.9.

Nodes	18
Arcs	54
Flows	306
Variables	1279
Constraints	1402

Table 7.6: Characteristics of the Tinet Italian backbone

Routing Optimization

The objective function value comparison shown in Table 7.7 highlights the congestion level reduction granted by the optimization processes.

In particular, the MPLS optimization allows dropping the maximum utilization by 10.6%, with respect to the ISP's IS-IS configuration, using only 105 LSPs. Histograms report the occupation of the links in three different scenarios:

- ISP's IS-IS metrics (Figure 7.10)
- IS-IS optimized metrics (with Tabu Search) (Figure 7.11)
- MPLS optimization (Figure 7.12)

Phase	u_{max}	% Gain (Def)	% Gain (Tis)	# LSP
Default	$u_{max} = 72\%$	–	–	0
Tiscali	$u_{max} = 66\%$	8.3%	–	0
IS-IS opt.	$u_{max} = 61\%$	15.2%	7.6%	0
MPLS opt.	$u_{max} = 59\%$	18.1%	10.6%	105

Table 7.7: Routing optimization for the Tinet network

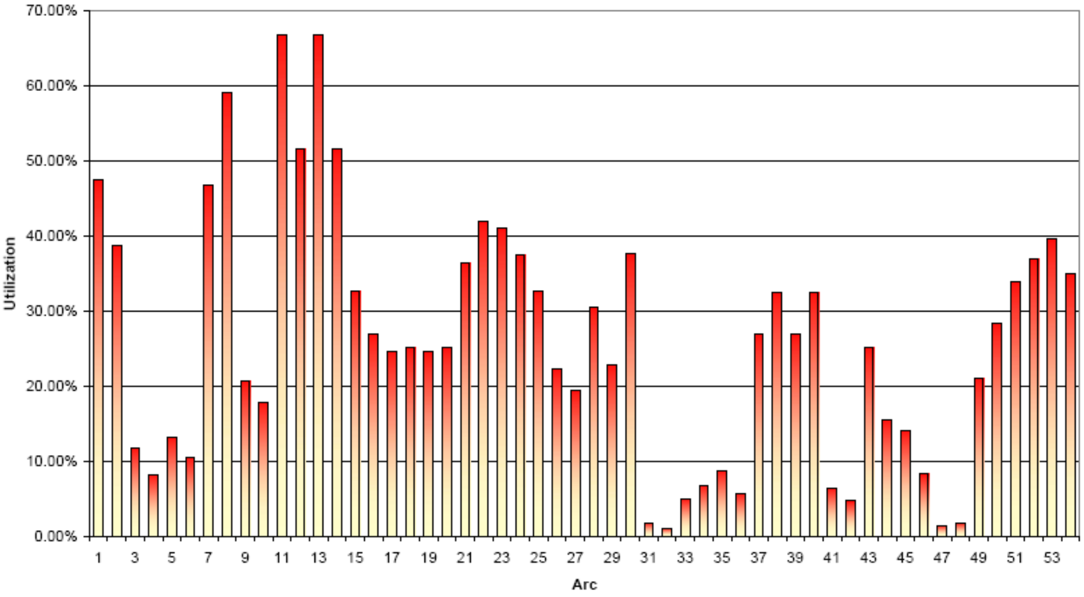


Figure 7.10: Link occupation level with Tinet's metrics

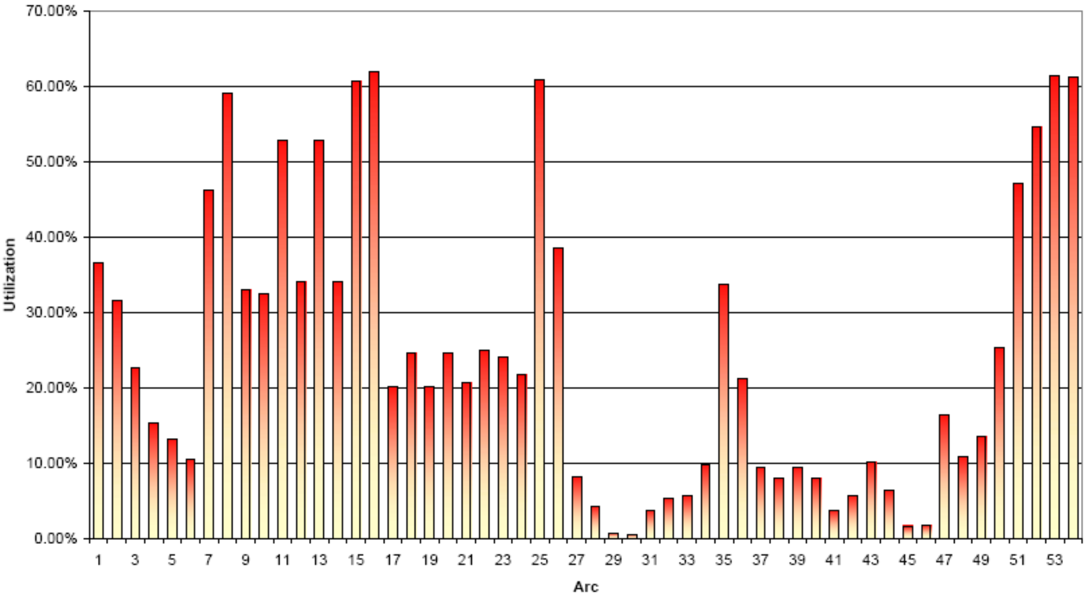


Figure 7.11: Link occupation level with IS-IS optimized metrics

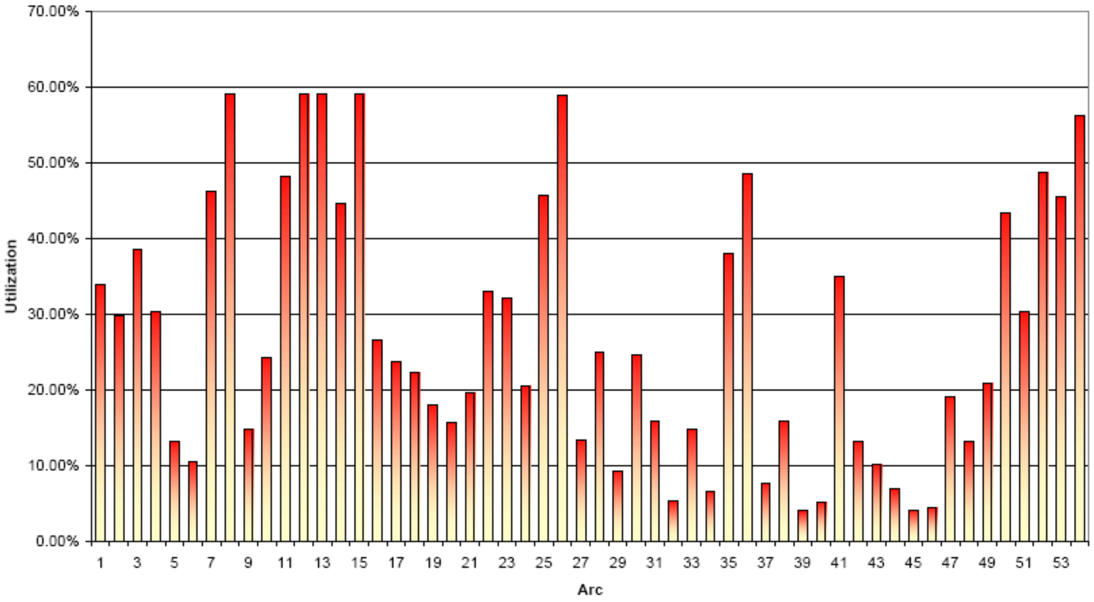


Figure 7.12: Link occupation level using MPLS LSPs

Survivability optimization

Simulation of a single edge failure has shown to TINet's network operators an alarming situation. In fact, two congestion scenarios arose if the original IS-IS metrics configuration is held, and Figure 7.13 displays such conditions when edges 10 and 12 fail.

The optimization of IS-IS metrics, considering the failure scenarios already produces good results, avoiding the congestion when the aforementioned edges fail as shown in Figure 7.14.

In the other hand, introducing the survivability constraints within the LP model, it is possible to reach a much better configuration, reducing the congestion by 29.1% with respect to the ISP's configuration and implementing only 86 LSPs, as shown in Table 7.8 and in Figure 7.15.

Phase	u_{max}	% Gain (Default)	% Gain (Tinet)	# LSP
Default	$u_{max} = 128\%$	–	–	0
Tiscali	$u_{max} = 117\%$	8.6%	–	0
IS-IS opt.	$u_{max} = 85\%$	33.6%	27.3%	0
MPLS opt.	$u_{max} = 83\%$	35.2%	29.1%	86

Table 7.8: Survivability scenarios comparison for the Tinet network

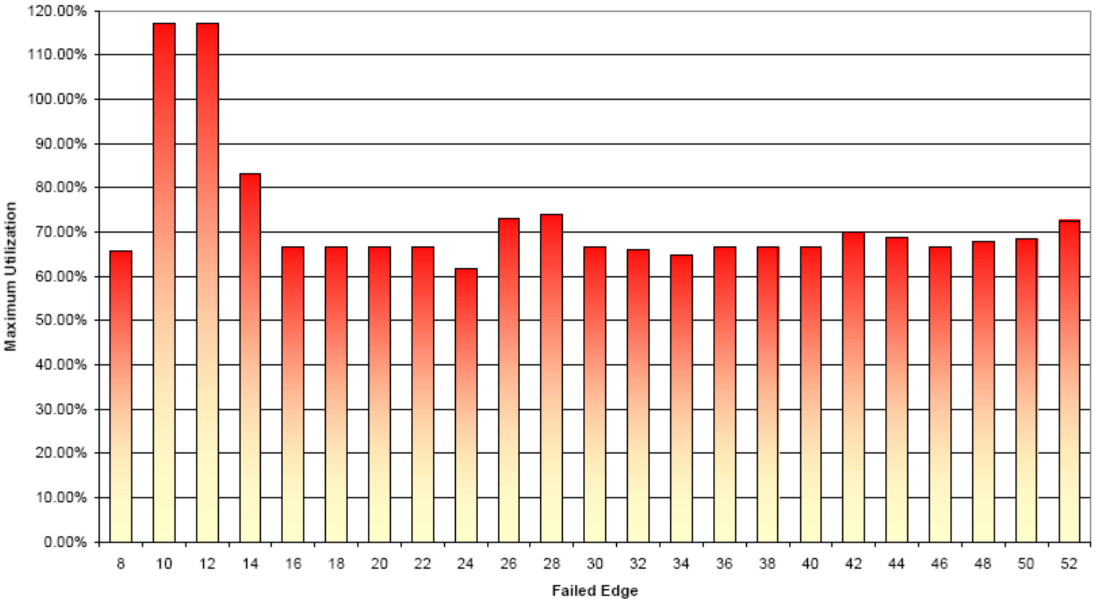


Figure 7.13: Objective function value in case of failure (Tinet metrics)

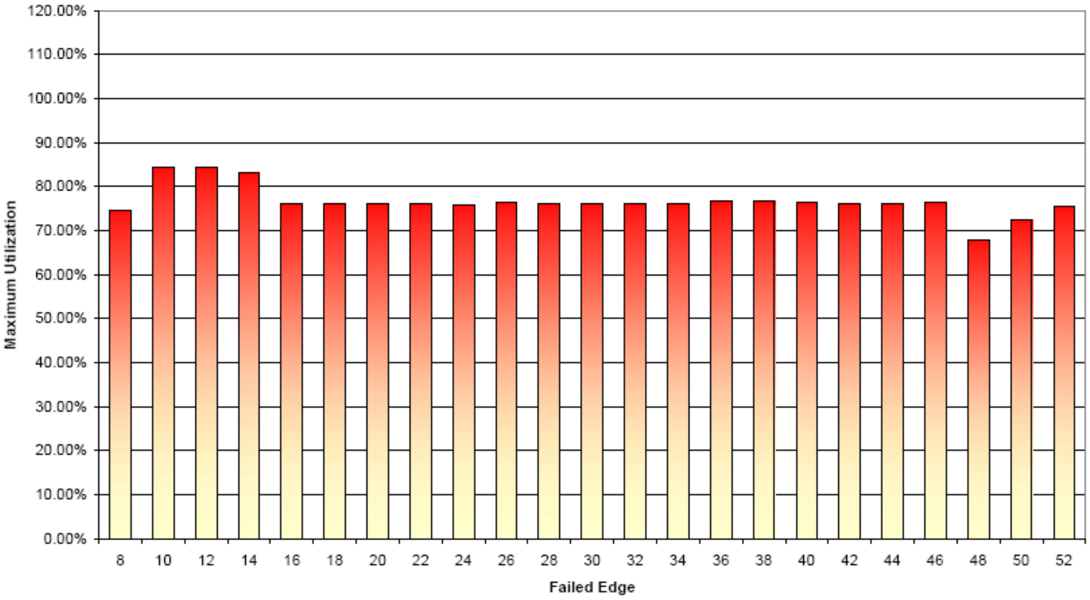


Figure 7.14: Objective function value in case of failure (IS-IS optimized metrics)

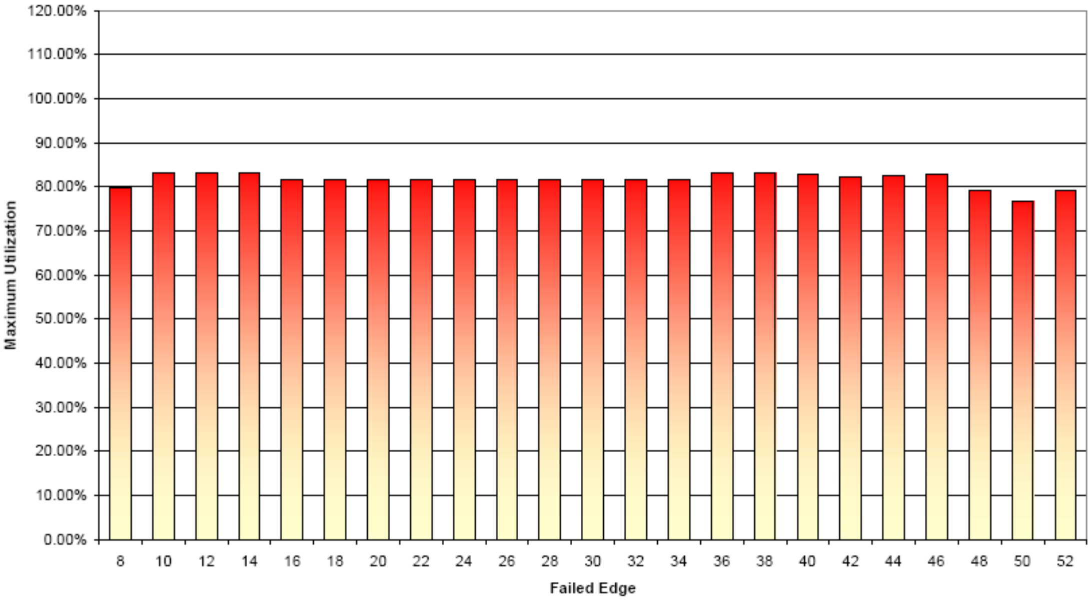


Figure 7.15: Objective function value in case of failure (LP results)

7.5 Conclusion and validation of results

It has be proven that the developed LP model is a good candidate for the solution of the congestion of a backbone network in case of a single edge failure.
Comparison with default configuration, with current ISP's configuration, and with results of an IS-IS optimization process have highlighted the good performances of the model. All results have been validated with a commercial simulation software called OPNET Modeler [15].

Chapter 8

Conclusions

Optimization problems in telecommunication have been widely studied, mainly focusing the improvement of traffic routing within the network.
On the other hand, methods to successfully deliver flows in case of failure have been slightly explored. I have presented three linear models that cover this lack using technological and topological constraints derived from the observation of real-world telecommunication networks.
In particular, the models use an optimal (or suboptimal) configuration of IGP protocol parameters as underlying condition and take advantages carried by Traffic Engineering MPLS techniques in order to find a survivable global configuration.
The main goals achieved in this dissertation can be summarized as follows:

- optimal configuration in normal working condition, solving the General Routing Problem in a network using IS-IS with complementary MPLS LSPs.
- optimal configuration under single link failure condition using IS-IS and MPLS LSPs.
- results shown that network operators presence is minimal and is only requested in the preliminary configuration. Survivability of the service delivery is guaranteed automatically.
- the rule of the “60%” has been revised

The empiric rule of “60%” states that an upgrade of the network is needed if the link occupation is greater than 60% of its capacity. I have shown that an optimal configuration of network parameters, pull down the utilization of the link with respect to the usual configuration made by network operators based on their experience. This result becomes relevant in economic terms because it reduces, and shifts in time, the investments required to upgrade the network.

Furthermore, the LP models are able to determine the optimal configuration in nearly real time, while the optimization processes based on heuristic techniques require numerous hours in order to find a feasible solution. Finally, my work has produced a traffic collector/analyzer/monitor tool that is currently used by an important Italian ISP (Tiscali).

References

[1] A^* algorithm. URL: `theory.stanford.edu/~amitp/GameProgramming/`. 49

[2] Autofocus. URL: `www.caida.org/tools/measurement/autofocus/`. 34

[3] CampusIO. URL: `net.doit.wisc.edu/~plonka/FlowScan/CampusIO.html`. 34

[4] CarrierIN. URL: `carrierin.sourceforge.net/`. 34

[5] CFlowd. URL: `www.caida.org/tools/measurement/cflowd/`. 30

[6] Cisco Netflow. URL: `www.cisco.com`. 28

[7] CPLEX. URL: `www.ilog.com/products/cplex/`. 15

[8] CUFlow. URL: `www.columbia.edu/acis/networks/advanced/CUFlow/CUFlow.html`. 34

[9] Flow-Tools. URL: `www.splintered.net/sw/flow-tools/`. 30

[10] FlowScan. URL: `www.caida.org/tools/utilities/flowscan`. 30

[11] Graph. URL: `sorsa.unica.it/crifor/it/software.php`. 65

[12] INTEC Broadband Communication Networks research group (IBCN). URL: `www.ibcn.intec.ugent.be`. 68

[13] JKFlow. URL: `users.telenet.be/jurgen.kobierczynski/jkflow/JKFlow.html`. 30, 34

[14] Jones Lustig Format for Multicommodity Flow Problems. URL: `www.di.unipi.it/di/groups/optimize/Data/MMCF.html`. 65

[15] OPNET. URL: `www.opnet.com`. 82

[16] OsiSolver. URL: `www.coin-or.org`. 65

[17] RFC 2702: Requirements for Traffic Engineering Over MPLS. URL: `ftp://ftp.rfc-editor.org/in-notes/rfc2702.txt`. 10

[18] RRDtool. URL: `oss.oetiker.ch/rrdtool/`. 34

[19] SubnetIO. URL: `net.doit.wisc.edu/~plonka/FlowScan/SubNetIO.html`. 34

[20] Tiscali. URL: `www.tiscali.com`. 27

[21] Tiscali International Network. URL: `www.tiscali.net`. 75

[22] ZIB - SNDlib. URL: `sndlib.zib.de/home.action`. 68

[23] S. Low A. Elwalid, C. Jin and I. Widjaja. Multi-path adaptive traffic engineering. *Computer Networks*, **40**:695–709, 2002. 16

[24] R. Sarkissian A. Lisser and J.-P. Vial. Solving LP relaxation for survivability problems in telecommunication networks. *Investigacion Operativa*, **9**:21–48, 2002. 17

[25] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. 18, 44, 49

[26] S. Beker, N. Puech, and V. Friderikos. A Tabu Search heuristic for the offline MPLS reduced complexity layout design problem. In *Networking*, pages 514–525, 2004. 16

[27] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, **16**(1). 49

[28] L. S. Buriol, M. G. C. Resende, and M. Thorup. Survivable IP network design with OSPF routing. *Networks*, **49**(1):51–64, 2007. 17

[29] V. Cerny. A thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Optimization Theory and Appl.*, **45**:41–51, 1985. 22

[30] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990. 47

[31] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, 1963. 43

[32] Edsger W. Dijkstra. A note on two problem in connexion with graphs. *Numerische Mathematik*, **1**:269–271, 1959. 5, 49

[33] M. Dorigo and G. Di Caro. New Ideas in Optimization: The Ant Colony Optimization Meta-Heuristic. pages 11–32. McGraw-Hill, 1999. 24

[34] Marco Dorigo. *Optimization, learning and natural algorithms*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1992. 24

[35] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. The MIT Press, 2004. 24

[36] U. Killat E. Mulyana. A Hybrid Genetic Algorithm Approach for OSPF Weight Setting Problem. In *Peroc. of the* 2^{nd} *Polish-German Teletraffic Symposium PGTS*, 2004. 15

[37] J. Edmonds and R.M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, **19**(2):248–264, 1972. 47

[38] A. Elwalid, C. Jin, S. H. Low, and I. Widjaja. MATE: MPLS adaptive traffic engineering. In *INFOCOM*, pages 1300–1309, 2001. 16

[39] S. Cerav Erbas and C. Erbas. A multiobjective off-line routing model for MPLS networks. In *Proc. of the 18th International Teletraffic Congress*, 2003. 16

[40] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming : Sequential Unconstrained Minimization Techniques.* John Wiley & Sons, New York, NY, USA, 1968. 43

[41] R. W. Floyd. Algorithm 97: Shortest path. *Communication ACM*, **5**(6), 1962. 49

[42] L. R. Ford Jr and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, **8**(3):399–404, 1956. 47

[43] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. 2000. 14, 17, 51, 52, 54

[44] B. Fortz and M. Thorup. Robust optimization of OSPF/IS-IS weights. *INOC 2003*, 2003. iv, 2, 15, 17, 52

[45] Antonio Frangioni. *Dual Ascent Methods and Multicommodity Flow Problems.* PhD thesis, Dipartimento di Informatica, Università di Pisa, 1997. 50

[46] D. M. Gay, N. K. Karmarkar, and K. G. Ramakrishnan. The Karmarkar algorithm: Adding wings to linear programming. *Record AT & T Bell Laboratories*, **64**(2):4–10, 1986. 43

[47] F. Glover and M. Laguna. *Tabu Search.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998. 20

[48] J. Harmatos. A heuristic algorithm for solving the static weight assignment optimisation problem in OSPF networks. In *Proc. of of Global Internet Conference*, 2002. 15

[49] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot. Analysis of link failures in an IP backbone. In *Proc. of ACM SIGCOMM Internet Measurement Workshop 2002, Marseille, France, Nov 2002*, Nov 2002. 12

[50] Donald B. Johnson. Efficient algorithms for shortest paths in sparse networks. *J. ACM*, **24**(1):1–13, 1977. 49

[51] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, **220, 4598**:671–680, 1983. 22

[52] C. C. Ribeiro L. S. Buriol, M. G. C. Resende and M. Thorup. A memetic algorithm for OSPF routing. In *Proc. of the* 6^{th} *INFORMS Telecom*, pages 187–188, 2002. 15

[53] Yu Liu. *Spare Capacity Allocation: Model, Analysis, and Algorithm.* PhD thesis, University of Pittsburgh, School of Information Sciences, 2001. 2, 18

[54] Yu Liu and D. Tipper. Spare capacity allocation for non-linear cost and failure-dependent path restoration. In *Proceeding of the* 3^{rd} *International Workshop on Design of Reliable Communication Networks, DRCN*, 2001. 18

[55] Yu Liu and D. Tipper. Successive survivable routing for node failures. In *Proceeding of IEEE Global Communications Conference*, 2001. 18

[56] G. F. Luger. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving.* Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA, 1997. 21

[57] M. Resende M. Ericsson and P. Pardalos. A genetic algorithm for the weight setting problem in OSPF routing. *Combinatorial Optimization*, **6**:299–333, 2002. 15

[58] H. H. Aly M. F. El-Hawary, M. N. El-Derini. Heuristics for Rerouting of Label Switched Paths in MPLS Based IP Networks. In *ISCC '03: Proceedings of the Eighth IEEE International Symposium on Computers and Communications*, 2003. 16

[59] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot. Characterization of failures in an IP backbone. In *IEEE Infocom 2004, IRC-TR-04-015, Hong Kong, China, March 2004*, 2004. 12

[60] Cristian Murgia. *Optimal Telecommunication Network planning with Topological and Quality constraint.* PhD thesis, Dipartimento di Ingegneria Elettrica ed Elettronica, Università di Cagliari, 2007. 17, 67

[61] A. Nucci, B. Schroeder, S.k Bhattacharrya, N. Taft, and C. Diot. IGP link weight assignment for transient link failures. In 18th *International Teletraffic Congress*, Berlin, Germany, 2003. 17

[62] D. Rajan and A. Atamtürk. A directed cycle based column-and-cut generation method for capacitated survivable network design. *Networks*, **43**:201–211, 2004. 63

[63] Anton Riedl. Optimized routing adaptation in IP networks utilizing OSPF and MPLS. In *IEEE 2003 International Conference on Communications (ICC), Anchorage, USA, May 2003*, May 2003. 16, 52

[64] R. Susitaival and S. Aalto. Adaptive load balancing with OSPF (extended version). Technical report, 2005. 15, 16

[65] R. Sustiatival, S. Aalto, and J. T. Virtamo. Adaptive load balancing using MPLS. In *MMB*, pages 15–24, 2004. 16

[66] D. Yuan. A bi-criteria optimization approach for robust OSPF routing. In 6th *IEEE Workshop on IP Operation Management (IPOM)*, 2003. 17